DESK: A Robotic Activity Dataset for Dexterous Surgical Skills Transfer to Medical Robots

Naveen Madapana[1*], Md Masudur Rahman[2*], Natalia Sanchez-Tamayo[1*], Mythra V. Balakuntala[3], Glebys Gonzalez[1], Jyothsna Padmakumar Bindu[3], L. N. Vishnunandan Venkatesh[3], Xingguang Zhang[1], Juan Barragan Noguera[1], Thomas Low [4], Richard M. Voyles[3], Yexiang Xue[2], and Juan Wachs[1]

Abstract— **Datasets are an essential component for training effective machine learning models. In particular, surgical robotic datasets have been key to many advances in semi-autonomous surgeries, skill assessment, and training. Simulated surgical environments can enhance the data collection process by making it faster, simpler and cheaper than real systems. In addition, combining data from multiple robotic domains can provide rich and diverse training data for transfer learning algorithms. In this paper, we present the DESK (DExterous Surgical SKills) dataset. It comprises a set of surgical robotic skills collected during a surgical training task using three robotic platforms: the Taurus II robot, Taurus II simulated robot, and the YuMi robot. This dataset was used to test the idea of transferring knowledge across different domains (e.g. from Taurus to YuMi robot) for a surgical gesture classification task with seven gestures/surgemes. We explored two different scenarios: 1) No transfer and 2) Domain transfer (simulated Taurus to real Taurus and YuMi robots). We conducted extensive experiments with three supervised learning models and provided baselines in each of these scenarios. Results show that using simulation data during training enhances the performance on the real robots, where limited real data is available. In particular, we obtained an accuracy of 55% on the real Taurus data using a model that is trained only on the simulator data, but that accuracy improved to 82% when the ratio of real to simulated data was increased to 0.18 in the training set.**

I. INTRODUCTION

Minimally invasive robotic surgery has evident advantages over traditional surgery, such as quick recovery, lower risks and lower catastrophic errors for patients, thereby becoming the standard of care for a wide variety of surgical procedures [1]. However, these techniques require residents to spend a substantial amount of time practicing surgical maneuvers in simulation environments. In particular, the surgical robotic simulators play a crucial role in the training process of residents and novice surgeons, leading to significant improvement of their technical skills gradually and over time [2]. The tasks that are predominantly presented for training using the simulation and bench-top models include, but are not limited to, peg transfer, pattern cut, suture, and needle passing [3].

There are other key-benefits in using simulation: (a) It provides unlimited amount of training time while avoiding human/animal tissue manipulation or expensive single-use mock models [4] and (b) It allows researchers to collect enormous amount of data at a lower cost and in a scalable manner [5]. While some Da-Vinci datasets are publicly available, including surgical maneuvers during a variety of procedures, there is a lack of datasets featuring other types of surgical robots which may be less popular. This limits the ability to generalize across other robotic platforms. This limitation is particularly critical when in-field deployable surgical robots are needed (e.g. military, disaster-relief scenarios). The scarcity of publicly available datasets prevents researchers from exploring novel strategies to transfer the knowledge gained in the surgical simulator to a variety of other robots, less common in the Operating Room (OR) but more suitable to field conditions.

The goal of this paper is to provide a dataset that can allow for "transfer learning" across robotic platforms. As a test-case study, we will rely on the peg transfer task, which is a task of primary importance in surgical skill learning. In this regard, this paper introduces a library of motions for the robotic peg transfer task obtained from multiple domains: robotic simulator and two real robots (Taurus and YuMi). We refer to this database as DExterous Surgical SKills Transfer dataset (DESK). In addition to providing the dataset, we will present a proof-of-concept for learning to classify surgemes with a very limited or no training information from the real robots. The goal is to transfer the knowledge gained from the simulator to accelerate the learning on real robots by leveraging on abundant data obtained from the simulation environment. This instance of transfer learning is known as domain adaptation, a scenario in transductive transfer learning [6], where the source and target tasks are the same, but the data distribution of target and source domain are different.

The main contributions of this work are to: 1) Provide an annotated dataset of surgical robotic skills (DESK) collected in three domains: two real robots with different morphology (Taurus II and YuMi) and one simulated robot (Taurus II), and 2) Transfer the knowledge gained from abundant simulation data to real robot's data in context of surgeme classification.

* These authors contribute equally to this work

[1]School of Industrial Engineering, Purdue University, West Lafayette, IN, 47907, USA nmadapan, sanch174, gonza337, zhan3275, barragan,jpwachs@purdue.edu

[2] Department of Computer Science, Purdue University, West Lafayette, IN, 47907, USA rahman64, yexiang@purdue.edu

[3]School of Engineering Technology, Purdue University, West Lafayette, IN, 47907, USA mbalakun, jpadmaku, lvenkate, rvoyles@purdue.edu

[4] SRI International thomas.low@sri.com

II. Background and Related work

The advancement of surgical robotic activity recognition and semi-autonomous surgeries benefits from the availability of robotic surgical datasets. Two such prominent datasets are the JIGSAWS [7] (JHU-ISI Gesture and Skill Assessment Working Set) and MISTIC-SL (Johns Hopkins Minimally Invasive Surgical Training and Innovation Center; Science of Learning Institute). These datasets comprise procedures preformed with the da Vinci Surgical System on a bench-top model, including synchronized video and kinematic data [8]. A main advantage of these datasets is that they allow elucidating patterns associated with skill learning. With this goal in mind, surgical tasks are decomposed into a finite set of maneuvers [7]. This process of decomposition is known as surgical skill modelling [9]. In this modelling technique, each surgical skill is represented as a sequence of atomic units referred as gestures or surgemes [7]. These datasets have been used to learn representations of surgical motions [10] and recognize surgical activities [11]. However, the MISTIC-SL dataset is not publicly available at the moment. In contrast, the JIGSAWS is publicly availiable and has been used for multiple applications such as motion generation of expert demonstrations [12], recognition of surgical gestures [13], and surgical trajectory segmentation [14]. Nonetheless, the data collection for the JIGSAWS dataset did not intentionally introduce variability in the environment and initial conditions of the task. In realistic surgical tasks, the videos and kinematics will vary greatly between demonstrations, stressing the need of surgical datasets that include variability in the experimental setup to facilitate generalization.

The segmentation and classification of time series data on surgical datasets has been evaluated on the JIGSAWS and MISTIC-SL dataset using several approaches. Previous supervised learning methods include the use of hidden Markov models [15], conditional random fields [16], and bag of spatio-temporal features [13]. More recent methods include the use of Recurrent Neural Networks for recognizing surgical activities [10], [11]. Nonetheless, these approaches were tested using data from the same distribution as the training data, and do not account for the disparity encountered from randomized initial conditions.

Several efforts have been made in autonomous classification and execution of surgical tasks using multiple surgical robots and simulation platforms [17], [18]. However, the obtained models are specific for a given platform and setup, and cannot be directly transferred to other robots/procedures. As surgical data sets become available, it is desirable to use principles of transfer learning to leverage previous models to accelerate learning in new domains.

Research in transfer learning and domain adaptation has leveraged simulation environments to boost learning in real systems [19], [20]. Knowledge transfer between a robot and a simulated environment is challenging due to the differences in data distribution. Methods of domain randomization have been proposed to increase generalization in real scenarios for models trained in simulation [5], [21], by randomizing object position and appearance over the training set.

Transfer learning between dissimilar robots has been extensively studied in the area of Reinforcement Learning (RL) [22], [23]. The work in [24] shows a modular policy strategy for networks that allows to jointly train data from different robots with a common task, or data coming for the same robot with varying tasks. In domain transfer scenarios, dimensionality reduction techniques can be used to transfer learning between different robots [25]. Bcsi et al. proposed a knowledge transfer approach that is agnostic to the robot parameters, were the source and the target dataset are reduced to a common lower-dimension manifold [25]. This method does not require any kinematic knowledge of the target domain. Instead, the users must propose a bijective mapping between each dataset and the lower dimensional manifold. In contrast with this approach, our method does not require mapping back to the original space. We take a dimensionality reduction approach that intends to preserve the common features and the transfer learning is done directly in the reduced space.

Our DESK dataset provides RGB images, depth and kinematic information for the peg transfer task from multiple domains including two real robots (SRI Taurus II and ABB YuMi) and a simulation environment (SRI Taurus II). These three robotic setups possess inherent variance in peg board configuration, object size and appearance. Furthermore, we provide the data annotations concerning atomic surgical actions (start frame ID, end frame ID and if the action was a success or a failure). Additional variability is added to the dataset by randomizing the pick and place locations for the pegs and orientation of the board, while leaving the order of the pegs to be transferred unrestrained. In addition, the dataset contains examples of success and failure of surgemes employed during the task and subsequent recovery maneuvers. This work provides a baseline for transfer learning, from simulated to real robots using the DESK dataset.

III. DESK Dataset

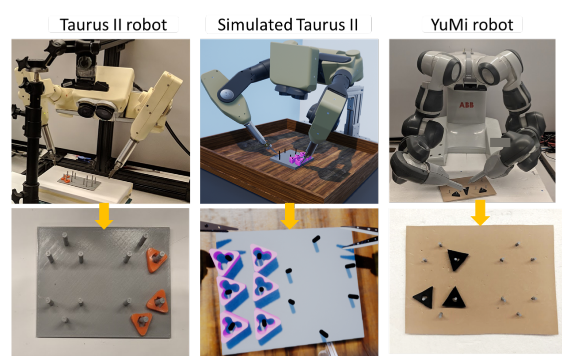

Fig. 1: Experimental setup for data collection on three different robotic domains.

A. *Peg transfer surgical training task*

The peg transfer task is one of the five tasks present in the Fundamentals of Laparoscopic Surgery [26] and has been commonly used to train residents [27], [28]. The task consists of picking an object from a peg board with one robotic arm, transferring it to the other arm and positioning the object over a target peg on the opposite side of the board. These tasks require a high level of sensorimotor skill due to the small clearance between pegs and objects, and the limited maneuverability of the manipulator caused by multiple objects in the workspace.

The peg transfer setup for the DESK dataset has two sets of numbered poles (from 1 to 6), each object has to be picked from its peg with one gripper, transferred to the other gripper and placed in a specified peg on the other side. Variability is introduced in the dataset by completely randomizing the following elements in the setup: 1) Initial and final positions of objects, 2) direction of the transfer (objects on the left side are transferred to the right and vice versa) and 3) position/orientation of the peg board. The experimental design for each of the robotic platforms was summarized in the table II. The data collection on all robotic platforms was performed by trained non-surgeon operators. Since multiple pegs were present in each trial, the order of the pegs to be transferred was selected by the user.

B. Data description

The kinematic data, the RGB video and the depth video were segmented according to surgical gestures (surgemes) observed in RGB video frames. A graphical tool was developed to facilitate the surgeme annotation based on RGB video recordings. A total of 7 surgemes were annotated for the peg transfer task. All the surgeme names, excluding the ones related to transfer, are self explanatory. The surgeme *Get Together* refers to the action of aligning and getting the grippers close to each other. The surgeme *Exchange* refers to the action of passing the peg from one gripper to the other. In addition, each surgeme was marked as a success or a failure. Table I and Figure 2 show the list of annotated surgemes for the DESK dataset. Each peg transfer video was associated with an annotation file that describes the following for each surgeme: name, the start and end frame, and whether the surgeme execution was a success (True) or a failure (False). In addition, timestamps were stored for each recording which allow to synchronize all the recordings (depth, kinematic and controller data) with respect to the RGB videos.

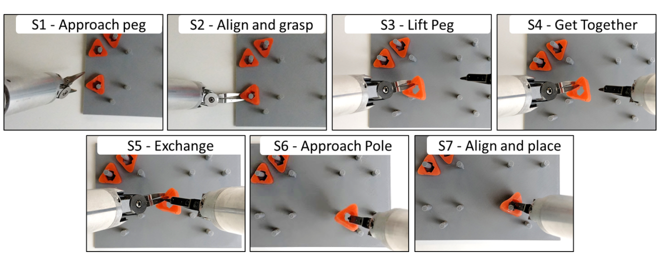

Fig. 2: Surgemes in the peg transfer task for the Taurus II robot.

C. Data collection with the Taurus robot

The Taurus II robot is controlled using the Razer Hydra® over a stereoscopic display. It is interfaced with two foot pedals that allow to switch control between the arms and the camera. A *clutch* pedal was used to toggle the robot between operation mode and standby mode to enable the user to reset to an ergonomic position for manipulation. In this setup, the robot only moved when the *clutch* of the foot pedal was pressed.

The data collected from the Taurus robot included RGB video and depth video recorded from the top view Realsense® camera. In addition, the kinematic data of the Taurus robot's end-effector was captured using 16 kinematic variables as shown in Table III. For each arm, we recorded: the rotation matrix of the wrist (nine values), the translation in x, y and z coordinates of the wrist with respect to the robot origin and the gripper state, which is a value between 30 and 100 (30 when completely closed and 100 when the gripper is completely open).

TABLE I: Surgical gestures in the peg transfer task. The columns indicate surgeme ID, name of the surgeme, number of instances present for each surgeme for the simulator, real Taurus and the YuMi robot.

ID	Surgeme name	# Sim	# Taurus	# YuMi
S1	Approach peg	192	111	117
S2	Align & grasp	206	115	123
S3	Lift peg	203	112	123
S4	Transfer peg - Get together	180	113	117
S5	Transfer peg - Exchange	175	113	118
S6	Approach pole	167	109	117
S7	Align & place	163	107	116

TABLE II: Experimental design for data collection.

	Taurus Simulator	Taurus II	YuMi
DOF	7	7	7
# of subjects	7	3	1
# of trials/subject	6	12	3
# of transfers/trial	6	3	40
# of total transfers	252	108	120

D. Data Collection with the Taurus Simulation Environment

The simulated Taurus robot is controlled using the Oculus Rift touch controllers. Similar to the Taurus robot, the simulated Taurus system has a foot pedal that enables the motion of the robot and allows to switch control between the arms and the stereoscopic camera.

This dataset includes the following recordings: the kinematic data of the robot's wrist and RGB videos recorded from the point of view of the user's virtual environment, stereo view of both the left and right robot's cameras (the two stereo videos can be used to compute the depth information). The kinematic data consists of 14 kinematic variables that represent the robot's end-effector pose, as displayed in Table III. It also includes the wrist orientation (yaw, pitch and roll angles), the translation in x, y and z coordinates with respect to the robot's origin, the joint angles (7 DOF) and the griper state (also a value between 30 and 100) for both the arms.

E. Data collection with the YuMi robot

The YuMi collaborative robot was adapted to surgical tasks using 3D printed gripper extensions [29]. The end-effectors of the robot are controlled using the HTC VIVE controllers. The recorded data includes RGB video and depth video obtained from the top view using a Realsense® camera. In addition, the kinematic data of each robot arm is captured using 20 kinematic variables that provide joint state information, translation of the tooltip in x,y,z coordinates, rotation matrix of the tooltip with respect to the robot's origin, and gripper state (see Table III).

The YuMi robot is significantly different from the Taurus robot. For instance, the Taurus II is designed specifically for small dexterous tasks such as bomb disposal and surgery, while the YuMi is suitable for larger workspaces and collaborative tasks. For this reason, the setup for peg transfer using the YuMi was scaled by a factor of 2 (the size of the peg board is larger). Other differences include robot morphology and the interface used for manipulation. The pitch, yaw, and roll angles of the Taurus are computed with respect to the wrist, which causes minimal movement of the wrist when the operator changes the orientation the tool. In contrast, the YuMi robot has a kinematic control that reorients with respect to the tooltip. Given the large distance between the tooltip and the wrist, even minor changes in the orientation of the tool causes large motions at the robot's wrist and the arm configuration.

TABLE III: Kinematic variables. Note that ts is the Unix timestamp, $\vec{J}$ is the vector of joint angles, $\vec{p}$ is the position vector (x, y and z), $\vec{\theta}$ be the Euler angles (yaw, pitch and roll), gs is the gripper state of the end-effector and R be the 3 x 3 rotation matrix.

Taurus		Taurus Simulator		YuMi	
ID	Variable	ID	Variable	ID	YuMi
1	ts	1	ts	1	ts
2-13	R and $\vec{p}$	2-4	$\vec{p}$	2-8	$\vec{J}$
	-	5-7	$\vec{\theta}$	9-11	$\vec{p}$
14-16	$\vec{p}$	8-14	$\vec{J}$	12-20	R
17	gs	15	gs	21	gs

IV. Experiments and Results

The data collected from the simulator and the real robots had different dimensions, as shown in Table III. The first step in the pipeline was to ensure that the dimensions of the data were equivalent in the three domains. Therefore, we reduced the feature dimension by considering the features that are commonly shared between these two domains (position, orientation and gripper status of the end-effector). Overall, we considered 14 features per frame (seven features in each arm).

The experiments conducted in this work are two-fold: 1. Train and test on the data obtained from the same domain (no-transfer scenario) and 2. Train on one domain and test on the other (domain-transfer scenario). Furthermore, we considered two kinds of classification tasks: 1. frame-wise and 2. sequence-wise. The former method associates each frame to a particular class label and treats each frame as a sample point (*frame-wise instances*), while the latter method considers the entire surgeme (sequence of frames) as a single sample point (*sequence-wise instances*). Next, we used three supervised learning methods for our experiments: 1. Support Vector Machines (SVM), 2. Random Forest (RF) and 3. Multi-layer Perceptron (MLP). These three learning techniques are commonly used in the machine learning community for creating the baselines for classification. We used the scikit-learn [30] implementation of these models for our experiments.

Hyperparameter setting. A linear kernel was used SVM classifier. For RF, we set $n\_estimators = 200$ (number of trees in the forest), and maximum depth = 10. For the MLP, we used a hidden layer of size = 100, $tanh$ as the activation function with *adam* as the optimizer.

Each surgeme instance consists of a variable number of frames. Thus, we re-sampled (via linear interpolation) the original instances to a fixed number of frames to generate *sequence-wise* instances. Next, we concatenated the seven features corresponding to each frame, for both the arms, to create a single feature vector. In our case, we set the number of frames to 40 and each *sequence-wise* instance is a 560 dimensional vector ($40 \times 7 \times 2$). In our experiments, we did not differentiate between successful and failed instances of the same surgeme class.

A. Surgeme classification

In the first experiment, our goal was to study the classification in the *no-transfer* scenario where the learning model is trained and tested on data coming from the same domain. In other words, the training and testing data follow the same distribution. We have two such scenarios: train and test on 1. simulator data ($S \rightarrow S$) and 2. real robot data ($R \rightarrow R$). In our experiments, we have used 60-40% split for training and testing respectively. Furthermore, we have used five-fold cross validation approach to tune the learning parameters (or weights) of the models.

Next, we performed the classification on both the frame-wise and sequence-wise instances as shown in the Table IV. The *sequence-wise* features contain the temporal information embedded into them. Hence, these features give superior accuracy except for the random forest approach on simulator data. Therefore, we used *sequence-wise* features for the experiments associated with the domain transfer. We conducted the two-sided paired t-test and found that SVM significantly outperforms the RF and MLP with the *sequence-wise* features ($p < 0.05$).

TABLE IV: Classification accuracy on the no-transfer scenario for both the frame-wise and sequence-wise features. S is the Taurus simulator, R_1 is Taurus robot, and R_2 is the YuMi robot.

	Sequence-wise			Frame-wise		
	RF	SVM	MLP	RF	SVM	MLP
$S \rightarrow S$	88 ± 2	87 ± 1	78 ± 4	86 ± 0	58 ± 1	73 ± 1
$R_1 \rightarrow R_1$	94 ± 2	92 ± 1	92 ± 2	95 ± 0	60 ± 0	92 ± 1
$R_2 \rightarrow R_2$	91 ± 1	93 ± 1	95 ± 1	88 ± 1	48 ± 1	86 ± 1

B. Domain transfer: Simulator to real robot

In this experiment, our goal is to build a learning model that was trained on the data obtained from one domain but tested on the data coming from a new domain. In other words, the input data distribution is different in the train and test datasets. We start with the assumption that it is much easier to collect the data from the simulator when compared to the data from the physical surgical robot. Therefore, the real robot's data is assumed to be limited or not available in extreme cases. Hence, we trained our models on the simulator data with very little or no data from the real robot and tested this model on the real robot's data.

To simulate the limited availability of the real data, we added a small fraction (α) of the real data into the training, where α was varied from zero to one. The value of α is defined as the ratio of number of examples of the real data to the simulator data present in the training. An $\alpha = 0$ indicates *complete transfer*, where there is no real data present in the training, while $\alpha = 1$ implies that the data from the simulator and real data are in equal proportions.

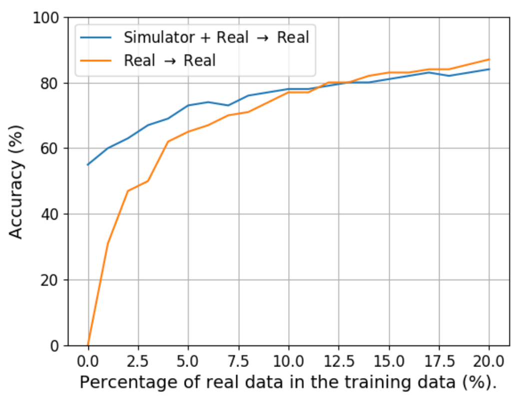

Fig. 3: Performance comparison for transfer learning from Taurus simulator to Taurus Robot using SVM. The blue curve indicates the transfer accuracy and the blue curve indicates accuracy on real data without transfer.

TABLE V: Domain transfer accuracy when the models are trained on the domain C but tested on domain R. Note that the domain R is the real Taurus domain C is the combination of S and R.

		RF		SVM		MLP	
α	#	$C \to R$	$R \to R$	$C \to R$	$R \to R$	$C \to R$	$R \to R$
0.00	0	34 ± 3	0 ± 0	55 ± 2	0 ± 0	40 ± 2	0 ± 0
0.03	24	53 ± 3	39 ± 4	67 ± 2	50 ± 5	57 ± 3	51 ± 4
0.06	49	66 ± 1	64 ± 4	74 ± 0	67 ± 3	68 ± 4	72 ± 2
0.09	73	77 ± 3	73 ± 5	77 ± 1	74 ± 3	74 ± 1	80 ± 3
0.12	98	78 ± 1	80 ± 3	79 ± 2	80 ± 1	78 ± 1	85 ± 1
0.15	123	84 ± 2	84 ± 2	81 ± 1	83 ± 0	82 ± 2	86 ± 1
0.18	147	85 ± 0	85 ± 0	82 ± 1	84 ± 1	79 ± 3	88 ± 1

Moreover, *accuracy* (γ - the percentage of test examples that are correctly classified) and confusion matrices are used as performance metrics to evaluate how well the model is performing on the unseen data. Let us define $\gamma^{M}_{A \to B}$ as the test accuracy of the model M trained on the data from domain A and tested on the data from domain B. Note that $\gamma^{M}_{A \to B}$ depends on the value of α. Also, let us define C as the domain that is a combination of both real and simulator domains. The goal of this study is to determine the behavior of $\gamma^{M}_{C \to R}$ (trained on both simulator and real data and tested on the real data) and $\gamma^{M}_{R \to R}$ (trained and test on real data) as the value of α increases.

Figure 3 shows the behavior of $\gamma^{SVM}_{R \to R}(\alpha)$ and $\gamma^{SVM}_{C \to R}(\alpha)$ for SVM classifier. In the extreme case when there was no real data in the training process (i.e. $\alpha = 0$ - *complete transfer*), the accuracy was approximately 55%. As the value of α gradually increases from 0 to 0.2, the value of $\gamma^{SVM}_{C \to R}$ increases and converges in comparison to $\gamma^{SVM}_{R \to R}$. At $\alpha = 0.18$, the values of $\gamma^{RF}_{C \to R}$ and $\gamma^{RF}_{R \to R}$ were approximately equal to 80%. Moreover, this plot shows that adding the simulator data into the training procedure considerably improved the accuracies on the real data.

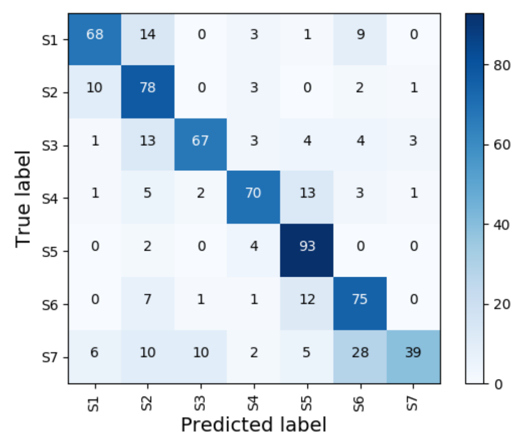

Fig. 4: Confusion matrix for transfer learning with SVM when $\alpha = 0.05$ for real Taurus robot data.

Table V shows the transfer accuracies obtained using three learning models for a range of values of α. Note that there are 1286 samples in the simulator domain. The second column (#) is the number of examples of the real data present in the training procedure. In the first row, it was assumed that there were no real data examples in the training, hence accuracy in $R \to R$ is 0. For illustration purposes, the confusion matrix presented in Figure 4 shows the transfer accuracies for all the classes when $\alpha = 0.05$. Overall, the Table V shows that it is beneficial to augment the training data with the samples obtained from robotic simulators. Irrespective of the learning model, it helps greatly enhance the performance of those models on the real robot's data.

In the last experiment, our goal was to verify the transfer across the robots i.e. to train the model on the simulation data of Taurus robot and test on the data of YuMi robot. The workspace, morphology and dimension of the data of these robots was significantly different. Hence, we used the principal component analysis (PCA) to rank the features based on the Eigen values and transform the data into the Eigen space. Next, we chose a fixed number of features (a value of 8 is used in our experiments) in the Eigen space in order to unify the feature dimensions. We followed a similar

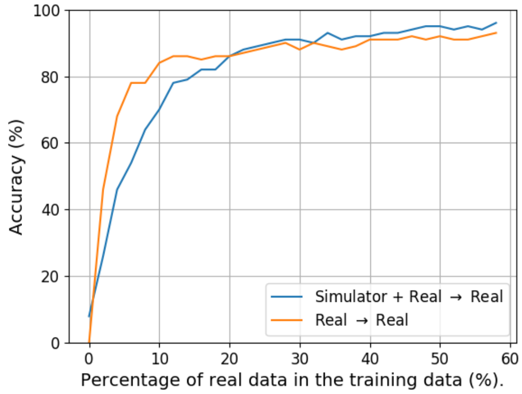

Fig. 5: Performance comparison for transfer learning from the Taurus simulator to the YuMi Robot.

experimental design to test the transfer of knowledge from the Taurus simulator to the YuMi domain. The accuracy of *complete transfer* scenario is close to the random accuracy indicating that the model is overfitting to the simulator, as there is no data from the real robot in the training. However, a slight increase in the number of examples of YuMi data in the training produces an swift increase in the transfer accuracies. For instance, when the number of YuMi examples = 100, the transfer accuracy is approximately 80% as shown in the Figure 5. When we add more than 200 examples of YuMi in the training, the transfer accuracies marginally surpasses the accuracies in $R \rightarrow R$ scenario. This shows that transferring the knowledge between two completely different robots is a challenging task and requires relatively more examples to achieve superior transfer accuracies.

V. Discussion and Future Work

The interaction modality of the Taurus robot and the Taurus simulated robot are completely different, however, the robot configuration is similar. Hence, transferring the knowledge to the physical domain is relatively easier in the case of the real Taurus robot in comparison to the YuMi robot. Thus, the *transfer accuracy* is significantly higher for the real Taurus. Furthermore, the *transfer accuracy* obtained on the YuMi robot increases with the increase in α and surpasses $R \rightarrow R$ scenario for $\alpha > 0.2$. In other words, the amount of real data needed to achieve the *transfer accuracy* of 80% is much higher for the YuMi ($\alpha = 0.3$) in comparison to the real Taurus ($\alpha = 0.12$). The relatively lower transfer accuracies obtained on YuMi data show that the transfer from one robot to another robot is a challenging task.

It was mentioned in the results that the *sequence-wise* features provide significantly better accuracies in comparison to the *frame-wise* features for the transfer learning tasks. Note that the sequence classification approach assumes that the test data is annotated beforehand with respect to the start and end of the surgemes. However, this assumption is not valid when we want to deploy these trained classifiers in real-time as the real-time data cannot be segmented beforehand. In contrast, the *frame-wise* classification approach does not require the surgemes to be segmented. Hence the *frame-wise* classifier can potentially act as the model that can be used to know the start and the end point of the surgemes.

In this regard, we have made our database publicly available to encourage researchers to further investigate the problem of transfer learning between the real robots or between the simulator and a real robot. In addition to the RGB-D videos and the kinematic data, we also provide bounding boxes of the objects and pegs for each frame. These annotations are created in a semi-autonomous manner i.e. first, the color-based image processing techniques were used to create the bounding boxes automatically and next, a human annotator was asked to verify those annotations and manually annotate the frames with erroneous annotations. The instructions to obtain the data are available at https://github.com/nmadapan/Forward_Project.git.

The potential future works and applications of our dataset include: 1. Incorporating the object bounding boxes and visual data (RGB-D images) into the feature vectors to improve the transfer accuracies, 2. Developing task specific deep models with the goal of transferring the knowledge from one robot to the other, 3. Learn to predict the surgemes from one physical robot instead of the simulator and test on another physical robot (in our case, train on YuMi and test on Taurus), 4. Learning to predict the surgemes only with the partial information, and 5. Developing learning models that do not require the surgemes to be segmented beforehand.

VI. Conclusions

The main goal of this paper is to learn to transfer the knowledge (surgical skills in our case) from one domain (simulator or a physical robot) to another domain (physical robot). Previous datasets concerned with surgical tasks were mainly focused on a unique robotic platform (e.g. da-Vinci). This limits researchers to explore novel ways to transfer the surgical skills learned from one platform to the other. Therefore, it is essential to have the data collected from various robotic platforms. Hence, we created a database DESK of surgical robotic skills (peg transfer task) collected from three domains: simulated Taurus robot, real Taurus robot and YuMi industrial robot. In addition, we proposed a simple, yet effective, technique to improve the learning for a surgical gesture classification task over real robot's data using the data obtained from the simulation. We presented three supervised models as baselines for surgeme classification. Results show that augmenting the training data with simulator data considerably improves the accuracies of prediction on the real data. More specifically, in the extreme case when there is no real Taurus robot's data present in the training, the *transfer accuracy* on the real Taurus data is 55%.

Acknowledgments

This work was supported by the Office of the Assistant Secretary of Defense for Health Affairs under Awards No. W81XWH-18-1-0769 and W81XWH-14-1-0042, and by the NSF RoSe-HUB Center under CNS-1439717. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the funders.

References

[1] A. Chellali, L. Zhang, G. Sankaranarayanan, V. S. Arikatla, W. Ahn, A. Derevianko, S. D. Schwaitzberg, D. B. Jones, M. DeMoya, and C. G. L. Cao, "Validation of the VBLaST peg transfer task: a first step toward an alternate training standard," *Surgical Endoscopy*, vol. 28, pp. 2856–2862, Oct. 2014.

[2] R. Korets, A. C. Mues, J. A. Graversen, M. Gupta, M. C. Benson, K. L. Cooper, J. Landman, and K. K. Badani, "Validating the Use of the Mimic dV-trainer for Robotic Surgery Skill Acquisition Among Urology Residents," *Urology*, vol. 78, pp. 1326–1330, Dec. 2011.

[3] A. J. Hung, I. S. Jayaratna, K. Teruya, M. M. Desai, I. S. Gill, and A. C. Goh, "Comparative assessment of three standardized robotic surgery training methods," *BJU International*, vol. 112, no. 6, pp. 864–871, 2013.

[4] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-Real Transfer of Robotic Control with Dynamics Randomization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, May 2018.

[5] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, Sept. 2017.

[6] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 1345–1359, Oct. 2010.

[7] Y. Gao, S. S. Vedula, C. E. Reiley, N. Ahmidi, B. Varadarajan, H. C. Lin, L. Tao, L. Zappella, B. Bjar, and D. D. Yuh, "JHU-ISI gesture and skill assessment working set (JIGSAWS): A surgical activity dataset for human motion modeling," in *MICCAI Workshop: M2CAI*, vol. 3, p. 3, 2014.

[8] Y. Gao, S. S. Vedula, G. I. Lee, M. R. Lee, S. Khudanpur, and G. D. Hager, "Unsupervised surgical data alignment with application to automatic activity annotation," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4158–4163, May 2016.

[9] G. P. Moustris, S. C. Hiridis, K. M. Deliparaschos, and K. M. Konstantinidis, "Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 7, no. 4, pp. 375–392, 2011.

[10] R. DiPietro and G. D. Hager, "Unsupervised Learning for Surgical Motion by Learning to Predict the Future," in *Medical Image Computing and Computer Assisted Intervention MICCAI 2018* (A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-Lpez, and G. Fichtinger, eds.), Lecture Notes in Computer Science, pp. 281–288, Springer International Publishing, 2018.

[11] 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 *Medical Image Computing and Computer-Assisted Intervention MICCAI 2016* (S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells, eds.), Lecture Notes in Computer Science, pp. 551–558, Springer International Publishing, 2016.

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

[13] 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*, vol. 64, pp. 2025–2041, Sept. 2017.

[14] S. Krishnan, A. Garg, S. Patil, C. Lea, G. Hager, P. Abbeel, and K. Goldberg, "Transition State Clustering: Unsupervised Surgical Trajectory Segmentation for Robot Learning," in *Robotics Research: Volume 2* (A. Bicchi and W. Burgard, eds.), Springer Proceedings in Advanced Robotics, pp. 91–110, Cham: Springer International Publishing, 2018.

[15] C. E. Reiley and G. D. Hager, "Task versus Subtask Surgical Skill Evaluation of Robotic Minimally Invasive Surgery," in *Medical Image Computing and Computer-Assisted Intervention MICCAI 2009* (G.-Z. Yang, D. Hawkes, D. Rueckert, A. Noble, and C. Taylor, eds.), Lecture Notes in Computer Science, pp. 435–442, Springer Berlin Heidelberg, 2009.

[16] L. Tao, L. Zappella, G. D. Hager, and R. Vidal, "Surgical Gesture Segmentation and Recognition," in *Medical Image Computing and Computer-Assisted Intervention MICCAI 2013* (K. Mori, I. Sakuma, Y. Sato, C. Barillot, and N. Navab, eds.), Lecture Notes in Computer Science, pp. 339–346, Springer Berlin Heidelberg, 2013.

[17] B. Kehoe, G. Kahn, J. Mahler, J. Kim, A. Lee, A. Lee, K. Nakagawa, S. Patil, W. D. Boyd, P. Abbeel, and K. Goldberg, "Autonomous multilateral debridement with the Raven surgical robot," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1432–1439, May 2014.

[18] D. Seita, S. Krishnan, R. Fox, S. McKinley, J. Canny, and K. Goldberg, "Fast and Reliable Autonomous Surgical Debridement with Cable-Driven Robots Using a Two-Phase Calibration Procedure," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6651–6658, May 2018.

[19] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, and V. Vanhoucke, "Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4243–4250, May 2018.

[20] K. Fang, Y. Bai, S. Hinterstoisser, S. Savarese, and M. Kalakrishnan, "Multi-Task Domain Adaptation for Deep Learning of Instance Grasping from Simulation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3516–3523, May 2018.

[21] S. James, A. J. Davison, and E. Johns, "Transferring End-to-End Visuomotor Control from Simulation to Real World for a Multi-Stage Task," *arXiv:1707.02267 [cs]*, July 2017. arXiv: 1707.02267.

[22] A. A. Rusu, M. Vecerik, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell, "Sim-to-real robot learning from pixels with progressive nets," *arXiv preprint arXiv:1610.04286*, 2016.

[23] M. E. Taylor, H. B. Suay, and S. Chernova, "Integrating reinforcement learning with human demonstrations of varying ability," in *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pp. 617–624, International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[24] C. Devin, A. Gupta, T. Darrell, P. Abbeel, and S. Levine, "Learning modular neural network policies for multi-task and multi-robot transfer," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2169–2176, IEEE, 2017.

[25] B. Bocsi, L. Csató, and J. Peters, "Alignment-based transfer learning for robot models," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, IEEE, 2013.

[26] E. M. Ritter and D. J. Scott, "Design of a Proficiency-Based Skills Training Curriculum for the Fundamentals of Laparoscopic Surgery," *Surgical Innovation*, vol. 14, pp. 107–112, June 2007.

[27] N. A. Arain, G. Dulan, D. C. Hogg, R. V. Rege, C. E. Powers, S. T. Tesfay, L. S. Hynan, and D. J. Scott, "Comprehensive proficiency-based inanimate training for robotic surgery: reliability, feasibility, and educational benefit," *Surgical Endoscopy*, vol. 26, pp. 2740–2745, Oct. 2012.

[28] R. A. Joseph, A. C. Goh, S. P. Cuevas, M. A. Donovan, M. G. Kauffman, N. A. Salas, B. Miles, B. L. Bass, and B. J. Dunkin, "Chopstick surgery: a novel technique improves surgeon performance and eliminates arm collision in robotic single-incision laparoscopic surgery," *Surgical Endoscopy*, vol. 24, pp. 1331–1335, June 2010.

[29] N. Sanchez-Tamayo and J. P. Wachs, "Collaborative Robots in Surgical Research: A Low-Cost Adaptation," in *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '18, (New York, NY, USA), pp. 231–232, ACM, 2018. event-place: Chicago, IL, USA.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.