2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Multi-Task Recurrent Neural Network for Surgical Gesture Recognition and Progress Prediction

Beatrice van Amsterdam[1], Matthew J. Clarkson[1], Danail Stoyanov[1]

***Abstract*— Surgical gesture recognition is important for surgical data science and computer-aided intervention. Even with robotic kinematic information, automatically segmenting surgical steps presents numerous challenges because surgical demonstrations are characterized by high variability in style, duration and order of actions. In order to extract discriminative features from the kinematic signals and boost recognition accuracy, we propose a multi-task recurrent neural network for simultaneous recognition of surgical gestures and estimation of a novel formulation of surgical task progress. To show the effectiveness of the presented approach, we evaluate its application on the JIGSAWS dataset, that is currently the only publicly available dataset for surgical gesture recognition featuring robot kinematic data. We demonstrate that recognition performance improves in multi-task frameworks with progress estimation without any additional manual labelling and training.**

I. Introduction

Automated surgical gesture recognition aims at automatically identifying meaningful action units within surgical tasks that constitute a surgical intervention. The process forms a fundamental step in the development of systems for surgical data science [1], objective skill evaluation [2, 3] and surgical automation [4, 5, 6]. The problem is however challenging because surgical gestures have high degree of variability due to multiple parameters in the operating surgeon's style and the patients' anatomy which alters the duration, kinematics and order of actions among different demonstrations [7].

Much research in the field, however, is based on the premise that many surgical tasks have well-defined structure and use specific action patterns to progress towards a surgical goal. Gesture flow has then been described through task-specific probabilistic grammars [8], which have been modelled with powerful statistical tools such as graphical models [9, 10] and neural networks [11, 12]. This work investigates if the recognition performance improves when the progress of the surgical task is modelled explicitly and learnt jointly with the action sequence, resulting in a more discriminative feature extraction process.

The effectiveness of multi-task learning [13] and surgical progress modelling has been demonstrated in previous work focused on surgical workflow analysis [14, 15], where the aim is to recognise surgical phases representing high-level surgical states. We adopt this approach with high-granularity gesture sequences and design a multi-task recurrent neural network for simultaneous gesture recognition and progress estimation. Differently from previous work, however, the task progress is based on the underlying action sequence rather than on time. We hypothesize that action-based progress estimation could help to learn action sequentiality despite duration variability and the presence of adjustment gestures and spurious motions, and thus reduce out-of-order predictions and over-segmentation errors. We also analyse different progress estimation strategies and highlight correlations between gesture and progress predictions.

We validate our algorithm on the kinematic data of the JIGSAWS dataset [16], featuring demonstrations of elementary surgical tasks collected from eight surgeons with different skill level using the da Vinci Surgical System (dVSS, Intuitive Surgical Inc.) [17]. Our experiments show that gesture recognition performance improves in multi-task frameworks with progress estimation at no additional cost, as the progress labels can be generated automatically from the data and available action labels.

A. Related Work

Gesture recognition from robot kinematics has been tackled through probabilistic graphical models such as Hidden Markov Models (HMMs) [9, 10, 18] and Conditional Random Fields (CRFs) [19, 20, 21]. These however rely on frame-to-frame and segment-to-segment transitions only, ignoring long-range temporal dependencies in the surgical demonstrations. Deep learning techniques have been recently used to capture complex, long-distance patterns through hierarchies of temporal convolutional filters [11, 22], LSTM networks [12] or deep Reinforcement Learning (RL) [23]. Besides, unsupervised [24, 25] and weakly-supervised [26] recognition have been shown through clustering, which reduces the dependency on annotations but at the expense of performance.

Surgical video rather than kinematics also embeds gesture information which can be extracted with spatio-temporal CNNs [27], 3D CNNs [28], multi-scale temporal convolutions [29, 30] or hybrid encoder-decoder networks with temporal-convolutional filters for local motion modelling and bidirectional LSTM for long-range dependency memorization [31].

Finally, a number of studies have approached surgical workflow analysis through multi-task learning. Examples

This work was supported by the Wellcome/EPSRC Centre for Interventional and Surgical Sciences (WEISS) at UCL (203145Z/16/Z), EPSRC (EP/P027938/1, EP/R004080/1,NS/A000027/1), the H2020 FET (GA 863146) and Wellcome [WT101957]. Danail Stoyanov is supported by a Royal Academy of Engineering Chair in Emerging Technologies (CiET1819/2/36) and an EPSRC Early Career Research Fellowship (EP/P012841/1).

[1]B. van Amsterdam, M. J. Clarkson and D. Stoyanov are with the Wellcome/EPSRC Centre for Interventional and Surgical Sciences (WEISS), University College London, London, United Kingdom. beatrice.amsterdam.18@ucl.ac.uk

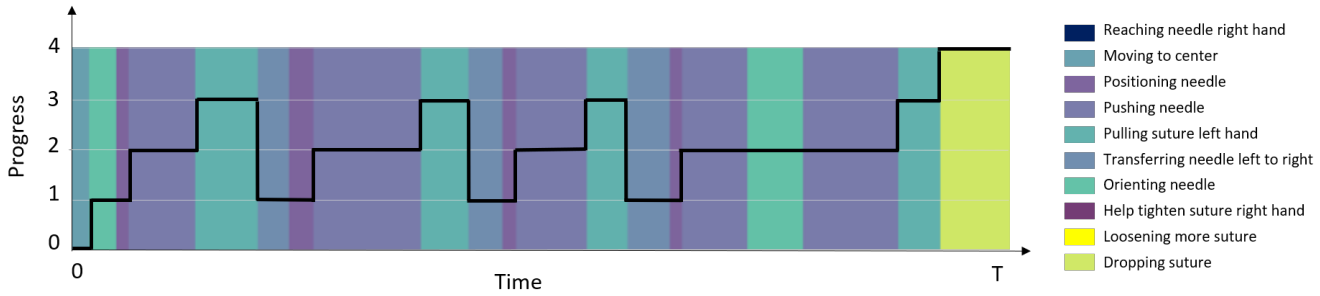

Fig. 1. Our definition of progress is dictated by the underlying action sequence. We identified five gestures that represent essential progressive stages in any complete suturing demonstration. The other classes represent adjustment gestures that serve to prepare or help to complete the execution of the essential gestures.

include systems for joint task and gesture classification [32], and models for joint phase recognition and tool detection [33] or progress estimation [15]. Phase recognition networks have also been pre-trained on auxiliary tasks such as prediction of the Remaining Surgery Duration (RSD) [14] or estimation of the frame temporal order [34], aiming to improve understanding of the temporal progression of the surgical workflow. Such approaches show that multi-task learning and progress modelling are beneficial for surgical workflow understanding and could support fine-grained analysis that requires discriminative feature extraction.

II. Methods

A. Dataset

We trained our network on the 39 suturing demonstrations of the JIGSAWS dataset, using the kinematic data (end-effector position, velocity, gripper angle) recorded at 30 Hz from the two Patient Side Manipulators (PSMs) of the dVSS. The trajectories were first smoothed with a low-pass filter with cut-off frequency $f_c = 1.5$ Hz against measurement noise [35], and then normalized to zero mean and unit variance to compensate for different units of measure. Finally, data were re-sampled from 30 Hz to 5 Hz for shorter computation time.

In order to learn the task progress, new ground truth labels were automatically generated from the available data and action labels. As a preliminary step, however, we carefully inspected the video recordings in order to identify possible imprecisions in the available annotations, that would affect the automatic generation of our progress labels. We identified and corrected 12 mistakes, affecting a total of 2356 data samples. Amendments to the original annotations are reported in the Appendix.

As illustrated in Fig. 1, our definition of progress is dictated by the underlying action sequence. Out of the 10 original action labels from JIGSAWS, we identified five gestures that constitute essential progressive stages in any complete suturing demonstration (*Reaching for the needle, Positioning the tip of the needle, Pushing the needle through the tissue, Pulling the suture, Dropping the suture*), generating a simplified probabilistic state machine that describes the commonly-observed workflow of the suturing task. The other classes represent adjustment gestures that serve to prepare or help to complete the execution of the essential gestures and that generally appear in variable order. We thus grouped fundamental gestures (performed by any of the two arms, even if JIGSAWS only features right-handed suturing demonstrations) and their corresponding adjustment gestures into 5 progress stages (from 0 to 4), as detailed below:

- Progress 0: G1 *Reaching for needle with right* + G5 *Moving to center of workspace*
- Progress 1: G2 *Positioning the tip of the needle* + G4 *Transferring the needle from left to right* before G2 + G8 *Orienting the needle* before G2
- Progress 2: G3 *Pushing the needle through the tissue* + G4/G8 before G3
- Progress 3: G6 *Pulling the suture with left* + G9 *Using right hand to tighten suture* + G10 *Loosening more suture* + G4/G8 before G6/G9/G10
- Progress 4: G11 *Dropping suture and moving to end points* + G4/G8 before G11

As the task evolution in time is affected by numerous factors, such as surgical skill and surgical context, we believe that activity-based progress could be better than time-based progress in reducing the kinematic feature variation for equal progress values. Moreover, it could help to learn action sequentiality despite the presence of adjustment gestures which occur in variable frequency and uncertain order.

B. Multi-task Recurrent Neural Network

Our multi-task architecture performs action recognition jointly with progress estimation. As the progress is quantized into 5 sequential steps, we estimate it using three different strategies: regression, standard classification and classification with ordered classes (or ordinal regression).

Notation: vectors are represented in bold lowercase letters (e.g. $\mathbf{y}$), scalars in lowercase letters (e.g. y), parameters and losses in uppercase letters (e.g. C).

Regression: As shown in Fig. 2, the kinematic features (K) are fed to a single-layer bidirectional LSTM with 1024 hidden units. Activations from the forward and backward

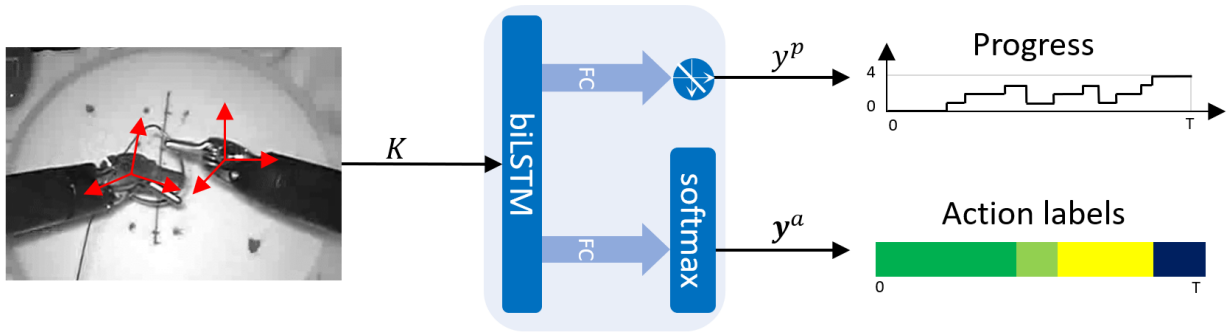

Fig. 2. Multi-task architecture for joint action recognition and progress regression. The kinematic features (K) are fed to a bidirectional LSTM cell, whose hidden units are connected to the regression node by a fully connected layer. The same hidden units are projected by a second fully connected layer into 10 logits with softmax activation function for action classification.

Fig. 3. Target encodings for progress regression, classification and ordered classification.

streams are concatenated into a 2048-dimensional vector and then connected to the regression node by a Fully Connected (FC) layer with linear activation function. The same 2048 features are also projected by a second fully connected layer into 10 logits with softmax activation function for action classification.

At each training iteration, we compute the regression loss using the Mean Absolute Error (MAE) over individual demonstrations:

$$MAE = \frac{1}{T}\sum_{t=1}^{T}|y_t^p - \hat{y}_t^p|, \tag{1}$$

and the classification loss using the Mean Cross Entropy (MCE) over individual demonstrations, as in [12]:

$$MCE = \frac{1}{T}\sum_{t=1}^{T}\left(-\sum_{c=1}^{C} y_{tc}^a \log(\hat{y}_{tc}^a)\right), \tag{2}$$

where T is the demonstration length (number of samples), C is the number of action classes, y_t^p and $\mathbf{y}_t^a$ are the regression and prediction nodes' output at timestamp t, and $\hat{y}_t^p$ and $\hat{\mathbf{y}}_t^a$ are the corresponding ground truths.

After model training, the regression output is rounded to the nearest integer for progress prediction, and the logit with largest activation is considered for action prediction.

Classification: To perform standard progress classification, we substitute the regression layer with a 5-logit fully connected layer with softmax activation function and MCE loss, thus obtaining a multi-hierarchical action recognition network (Fig. 2). After model training, the logit with largest activation is considered for progress prediction.

Ordered classification: Standard classification considers independent categories and does not penalize major ordering mistakes. In order to represent the succession of progress classes, we thus encoded the target vectors with the ordinal formulation of [36] as represented in Fig. 3, and substituted the categorical MCE loss with the Mean Binary Cross Entropy (MBCE) loss (i.e. Sigmoid activation function and MCE loss). MBCE sets up an independent binary classifier for each class and, in combination with the ordinal target encoding, generates a larger loss the further the prediction is from its ground truth. After model training, progress predictions are obtained from the output $\mathbf{y}^p$ of this classifier by finding the first index k where y_k^p<0.5.

In all the three cases, the final multi-task loss (L) is a weighted combination of the two single-task losses ($L_1 = MCE$, $L_2 = MAE$ *or* MCE *or* $MBCE$):

$$L = w_1 L_1 + w_2 L_2 \tag{3}$$

However, multi-task networks are generally difficult to train, as task imbalances may lead to the generation of shared features that are not useful across all tasks. In order to automatically balance our model training, we used the GradNorm algorithm [37] for gradient normalization, that has been shown to improve accuracy and reduce overfitting across multiple tasks when compared to single-task networks. GradNorm dynamically updates the single-task loss weights (w_1, w_2) during training by optimizing an additional loss ($Lgrad$), which aims at regularizing the training rate of the individual tasks:

$$Lgrad = \sum_{i=1}^{2}\left|g_{\mathbf{w}}^{(i)} - \bar{g}_{\mathbf{w}} \times (r_i)^{\alpha}\right| \tag{4}$$

$g_{\mathbf{w}}^{(i)} = ||\bigtriangledown_{\mathbf{w}} w_i L_i||$ is the L_2-norm of the gradient of the weighted single-task loss $w_i L_i$ with respect to the network

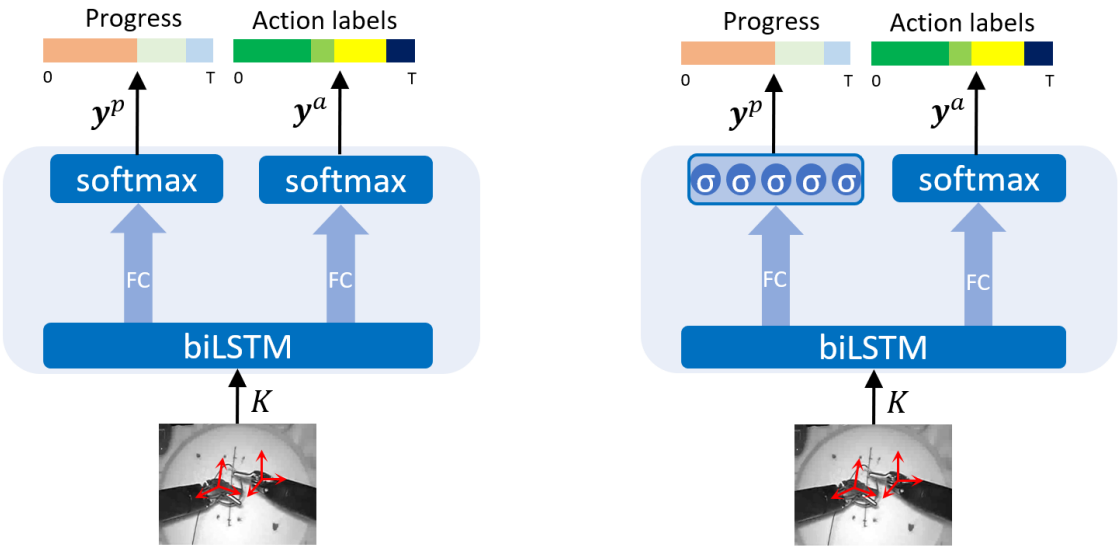

Fig. 4. On the left: multi-task architecture for joint action recognition and progress classification. The LSTM hidden units are connected to 5 logits with softmax activation and MCE loss for progress classification. On the right: multi-task architecture for joint action recognition and ordered progress classification. The LSTM hidden units are connected to 5 logits with sigmoid activation and MBCE loss for ordered progress classification.

weights **w**.

$\bar{g}_{\mathbf{w}}$ is the average gradient norm across all tasks.

$r_i = (L_i/L_i^0)/\bar{L}$ is the relative inverse training rate of task i, with L_i^0 the single-task loss at the first training iteration and $\bar{L}$ the average loss across all tasks.

α is a balancing hyperparameter to be tuned.

C. Evaluation Setup

As in [22], we evaluated our network recognition performance using accuracy, i.e. the percentage of correctly labelled frames, normalized segmental Edit score, which determines the precision of the predicted temporal ordering of actions, and segmental F1@10 score, which penalizes over-segmentation errors but is not sensitive to minor temporal shifts between predictions and ground truth. Progress regression was evaluated with MAE, normalized with respect to the full range of progress values ($\frac{MAE}{4-0} * 100$).

We followed the standard JIGSAWS Leave One User Out (LOUO) cross-validation setup [16]: for every fold, all the trials performed by a single user are kept out as the test set and the other demonstrations are used to train our model.

III. Experiments and Results

We used the open source TensorFlow implementation of the Bidirectional LSTM presented in [12] as our baseline (A). We also relied on the provided training parameters, since they were carefully tuned on the same dataset. Given the stochastic nature of the optimization process, all experiments were performed three times and results were averaged. All runs were trained on NVIDIA Tesla V100-DGXS GPU, with training time of about 1 hour per run.

Our multi-task network for joint action recognition and progress regression (APr) was learnt on the multi-task loss L using Gradient Descent (GD) with Momentum 0.9, batch size 5 and initial learning rate 0.1. The multi-task architectures for standard progress classification (APc) and ordered progress classification (APoc) were instead trained with GD, batch size 5 and initial learning rate 1.0. We always applied learning rate decay of 0.5 after 80 iterations and stopped the training after 120 iterations. We used gradient clipping to avoid exploding gradients and dropout regularization with dropout rate of 0.5, as for the baseline (A). The single-task loss weights (w_1, w_2) were updated at a learning rate of 0.025 using GD on the regularization loss ($Lgrad$), with α set to 1.5. Testing was performed after 100, 110 and 120 training iterations and results were averaged. We trained all networks on the pre-processed kinematic data with revised annotations.

Comparison between A, APr, APc and APoc is presented in Table I. Multi-task performance is evaluated with ($\alpha = 1.5$) and without (w_1=w_2=1) GradNorm regularization. Scores are reported as mean values across the 8 validation folds and corresponding standard deviations, which are strongly representative of inter-surgeon style variability in the LOUO setup. All three multi-task architectures outperform the single-task baseline on the segmental scores (Edit and F1@10), which seems to confirm the hypothesis that action-based progress estimation could help to learn action sequentiality and to reduce out-of-order predictions and over-segmentation errors. Even if none of the proposed architectures clearly stands out from the others, APoc generates slightly better results, which could be explained by stronger penalization of major ordering mistakes than standard classification, and easier optimization goal than regression of a discontinuous progress function. The architecture that benefits the most from multi-task gradient normalization is APr, as it is perhaps more challenging to balance two

TABLE I

GESTURE RECOGNITION (A) AND PROGRESS ESTIMATION (P) PERFORMANCE. SCORES ARE REPRESENTED AS MEAN(STD).

	Scores	A	APr		APc		APoc		Pr	Pc	Poc
			w_1=w_2=1	α=1.5	w_1=w_2=1	α=1.5	w_1=w_2=1	α=1.5			
A	**Accuracy**	85.3(5.8)	85.1(6.6)	85.2(6.3)	85.7(5.7)	85.8(5.6)	85.5(5.8)	**86.0(5.4)**	-	-	-
	Edit	83.1(7.4)	84.2(6.7)	85.9(6.4)	85.4(5.7)	85.7(5.7)	**86.2(6.3)**	86.1(6.2)	-	-	-
	F1@10	88.5(5.7)	89.0(5.5)	90.1(5.4)	90.1(4.9)	90.2(4.9)	90.5(5.0)	**90.7(4.8)**	-	-	-
P	**MAE**	-	5.3(1.5)	**5.1(1.5)**	-	-	-	-	6.2(1.1)	-	-
	Accuracy	-	-	-	89.0(2.7)	89.1(2.8)	87.9(3.5)	88.4(3.1)	-	**89.2(2.8)**	87.0(4.0)
	Edit	-	-	-	**89.3(6.7)**	89.2(6.6)	83.7(4.8)	83.6(4.5)	-	87.8(7.8)	87.3(5.7)
	F1@10	-	-	-	93.2(4.4)	**93.2(3.4)**	90.0(3.1)	90.0(3.1)	-	91.9(4.9)	91.9(3.6)

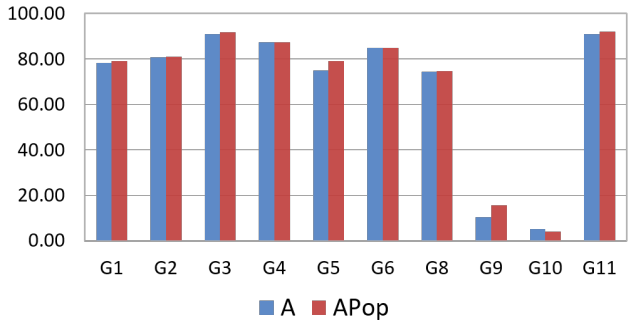

Fig. 5. Recognition accuracy [%] of individual gestures.

different loss functions (MCE for classification and MSE for regression) than two similar or identical ones. However, balanced multi-task networks rely on a large number of hyperparameters, including optimization parameters for the regularization loss. We believe that results could be improved and differences between the three proposed architectures could be emphasized with more extensive parameter tuning, as well as with larger datasets.

Fig. 5 shows recognition accuracies of individual gestures from A and APoc. APoc consistently matches or outperforms A, even if improvement is only marginal. Results in Table I, however, showed that the advantage of the proposed method relies in the regularization of the predicted sequences, which mainly affects the segmental scores and only marginally the framewise evaluation metrics. Some gestures, such as G9 and G10, are extremely challenging to recognize in both cases, as they are under-represented in the dataset.

In addition to recognising surgical gestures, our multi-task architectures segment the surgical demonstrations into 5 fundamental progressive steps of the suturing task, reaching an average accuracy of 89.1% with standard classification (Table I). For APr and APc, but not for APoc, all evaluation scores improved with respect to their single-task counterparts Pr and Pc[1]: not only higher-level progress understanding can help gesture recognition, but gesture recognition can reciprocally boost progress prediction.

[1]Pr, Pc and Poc were trained once with the same hyperparameters as their multi-task counterpart. Weight decay was however anticipated and training was stopped after 80 iterations.

TABLE II

COMPARISON WITH RELATED WORK ON ROBOT KINEMATICS.

	Accuracy	Edit
TCN[11]	79.6	85.8
TCN+RL[23]	82.1	**87.9**
BiLSTM[12]	83.3	81.1
APoc	85.3	84.5
APc	**85.5**	85.3

Fig. 6 illustrates an example of recognition output where predictions generated by the multi-task network show reduced over-segmentation with respect to the baseline, as quantified by the segmental score improvement previously reported. It is also interesting to visualize the relationship between gesture and progress predictions, as the segmentations boundaries are frequently aligned (Fig. 7), and poor progress estimation often corresponds to poor gesture recognition, and vice versa (Fig. 8).

We also trained APc and APoc with the original annotations of JIGSAWS, in order to compare our multi-task models to the original single-task baseline [12] and to related work on robot kinematics. Our investigation, however, was carried out on a simple LSTM architecture, and we suggest the proposed multi-task approach could be applied on top of more complex architectures to boost performance. Results in Table II highlight sensitivity of our models to action annotation noise, which partially spoils the automatic generation of progress labels. This results in performance degradation with respect to the previous experiments, especially for APoc. Nonetheless, the proposed networks significantly outperform [12] both in accuracy and Edit score, and reach competitive performance with respect to related work on robot kinematics.

Finally, we substituted the Bidirectional LSTM cell in APc with a Forward LSTM cell for online recognition. We reached accuracy and Edit score of 82.2 and 76.2 respectively, improving upon the original single-task baseline [12] (Table III).

Our results support the hypothesis that joint surgical gesture recognition and progress estimation can induce more robust feature learning than gesture recognition alone, and boost performance in both online and offline applications.

TABLE III

ONLINE GESTURE RECOGNITION PERFORMANCE.

	Accuracy	Edit
LSTM[12]	80.5	75.3
APc	**82.2**	**76.2**

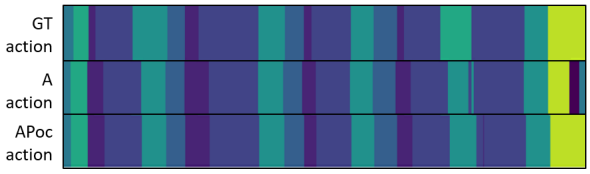

Fig. 6. Example of recognition output where predictions generated by the multi-task network (APoc) show reduced over-segmentation with respect to the baseline (A), as quantified by improved segmental scores. The ground truth segmentation (GT) is shown at the top.

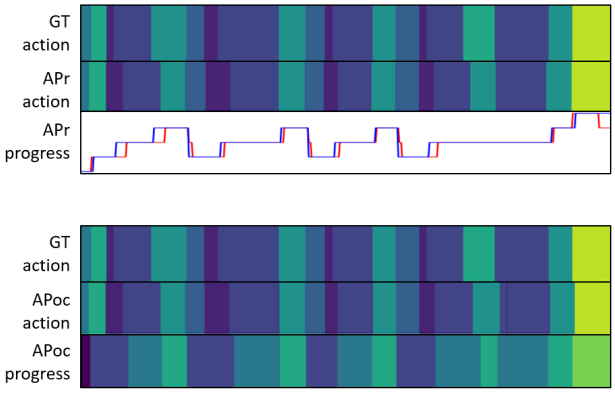

Fig. 7. Comparison between action and progress predictions (APr progress prediction in red, progress ground truth in blue). The predicted segmentation boundaries are frequently aligned.

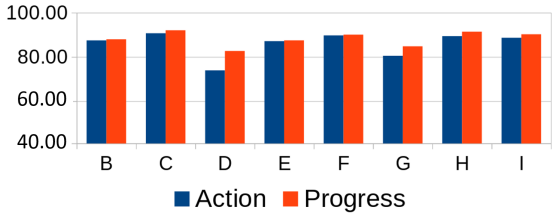

Fig. 8. Action and progress prediction accuracy [%] from APoc for each cross-validation fold (B, C, D, E, F, G, H, I). Poor progress estimation often corresponds to poor gesture recognition, and vice versa.

IV. CONCLUSIONS

In this paper, we performed joint recognition of surgical gestures and progress prediction from robot kinematic data. Differently from prior work, the progress labels were defined on the underlying action sequence rather than on time, in order to reduce kinematic feature variation for equal progress values. Moreover, adjustment gestures did not contribute to the progress advancement. We assumed that action-based progress prediction could help to recognize surgical gestures in well-structured tasks such as suturing and knot tying, which are generally performed several times during surgical interventions. We analysed different progress estimation strategies, and demonstrated on the suturing demonstrations of the JIGSAWS dataset that the proposed multi-task networks outperform the single-task baseline in terms of Edit score and F1@10 score, indicating a reduction in out-of-order predictions and over-segmentation errors. Since action-based progress does not depend on time nor on adjustment gestures, we conjecture this approach could also be effective beyond JIGSAWS in unconstrained environments, such as real surgical interventions or free surgical training sessions, where demonstrations do not have standardized length, right and left hands are often used interchangeably, and adjustment gestures, pauses and undefined motions are more frequent. In this scenario, contextualization of surgical motion into high-level progress stages could help to better recognize the surgical actions. The limitation of this method, however, is in the recognition of unstructured tasks such as blunt dissection, where action-based progress can not be clearly defined. In the presence of frequent and scattered mid-task failures and restarting, the ordered classification method might also lose its advantage to the standard classification method.

As suggested in [14], further investigation could be performed on alternative multi-task integration modalities, such as pre-training on the auxiliary task for feature extraction or fine-tuning on the target task. This might potentially match or even improve upon multi-task training, at the cost of additional training time. Another study could model the progress in time of the individual gestures, which could improve understanding of gesture evolution and duration. Moreover, the integration of visual features extracted from surgical videos could boost both action recognition and progress estimation, as video data encode complementary information about the surgical tools and the state of the environment.

Finally, evaluation of the proposed methodology was performed on the JIGSAWS dataset, which is currently the only publicly available dataset for surgical gesture recognition featuring robot kinematics. However, JIGSAWS is small and contains a limited range of surgical motions. New surgical data will be collected in the future, and extensive evaluation will be carried out on larger datasets of robotic surgical demonstrations.

APPENDIX

Amendments to the original annotations of JIGSAWS:

Demonstration	Start	End	Label
Suturing_B004	2650	2860	G3
Suturing_C002	1596	1685	G4
Suturing_D003	1013	1250	G9
Suturing_D003	1251	1339	G4
Suturing_D004	0099	0166	G5
Suturing_D004	0167	0275	G8
Suturing_D004	0956	1020	G4
Suturing_E003	1095	1267	G4
Suturing_F001	2401	2498	G6
Suturing_G001	1132	1353	G6
Suturing_G001	7628	8181	G8
Suturing_I003	0800	1250	G3

REFERENCES

[1] L. Maier-Hein, S. S. Vedula, S. Speidel, N. Navab, R. Kikinis, A. Park, M. Eisenmann, H. Feußner, G. Forestier, S. Giannarou, M. Hashizume, D. Katic, H. Kenngott, M. Kranzfelder, A. Malpani, K. März, T. Neumuth, N. Padoy, C. M. Pugh, N. Schoch, D. Stoyanov, R. H. Taylor, M. Wagner, G. D. Hager, and P. Jannin, "Surgical data science for next-generation interventions," *Nature Biomedical Engineering*, vol. 1, pp. 691–696, 2017.

[2] S. S. Vedula, A. O. Malpani, L. Tao, G. Chen, Y. Gao, P. Poddar, N. Ahmidi, C. Paxton, R. Vidal, S. Khudanpur, G. D. Hager, and C. C. G. Chen, "Analysis of the Structure of Surgical Activity for a Suturing and Knot-Tying Task," *PLOS ONE*, vol. 11, 2016.

[3] C. E. Reiley and G. D. Hager, "Task versus Subtask Surgical Skill Evaluation of Robotic Minimally Invasive Surgery," *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 435–442, 2009.

[4] C. E. Reiley, E. Plaku, and G. D. Hager, "Motion generation of robotic surgical tasks: Learning from expert demonstrations," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 967–970, 2010.

[5] A. Murali, S. Sen, B. Kehoe, A. Garg, S. McFarland, S. Patil, W. D. Boyd, S. Lim, P. Abbeel, and K. Goldberg, "Learning by observation for surgical subtasks: Multilateral cutting of 3D viscoelastic and 2D Orthotropic Tissue Phantoms," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1202–1209, 2015.

[6] D. Nagy, "A DVRK-based Framework for Surgical Subtask Automation," *Acta Polytechnica Hungarica*, vol. 16, no. 8, pp. 61–78, 2019.

[7] C. G. Cao, C. L. MacKenzie, and S. Payandeh, "Task and motion analyses in endoscopic surgery," in *American Society of Mechanical Engineers, Dynamic Systems and Control Division DSC*, 1996.

[8] N. Ahmidi, L. Tao, S. Sefati, Y. Gao, C. Lea, B. B. Haro, L. Zappella, S. Khudanpur, R. Vidal, and G. D. Hager, "A Dataset and Benchmarks for Segmentation and Recognition of Gestures in Robotic Surgery," *IEEE Transactions on Biomedical Engineering*, 2017.

[9] B. Varadarajan, C. Reiley, H. Lin, S. Khudanpur, and G. Hager, "Data-Derived Models for Segmentation with Application to Surgical Assessment and Training," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 426–434, 2009.

[10] L. Tao, E. Elhamifar, S. Khudanpur, G. D. Hager, and R. Vidal, "Sparse hidden Markov models for surgical gesture classification and skill evaluation," *International conference on information processing in computer-assisted interventions*, vol. 7330 LNCS, pp. 167–177, 2012.

[11] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks: A unified approach to action segmentation," *European Conference on Computer Vision (ECCV)*, vol. 9915 LNCS, pp. 47–54, 2016.

[12] R. DiPietro, C. Lea, A. Malpani, N. Ahmidi, S. S. Vedula, G. I. Lee, M. R. Lee, and G. D. Hager, "Recognizing Surgical Activities with Recurrent Neural Networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 551–558, 2016.

[13] S. Ruder, "An Overview of Multi-Task Learning in Deep Neural Networks," *arXiv preprint arXiv:1706.05098*, 2017.

[14] G. Yengera, D. Mutter, J. Marescaux, and N. Padoy, "Less is More: Surgical Phase Recognition with Less Annotations through Self-Supervised Pre-training of CNN-LSTM Networks," *arXiv preprint arXiv:1805.08569*, 2018.

[15] X. Li, R. S. Burd, Y. Zhang, J. Zhang, M. Zhou, S. Chen, Y. Gu, Y. Chen, I. Marsic, and R. A. Farneth, "Progress Estimation and Phase Detection for Sequential Processes," *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, pp. 1–20, 2017.

[16] Y. Gao, S. S. Vedula, C. E. Reiley, N. Ahmidi, B. Varadarajan, H. C. Lin, L. Tao, L. Zappella, B. Béjar, D. D. Yuh, C. C. G. Chen, R. Vidal, S. Khudanpur, and G. D. Hager, "JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS): A Surgical Activity Dataset for Human Motion Modeling," *Modeling and Monitoring of Computer Assisted Interventions (M2CAI) - MICCAI Workshop*, 2014.

[17] G. S. Guthart and J. K. S. Jr, "The Intuitive TM telesurgery system: overview and application," *Robotics and Automation*, vol. 1, no. April, pp. 618–621, 2000.

[18] S. Sefati, N. J. Cowan, and R. Vidal, "Learning Shared, Discriminative Dictionaries for Surgical Gesture Segmentation and Classification," *Modeling and Monitoring of Computer Assisted Interventions (M2CAI) - MICCAI Workshop*, pp. 1–10, 2015.

[19] L. Tao, L. Zappella, G. D. Hager, and R. Vidal, "Surgical gesture segmentation and recognition," *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, vol. 8151 LNCS, no. PART 3, pp. 339–346, 2013.

[20] C. Lea, G. D. Hager, and R. Vidal, "An Improved Model for Segmentation and Recognition of Fine-Grained Activities with Application to Surgical Training Tasks," *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1123–1129, 2015.

[21] E. Mavroudi, D. Bhaskara, S. Sefati, H. Ali, and R. Vidal, "End-to-end fine-grained action segmentation and recognition using conditional random field models and discriminative sparse coding," *IEEE Winter Conference on Applications of Computer Vision (WACV)*, vol. 2018-Janua, pp. 1558–1567, 2018.

[22] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2017-Janua, pp. 1003–1012, 2017.

[23] D. Liu and T. Jiang, "Deep Reinforcement Learning for Surgical Gesture Segmentation and Classification," *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, vol. 11073 LNCS, pp. 247–255, 2018.

[24] S. Krishnan, A. Garg, S. Patil, C. Lea, G. D. Hager, P. Abbeel, and K. Y. Goldberg, "Transition state clustering: Unsupervised surgical trajectory segmentation for robot learning," *The International Journal of Robotics Research*, vol. 36, pp. 1595–1618, 2015.

[25] M. J. Fard, S. Ameri, R. B. Chinnam, and R. D. Ellis, "Soft Boundary Approach for Unsupervised Gesture Segmentation in Robotic-Assisted Surgery," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 171–178, 2017.

[26] B. van Amsterdam, H. Nakawala, E. D. Momi, and D. Stoyanov, "Weakly Supervised Recognition of Surgical Gestures," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9565–9571, IEEE, IEEE, 2019.

[27] C. Lea, A. Reiter, R. Vidal, and G. D. Hager, "Segmental spatiotemporal CNNs for fine-grained action segmentation," *European Conference on Computer Vision (ECCV)*, vol. 9907 LNCS, pp. 36–52, 2016.

[28] I. Funke, S. Bodenstedt, F. Oehme, F. von Bechtolsheim, J. Weitz, and S. Speidel, "Using 3D Convolutional Neural Networks to Learn Spatiotemporal Features for Automatic Surgical Gesture Recognition in Video," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2019.

[29] P. Lei and S. Todorovic, "Temporal Deformable Residual Networks for Action Segmentation in Videos," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6742–6751, 2018.

[30] J. Wang, Z. Du, A. Li, and Y. Wang, "Atrous temporal convolutional network for video action segmentation," *IEEE International Conference on Image Processing (ICIP)*, pp. 1585–1589, 2019.

[31] L. Ding and C. Xu, "TricorNet: A Hybrid Temporal Convolutional and Recurrent Network for Video Action Segmentation," *arXiv preprint arXiv:1705.07818*, 2017.

[32] D. Sarikaya, K. A. Guru, and J. J. Corso, "Joint Surgical Gesture and Task Classification with Multi-Task and Multimodal Learning," *arXiv preprint arXiv:1805.00721*, 2018.

[33] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. De Mathelin, and N. Padoy, "EndoNet: A Deep Architecture for Recognition Tasks on Laparoscopic Videos," *IEEE Transactions on Medical Imaging*, vol. 36, no. 1, pp. 86–97, 2017.

[34] S. Bodenstedt, M. Wagner, D. Katić, P. Mietkowski, B. Mayer, H. Kenngott, B. Müller-Stich, R. Dillmann, and S. Speidel, "Unsupervised temporal context learning using convolutional neural networks for laparoscopic workflow analysis," *arXiv preprint arXiv:1702.03684*, vol. abs/1702.0, 2017.

[35] F. Despinoy, D. Bouget, G. Forestier, C. Penet, N. Zemiti, P. Poignet, and P. Jannin, "Unsupervised Trajectory Segmentation for Surgical Gesture Recognition in Robotic Training," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 6, pp. 1280–1291, 2016.

[36] J. Cheng, Z. Wang, and G. Pollastri, "A neural network approach to ordinal regression," in *IEEE International Joint Conference on Neural Networks*, pp. 1279–1284, IEEE, 2008.

[37] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks," in *International Conference on Machine Learning (ICML)*, 2017.