2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Supervised Autoencoder Joint Learning on Heterogeneous Tactile Sensory Data: Improving Material Classification Performance

Ruihan Gao[1,2], Tasbolat Taunyazov[1,3], Zhiping Lin[2], Yan Wu[1]

***Abstract*— The sense of touch is an essential sensing modality for a robot to interact with the environment as it provides rich and multimodal sensory information upon contact. It enriches the perceptual understanding of the environment and closes the loop for action generation. One fundamental area of perception that touch dominates over other sensing modalities, is the understanding of the materials that it interacts with, for example, glass versus plastic. However, unlike the senses of vision and audition which have standardized data format, the format for tactile data is vastly dictated by the sensor manufacturer, which makes it difficult for large-scale learning on data collected from heterogeneous sensors, limiting the usefulness of publicly available tactile datasets. This paper investigates the joint learnability of data collected from two tactile sensors performing a touch sequence on some common materials. We propose a supervised recurrent autoencoder framework to perform joint material classification task to improve the training effectiveness. The framework is implemented and tested on the two sets of tactile data collected in sliding motion on 20 material textures using the iCub RoboSkin tactile sensors and the SynTouch BioTac sensor respectively. Our results show that the learning efficiency and accuracy improve for both datasets through the joint learning as compared to independent dataset training. This suggests the usefulness for large-scale open tactile datasets sharing with different sensors.**

I. INTRODUCTION

Tactile sensing is arguably the first developed and most widely used sensation for humans to interact with the environment, given the widespread of nerve ending, corpuscles, and receptors that together sense temperature, pain, pressure, vibration and many other subtle sensations. Equipping robots with tactile sensing and understanding contributes another dimension to robots' perception other than conventional modalities such as vision and audition. By providing information of vibration, force, torque, and sometimes temperature through physical contact with the environment, the sense of touch is crucial for real-time planning and stable movement [1]. Perceptual understanding from rich tactile data also provides additional guarantee for safety, especially when robots are increasingly expected to move out of fixed workstations and interact closely with humans in our daily environment [2].

*This research is partially supported by the Agency for Science, Technology and Research (ASTAR) under its AME Programmatic Funding Scheme (Project #A18A2b0046).

[1]Robotics & Autonomous Systems Department, A*STAR Institute for Infocomm Research, Singapore. Email: {gao_ruihan,wuy}@i2r.a-star.edu.sg

[2]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Email: EZPLin@ntu.edu.sg

[3]School of Computing, National University of Singapore, Singapore. Email: ttaunyazov@u.nus.edu

Extensive research has been carried out in the tactile space including applications such as robotic manipulation [3] [4], object detection [5], and material classification [6]. More recent works focus on object identification [7] [8], slip detection [9] [10] [11], and grasp control [12] [13]. Furthermore, some works propose to integrate tactile sensing with vision to supplement the texture information with more apparent features such as shape and color [14] [15].

However, one challenge of tactile sensing is the lack of a unified representation. Unlike images that has a universal lattice RGB representation with pixel value ranging from 0 to 255, tactile sensors all have their unique design, implementation and distinct representation for properties such as pressure, vibration, and temperature. The scales of the sensor readings also vary according to the sensor mechatronics. Due to this huge variance, research in tactile understanding is grossly in silos. Each group tends to collect its own datasets from scratch and it is hard to draw any inference from other few publicly available datasets.

Common types of exploratory movements for tactile sensing are pressure, static contact, lateral sliding, etc [16]. As pressure and lateral sliding require time to execute, temporal information plays an important role in implicating the texture properties. However, the duration and the temporal resolution may vary across experiments. It is also laborious to properly align the signal samples, despite some possible but computationally expensive methods such dynamic time warping [17] and resampling with delay estimation [18].

In addition, although a few tactile sensors have been used in research, such as the Gelsight [19], iCub RoboSkin [20], and SynTouch BioTac®, each sensor has a unique implementation. To the best of our knowledge, no work is devoted to improve the learnability of data collected on one sensor using datasets from others. This in consequence limits the usefulness of open tactile datasets and generalizability of existing tactile representation models across sensors.

To explore the joint learnability and shareability of features learned from heterogeneous tactile sensors, we propose a supervised recurrent autoencoder framework to perform joint material classification task. Two datasets are used for validation – one collected with the BioTac sensor and one open dataset on iCub RoboSkin [21]. The joint training model is benchmarked against independently trained ones. While the RoboSkin dataset was collected for the purpose of validating the effectiveness of the representation model under a loosely controlled setup with a cost-effective sensor which has higher noise in the dataset, the BioTac dataset is aimed to provide a clean dataset at the other end of the spectrum

under strict control of the setup and procedure with a highly precise sensor. Our findings show that the compressed latent representation can be shared and used to boost the learning efficiency and accuracy. In summary, the main contributions of this paper are threefold:

- A dataset consists of 20 materials with 50 samples each is made publicly available;
- A recurrent autoencoder unit that compresses both spatial and temporal information is proposed and tested on the two datasets;
- A supervised autoencoder joint learning framework is proposed and validated to improve the learning performance as compared to independent training.

The rest of this paper is organized as follows: In Section II, related works on texture classification are reviewed. The proposed supervised recurrent autoencoder framework is described in Section III, together with implementation details of independent and joint training approaches. Section IV presents the experimental results and analysis. Section V concludes the paper with future work.

II. RELATED WORK

The early attempts to recognize tactile features mainly apply signal processing techniques. In [22], FFT and a learning vector quantization technique discriminate the sound signals produced when an electret piezoelectric microphone moves through a material. In [6], a bio-mimetic approach is used to capture the changing rate of stimuli for material classification with strain gauges and polyvinylidene fluorides.

Along with the widespread of machine learning methods, more feature-based algorithms have been applied to automatic texture classification. Optimal parameters (velocity and force) of exploratory movements have been chosen based on Bayesian exploration; interpretable features like traction, fineness, roughness are hand-crafted based on the sensor readings [23]. In [24], further improvement is achieved by taking multi-modal readings (force, vibration, and temperature) of the sensor into consideration.

Other ways of interpreting the tactile signal have also been proposed. For example, readings of tactile sensor units in a sensor array can be treated as pixels in an image with some signal processing technique; consequently, taxel image can be formed from discrete sensor readings. For example, [25] makes use of deep convolutional neural network to process the signals and achieves classification accuracy up to 97.3%. Integrated representation has also been explored by combining deep convolutional features of camera images and Gelsight sensor data [26] with a proposed data compression method called Deep Maximum Covariance Analysis (DMCA). To further improve the classification results, [21] proposes to incorporate different exploratory movements on the same materials learned by a Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM), achieving an accuracy of 98%.

In summary, various tactile sensors have been designed to model the environment and the main focus of texture classification would arguably lie in constructing abundant features from sensor readings and formulating efficient algorithms, statistical models or more intriguing networks, to capture the distinguished characteristics for each label. While most models stand with their own merits, however, to the best of our knowledge, little attention has been devoted to making inferences of datasets from different tactile sensors which makes prior work difficult to use.

III. METHODOLOGY

Borrowing the approach of domain adaptation [27], in this work, we hypothesise that, at a more abstract level, there exists a representation that distinguishes material textures and is common to data collected from different sensors through a common exploratory action within some noise limit. The following sections present and discuss our approach, experiment design, and implementation to find such latent representation that may enhance model performance and enable datasets from different sensors to be jointly learned.

To examine our hypothesis, we propose and evaluate a Recurrent AutoEncoder framework with Classifier (RAEC). It comes with a "header" network that is unique to receive outputs from different sensor. This network can be treated as a pre-processing unit to extract basic common features. The basic structure of the framework is shown in Fig. 1, where E and D stand for the header at the encoder and decoder side, respectively.

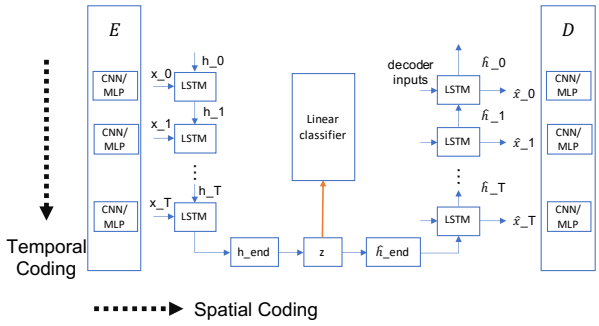

Fig. 1: The recurrent autoencoder framework with classifier

Based on this basic framework, we explore variation on training steps and the amount of data provided for the dataset. The following subsections discuss the framework and training procedures in more details.

A. Raw Input Features

For the texture classification task, all models are trained to predict a label y within C classes. The input data are first passed to the corresponding "header" network and transformed to features. We choose lateral sliding as the common exploratory movement, of which each feature comes in as a sequence of $x_i(t)_{t=0}^{t=T-1}$, where T is the total sequence length. Since multiple features are present in both BioTac and RoboSkin sensors, one input data can be expressed as a set of input features, i.e. $X_i = \{x_i(t)\}_{t=0}^{t=T-1}$. The ground-truth y for each input is labelled according to the given material.

B. Header Network

The "header" networks are attached to the front of the encoder and mirrored to the end of the decoder respectively. They are designed to adapt to unique input data from different sensors. Different neural network blocks have been tested. Convolutional Neural Network (CNN) is chosen for RoboSkin data to capture features in taxel images reproduced from the Skin readings. Its model parameters for each layer follow the original implementation in [21]. For BioTac data, however, since the 19 electrodes are sparsely and not evenly distributed which are less reasonable to be treated as an image, a simple multi-layer perceptron (MLP) is implemented as the header.

C. Recurrent Autoencoder

Instead of implementing a direct feedforward network, we propose a supervised recurrent autoencoder that transforms the input to a latent representation and classifies the object with a simple linear predictor. The supervised learning setting with autoencoder has been shown to regularize the solution and to enhance the stability of representation learning [28]. The encoder-decoder structure forces the network to distill the input so that only important information is kept in the latent representation, shown as z in Fig. 1. This distillation facilitates data compression and noise reduction. In this work, we also add LSTM blocks to form a recurrent autoencoder. Featuring its gating mechanism, LSTM has the edge over processing sequential data and preserving long term dependencies. Its forget gate automatically filters out trivial information and the input gate adds new information in; the cell state represents the memory of LSTM and is finally converted to output with activation function. We believe that embedding the spatial and temporal information in one latent vector assists to handle phase shift and to capture the repetitive pattern one may encounter during sliding motion. Therefore, it automatically facilitates the time alignment between different models, eliminating the need to perform one-to-one rigorous mapping among sequences.

In this work, the number of LSTM cells are set according to the input sequence length, i.e. 400 for BioTac data and 75 for RoboSkin data. All cells have a hidden size of 90 and a hidden layer of depth 1, and the size of the latent vector z is chosen to be 40.

D. Classifier

Based on the autoencoder structure, it is assumed that the latent vector z represents well the intrinsic texture features. Therefore, it is mapped to a distribution over C classes (20 in this work) with a linear layer and the model is trained to minimize multi-class cross-entropy loss.

$$\text{classification loss} = -\sum_{i=1}^{N}\sum_{c=1}^{C}\mathbb{I}(c, y_i)\log(p_{X_i,c}) \quad (1)$$

where y_i is the ground truth label for sample i, $\mathbb{I}(\cdot,\cdot)$ is a binary indicator function, and $p_{X_i,c}$ represents the predicted probability of the class c for sample X_i, which is obtained from a softmax layer.

E. Independent and Joint Training

As aforementioned, the BioTac data and RoboSkin data use different header networks. For independent training, each dataset is separately trained with a weighted combination of reconstruction loss and classification loss. The reconstruction loss minimizes the difference between the decoded output and the input data, and is designed to capture the distinctive input pattern. It has also been shown to improve generalizability as a method of regularization [28]. The classification loss is computed as negative log likelihood loss between the output and the ground-truth label.

To testify the interoperability of the latent representation, joint training is carried out between two models, i.e. BioTac model and RoboSkin model, in a semi-supervised approach. While the network structures remain intact for both models, the loss function is modified to also include a mean square error (MSE) loss term to compute the difference between the two latent representations given the same ground-truth label. In order to ensure the stability of training, the MSE loss only contributes to the training after the first 20 epochs. This additional loss aims to minimize the disparity between the two latent representations and therefore intends to bond them together or more likely draw them to some shared hyperspace where the materials can be differentiated by more abstract and generalizable features. Fig. 2 shows the block diagram and weighted losses; the dotted line of MSE loss is only added for joint training.

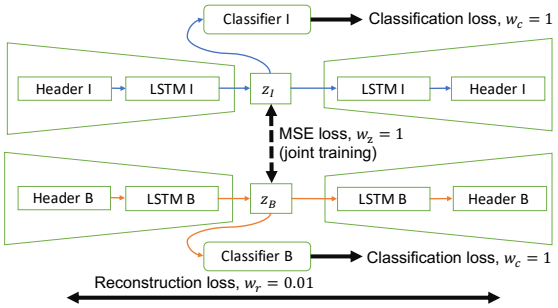

Fig. 2: Block diagram with loss weightage

All the methods were implemented in Python using the PyTorch machine learning library [29]. The models are trained to minimize the loss functions as follows:

For independent training,

$$\begin{aligned}\mathbb{L}_{\mathbb{B},\mathbb{I}} &= \text{reconstruction loss} + \text{classification loss} \\ &= w_r \frac{1}{N}\sum_{i=1}^{N}(X_i - \hat{X}_i)^2 \\ &+ w_c \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\mathbb{I}(c, y_i)\log(p_{X_i,c})\end{aligned} \quad (2)$$

Eqn. (2) consists of a reconstruction loss to penalize disparities between the decoder output $\hat{X}_i$ and the input

data X_i, and a cross entropy loss to penalize classification error. N represents the number of input data in a mini-batch. w_r, w_c are weights for the reconstruction loss and the classification loss, respectively. They are chosen to be 0.01 and 1 in order to scale two losses into similar order of magnitude.

For joint training, the loss function is the same as independent training for the first 20 epochs and back-propagate along individual models separately. After 20 epochs, the MSE loss for two latent representations is added as shown in Eqn. (3):

$$\mathbb{L} = \mathbb{L}_{\mathbb{B}} + \mathbb{L}_{\mathbb{I}} + w_z \mathbb{L}_{\mathbb{C}} = \mathbb{L}_{\mathbb{B}} + \mathbb{L}_{\mathbb{I}} + w_z \frac{1}{N} \sum_{i=1}^{N} (z_B - z_I)^2 \quad (3)$$

where $\mathbb{L}_{\mathbb{B}}$ and $\mathbb{L}_{\mathbb{I}}$ represent the individual loss for BioTac model and iCUb model. $\mathbb{L}_{\mathbb{C}}$ represent the MSE loss between two latent spaces, z_B for BioTac model and z_I for iCub model. w_z represents its weight, which is set to 1.

All data are divided as 60% for training, 20% for validation, and 20% for testing. The Adam optimizer [30] is used with a learning rate of 0.0005. All models are trained for a maximum of 2000 epochs, with a batch size of 32 samples. The finalized model is chosen based on the highest validation accuracy. A dropout rate of 0.2 is used to avoid over-fitting.

After training all the models, the accuracy (Acc) on the test dataset is computed and used as the evaluation metric.

$$\text{Acc} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y_i})^2 \quad (4)$$

where y_i and $\hat{y_i}$ are the prediction output and the ground-truth label respectively.

IV. EXPERIMENTS

This section explains the details of our experiments, from robot set up for data collection, independent and joint training, to data reduction testings.

A. Data Collection Setup

We hypothesize that sliding is essential to tactile perception as it provides sequential feedback that can be used to infer the pattern and texture. Therefore, in addition to the online RoboSkin dataset [21], we set up the experiment to collect lateral sliding data with BioTac sensor for material texture classification.

The BioTac sensor is attached as a passive end-effector on the KUKA LBR iiwa 14 manipulator. The movement is designed in such a way that the BioTac sensor gradually comes into contact with the material and then slides with a constant linear velocity and a constant contact force controlled by the on-board controller. This predefined motion path is repeated for 20 different types of materials with 50 samples each. The material snapshots are shown in Fig. 3. The dataset is accessible* with all modalities of BioTac sensor, together with algorithmic and auxiliary robot information.

*https://dexrob.github.io/dexrob/supervised_ae_iros_2020/

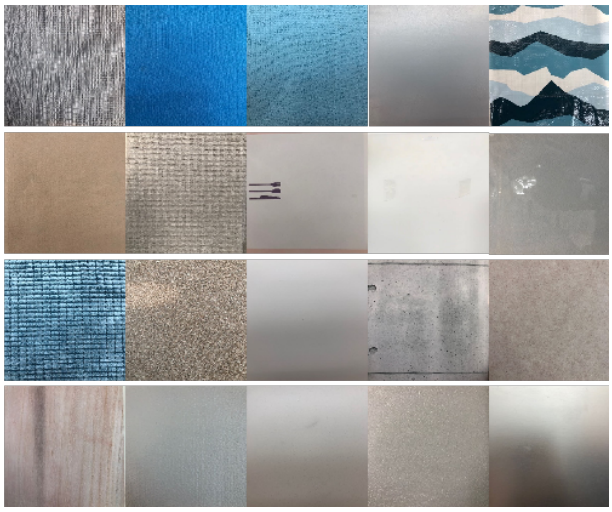

Fig. 3: Snapshots of 20 materials used in this work (left to right, top to bottom): carpet net, cotton, bath towel, leather fake, polypropileno, felt, soft material, paper 2, polypropileno smooth, polypropylene thin, soft material, cork, eva, paper 1, fiber board, wood hard, styrofoam, sponge soft, foam, metal

1) Constant Force Control: Each material is firmly attached to a metal plate placed on a levelled platform within the robot's workspace. According to [23], the change in fluid pressure of the BioTac sensor is linear to the magnitude of the contact force within the range of $0N - 2N$, as shown in Fig. 4.

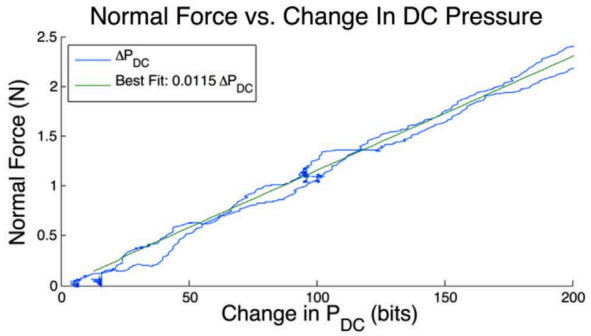

Fig. 4: Relationship between the normal force and the change in P_{DC} reading [23]

This linear property is used to set a conditional execution of the sliding movement. In general, the KUKA robot executes position control and follows a path defined in Cartesian space consisting of the following segments. It starts from a global home position, moves to a set-down point around 1cm above the material, and gradually lower down with a linear speed of 0.5mm/s until the P_{DC} reading of the BioTac sensor reaches a certain threshold (40 in the current setting) and triggers the sliding motion. The sliding is executed with a

linear motion command to a predefined target at a distance of 20cm from the starting point. A sample of the force profile during the slide is shown in Fig. 5. It is shown that the pressure increases over 40 and the friction changes from static to dynamic and remains approximately constant as the sensor slides through different parts of the material. Finally, the robot arm is lifted by 10cm and eventually brought back to the global home position.

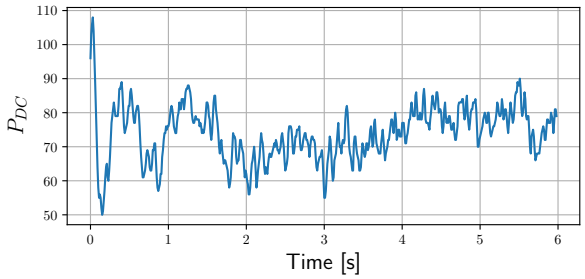

Fig. 5: A sample force profile

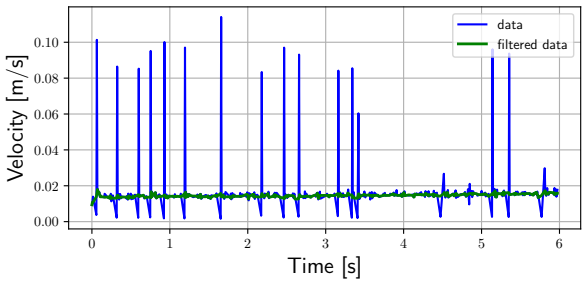

Fig. 6: A sample velocity profile

The robot and sensor setup is shown in Fig. 7 with the full motion path drawn in color for demonstration.

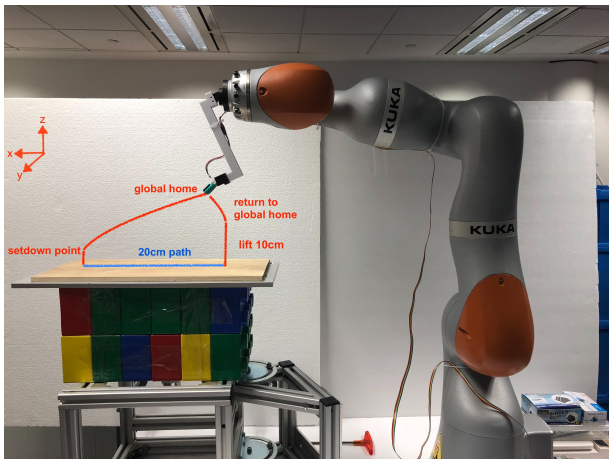

Fig. 7: Robot setup and predefined motion path

2) Velocity Control: Since the robot executes position control, the velocity is controlled by setting a proper speed limit in Cartesian space. The Cartesian frame is shown in the upper left corner of Fig. 7. The downward motion is executed in z direction in a speed of 0.5mm/s and the horizontal sliding is executed in x direction with a maximum speed limit of 2.5cm/s, following the optimal setting stated in [23].

The actual Cartesian velocity is calculated from the joint velocity readings using KUKA iiwa's Jacobian matrix. A snapshot of the velocity profile for the linear sliding motion is shown in Fig. 6, where the blue line represents the raw data, and the green line represents the processed data filtered with a median filter with a kernel size of three. It is noted that after taking consideration of a few spikes that can be explained by joint control, the speed can be approximated as a constant value relatively smaller than the maximum limit.

3) BioTac Reading: As shown in Fig. 8, the BioTac sensor is a mult-modal tactile sensor that senses the deformation of the elastomeric skin through changing impedance of the 19 electrodes when the conductive fluid path diverts [31]. There are other modalities such as the vibration and temperature, but they are currently not used in this work. At the beginning of each experiment, the sensor is calibrated to ensure consistent readings. The sensor sampling rate is $100Hz$. For each collection, the start and end timing of the sliding motion are recorded to crop the sensor readings. Each sliding motion lasts about 8s and only the mid-400 frames are used the dataset to remove any transient data. The whole system is controlled in Robot Operating System (ROS).

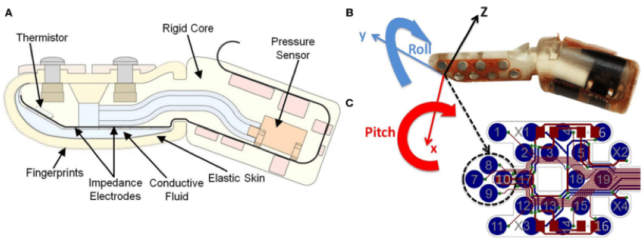

Fig. 8: Schematic diagram of BioTac sensor and the distribution of its 19 electrodes [32]

B. Independent and Joint Training

In this experiment, we take reference of the state-of-the-art CNN-LSTM architecture proposed in [21] and use it as baseline. Similar architecture is reproduced for BioTac data by replacing the CNN layers with a single layer perceptron. Each experiment is repeated for 4 times with different splitting of training and testing datasets and the results are shown in Table I. Note that the accuracy for RoboSkin data is slightly better than what was reported in [21] and that may be due to the upgrade of library packages and different parameters chosen for LSTM.

The X in $X-LSTM$ represents either CNN or MLP network for RoboSkin and BioTac respectively, equivalent to the "header" network used in RAEC. It is shown that compared to directly applying LSTM, adding the autoencoder structure to transform the data to a latent space improves the performance by about 1% for BioTac data and 3% for RoboSkin data. The more significant improvement for RoboSkin can be explained because RoboSkin data is noisier

TABLE I: Results of three architectures for BioTac dataset and RoboSkin dataset

Method	Acc_B	Acc_I
X-LSTM	94.50%	90.00%
RAEC (independent)	95.31%	91.14%
RAEC (joint)	**96.01%**	**93.23%**
Mapping to known space	94.79%	91.67%

TABLE II: Results for swapping classifier test

	Acc_B	Acc_I
Before swapping	96.01%	93.23%
After swapping	95.31%	93.23%

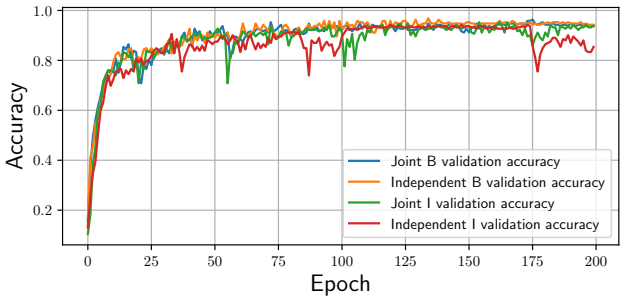

Fig. 9: Validation accuracy comparison between independent training and joint training

bearing a more loosely constrained experiment setup [21]; therefore autoencoder helps to remove the noise of the data.

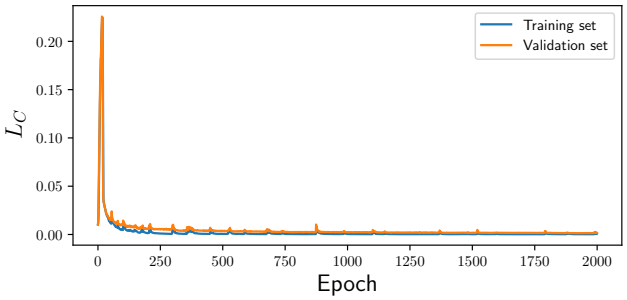

Fig. 10: MSE loss of latent embedding of two sensors during joint training

The comparison between RAECs obtained by independent training and joint training shows that binding the latent representation of two datasets actually boost the performances of both. The accuracy increases by about another 1% and RoboSkin model seems to gain more benefits in joint training. In terms of learning speed, Fig. 9 presents the validation accuracy for the first 200 training epochs and it is shown that both models converge faster with joint training. Fig. 10 also shows that the MSE of latent embedding of two sensors are minimized during joint training process of 2000 epochs.

A complementary experiment has also been conducted as follows. One RAEC is trained independently and then fixed; then the other RAEC is trained in an unsupervised way, i.e. only the reconstruction loss and MSE loss between two latent spaces are minimized. The result is shown in the last row of Table I. It is noticed that RoboSkin model improves, while BioTac model degrades by mapping to the other known latent space, which is explainable by our observation that BioTac data is cleaner than RoboSkin data.

A quick test of swapping the linear classifier between two models have also been implemented. As shown in Table II, after interchanging two classifiers, the performance is retained. It helps to substantiate our claim that a unified representation between two sensor datasets can be achieved.

C. Data Reduction Test

To further explore how joint training can contribute to the learning process, a data reduction test is designed to gradually reduce the amount of training data of one of the datasets. Specifically, one set of data are kept as a whole, while the other is gradually reduced to a portion (training ratio) of 90%, 50%, 25%, 17%, and 11%. During the training, the model with full data is always updated, while the other is updated time to time, depending on the training ratio. The results are presented in Table III and Table IV respectively. Two independent models are also tested with reduced training data and the results are listed as the last column for the sake of comparison.

TABLE III: Partial reduction on BioTac dataset

Training Ratio	Acc_B	Acc_I	$Acc_B(independent)$
90%	94.27%	93.75%	93.75%
50%	90.63%	93.23%	91.67%
25%	83.33%	93.75%	86.46%
17%	77.08%	91.67%	78.65%
11%	70.31%	92.19%	69.79%

TABLE IV: Partial reduction on RoboSkin dataset

Training Ratio	Acc_B	Acc_I	$Acc_I(independent)$
90%	94.27%	91.15%	90.62%
50%	96.88%	86.98%	82.29%
25%	96.88%	73.96%	71.35%
17%	94.28%	58.85%	57.81%
11%	96.88%	45.31%	46.35%

It is observed that performance drops generally against any data reduction. However, it is worth noting that the impact of data reduction on the BioTac dataset is higher for the RoboSkin dataset and the complete opposite for the other way round. We hypothesise that the difference in noise level present in the datasets contributes to such outcomes. This further suggests that heterogeneous tactile datasets taken from different sensors on the same tasks can be combined using the proposed joint training approach to improve training results provided that the available datasets contain clean and/or sufficiently large number of training samples.

V. CONCLUSIONS

In this study, we demonstrate that accurate and efficient texture classification for a collection of mundane materials

can be achieved with the proposed recurrent autoencoder framework. In addition, jointly training two models with diverse input data collected by different tactile sensors can complement each other and boost the performances of both in terms of accuracy and converging rate. The obtained latent vector can arguably capture and compress the temporal and spatial information as a succinct representation which provides information that can be leveraged to facilitate the learning process of other models. We also provided a complete multi-modality dataset for lateral sliding of the BioTac sensor.

Based on the current data and framework, future improvements can be made to further enhance the robustness and generability. Multi-modality integration can be explored and unique features such as temperature may introduce a new dimension to the latent representation. Current network structures do not share parameters at LSTM layers because one-to-one mapping in temporal scale is hard for two sensors. Networks relying less on temporal scale, like Spiking Neural Network (SNN) can be explored in the future. Moreover, if new sensor is available, the model can be extended to include more diverse input, and evaluate the applicability of the latent representation, which promisingly indicates explainable tactile features that are shared by and transferable among various tactile sensors.

REFERENCES

[1] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini, "Tactile sensing—from humans to humanoids," *IEEE transactions on robotics*, vol. 26, no. 1, pp. 1–20, 2009.

[2] E. Broadbent, "Interactions with robots: The truths we reveal about ourselves," *Annual review of psychology*, vol. 68, pp. 627–652, 2017.

[3] J. Tegin and J. Wikander, "Tactile sensing in intelligent robotic manipulation–a review," *Industrial Robot: An International Journal*, 2005.

[4] J. Sanchez, C. M. Mateo, J. A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Online shape estimation based on tactile sensing and deformation modeling for robot manipulation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 504–511, IEEE, 2018.

[5] T. Tsujimura and T. Yabuta, "Object detection by tactile sensing method employing force/torque information," *IEEE Transactions on robotics and Automation*, vol. 5, no. 4, pp. 444–450, 1989.

[6] N. Jamali and C. Sammut, "Majority voting: Material classification by tactile sensing using surface texture," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 508–521, 2011.

[7] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard, "Object identification with tactile sensors using bag-of-features," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 243–248, IEEE, 2009.

[8] G. Heidemann and M. Schopfer, "Dynamic tactile sensing for object identification," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 1, pp. 813–818, IEEE, 2004.

[9] Z. Su, K. Hausman, Y. Chebotar, A. Molchanov, G. E. Loeb, G. S. Sukhatme, and S. Schaal, "Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 297–303, IEEE, 2015.

[10] R. Fernandez, I. Payo, A. S. Vazquez, and J. Becedas, "Micro-vibration-based slip detection in tactile force sensors," *Sensors*, vol. 14, no. 1, pp. 709–730, 2014.

[11] T. Taunyazoz, W. Sng, H. H. See, B. Lim, J. Kuan, A. F. Ansari, B. Tee, and H. Soh, "Event-driven visual-tactile sensing and learning for robots," in *Proceedings of Robotics: Science and Systems*, July 2020.

[12] J. M. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K. J. Kuchenbecker, "Human-inspired robotic grasp control with tactile sensing," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1067–1079, 2011.

[13] Y. Bekiroglu, K. Huebner, and D. Kragic, "Integrating grasp planning with online stability assessment using tactile sensing," in *2011 IEEE International Conference on Robotics and Automation*, pp. 4750–4755, IEEE, 2011.

[14] P. K. A. A. T. Miller and P. Y. O. B. S. Leibowitz, "Integration of vision, force and tactile sensing for grasping," *Int. J. Intell. Mach.*, vol. 4, pp. 129–149, 1999.

[15] F. Sun, C. Liu, W. Huang, and J. Zhang, "Object classification and grasp planning using visual and tactile sensing," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 969–979, 2016.

[16] Z. Kappassov, J.-A. Corrales, and V. Perdereau, "Tactile sensing in dexterous robot hands," *Robotics and Autonomous Systems*, vol. 74, pp. 195–220, 2015.

[17] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series.," in *KDD workshop*, vol. 10, pp. 359–370, Seattle, WA, 1994.

[18] S. E. Schmidt, K. Emerek, A. S. Jensen, C. Graff, J. Melgaard, J. J. Struijk, *et al.*, "Temporal alignment of asynchronously sampled biomedical signals," in *2016 Computing in Cardiology Conference (CinC)*, pp. 1–4, IEEE, 2016.

[19] W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, p. 2762, 2017.

[20] A. Schmitz, M. Maggiali, L. Natale, B. Bonino, and G. Metta, "A tactile sensor for the fingertips of the humanoid robot icub," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2212–2217, IEEE, 2010.

[21] T. Taunyazov, H. F. Koh, Y. Wu, C. Cai, and H. Soh, "Towards effective tactile identification of textures using a hybrid touch approach," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 4269–4275, IEEE, 2019.

[22] W. W. Mayol-Cuevas, J. Juarez-Guerrero, and S. Munoz-Gutierrez, "A first approach to tactile texture recognition," in *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218)*, vol. 5, pp. 4246–4250, IEEE, 1998.

[23] J. A. Fishel and G. E. Loeb, "Bayesian exploration for intelligent identification of textures," *Frontiers in neurorobotics*, vol. 6, p. 4, 2012.

[24] D. Xu, G. E. Loeb, and J. A. Fishel, "Tactile identification of objects using bayesian exploration," in *2013 IEEE International Conference on Robotics and Automation*, pp. 3056–3061, IEEE, 2013.

[25] S. S. Baishya and B. Bäuml, "Robust material classification with a tactile skin using deep learning," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8–15, IEEE, 2016.

[26] S. Luo, W. Yuan, E. Adelson, A. G. Cohn, and R. Fuentes, "Vitac: Feature sharing between vision and tactile sensing for cloth texture recognition," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2722–2727, IEEE, 2018.

[27] A. Rozantsev, M. Salzmann, and P. Fua, "Beyond sharing weights for deep domain adaptation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 4, pp. 801–814, 2018.

[28] L. Le, A. Patterson, and M. White, "Supervised autoencoders: Improving generalization performance with unsupervised regularizers," in *Advances in Neural Information Processing Systems 31* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), pp. 107–117, Curran Associates, Inc., 2018.

[29] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[31] L. Chia-Hsien, J. Fishel, and G. Loeb, "Estimating point of contact, force and torque in a biomimetic tactile sensor with deformable skin," *Tech. Rep.*, 2013.

[32] Z. Su, J. A. Fishel, T. Yamamoto, and G. E. Loeb, "Use of tactile feedback to control exploratory movements to characterize object compliance," *Frontiers in neurorobotics*, vol. 6, p. 7, 2012.