Deep Reinforcement Learning for Surgical Gesture Segmentation and Classification

Daochang Liu and Tingting Jiang

National Engineering Lab for Video Technology, Cooperative Medianet
Innovation Center, School of EECS, Peking University, Beijing 100871, China
{daochang, ttjiang}@pku.edu.cn

Abstract. Recognition of surgical gesture is crucial for surgical skill
assessment and efficient surgery training. Prior works on this task are
based on either variant graphical models such as HMMs and CRFs, or
deep learning models such as Recurrent Neural Networks and Temporal
Convolutional Networks. Most of the current approaches usually suffer
from over-segmentation and therefore low segment-level edit scores. In
contrast, we present an essentially different methodology by modeling
the task as a sequential decision-making process. An intelligent agent
is trained using reinforcement learning with hierarchical features from a
deep model. Temporal consistency is integrated into our action design
and reward mechanism to reduce over-segmentation errors. Experiments
on JIGSAWS dataset demonstrate that the proposed method performs
better than state-of-the-art methods in terms of the edit score and on
par in frame-wise accuracy. Our code will be released later.

Keywords: surgical gesture segmentation, surgical gesture classification, deep reinforcement learning, time series analysis

1   Introduction

Joint surgical gesture segmentation and classification is fundamental for objective surgical skill assessment and for improving efficiency and quality of surgery
training [1]. The goal is to segment robotic kinematic data or video sequence
and to classify segmented pieces into surgical gestures, such as *reaching for the
needle*, *orienting needle* and *pushing needle through the tissue*, etc.

Variant temporal models have been exploited in prior works on surgical gesture segmentation and classification. One branch of works has been based on
hidden Markov models (HMMs) [2–4], differing from each other in how the
emission probability is modeled. HMM-based methods assume that gesture label at frame t is only conditioned on previous frame $t-1$, leaving long-term
dependency unconsidered. Another branch has been based on conditional random fields (CRFs) [5–7] and their extensions, which obtains the gesture sequence
by minimizing an overall energy function. Although these methods capture temporal patterns by the pairwise potentials in their energy functions, they produce
severe over-segmentation and therefore suboptimal segmental edit scores. In recent years, a third branch using deep learning has set new benchmarks for this

task. Recurrent neural networks, in particular LSTMs, were applied in [8]. A memory cell is maintained in LSTM to remember and forget action changes over time. [9] proposed a spatiotemporal CNN, in which the spatial component described relationships of objects in the scene, and a long temporal convolutional filter captured how the relationships change temporally. Thereupon, [10, 11] went further and built a hierarchical encoder-decoder network called Temporal Convolutional Network (TCN) composed of long temporal convolutional filters, upsampling/downsampling layers, and normalization layers. In spite of the promising performance improvement achieved, these methods are only driven by frame-wise accuracy due to their cross-entropy training loss.

Unlike prior works, we propose an essentially different deep reinforcement learning approach for joint surgical gesture segmentation and classification, which is driven by both frame-wise accuracy and segment-level edit score. Reinforcement learning has gained remarkable success recently in domains like playing Go [12], Atari games [13], and anatomical landmark detection in medical images [14], etc. However, reinforcement learning has not been applied in surgery gesture segmentation in existing works. We formulate the task as a sequential decision-making process and train an agent to operate in a human-like manner. The agent looks through the surgical data sequence from the beginning, segment the sequence step by step and classify frames simultaneously. To highlight, our agent learns a strategical policy—skim fast in the middle of segments and examine attentively at segment boundaries, which resembles human intelligence. Additionally, current deep learning methods like RNN and TCN handle temporal consistency *implicitly* by memory cells or temporal convolutions. On the contrary, we enforce temporal consistency *explicitly* by the design of action and reward. The reward consists of two terms that guide the agent to high accuracy and high edit score respectively. To combine reinforcement learning with the hierarchical representation learned by deep neural networks, features extracted by TCN are utilized as powerful state representation for the agent. The proposed method is tested on the suturing task of the JIGSAWS dataset [1, 15]. Experiments show that our method outperforms state-of-the-art methods in terms of edit score. In summary, our contributions are three-fold:

- Joint surgical gesture segmentation and classification is formulated, to our best knowledge for the first time, as a sequential decision-making problem using deep reinforcement learning.
- The two evaluation metrics of frame-level accuracy and segment-level edit score are both incorporated into the rewarding mechanism.
- Our method outperforms state-of-the-art methods in terms of edit score while retaining comparable frame-wise accuracy.

2 Preliminary

Reinforcement learning (RL) is a computational approach to decision-making problems with definite goals [16], where an artificial agent learns its policy from interactions with the environment. At each time step, the agent observes the
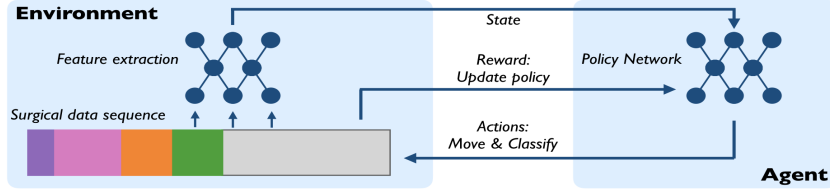
Fig. 1. Overview diagram of the proposed method. An agent perceives the environment, selects an action, and update its policy to maximize future rewards.

state of environment and selects an action accordingly, which in turn affects the environment. The agent earns a numerical reward for each action and updates its policy to maximize future reward. This sequential decision-making process is formalized as a Markov Decision Process (MDP) [16] $\mathcal{M} := (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ denotes state transition probabilities, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ denotes a reward function for each action performed in certain state, and $\gamma \in [0, 1]$ is the discount factor balancing between immediate and long-term reward. The agent learns from experience to optimize its policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, which is stochastic in general. The goal is to maximize discounted future reward accumulated from current time step to the end of learning episode.

3 Proposed Method

The input to our model is a data sequence $\{x_t\}$, which can be either visual features extracted from surgery video frames or kinematic data frames collected from surgical robots, together with its ground truth gesture label sequence $\{y_t\}$. Each $x_t \in \mathbb{R}^{n_x}$ and each $y_t \in \{1, 2, \ldots, n_y\}$, where n_x is the dimensions of features, n_y is the number of gesture classes, and $1 \leq t \leq n_t$. Note that the number of frames n_t may differ between each data sequence.

We propose to model joint surgical gesture segmentation and classification as a sequential decision-making problem, illustrated in Fig. 1. An agent is built to interact with the visual or kinematic data sequence, i.e., the environment. Initially positioned at the beginning of data sequence, the agent selects a proper step size and moves forward at each time step, concurrently classifying frames stepped over. Following the standard paradigm of reinforcement learning, we formalize the task as a Markov Decision Process. The action, the state and the reward of our proposed MDP model are detailed as follows.

Action. The action of the proposed model includes two subactions, which are to decide how far to move forward and to choose which class label to give. The action set is defined as:

$$\mathcal{A} := \mathcal{K} \times \mathcal{C} \tag{1}$$

where $\mathcal{K}$ represents a predefined set of optional step sizes, and $\mathcal{C}$ represents the set of gesture classes. From the start, the agent keeps selecting an action

pair (k, c) from action set $\mathcal{A}$ to walk through and classify the data sequence until reaching the end. Step size set $\mathcal{K}$ is defined to contain one small step and one large step: $\mathcal{K} := \{k_s, k_l\}$. This binary design enables the agent to alter its step size based on the confidence in the gesture label to give. The agent can adopt the smaller step when the state is not discriminative enough such as at the boundaries between gestures, and adopt the larger step otherwise. At each action, the k frames stepped over by the agent are labeled with the same class c, explicitly enforcing temporal consistency.

State. Raw features $\{x_t\}$ are challenging for the agent to fully understand the surgical activity. Therefore we define the state to be a combination of high-level representation and other auxiliary information, which assists the agent to make a better decision. We utilize TCN [10, 11] to extract such high-level representation from raw features.

The state observed at frame t is the concatenation of current and future feature vectors extracted by TCN, gesture transition probabilities from a language and duration model [17], and a one-hot vector, which is formalized below:

$$s^t := (s_{tcn}^t, s_{tcn}^{t+k_s}, s_{tcn}^{t+k_l}, s_{trans}, s_{hot}) \tag{2}$$

where $s_{tcn}^t, s_{tcn}^{t+k_s}, s_{tcn}^{t+k_l}$ are respectively TCN features at the current frame, k_s frames later and k_l frames later, s_{trans} are probabilities of transition into each gesture computed from a statistical language model, $s_{hot} \in \{0, 1\}^{n_y}$ is a one-hot vector such that the gesture class given by last action is 1 and all others are 0.

Statistical Language and Duration Model. Similar to [17], we use a statistical model to describe the length and contextual pattern of gestures. Specifically, we use Gaussian distributions for gesture durations and a bigram language model for gesture transitions, assuming the gesture class depends only on one previous class. The statistical model is formalized as:

$$p(i \mid j, l) := \begin{cases} \frac{N(j,i)}{N(j)} \cdot CDF_j(l) & \text{if } i \neq j \\ 1 - CDF_j(l) & \text{if } i = j \end{cases} \tag{3}$$

where $p(i \mid j, l)$ is the probability of transition from gesture j to gesture i given l—how many frames the agent has stayed in gesture j, $N(j, i)$ and $N(j)$ are respectively occurrence counts of ordered gesture pair (j, i) and gesture j alone in training data, CDF_j stands for the cumulative distribution function of a Gaussian distribution modeling the length of gesture j. The Gaussian distributions are parameterized by maximum likelihood estimation using training data. Then s_{trans} is set as the probabilities of transition to each gesture, given the gesture class of last action and how long the agent has stayed in this gesture.

Reward. The reward is numerical feedback for each action performed by the agent. Given action pair (k, c) performed at frame t, the reward in our MDP model is designed as:

$$r(s^t, (k, c)) := \alpha k - \sum_{t'=t}^{t+k-1} \mathbb{1}(y_{t'} \neq c) \tag{4}$$

where the first term encourages the agent to adopt the larger step, the second term penalizes the errors caused by this action, and α is a weight parameter balancing the two terms. The two evaluation metrics of accuracy and edit score are both incorporated in this reward inherently. While the second term serves as straightforward guidance for the agent to achieve high frame-wise accuracy, the first term is crucial for a good edit score at segment-level. Preference for the larger step can mitigate the jittering between gestures and therefore can reduce over-segmentation errors, which is validated in our experiments section.

Policy Learning. We use a multilayer perceptron (MLP) to model the policy, whose input layer has the same number of units as the dimensions of the state, and output layer has the same number of units as the dimensions of action space. The policy network takes environment state as input and output a distribution over action space. With the MDP process well defined, any standard reinforcement learning method can be applied to policy learning. We choose Trust Region Policy Optimization (TRPO) [18] since it is theoretically guaranteed to improve the policy monotonically.

4 Experiments

We evaluate the proposed method on JIGSAWS [1, 15], a public benchmark dataset recorded using the *da Vinci* surgical system. We use the video and kinematic data from the suturing task, which contains 39 sequences performed by eight subjects with varying skill levels. For video data, we use features extracted from each frame image by a spatial CNN [9], which is consistent with TCN [11]. As for kinematic data, we pick the same subset of features as in [11], which are position, velocity and gripper angle for both slave manipulators.

Experiments Setup. The standardized leave-one-user-out (LOUO) evaluation setup of JIGSAWS is followed in our experiments. Trials performed by a single user are left out as testing set while all remaining trials are used as training set, resulting in 8-fold cross-validation. Since our RL based method is inherently stochastic, we train the TCN for state feature extraction five times, train the agent using TRPO three times, and test on each data sequence ten times, with $5 * 3 * 10 = 150$ runs in total. Results are averaged over these 150 runs. We also include an ablation study to measure the impact of each component of the state.

We include three evaluation metrics: accuracy, edit score, and F1 score. Accuracy is the percentage of correctly labeled frames, measuring the performance at the frame level. Edit score is the normalized Levenshtein distance between predicted gesture sequence and ground truth, measuring the performance at the segment level, which is between 0 and 100 (the higher the better). F1 score is introduced for this task in [10]. Each predicted gesture segment is considered to be true or false positive according to whether its Intersection over Union (IoU) with respect to the corresponding ground truth segment is above a threshold. Then F1 score is the harmonic mean of the precision and recall: $F1 = \frac{2*precision*recall}{precision+recall}$.

Implementation Details. The proposed model is implemented with Python and *OpenAI Baselines* library [19]. The policy network is of one hidden layer and

Table 1. Results on the suturing task of JIGSAWS. F1@{10,25,50} stands for F1 score with the IoU threshold set to 10%, 25% and 50%. TCN* stands for our re-implemented version of TCN with several minor changes for state feature extraction. The four entries at the bottom are results of our RL method with partial state or full state.

Method	Video			Kinematic		
	Acc	Edit	F1@{10,25,50}	Acc	Edit	F1@{10,25,50}
SD-SDL [3]	-	-	-	78.6	83.3	-
Bidir LSTM [8]	-	-	-	83.3	81.1	-
LC-SC-CRF [7]	-	-	-	**83.4**	76.8	-
Seg-ST-CNN [9]	74.7	66.6	-	-	-	-
TCN [11]	81.4	83.1	-	79.6	85.8	-
TCN*	**81.71**	86.63	91.0, 89.5, 82.0	82.57	86.58	90.5, 89.3, **82.4**
RL (no tcn)	32.35	15.04	13.1, 6.1, 3.4	32.90	17.21	14.8, 7.5, 4.6
RL (no future)	80.84	81.58	87.5, 85.6, 77.4	81.48	81.07	86.5, 84.6, 77.4
RL (no trans)	81.32	87.30	91.5, 90.1, 81.8	82.06	87.12	90.7, 89.1, 81.8
RL (full)	81.43	**87.96**	**92.0**, **90.5**, **82.2**	82.07	**87.86**	**91.1**, **89.5**, 82.3

64 hidden units. We set k_s to be the minimum gesture length in training set, and k_l to be the minimum of mean gesture lengths for each class, which are 4 and 21 frames for example. The discount factor γ and the reward weight α are set to 0.9 and 0.1 respectively in experiments. Besides, we re-implement the TCN for state feature extraction using PyTorch with several minor changes, which can be regarded as a baseline for our RL method. The convolutional layers in the decoder of TCN are replaced with transposed convolutional layers. And due to the data is highly imbalanced, we use weighted cross-entropy as the training loss of TCN instead. Activations before the last fully-connected layer are used as state features, which is of 32 dimensions. Our code will be released later.

Result. Experiment results on both video and kinematic data are shown in Table. 1. Our RL based approach is compared to the original TCN, the modified TCN and several other recent works. All results of prior works are excerpted from [11]. Compared to existing works, the proposed method achieves higher edit score and F1 score at a negligible cost of accuracy. We present the result of ablation study on the state design as well. Each of following components is removed from the state to justify its necessity: 1) all TCN features 2) TCN features at future frames 3) transition probabilities from the statistical model. Results show that all these components are required to achieve the best performance. And the high-level representation extracted by TCN is the most important one.

A prediction example produced by our agent is provided in Fig. 2. We plot the history of steps of the agent, finding it interesting that the agent tends to select the larger step in the middle of gestures and the smaller step at the boundaries. Such behavior pattern verifies our intuitions on the action and reward design.

Does the Larger Step Really Benefit? To further validate our reward design that encourages the larger step, we complete a comparative experiment on the step size. We set the step size set $\mathcal{K}$ to contain only a single option, and set this option to 1, 2, 4, 8, 16, 32 frames. From the result in Table. 2, the larger step the agent can take, the higher edit score is achieved unless the step

Fig. 2. Prediction example: (1) ground truth sequence (2) step history of the agent (3) predicted sequence. Each color corresponds to a gesture. Our agent learns to skim fast in the middle of gestures and examine cautiously at the boundaries.

Table 2. Experiments on step size

Step Size	Video		Kinematic	
	Acc	Edit	Acc	Edit
1	81.65	80.09	82.31	80.12
2	81.67	83.46	82.24	82.47
4	81.53	86.04	82.16	85.35
8	80.95	87.56	81.91	87.27
16	79.12	88.28	79.81	88.13
32	72.94	84.05	72.86	84.04
4 & 21	81.43	87.96	82.07	87.86

is excessively large. But larger steps such as 8, 16 and 32 degrade the accuracy considerably. Our binary design of step sizes achieves the promising result on both metrics.

5 Conclusion and Future Work

In this work, we proposed a novel method based on deep reinforcement learning for joint surgical gesture segmentation and classification. An artificial agent is trained to act in a human-like manner. By the state, the action and the reward formulation, temporal consistency is explicitly stressed and over-segmentation errors are reduced. The proposed method outperforms the state-of-the-art on JIGSAWS dataset in terms of edit score, while retaining comparable frame-wise accuracy. Future work can be made in following two aspects: 1) developing the step size options into a continuous set 2) combining the state feature extraction network and the policy network for an end-to-end model.

References

1. Ahmidi, N., Tao, L., Sefati, S., Gao, Y., Lea, C., Haro, B.B., Zappella, L., Khudanpur, S., Vidal, R., Hager, G.D.: A dataset and benchmarks for segmentation

and recognition of gestures in robotic surgery. IEEE Transactions on Biomedical Engineering **64**(9) (2017) 2025–2041

2. Tao, L., Elhamifar, E., Khudanpur, S., Hager, G.D., 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) 167–177

3. Sefati, S., Cowan, N.J., Vidal, R.: Learning shared, discriminative dictionaries for surgical gesture segmentation and classification. In: MICCAI Workshop: M2CAI. Volume 4. (2015)

4. Varadarajan, B., Reiley, C., Lin, H., Khudanpur, S., Hager, G.: Data-derived models for segmentation with application to surgical assessment and training. In: MICCAI, Springer (2009) 426–434

5. Tao, L., Zappella, L., Hager, G.D., Vidal, R.: Surgical gesture segmentation and recognition. In: MICCAI, Springer (2013) 339–346

6. Lea, C., Hager, G.D., Vidal, R.: An improved model for segmentation and recognition of fine-grained activities with application to surgical training tasks. In: WACV, IEEE (2015) 1123–1129

7. Lea, C., Vidal, R., Hager, G.D.: Learning convolutional action primitives for fine-grained action recognition. In: ICRA, IEEE (2016) 1642–1649

8. DiPietro, R., Lea, C., Malpani, A., Ahmidi, N., Vedula, S.S., Lee, G.I., Lee, M.R., Hager, G.D.: Recognizing surgical activities with recurrent neural networks. In: MICCAI, Springer (2016) 551–558

9. Lea, C., Reiter, A., Vidal, R., Hager, G.D.: Segmental spatiotemporal cnns for fine-grained action segmentation. In: ECCV, Springer (2016) 36–52

10. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. In: CVPR. (2017)

11. Lea, C., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks: A unified approach to action segmentation. In: ECCV, Springer (2016) 47–54

12. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al.: Mastering the game of go with deep neural networks and tree search. Nature **529**(7587) (2016) 484–489

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

14. Ghesu, F.C., Georgescu, B., Mansi, T., Neumann, D., Hornegger, J., Comaniciu, D.: An artificial agent for anatomical landmark detection in medical images. In: MICCAI, Springer (2016) 229–237

15. Gao, Y., Vedula, S.S., Reiley, C.E., Ahmidi, N., Varadarajan, B., Lin, H.C., Tao, L., Zappella, L., Béjar, B., Yuh, D.D., et al.: JHU-ISI gesture and skill assessment working set (JIGSAWS): A surgical activity dataset for human motion modeling. In: MICCAI Workshop: M2CAI. Volume 3. (2014) 3

16. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. Volume 1. MIT press Cambridge (1998)

17. Richard, A., Gall, J.: Temporal action detection using a statistical language model. In: CVPR. (2016) 3131–3140

18. Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P.: Trust region policy optimization. In: International Conference on Machine Learning. (2015) 1889–1897

19. Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y.: OpenAI Baselines. https://github.com/openai/baselines (2017)