A Deep Learning Approach to Intrinsic Force Sensing on the da Vinci Surgical Robot

Nam Tran[1,2], Jie Ying Wu[1], Anton Deguet[1], Peter Kazanzides[1]

***Abstract*— In robot-assisted minimally-invasive surgery (RAMIS), force estimation remains a challenging issue. We seek to estimate external forces based on available measurements from the joint encoders and motor currents. To this end, we propose a deep learning approach for end-to-end force estimation on the da Vinci Surgical System that is trained using data collected by both moving an instrument in free space and by palpating a tissue phantom that has an embedded force sensor for ground truth. The trained neural network provides reasonable force estimates (within about 1N to 2N precision given a full range of 10N) and is generalizable to other regions of the robot workspace. We further show that our proposed system can provide useful haptic feedback in a pilot study to differentiate stiffness in various tissue phantoms.**

***Index Terms*— deep learning, neural networks, force estimation, surgical robotics, dVRK**

I. Introduction

Proven to reduce post-operation trauma as well as other risks, robot-assisted minimally invasive surgery (RAMIS) has become a standard and well-established procedure. In RAMIS, organs are manipulated via robotic instruments inserted through small incisions, while an endoscopic camera provides operators with live video. Real-time force sensing is a desired capability in RAMIS to enable haptic feedback for surgeons, giving the useful sense of touch that will help to further decrease trauma during and after the operation. However, integrating force sensing for haptic feedback in surgical robotic systems remains an open problem [1]. Attaching a force sensor is challenging due to space, sterilizability, and cost constraints.

This work presents a deep neural network to estimate interaction forces, without a force sensor, on the da Vinci Research Kit (dVRK) [2], which is an open-source platform based on the first generation da Vinci® Surgical System (Intuitive Surgical Inc., Sunnyvale, CA). The da Vinci testbench used in our system has two components: a surgeon's console and a patient side console. The surgeon's console is a Master Tool Manipulator (MTM), providing 7 DOF for dexterous and natural tool manipulation, with an additional DOF for gripper control. On the patient side, there is a Patient Side Manipulator (PSM), controlled by the surgeon through the MTM. The PSM does not sense interaction forces, but our deep learning model provides end-to-end force estimation by receiving joint velocities and joint torques as inputs and yielding Cartesian force in the X, Y and Z directions as outputs. The estimated force from the PSM can then be used to provide haptic feedback to the surgeon.

Internal dynamics of a surgical robot can be difficult to model. Friction, wear and tear as well as other disturbances within the internal structure of a robot are highly nonlinear. On the other hand, a neural network can be easily retrained with a force sensor and teleoperation data gathered from a new surgical robot.

We perform both offline and online evaluation for our neural network, comparing the neural network output to forces measured by an external force sensor. We also carry out a pilot study to evaluate whether the estimated force can help operators to distinguish between phantom structures emulating human tissues of different stiffness.

II. Related Work

Previous works to estimate forces have been based on either intrinsic feedback (positions, velocities, torques) or extrinsic (external sensor) feedback, such as the endoscope images, with approaches that are model-based, data-driven or a combination of the two.

Our proposed method uses intrinsic feedback and is completely data-driven. A similar approach was used by Abeywardena et al. [3] but focused only on the estimation of the grip force in a da Vinci instrument. Recently, Yilmaz et al. [4] presented a neural network to estimate the internal (dynamics) torque of the da Vinci; the internal torque was then subtracted from the measured torque (based on the motor currents) to estimate the external torque. Thus, this approach can be considered as a combination of a data-driven method and a model-based method. Instead of using deep learning, some researchers use a combination of machine learning and simulation/modeling, making use of a torque observer [5] and neuro-evolutionary fuzzy systems [6].

Other methods use intrinsic feedback in model-based approaches, including modeling of the Lagrangian dynamics of the robot [7], changes in cable tension [8] and/or friction in the transmission [9]. These approaches suffer from inaccuracy in the estimation of the model parameters, as well as from unmodeled effects.

Some researchers adopt external feedback by using a miniature force sensor to provide force estimation [10], [11], [12]. These implementations require either creating a customized tooltip or attaching the force sensor on an existing tooltip/camera. This adds another layer of complexity to the system and raises questions about cost, sterilizability and compatibility with other medical devices [13]. Other

*This work was supported by an REU supplement to NSF NRI 1637789.

[1]Dept. of Computer Science, Johns Hopkins University, Baltimore, MD, {jieying, anton.deguet, pkaz}@jhu.edu

[2]Dept. of Computer Science & Engineering, Santa Clara University, Santa Clara, CA, nvtran@scu.edu

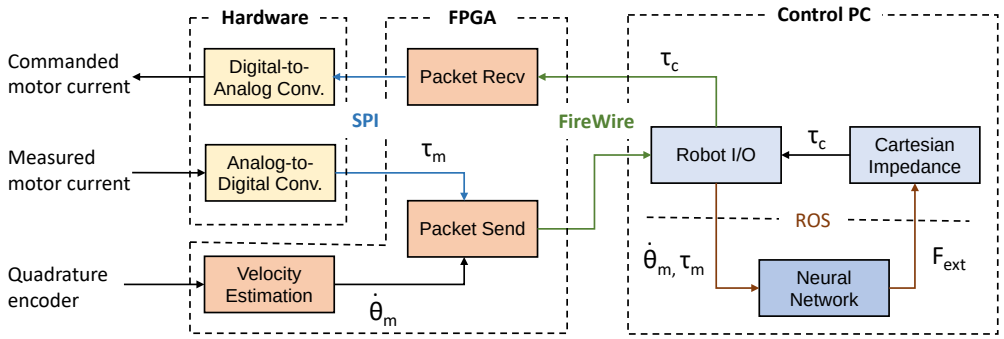

Fig. 1 Block diagram of dVRK with neural network for external force estimation and haptic feedback via Cartesian impedance controller

approaches based on external feedback involve extracting visual cues from tissue deformation and using computer vision to estimate forces [14], [15], [16], [17], [18], [19]. These demonstrate high accuracy in their setup and show the potential of neural networks and computer vision algorithms to extract force data; however, since they rely on visual features, they may not generalize well to new tissues with different appearances. Also, as many of these implementations are only shown to estimate force with a single component (usually Z-axis), they may not be able to provide realistic multidimensional haptic feedback for dexterous and natural palpation movements.

Our contributions are as follows: (1) a purely data-driven end-to-end force estimation algorithm, based on deep learning, that has the potential to generalize across tissues of different stiffnesses, workspace configurations, and users' behavior, and (2) a haptic feedback system integrated into the dVRK infrastructure that has been validated by a pilot study. Our method uses signals already measured by the system and does not require additional sensors at runtime (an external force sensor is required when collecting training data).

III. Methodology

A. Approach

In this work, we investigate an end-to-end approach, where the neural network estimates the external forces given the internal position/velocity and torque measurements. The neural network is trained by teleoperating the robotic instrument in both free space and in contact with a phantom. A force sensor placed under the phantom provides the ground truth for training. Although using the measured joint position can enable the network to better learn the robot dynamics, such an approach may lead the neural network to learn the position of the phantom. To prevent this, we do not pass the position of the robot to the neural network, and use only the joint velocities and torques, as shown in Fig. 1 and in Eq. 1:

$$F_{ext} = \text{NN}\left(\dot{\theta}_m, \tau_m\right) \tag{1}$$

In our preliminary testing, this worked about as well as an implementation that also used the measured joint positions, which indicates that joint positions have a small effect on the robot dynamics. For instance, the patient side manipulator (PSM) of the da Vinci contains mechanical counterbalances and therefore the effect of gravity is minimal. If necessary, a gravity correction term can be added to improve the force estimation.

B. Deep Learning Model

We use the Keras library to implement a fully-connected feedforward neural network, with Tensorflow as backend [20], [21]. We started from a single layer network and empirically increased the network complexity and searched for the optimal hyperparameters. The resulting network has 5 hidden layers and each hidden layer has 250 neurons. Fig. 3 shows the complete data flow of the neural network. The input dimension is 1200. The output dimension is 3, as the ground truth Cartesian force has 3 components: X, Y and Z. We use Leaky ReLU as the intermediate activation function. The output layer uses the sigmoid activation function and the values are scaled to the measured minimum and maximum detected forces in the training data. We initialize our weights using the `He uniform` initializer. Additionally, we choose SGD over Adam, as it provides more effective generalization in our tests. The learning parameters are set as follows: learning rate: 0.01, decay: 10^{-6} and momentum: 0.9. Nesterov's accelerated gradient descent is used for faster convergence. The loss function is set to MAE (mean absolute error), as when the network is used to provide haptic feedback, transient and large-magnitude errors are not as noticeable as long-lasting and small-magnitude errors.

C. Setup

Fig. 2 shows the complete setup in horizontal configuration for both contact and non-contact scenarios as well as the vertical configuration. The horizontal configuration is used for both training and evaluation. However, the vertical configuration is used only for evaluation. We use a Gamma Force/Torque sensor (ATI Industrial Automation, Apex, NC) to measure the Cartesian force acting on the phantom. The force sensor's sampling rate is set at 1,000 Hz. The phantom is placed above the force sensor, secured by a large binder clip, and its orientation is manually aligned with the robot world coordinate system.

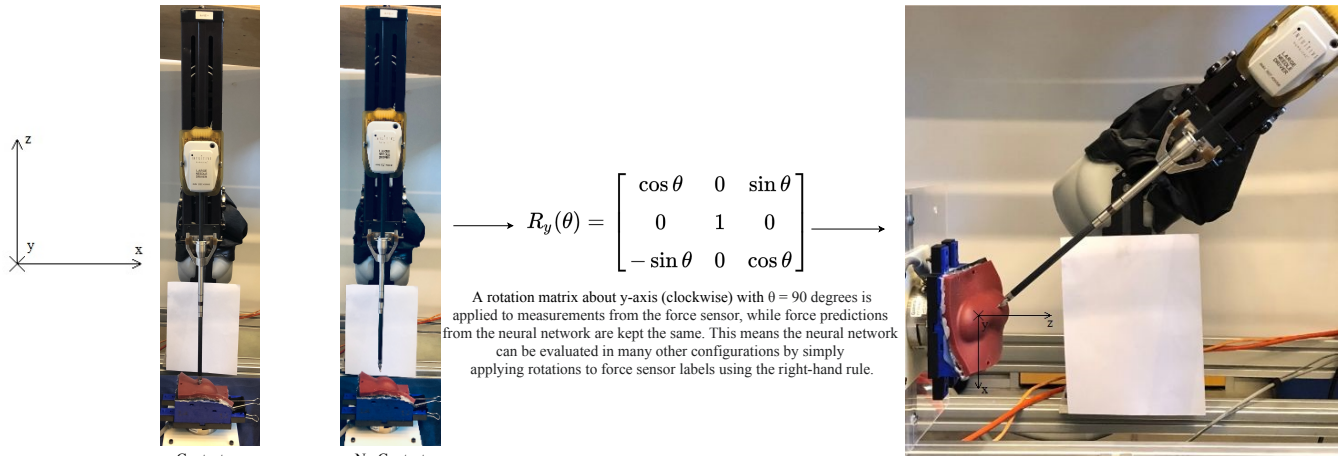

Fig. 2 dVRK testbench is shown with da Vinci robot arm in both horizontal and vertical configurations. The phantom is always placed directly on the ATI network force/torque sensor, all secured by a large binder clip.

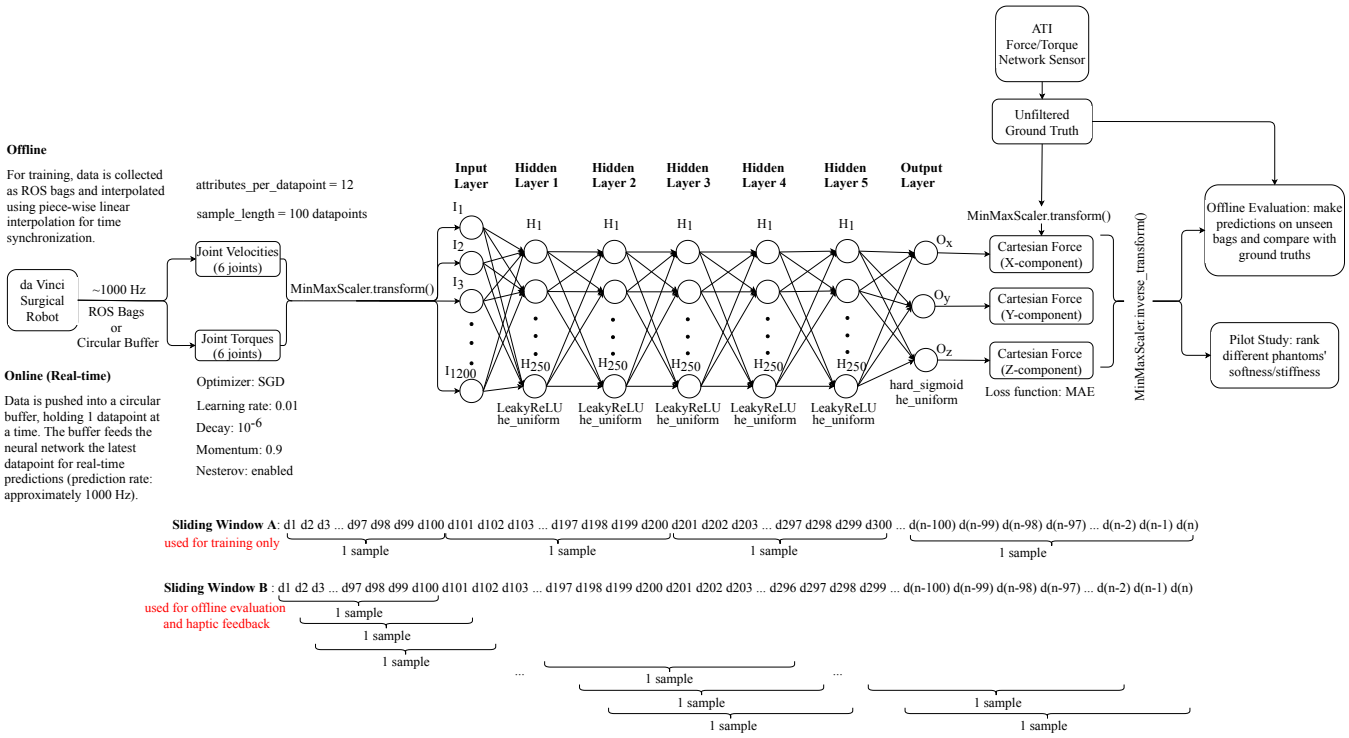

Fig. 3 Overview of dataflow, neural network architecture and sliding window methods. Sliding window method A (shifting by 100 raw datapoints) is used for training only, while method B (shifting by 1 raw datapoint) is used for both offline and online evaluation.

D. Data Collection

We teleoperate the PSM and smoothly palpate the phantom at different intervals, imitating realistic movements in a surgery. We also include spontaneous movements with varying velocities and contact forces. As shown in Fig. 5, we try to have a balanced mix of no-contact (free space movement without any touching by the robot arm) and with-contact (smooth pokes at various angles and intervals) data. When gathering no-contact data, we put the PSM in various positions, including near joint limits. All this ensures a diverse range of data for effective learning, helping the neural network to discern contact from no-contact and to estimate force accurately at as many PSM positions in the workplace as possible. The robot gripper is not used and kept closed during teleoperation, as we are more interested in the palpation force of the robotic arm with respect to the phantom. We capture the data through Rosbags [22].

Within each bag, there is a timestamp for each data value.

Fig. 4 dVRK testbench horizontal configuration for pilot study. A small round marker is attached to the tip to prevent damage to soft phantoms. Also depicted are the phantoms used in the pilot study, with increasing order of stiffness from left to right. The neural network is not trained on them. They are set to equal height and covered with a drape during the pilot study.

The da Vinci's sampling rate is set to 1,000 Hz. For this work, we use only the robot's joint velocity and joint torque data. Velocities are estimated from the encoders using the algorithm in [23] and torques are calculated using motor current measurements.

Table I summarizes the dataset sizes used in this paper and the approximate corresponding duration given the sampling rate of 1000 Hz. The training and the validation sets have a standard ratio of 80 to 20. The validation set is not trained on, but is used after every epoch during training to minimize overfitting. Both offline evaluation sets have approximately the same size for ease of comparison and analysis. As evidenced by the duration, many ROS bags are collected over the course of several weeks in order to form such large datasets.

E. Data Processing

Although both the da Vinci and the force sensor's sampling rates are set to 1,000 Hz, the topics are not synchronized with each other. As shown in Fig. 3, given that there is a large amount of data generated due to the high sampling rate, we use piece-wise linear interpolation, which is sufficiently precise in aligning data from both sources.

Since we keep the gripper closed, we consider only the velocity and joint torque of the remaining six joints. Fig. 3 also depicts two sliding window strategies. Sliding window method A is only used for training, while method B is employed for both offline and online evaluation. We notice that using method B for training yields poorer predictive performance than using method A. Additionally, method B can capitalize on the high sampling rate of the da Vinci robot, which is particularly necessary during online evaluation in order to provide fast and smooth feedback. For both methods, each raw datapoint has 12 attributes. We group every 100 consecutive raw data points into a sample, yielding 1200 attributes per row for input to the neural network. For ground truth Cartesian force values, for every row of data (sample) previously mentioned, we use the force vector label (containing X, Y and Z components) of the 100^{th} raw datapoint as the target prediction for that entire row. The only difference between the two sliding window strategies is that method A shifts the window by 100 raw datapoints, while method B shifts the window by 1 raw datapoint. Overall, the sliding window mechanism serves to give the neural network a sense of memory, enabling it to take into account backlash and hysteresis to make effective predictions for the current timestep.

TABLE I Summary of datasets. The number of raw datapoints for each set is shown, along with the corresponding duration given the sampling rate of 1000 Hz. H: Horizontal Configuration. V: Vertical Configuration

Set	Size	Duration
Training (H)	17,862,662	5.0 hours
Validation (H)	4,465,655	1.2 hours
Offline Evaluation (H)	359,921	5.9 minutes
Offline Evaluation (V)	345,060	5.8 minutes

IV. Experiments

A. Offline Evaluation

As shown in Fig. 2, we collect data in both horizontal and vertical configurations to evaluate our neural network. Fig. 5 demonstrates our data collection method, in which we ensure a wide range of no-contact and contact scenarios, such as poking with the tip or edge of the gripper. The phantom is the same as the one used in training.

B. Haptic Feedback Pilot Study

For online validation, we conduct a pilot study in horizontal configuration with three scenarios: (1) no force feedback, (2) force feedback from the neural network, and (3) force feedback from the force sensor under the phantom. The participants of this pilot study are the last three authors of this paper. Note that the first and second authors collected all the training data, so the pilot study also tests the neural network's potential to generalize to multiple users. The participants are asked to palpate three phantoms of low, medium and high stiffnesses, shown in Fig. 4. These phantoms are covered under a drape and their order is shuffled before a participant is engaged in a test scenario. Their heights are set to be equal, and decoys under the drape are used to prevent participants

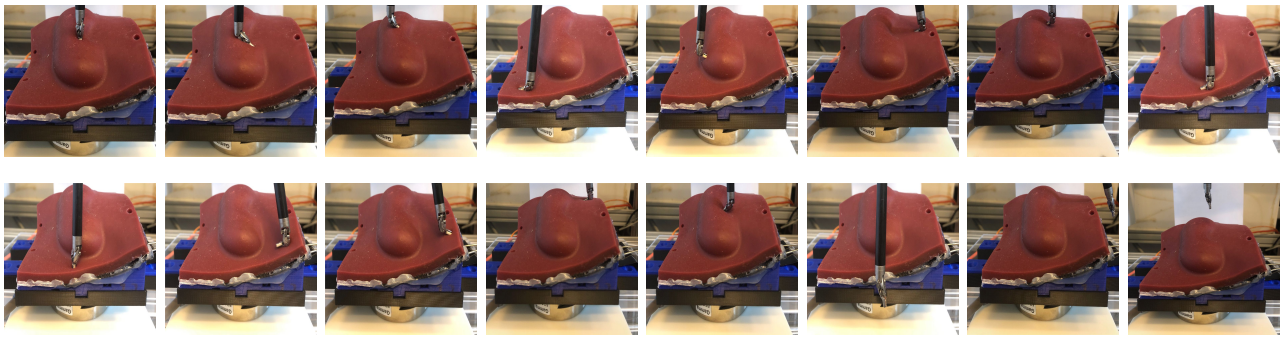

Fig. 5 Various scenarios on the same phantom are shown during data collection as well as offline evaluation. There is neither gripping nor pulling during teleoperation, as we are more interested in the palpation force with respect to the phantom.

from guessing based on visual cues, as shown in Fig. 4. For the no feedback case, the participants teleoperate with no haptic feedback upon contact. For the neural network feedback case, teleoperation occurs with force feedback from the neural network. For the force sensor feedback case, teleoperation occurs with force feedback drawn directly from the force sensor with no modification. Force feedback, which occurs at approximately 1000 Hz, is implemented by reversing the direction of the predicted force (either drawn from the neural network or the force sensor) and then applying it on the MTM.

The participants are aware of the scenario they are currently in (no feedback, neural network feedback or force sensor feedback). In each scenario, while the test phantoms are presented to participants at random, participants are allowed to revisit previous phantoms as much as they need. Before proceeding to the next scenario, participants must give a final ranking of the three phantoms in the current scenario. A participant must be able to rank the stiffness of all 3 phantoms correctly and confidently in order to be given an A (able to discriminate). A participant who is able to do so with low confidence is given a G (correct but uncertain guess). A participant who is not able to rank all three phantoms correctly is given a U (unable to discriminate). Redoing a previously completed scenario is not allowed. The same process is applied for all three scenarios (no feedback, neural network feedback and force sensor feedback). Each participant takes part in the pilot study individually, with no other participants present for all three scenarios.

V. Results

A. *Offline Force Prediction*

Fig. 6 shows offline evaluation results for both horizontal (left) and vertical (right) configurations. For the vertical configuration, as mentioned in Fig. 2, a rotation is applied to the force sensor data so that the readings remain aligned with the robot world coordinate system. The red curves show the error, calculated by subtracting predicted values from measured values. Zoomed insets depicting predicted values (blue) and ground truth values (orange) are used to highlight areas with particularly good and bad alignment, revealing the strengths and weaknesses of our neural network in different intervals.

Table II summarizes the offline evaluation results, showing both Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). We notice that the neural network performs better in the horizontal configuration than the vertical configuration, evidenced by the lower MAEs and RMSEs of plots on the left hand side. As shown on the horizontal configuration plots, the prediction curve (blue) closely embraces the ground truth curve (orange). Minor errors, in which the network incorrectly predicts a force when there is none, are also detected, as can be seen in the Y-component and Z-component plots of the horizontal configuration and the X-component plot of the vertical configuration.

There is also a considerable amount of sudden spikes in the red error curve for both horizontal and vertical configurations. Many of these error spikes do approach the highest magnitudes of force exerted in all three components across horizontal and vertical configurations, as evidenced in both Fig. 6 and Table II. While some of these error spikes are due to the neural network completely missing the estimation, in situations where the neural network obtains a fairly precise estimation of the force, a small phase shift or lag in the prediction curve can also cause high-magnitude error spikes. An example of such a lag can be seen in zoomed inset between 180s to 210s in the horizontal configuration's Y-component plot (Fig. 6). When the configuration is switched to be vertical, the neural network generalizes well in the Y-axis but shows more error in predictions in the X and Z-axes.

The prediction error of the neural network seems to follow a normal distribution. From Table II, we notice that the mean value of prediction error is very close to 0N. Also, the low standard deviation values in Table II are reflected in the bottom figures of Fig. 6 by the concentration of the probability density curves' peaks near 0N. The density of prediction error further away from 0N quickly becomes negligible. Overall, the prediction is smooth and accurate to about 1N for the horizontal configuration and 2N for the vertical configuration, with the maximum magnitude of the

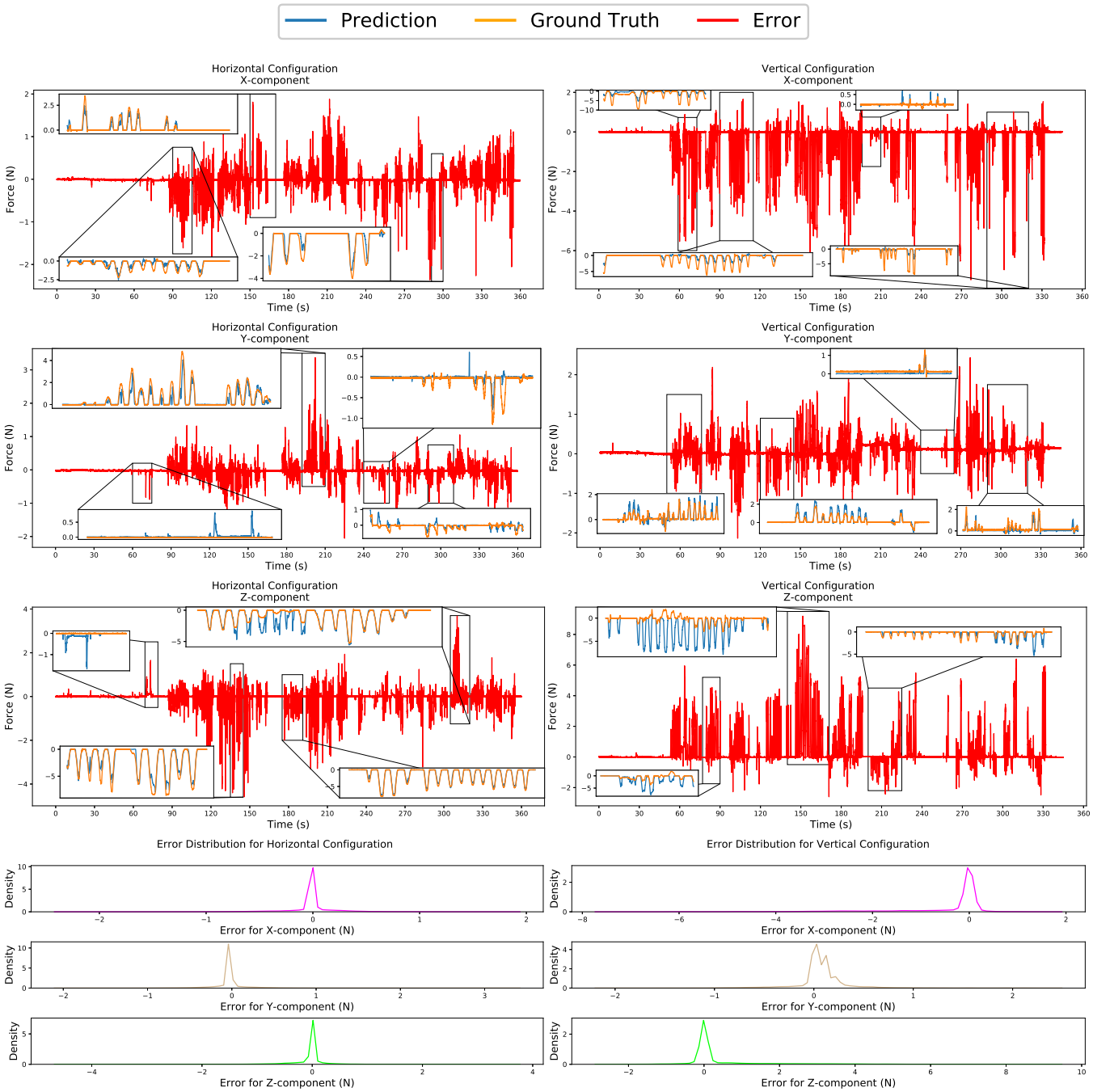

Fig. 6 Offline evaluation results and statistical analysis for horizontal (left) and vertical (right) configurations as depicted in Fig. 2. The results are generated by palpating with different angles, intervals and intensities as previously shown in Fig. 5. The red curve shows the prediction error with respect to ground truth data, with zoomed insets depicting actual predictions (blue curve) versus ground truth values (orange curve) to highlight the neural network's strengths and weaknesses in different intervals. For the top six plots, the Y-axes show force (N) and the X-axes depict time (s). The six plots at the bottom show the error distribution for both horizontal and vertical configurations, normalized so that the probability density integrates to 1 for each curve. The density curve resembles a bell curve, revealing a normal distribution with the prediction error concentrating near 0N for both horizontal and vertical configurations.

TABLE II Summary of key metrics for offline evaluation results (in N). MAE, RMSE, Mean and STDev are calculated on the prediction error. Min and Max refer to minimum and maximum recorded ground truth force in the corresponding configuration.

Configuration	Horizontal		Vertical	
X-component	MAE:	0.12	MAE:	0.50
	RMSE:	0.26	RMSE:	1.20
	Mean:	-0.02	Mean:	-0.47
	STDev:	0.26	STDev:	1.11
	Min:	-5.92	Min:	-9.95
	Max:	4.26	Max:	1.01
Y-component	MAE:	0.11	MAE:	0.16
	RMSE:	0.22	RMSE:	0.28
	Mean:	-0.03	Mean:	0.07
	STDev:	0.22	STDev:	0.27
	Min:	-3.67	Min:	-3.44
	Max:	4.83	Max:	3.15
Z-component	MAE:	0.16	MAE:	0.52
	RMSE:	0.40	RMSE:	1.28
	Mean:	-0.07	Mean:	0.43
	STDev:	0.40	STDev:	1.21
	Min:	-9.00	Min:	-7.02
	Max:	1.31	Max:	2.67

TABLE III Pilot study results. A: able to discriminate all three phantoms correctly and confidently, G: correct but uncertain in discrimination of all three phantoms, U: unable to discriminate all three phantoms

Participant	Test Scenarios		
	No Feedback	*NN Feedback*	*ATI Feedback*
I	U	A	A
II	G	A	A
III	A	A	A

exerted force being approximately 10N.

B. Haptic Feedback Pilot Study

This subsection demonstrates that the neural network's predictions, though prone to errors, are actually usable for palpating and stiffness discrimination in real-time conditions. Table III outlines the pilot study results, showing three scenarios from left to right respectively: no feedback, neural network feedback and force sensor feedback. Though allowed to revisit previous phantoms, the participants are able to finish all their scenarios in about 10 minutes (including the time to rearrange phantoms before each scenario). Participants of the user study are able to rank the stiffnesses of the three phantoms correctly when palpating using force feedback from the neural network. With no force feedback, only one user is able to discriminate correctly and confidently, while the remaining participants are either unable to rank or only able guess correctly with high uncertainty. As expected, in the force sensor feedback case, all users are able to rank the phantoms correctly and confidently. The neural network feedback does not feel as smooth as the feedback from the sensor, yet that does not affect the overall sense of touch and how the users rank the phantoms.

VI. Discussion

We have considered other deep learning approaches, such as recurrent neural networks (RNNs), that yield highly accurate force predictions. However, we decided to use fully-connected feedforward networks because they allow for rapid training and deployment, while still delivering sufficiently robust and usable predictions. Ease of training and deployment are crucial to us as we strive for complete generalizability of force sensing to different phantoms, different instruments and perhaps even different PSMs in our future work. Nevertheless, inspired by the powerful concept of memory from RNNs, we do implement the sliding window mechanism for our neural network to take advantage of the raw inputs' temporal structure.

Our results demonstrate that our neural network is capable of providing useful force feedback without sensors at test time. We choose not to use joint positions in order to avoid the risk that the neural network learns a palpation map that is specific to the training phantom. Through the palpation study, we show that the network has the potential to generalize to new phantoms on which it has not been trained. We do notice that a user's ability to discriminate seems very hampered without feedback. The network's predictions help users feel the stiffness of the phantoms with performance comparable to that obtained with use of a physical force sensor.

Since the neural network has to learn a diverse range of contact scenarios as shown in Fig. 5, it gains some generalizability, with sacrifices in precision. Nevertheless, since RAMIS operations typically involve a wide array of contact situations, it is more imperative that our neural network is able to handle them all within reason. We discovered during the haptic feedback pilot study that consistent magnitude errors do not pose a significant issue. Rather, it is the neural network's flexibility allowing feedback at various contact angles, as depicted in Fig. 5 and Fig. 4, that provides a realistic sense of touch for the participants of the pilot study.

Although our study only analyzes two configurations, it can be argued that the data generated from these two configurations are quite representative of all possible scenarios in our testbench. For example, another potential vertical configuration is the mirror of the current vertical configuration, in which the phantom is rotated 90 degrees anticlockwise from the horizontal configuration. In this case, the same rotation matrix can be applied, but with a different angle, to transform the force sensor labels to the approximate coordinate space of the neural network's sense of direction.

While our network cannot learn the complete dynamics model as the mass/inertia matrix, coriolis/centrifugal forces, and gravity depend on the joint positions, our results suggest that the effect of the da Vinci PSM dynamics is relatively small. Other factors that depend primarily on velocity, such as friction, may have greater influence. Elements such as friction, backlash and hysteresis can be challenging to model accurately, as they are dependent on the individual robot arm. By casting the problem of internal dynamics modeling as a deep learning problem, we are able to reduce the complexity and create a predictive system that is robust to variations.

We use a force sensor mounted under the phantom to approximate the force at the instrument tip. There may be some discrepancy, for example, due to the deformation of the intervening phantom. We are currently working to quantify

this possible discrepancy by using a different robot, with a wrist-mounted force sensor, to palpate the phantom while reading values from both force sensors.

Although our study has demonstrated the viability of an end-to-end neural network for estimating external force, one question remaining is how to implement such a system in practice. In particular, it remains to be seen how well a neural network trained on one PSM and one instrument will work for a different PSM and/or a different instrument, even of the same type. Retraining the neural network in the field, for example, prior to or during a surgery, would be challenging if the training requires a force sensor placed in the environment. One approach is to investigate whether it is possible to fine-tune the model using only free-space manipulations to learn the internal dynamics of a new robot arm and/or instrument.

VII. Conclusion

We have demonstrated that a neural network can provide useful force estimation in both offline and online evaluation. Future work will further evaluate the method's generalizability, such as testing on a different da Vinci arm and instruments of the same type to see whether different amounts of wear and tear would affect accuracy. Also, more tissue handling methods such as pulling, lifting and gripping the tissue can be explored. It would be interesting to do online learning so that the network can adapt to a new phantom as the user is palpating it, thus streamlining the training and validation process. This approach could be the first step in creating a general force estimation method that can be adapted at test time to a variety of robot architectures and procedures.

References

[1] A. Marbán, A. Casals, J. Fernández, and J. Amat, *Haptic Feedback in Surgical Robotics: Still a Challenge*. Cham: Springer International Publishing, 2014, pp. 245–253. [Online]. Available: https://doi.org/10.1007/978-3-319-03413-3_18

[2] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da Vinci® Surgical System," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6434–6439.

[3] S. Abeywardena, Q. Yuan, A. Tzemanaki, E. Psomopoulou, L. Droukas, C. Melhuish, and S. Dogramadzi, "Estimation of tool-tissue forces in robot-assisted minimally invasive surgery using neural networks," *Frontiers in Robotics and AI*, vol. 6, p. 56, 2019. [Online]. Available: https://www.frontiersin.org/article/10.3389/frobt.2019.00056

[4] N. Yilmaz, J. Y. Wu, P. Kazanzides, and U. Tumerdem, "Neural network based inverse dynamics identification and external force estimation on the da Vinci Research Kit," in *Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2020.

[5] A. Colomé, D. Pardo, G. Alenya, and C. Torras, "External force estimation during compliant robot manipulation," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2013, pp. 3535–3540.

[6] A. Mozaffari, S. Behzadipour, and M. Kohani, "Identifying the tool-tissue force in robotic laparoscopic surgery using neuro-evolutionary fuzzy systems and a synchronous self-learning hyper level supervisor," *Applied Soft Computing*, vol. 14, pp. 12 – 30, 2014, special issue on hybrid intelligent methods for health technologies. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494613003177

[7] H. Sang, J. Yun, R. Monfaredi, E. Wilson, H. Fooladi, and K. Cleary, "External force estimation and implementation in robotically assisted minimally invasive surgery," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 13, no. 2, p. e1824, 2017.

[8] L. Yu, W. Wang, and F. Zhang, "External force sensing based on cable tension changes in minimally invasive surgical micromanipulators," *IEEE Access*, vol. 6, pp. 5362–5373, 2018.

[9] M. Mahvash and A. Okamura, "Friction compensation for enhancing transparency of a teleoperator with compliant transmission," *IEEE Trans. on Robotics*, vol. 23, no. 6, pp. 1240–1246, Dec. 2007.

[10] P. Puangmali, H. Liu, L. D. Seneviratne, P. Dasgupta, and K. Althoefer, "Miniature 3-axis distal force sensor for minimally invasive surgical palpation," *Ieee/Asme Transactions On Mechatronics*, vol. 17, no. 4, pp. 646–656, 2011.

[11] A. Faragasso, J. Bimbo, Y. Noh, A. Jiang, S. Sareh, H. Liu, T. Nanayakkara, H. A. Wurdemann, and K. Althoefer, "Novel uniaxial force sensor based on visual information for minimally invasive surgery," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1405–1410.

[12] M. C. Yip, S. G. Yuen, and R. D. Howe, "A robust uniaxial force sensor for minimally invasive surgery," *IEEE transactions on biomedical engineering*, vol. 57, no. 5, pp. 1008–1011, 2010.

[13] P. Puangmali, K. Althoefer, L. D. Seneviratne, D. Murphy, and P. Dasgupta, "State-of-the-art in force and tactile sensing for minimally invasive surgery," *IEEE Sensors Journal*, vol. 8, no. 4, pp. 371–381, 2008.

[14] C. Gao, X. Liu, M. Peven, M. Unberath, and A. Reiter, "Learning to see forces: Surgical force prediction with RGB-point cloud temporal convolutional networks," in *MICCAI Workshop on Computer Assisted and Robotic Endoscopy (CARE)*, Sept. 2018, pp. 118–127.

[15] A. I. Aviles, A. Marban, P. Sobrevilla, J. Fernandez, and A. Casals, "A recurrent neural network approach for 3d vision-based force estimation," in *2014 4th International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, 2014, pp. 1–6.

[16] A. I. Aviles, S. M. Alsaleh, P. Sobrevilla, and A. Casals, "Force-feedback sensory substitution using supervised recurrent learning for robotic-assisted surgery," in *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2015, pp. 1–4.

[17] A. I. Aviles, S. M. Alsaleh, J. K. Hahn, and A. Casals, "Towards retrieving force feedback in robotic-assisted surgery: A supervised neuro-recurrent-vision approach," *IEEE Transactions on Haptics*, vol. 10, no. 3, pp. 431–443, 2016.

[18] N. Gessert, J. Beringhoff, C. Otte, and A. Schlaefer, "Force estimation from OCT volumes using 3D CNNs," *CoRR*, vol. abs/1804.10002, 2018. [Online]. Available: http://arxiv.org/abs/1804.10002

[19] E. Noohi, S. Parastegari, and M. Žefran, "Using monocular images to estimate interaction forces during minimally invasive surgery," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 4297–4302.

[20] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[21] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[22] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*. Kobe, Japan, 2009.

[23] J. Y. Wu, Z. Chen, A. Deguet, and P. Kazanzides, "FPGA-based velocity estimation for control of robots with low-resolution encoders," in *Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2018, pp. 6384–6389.