Drift-Free Latent Space Representation for Soft Strain Sensors

Thomas George Thuruthel[1], Kieran Gilday[1] and Fumiya Iida[1]

***Abstract*— Soft strain sensors are becoming increasingly popular for obtaining tactile information in soft robotic applications. Diverse technological solutions are being investigated to design these sensors. Simultaneously, new methods for modeling these sensor are being proposed due to their highly nonlinear, time varying properties. Among them, machine learning based approaches, particularly using dynamic recurrent neural networks look the most promising. However, these complex networks have large number of free parameters to be tuned, making it difficult to apply them for real-world applications. This paper introduces the concept of transfer learning for modelling soft strain sensors, which allows us to utilize information learned in one task to be applied to another task. We demonstrate this technique on a passive anthropomorphic finger with embedded strain sensors used for two regression tasks. We show how the transfer learning approach can drastically reduce the number of free parameters to be tuned for learning new skills. This work is an important step towards scaling of sensor networks (algorithm-wise) and for using soft sensor data for high-level control tasks.**

I. INTRODUCTION

Arguably, all tactile sensing capabilities like contact localization, deformation sensing, force/pressure sensing, etc. can be developed using strain-based sensors [1]. Hence, soft strain sensors are appealing as a general purpose unobtrusive tactile sensing solution. Soft strain sensors are commonly developed using conductive elastomers like Conductive Polydimethylsiloxane [2], [3], liquid metal channels [4], etc. Due to the nonlinear viscoelastic properties of the materials involved in their manufacturing, all of these sensors exhibit complex time varying behavior up to varying extends [5]. Hence, the design and modelling of these sensing systems is currently one of the challenging problems in the field.

The design of soft sensory networks is one of the least addressed problems. Current techniques are based on heuristics and human knowledge [6]. Learned models, due to their black-box nature, are not applicable for sensor design, or at least there have been no attempts to do so. Analytical models, on the other hand, are difficult to develop for complex systems [7]. Added to the fact that current sensor technologies are varied and often individually manufactured, the performance of a sensor network and their information content can have higher variability. This means that for a high dimensional sensory networks embedded in a complex body, as in this work (Figure 2), it becomes difficult to know *a priori* the sensing capabilities of the system. This is especially true when each sensing unit has a large sensing region, which is enforced to simplify the wiring procedure. This means that these soft sensor networks can respond to an infinitely large number of tactile cues. By restricting the kind of interactions the body can perform, desired tactile information can be obtained.

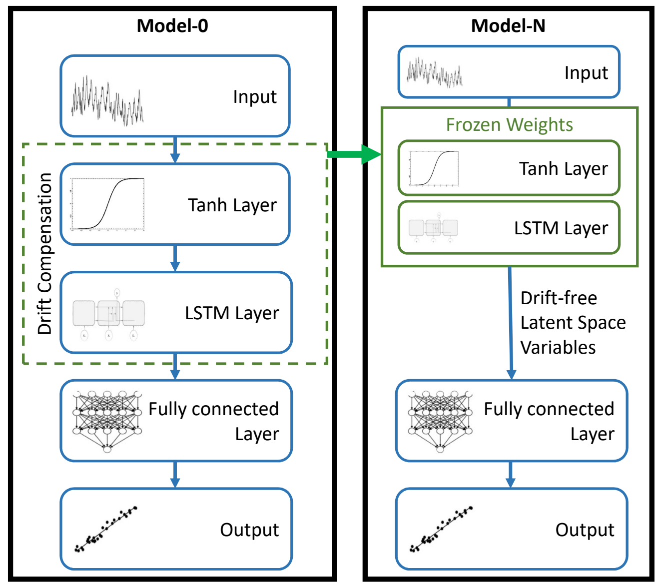

Fig. 1. Proposed transfer learning approach for quick drift-free representation of soft strain sensors

For a certain set of interactions performed by the body, certain set of tactile information is obtained. The mapping from the parameterized set of actions and the sensor readings is well defined. The inverse mapping may or may not be well defined. Assuming that a dense and distributed sensory network can be fabricated, the design problem can be partially solved and be condensed to an action parameterization problem.

Machine learning-based solutions have shown promise for the inverse mapping problem. Han et. al. used a type of dynamic recurrent neural network (RNN) called long short-term memory (LSTM) to estimate pressure values and discrete contact localization [8]. This work was further extended to a full-body wearable suit for human motion detection [9]. Similarly, LSTMs were used for proprioception and contact force estimation in [1]. Although, they are highly effective in learning time-series functions, they have large number of network parameters to be estimated. Hence, they are data hungry and prone to overfitting. Classic RNNs could reduce the number of network parameters to be tuned, however, they

[1] Thomas George Thuruthel and Fumiya Iida are with the Bio-Inspired Robotics Lab, Department of Engineering, University of Cambridge, UK. tg444@cam.ac.uk, fi224@cam.ac.uk

Source code and data for this paper can be found in : https://github.com/tomraven1/laten_space_robosoft

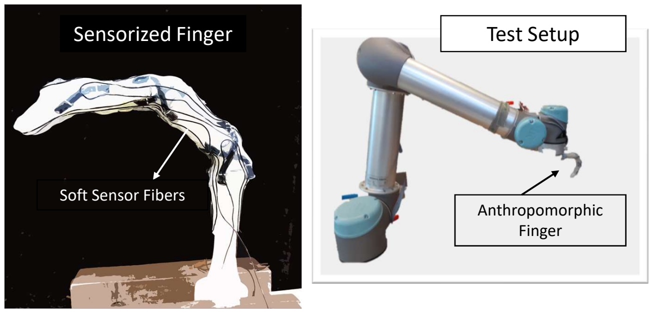

Fig. 2. Experimental setup used for this work. The passive anthropomorphic finger has six strain sensors embedded inside and is attached to a UR5 robotic arm.

are highly prone to the vanishing gradient or the gradient explosion problem.

A. Drift-free latent space representation

In summary, given a dense and distributed sensor network and using any kind of recurrent neural network architecture, various tactile sensing capabilities can be achieved for their corresponding range of interactions. Therefore multiple models can be learned for estimation based on the type of interaction. However, this means that for every new model, a large number of samples have to be obtained and the network has to be retrained. This is cumbersome for data hungry models networks like LSTMs. Yet, the reason for using recurrent networks is due to the time varying dynamics of the soft sensors, which can be assumed to be consistent across all models. Hence, the information about the sensor drift and noise should be transferable across various models. This paper presents the concept of transfer learning for such cases [10]. Transfer learning is a concept in machine learning where learned knowledge from solving one problem is leveraged to solve another *related* problem. Similarity between two problems depends on the functional form of the underlying function that machine leraning model is trying to approximate.

A simple methodology for transferring drift compensation information from one learned model to the other is presented here (see figure 1). This is based on the idea that the latent-space representation of the LSTM layer would have to contain drift-free information of the tactile sensors before it is sent to the fully-connected layer. Note that the fully-connected layer does not have any recurrent connections and has to predict the desired outputs using only the current information provided by the LSTM layer. Therefore all the layers preceding the fully-connected layer can be 'recycled' as such for further models. The only caveat being that task-0 should involve the usage of all the sensors that will be used for the later tasks. We are referring to the layer before the fully connected layer as the recurrent layer and the rest as the static layer for conciseness.

The proposed methodology provides an easy way to reduce the number of parameters to be tuned and subsequently the data requirements of learning-based approaches for drift compensation. Additionally the latent space representations are highly amenable for the reinforcement learning problems where the tactile sensor information is used to decide the next action [11]. Latent space representations have been previously used for a low-dimensional sensor-space representation by [12]. It was observed that control policies learned using this latent space inputs required fewer roll-outs and were more robust to noise. These methods come under the emerging concept of state representation learning [13].

II. EXPERIMENTAL SETUP

The experiments are conducted on a passive anthropomorphic finger (Figure 2). The skeleton of the finger is 3D printed and attached with compliant joints. The skeleton is then spin coated with a layer of Ecoflex-10. The strain sensors strands are then placed on the skin uniformly and with varying lengths. We use Conductive Thermoplastic Elastomer (CTPE), a thermoplastic elastic matrix which is homogeneously mixed with carbon black powder under high pressure and temperature for our purpose [14]. Six such sensors are placed. All the sensor ends come to the base of the finger for ease of wiring. Silver Conductive Adhesive Epoxy (MG Chemicals) is used to connect the soft sensor wire and the metallic wire. The finger is around 12 cm in length in the unbend state. The passive finger is then mounted on a UR5 robot arm (Universal Robotics). The sensors readings are measured using the Native Instruments USB-6212 data acquisition system after going through a voltage divider circuit.

A passive finger is used for this study to remove uncertainties produced in the system due to the finger actuation system. As we employ an industrial grade robotic arm for our motions, the repeatability and reliability of the finger motion can be guaranteed. Therefore all the observed noise in the system and other nonlinear time varying effects can be attributed to the sensor dynamics itself.

III. LEARNING PROCEDURE

A. LSTM Network

The learning of the sensor models is done using a LSTM network as shown in Figure 1. The inputs to the LSTM network are the sensor signals and the desired output depends on the task. The input layer takes in the raw sensor values (of size n), normalizes the values and passes them to the Tanh Layer. The Tanh layer uses the hyperbolic tangent activation function to bound the inputs to the LSTM layer. There are no weights multiplied to the layer and hence this layer does not have any tunable parameters. The number of inputs, n, is 6 in our case.

From the Tanh Layer the processed signals $x_t \in \mathbf{R}^n$ go to the LSTM layer of size h (50, in our case). With every forward pass of the LSTM network the states of the network at time step t are updated as follows:

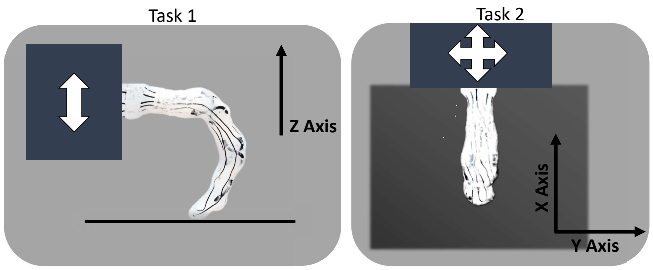

Fig. 3. The two tasks evaluated for the study. Task-1 is a simple linear motion of the finger in the Z Axis. Task-2 is a planar sliding motion performed on a flat surface. The three DoF combination of the both is used as Task-0.

$$
\begin{aligned}
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ \sigma(c_t)
\end{aligned}
$$

Where i_t is the input gates activation vector, f_t is the forget gates activation vector, o_t is the output gates activation vector, c_t is the cell state vector and h_t is the output vector. $W \in \mathbf{R}^{n\times h}, U \in \mathbf{R}^{h\times h}$ and $b \in \mathbf{R}^{h}$ are the corresponding weight and bias matrices to be learned during training for each of the cell, input gate, output gate and forget gate. Hence the number of free parameters in the LSTM network would be $4 \times (h^2 + h * n + h)$. The initial condition for the dynamic network is set as $c_0 = 0$ and $h_0 = 0$. For all the tasks, we use a LSTM layer of fixed size of 50. That corresponds to a number of $4 * (50^2 + 50 * 6 + 50)$ number of parameters to be tuned in the LSTM layer only.

The output vector (h_t) from the LSTM network goes into the fully connected layer which multiplies the inputs it gets with a weight matrix and adds a bias as shown below:

$$O_t = W_{fc} h_t$$

Where $O_t \in \mathbf{R}^{m}$ is the desired output from the network and $W_{fc} \in \mathbf{R}^{h\times m}$ is the weight matrix for the fully connected layer. Note that a nonlinear activation function can be added to this layer, if needed.

B. Transfer Learning

We try to evaluate our transfer learning procedure on two different tasks. In this paper we look into two regression tasks, where the objective is to estimate the state of the robot using the strain sensor data. The same approach can be extended to more complex tasks, like in a reinforcement learning framework.

Task-1 is a one degree of freedom motion, where the robotic arm moves in straight line in the Z-axis (See figure 3). The motions are random and the learned model has to predict the Z axis position of the UR5 end-effector using only the strain sensor data. Note that this is not a trivial task due to the noise, non-linearity and drift in the sensor data.

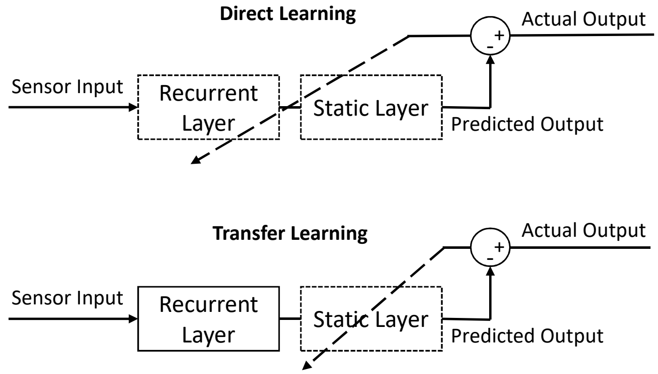

Fig. 4. Comparison between the direct learning approach and the proposed transfer learning approach. The layers before the fully connected layer are referred to as the recurrent layer and the rest as the static layer for conciseness.

The second task is a two DoF task with more computational complexity. The robotic arm moves along the XY plane with a fixed Z-axis position such that the finger is in contact with the flat surface and slides along the surface. As the finger slides along the surface the learned model has to implicitly keep track of the sensor drift state and the arm state. The inputs to the network, hence, is the same 6 sensor values and the output is the current X and Y coordinates of the UR5 end-effector. To achieve accurate predictions, the learned model has to implicitly estimate the deformations of the finger in the two directions, predict the time at which the finger slips and integrate the moved distance to the current estimate of the global arm location. To simplify the learning procedure and due to the low sampling rate of the sensors, the velocity of the UR5 arm is restricted within a range.

For obtaining the initial network for transfer learning (Model-0), a task-0 is defined. This is essentially a combination of task-1 and task-2, involving motions in all three directions. The mapping between the sensor readings and the three motion variables in not well-defined, in this case, and not accurately learnable. However, it still serves as a good source to obtain drift-free latent space representations. The tanh layer and the LSTM layer of model-0 is then fixed to be later trained on task-1 and task-2 (See figure 8).

All the models use a data sample of 30,000 points collected at 5 Hz. The MATLAB deep learning toolbox is used for modelling and training the LSTM networks. The Adam optimization algorithm is used for training and all the models are trained till 200 epochs. A test set which contains the last 20 percent of the sample data is used for estimating the performance of each model. The same training parameters are used for training the subsequent models using the transfer learning approach. The parameters to be tuned are however reduced to the parameters of the fully connected layer. The number of variables hence would be proportional to the size of the LSTM layer multiplied by the size of the output variables. Note that this is a drastic reduction in the number of parameters as the LSTM layers have parameter space which increases quadratically with the size of the network.

IV. EXPERIMENTAL RESULTS

A. Direct Learning

The output predictions of the LSTM network using the direct learning method is shown for both the tasks in Figure 5 and Figure 6. Task-1 has very good prediction accuracy because of the simplicity of the target and the obvious relation between the strain values and the finger deformation. The mean test accuracy of the task was 0.4 mm for a total deformation covering 10 mm. The sensor readings during the task are also shown in Figure 5. The drifting of the sensors can be observed. As the motion is one-dimensional, high correlation among the sensors can be found. In other words, for the given task, the sensor configuration is redundant. Hence, such tasks may not be appropriate for the developing the base model (model-0) for transfer learning. This is because the latent space might have a low-dimensional representation of the sensor space due to the redundancy in the base task.

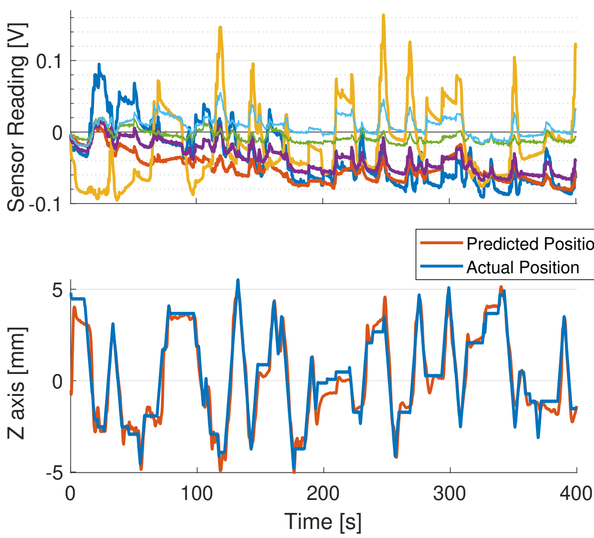

Fig. 5. Network performance for Task-1. The corresponding sensor values are shown above.

Task-2 is much more complex than task-1 and hence the performance of the learned model is poorer compared to Task-1. The mean test accuracy of the localization test was 27mm for a sliding surface of 140x80 mm size. The sensor readings during the task are also shown in Figure 6. Although, there are still sensor redundancies in this task, it will be less than task-1. Nonetheless, new sample data is obtained for learning the base model (model-0) to reduce the sensor correlations. This task-0 is simply the combination of task-1 and task-2, which means the finger moves in a three dimensional space of 140x80x10mm. The accuracy of this model is poor due to the complexity of the task, however, a drift-free latent space representation is still obtained for our purpose as show in the next subsection.

B. Latent Space Representation

After training the LSTM network on task-0 to obtain model-0, the weights of the all the layers preceding the

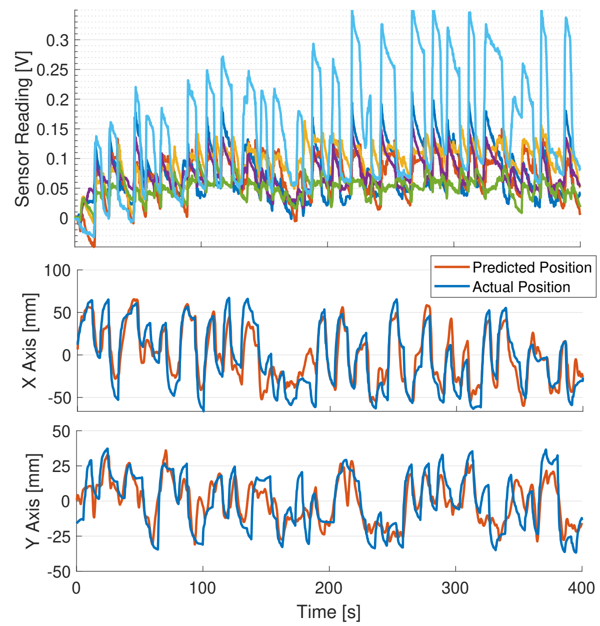

Fig. 6. Network performance for Task-2. The corresponding sensor values are shown above.

fully connected layer is fixed. This is essentially the dynamic component of the network. Providing new inputs from new tasks to this network would result in arbitrary output values and latent space values.

An example of the latent space values of the model-0, when sensor input from task-2 is given to the network is shown in Figure 7. This latent space representation is then fed to a fully connected layer (static layer) to obtain the desired outputs. In this way we are condensing the temporal information contained in the 6 sensors into a static variable of size 50. As the number of sensors increase, the advantage of such an approach becomes increasingly important. The human perceptive system contains a dense and redundant array of tactile receptors [15]. It is hence unnecessary to obtain all the information from these sensors for high-level processing. There are neuronal circuitry in the body that compresses the information from these mechanoreceptors. Likewise, the proposed transfer learning approach can be used to not only remove noise and drift-effects, but also to reduce the dimensionality of the sensory input provided to the high-level processing system [16].

The comparison of the performance between the two methods are shown in Figure 8. As expected, the direct learning approach leads to better accuracy. This is not surprising as the transfer learning method is severely limited by the outputs from the LSTM layer and the number of free parameters to tune. These differences can however be expected to drastically reduce when the sensor dimensionality increases, the task complexity increases and when data samples become scarce. It is already evident from the test results of task-2 as shown in Table I. The biggest difference between the two methods lie in the number of free parameters to be tuned as

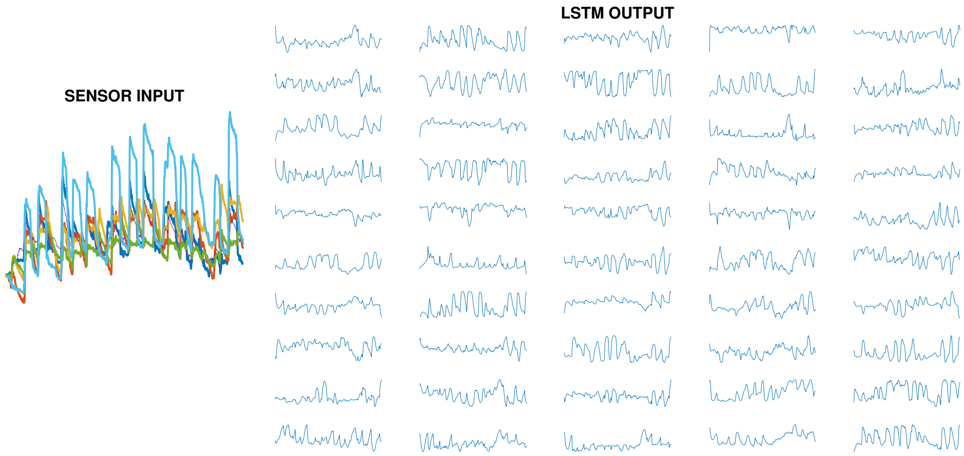

Fig. 7. Latent space output to the 6 dimensional sensor input sequence.

its subsequent requirement on the number of samples. For this study we use the same number of samples (30,000) for training the direct model and the transfer learning model, it can be deduced that the transfer learning approach would require much fewer samples. It must, however, be noted that the transfer learning approach requires a model-0 which is trained in the same was as the direct learning approach. Again, the advantages become more apparent when there are numerous tasks to be learned or the latent space representation is used for a new task where sample data is expensive to obtain. This is especially true for the case of direct reinforcement learning strategies for acquiring tactile manipulation skills [17].

TABLE I
COMPARING DIRECT TRAINING WITH TRANSFER LEARNING

	Direct training	**Transfer Learning**
No. Parameters (task 1)	11450	50
No. Parameters (task 2)	11500	100
Training Time (task 1)	344 s	93 s
Training Time (task 2)	355 s	97 s
Test accuracy (task 1)	0.43 mm	0.66 mm
Test accuracy (task 2)	27 mm	26 mm

V. CONCLUSIONS

This paper presents the concept of transfer learning for soft strain sensor modelling. The principle advantage of such an approach is its capability to quickly adapt to novel tasks, scenarios and conditions. This is valuable for learned models involving dynamic recurrent neural networks, which are data hungry and difficult to train. By reducing the number of free parameters, the generalizing capability of the network is also reduced. However, it is not significant with respect to the reduction in training effort as the first dynamic layers of the network has already been tuned to compensate for the complex time-varying noisy sensor data. The true potential of the approach would be more apparent with higher sensor dimensionality; as found in the human somatosensory system or in learning problems where each sample data point is expensive to obtain. Tactile sensor data is essential to close-the-loop for certain control tasks. Here, the mapping between the soft robot state and the control action cannot be easily framed as a regression problem (refer to [18] for a work around). These problems can be framed as a reinforcement learning problem, but each sample data point have to be obtained from real-world roll-outs which can last from few seconds to hours. This makes it almost impossible to tune all the parameters of a RNN, like the LSTM network shown here. The transfer learning approach shown here could be one of the methods obtain the state information about the robot without the additional complexity of tuning a RNN. Note that the first model can be easily learned using a regression task as described in this paper.

This work uses LSTM networks to learn our dynamic mapping from the sensors to the robot state, but the same approach can be applied to any other RNN architecture. We employed LSTMs due to their ease of training and their wide usage in such applications. Another viable candidate that can be particularly suited for this approach is the reservoir computing framework [19].

The design of sensor networks for particular tasks or in an optimal fashion is still a challenge not addressed in this work. We rely on human knowledge for embedding the sensors and through trial-and-error decode the sensory capabilities of the system. Future work would involve designing sensory architecture optimized for certain tasks either based on mathematical models or information theoretics.

ACKNOWLEDGMENT

The authors would like to thank Frank Clemens (EMPA) for providing assistance and for providing the sensor materials. This work was supported by the SHERO project, a Future and Emerging Technologies (FET) programme of the European Commission (grant agreement ID 828818).

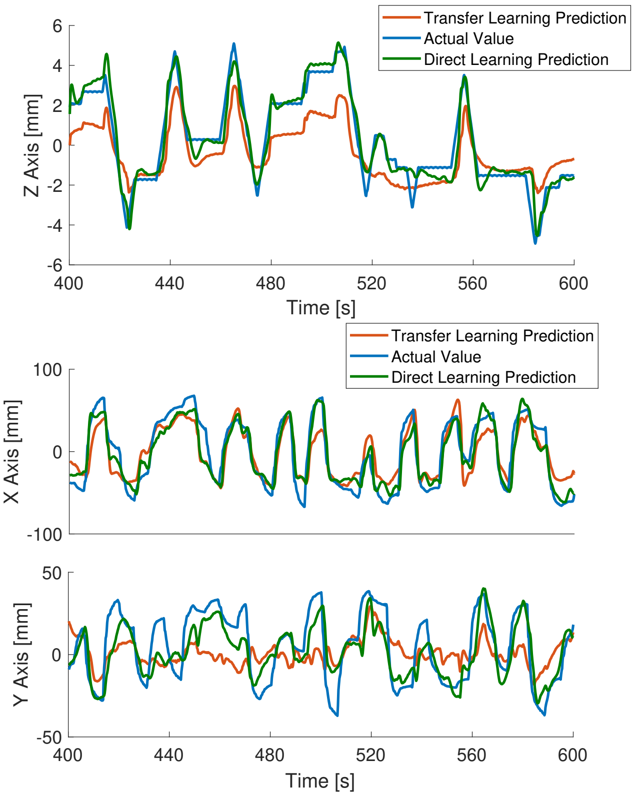

Fig. 8. Comparing the performance of the transfer learning approach with the direct learning approach for the two tasks. Task-1 is shown above and Task-2 is shown below.

REFERENCES

[1] T. G. Thuruthel, B. Shih, C. Laschi, and M. T. Tolley, "Soft robot perception using embedded soft sensors and recurrent neural networks," *Science Robotics*, vol. 4, no. 26, p. eaav1488, 2019.

[2] X. Niu, S. Peng, L. Liu, W. Wen, and P. Sheng, "Characterizing and patterning of pdms-based conducting composites," *Advanced Materials*, vol. 19, no. 18, pp. 2682–2686, 2007.

[3] N. Lu, C. Lu, S. Yang, and J. Rogers, "Highly sensitive skin-mountable strain gauges based entirely on elastomers," *Advanced Functional Materials*, vol. 22, no. 19, pp. 4044–4050, 2012.

[4] J. T. Muth, D. M. Vogt, R. L. Truby, Y. Mengüç, D. B. Kolesky, R. J. Wood, and J. A. Lewis, "Embedded 3d printing of strain sensors within highly stretchable elastomers," *Advanced Materials*, vol. 26, no. 36, pp. 6307–6312, 2014.

[5] L. Wang, F. Ma, Q. Shi, H. Liu, and X. Wang, "Study on compressive resistance creep and recovery of flexible pressure sensitive material based on carbon black filled silicone rubber composite," *Sensors and Actuators A: Physical*, vol. 165, no. 2, pp. 207–215, 2011.

[6] H. Wang, M. Totaro, and L. Beccai, "Toward perceptive soft robots: Progress and challenges," *Advanced Science*, vol. 5, no. 9, p. 1800541, 2018.

[7] U. Culha, S. Nurzaman, F. Clemens, and F. Iida, "Svas3: strain vector aided sensorization of soft structures," *Sensors*, vol. 14, no. 7, pp. 12 748–12 770, 2014.

[8] S. Han, T. Kim, D. Kim, Y.-L. Park, and S. Jo, "Use of deep learning for characterization of microfluidic soft sensors," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 873–880, 2018.

[9] D. Kim, J. Kwon, S. Han, Y.-L. Park, and S. Jo, "Deep full-body motion network for a soft wearable motion sensing suit," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 1, pp. 56–66, 2018.

[10] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[11] H. Van Hoof, T. Hermans, G. Neumann, and J. Peters, "Learning robot in-hand manipulation with tactile features," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 121–127.

[12] H. Van Hoof, N. Chen, M. Karl, P. van der Smagt, and J. Peters, "Stable reinforcement learning with autoencoders for tactile and visual data," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 3928–3934.

[13] T. Lesort, N. Díaz-Rodríguez, J.-F. Goudou, and D. Filliat, "State representation learning for control: An overview," *Neural Networks*, vol. 108, pp. 379–392, 2018.

[14] M. Melnykowycz, B. Koll, D. Scharf, and F. Clemens, "Comparison of piezoresistive monofilament polymer sensors," *Sensors*, vol. 14, no. 1, pp. 1278–1294, 2014.

[15] J. C. Stevens, "Aging and spatial acuity of touch," *Journal of gerontology*, vol. 47, no. 1, pp. P35–P40, 1992.

[16] O. Kroemer, C. H. Lampert, and J. Peters, "Learning dynamic tactile sensing with robust vision-based training," *IEEE transactions on robotics*, vol. 27, no. 3, pp. 545–557, 2011.

[17] S. Tian, F. Ebert, D. Jayaraman, M. Mudigonda, C. Finn, R. Calandra, and S. Levine, "Manipulation by feel: Touch-based control with deep predictive models," *arXiv preprint arXiv:1903.04128*, 2019.

[18] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, "Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 124–134, 2018.

[19] H. Jaeger, "Echo state network," *scholarpedia*, vol. 2, no. 9, p. 2330, 2007.