Single to Multi: Data-Driven High Resolution Calibration Method for Piezoresistive Sensor Array

Min Kim, Hyungmin Choi, Kyu-Jin Cho, and Sungho Jo

***Abstract*—Accurately detecting multiple simultaneous touches is crucial for various applications using piezoresistance sensor arrays. However, calibrating them is difficult due to their nonlinearity and hysteresis. While data-driven deep learning approaches could model complex sensor patterns, the required amount of labeled data increases exponentially as the number of contact points or sensor subelements increases. In this letter, we propose a novel supervised learning framework, Local Message Passing Network, that only needs single touch data to calibrate multiple contact points into a high resolution pressure map. The individual sub-local networks eliminate domain shift problems, while a message passing mechanism enables them to correctly learn correlations between neighboring sensor subelements. The performances of the proposed model were tested on labeled single- and double-pressure data and compared with previous deep learning calibration methods. Experimental results show that our framework can expand prior knowledge of single touch data to calibrate multi-touch sensor inputs into high resolution pressure maps.**

***Index Terms*—Calibration and identification, deep learning methods, force and tactile sensing.**

I. Introduction

With the ability to detect externally applied pressure from changes in electrical resistance, piezoresistive sensor arrays have been widely used in the industrial fields [1]–[5]. However, electromechanical hysteresis and non-linearity are reported to be major drawbacks of resistance-type sensors [6], [7]. While there exists conventional calibration models used to calibrate tactile sensor such as linear regression, polynomial regression and decision trees [8]–[10], these drawbacks make them hard to accurately calibrate sensor values into general pressure units.

Furthermore, the problems get even more complicated when we calibrate an array-like tactile sensor consisting of multiple sensor subelements. When using a tactile sensor, users do not always press the center of the sensor subelement and pressures are applied in between sensor subelements. Therefore, for a tactile sensor having an array-like structure, an ideal calibration method should be able to distinguish pressure applied in between subelements and generate a pressure map having a higher resolution than the sensor structure.

Meanwhile, deep neural networks, as an important breakthrough in machine learning, have significantly improved the ability to model complex patterns. Naturally, several deep neural networks have been used to model hysteresis and non-linearity to accomplish various applications such as learning human body motions [11], [12], hand gestures [13], [14] and classifying contact objects [15], [16]. For general purpose calibration, different learning-based models, such as RBF-Net [17], recurrent neural network (RNN) [18], convolutional neural network [19], VGG-16 [20] are used to model complex behavior of soft sensors.

Unfortunately, these previous calibration approaches were built on an assumption that only a single point of pressure is applied to the sensor. This assumption greatly limits the range of applications these models can be used for. As illustrated in Fig. 1(a), multiple contact points are crucial for fully understanding their interactions with surrounding objects. Furthermore, as demonstrated in Fig. 1(b), pressures can be applied to the side of a sensor subelement. The sensor behavior patterns change even with a slight shift in pressing location due to different sensor deforming patterns. However, multi-touch calibration is practically difficult when using learning-based approaches. Unlike conventional approaches that separately calibrate individual sensor subelements, learning-based approaches regard the entire sensor values as a single input vector. If we simply calibrate subelements separately with individual learning-based models, it will lose its ability to calibrate pressure applied in between subelements.

A naive approach for calibrating multi-touch using a learning-based approach would be to collect all different cases of simultaneous touches and use them for training a model. However, if we were trying to obtain two simultaneous touches from the 5×5 matrix-like sensor depicted in Fig. 1(c), the required number of labeled data increases exponentially as the number of simultaneous pressing combinations increases. As a result, fully training a model this way would need an infeasible amount of memory to store and computational time to process the data. An approach that avoids collecting multi-touch data and only uses a single touch data to train the model would be invaluable, but existing models have found this task to be arduous as the

Manuscript received December 17, 2020; accepted March 22, 2021. Date of publication April 2, 2021; date of current version April 19, 2021. This letter was recommended for publication by Associate Editor G. Huang and Editor L. Pallottino upon evaluation of the reviewers' comments. This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (MSIT) under Grant NRF-2016R1A5A1938472. *(Min Kim and Hyungmin Choi contributed equally to this work.) (Corresponding authors: Kyu Jin Cho; Sungho Jo.)*

Min Kim and Sungho Jo are with the School of Computing, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Republic of Korea (e-mail: m-kim@kaist.ac.kr; shjo@kaist.ac.kr).

Hyungmin Choi and Kyu-Jin Cho are with the Biorobotics Laboratory, Department of Mechanical Engineering, Seoul National University, Seoul 08826, Republic of Korea (e-mail: tndhals@snu.ac.kr; kjcho@snu.ac.kr).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2021.3070823, provided by the authors.

Digital Object Identifier 10.1109/LRA.2021.3070823

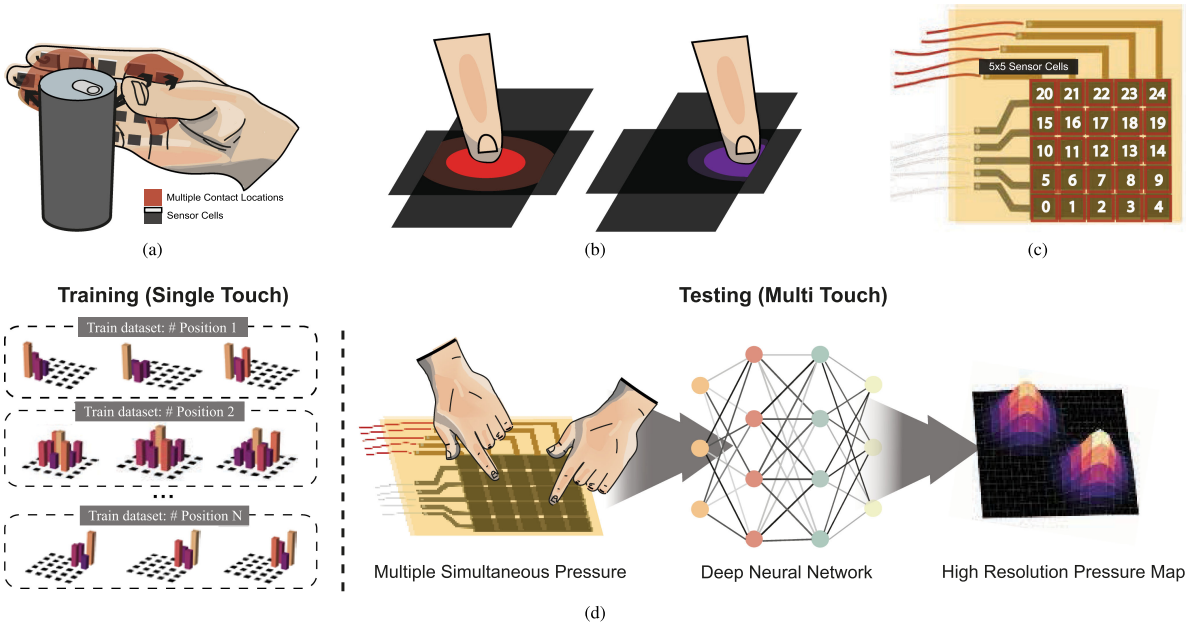

Fig. 1. Motivation and design of the LoMP algorithm. (a) Schematic depiction of the multi contact applications of sensor arrays. (b) Illustration of different pressing locations within a subelement. (c) 5 × 5 Piezoresistive sensor array with subelement numbers. (d) Design of the proposed LoMP training and testing framework.

data distributions of the training dataset and the true dataset are different. Differences in the distributions of the single touch and multi touch datasets cause a domain shift [22] so that the model is unable to clearly distinguish multiple touches.

In this article, we introduce Local Message Passing Network (LoMP), a novel supervised learning framework. Trained only with single touch data, the model is designed to calibrate multi touch cases at testing time as illustrated in Fig. 1(d). Instead of creating a network that takes all sensor values at a time, we created several small subnetworks that only consider local sensor inputs. Each subnetwork within LoMP is only responsible for calibrating pressure applied to a designated subarea of the sensor. We present our model can calibrate sensor values correctly even when multiple points of pressure are applied to several areas. With message passing mechanism, our subnetworks utilize neighboring messages to identify the pressure locations among sensor subelements.

The rest of the letter is organized as follows. Section II presents the sensor information and data acquisition setting. Section III explains the proposed Local Message Passing Network. Section IV discusses the performance evaluation of the proposed methods. Finally, in Section V we conclude the letter and present future works.

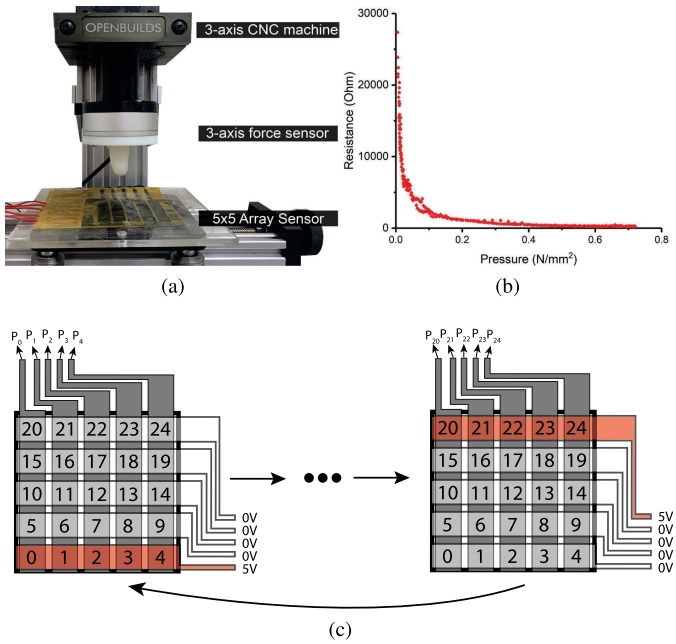

Fig. 2. Data Acquisition Methods. (a) Experiment setup for training data collection.(b) Characteristic curve of piezoresistive film (c) Illustration of the sensor value reading process.

II. Preliminary

A. Used Sensor Information

To demonstrate the effectiveness of our methods, we produced a 5 × 5 soft pressure matrix sensor with Velostat (3 M Co.), a piezoresistive material that is both flexible and low-cost. The piezoresistive characteristic curve of the Velostat has been measured using 1cmx1cm sandwich shape sensor design and voltage divider circuit proposed in [21] (Fig. 2(b)). The sensor comprises three layers: two electrode layers cover the middle piezoresistive layer. The overall size is 115 mm x 115 mm with less than 0.2 mm thickness. The sensing area of the sensor is 60 mm x 60 mm, with each subelement size of 11.5 mm x 11.5 mm.

B. Data Acquisition

The experiment setup for data acquisition is depicted in Fig. 2(a). The sensor was placed on a flat acrylic plate to avoid

the shape of the sensor becoming distorted from the unsmooth surface of the tabletop of a CNC milling machine (MiniMill, OpenBuilds). A three-axis commercial load cell (RFT60-HA, Robotous) was attached to the middle of cylindrical indenter to obtain true pressure values. The cylindrical indenter is covered by silicon coating made of Dragon Skin 10 to create a human finger-like indenter.

The sensor values are recorded through an Arduino device with the operation sequences depicted in Fig. 2(c). Each row of sensor is connected to the digital output of the MCU(Arduino Leonardo), and each column is connected to the junction where the column of the sensor, analog input of the MCU and reference resistor(330 O) connected to the GND gather. Firstly, the MCU sets high input voltage(5 V) to the bottom row and low input voltage(0 V, GND) to the other rows of the sensor. After supplying input voltages to each row, the microprocessor measures the voltage in each column of the sensor, which is related to the resistance between the high voltage row and the measured column, i.e. 0 to 4 subelements in this case. Finishing the measurements, the MCU sets low voltages on the row with high voltage values(bottom row) and high voltages on the immediate above row(4th row), and measures the analog inputs of each column(5 to 9 subelements' values). To read the sensor values of all subelements once, the above tasks were repeated until the top row became high voltage rows. After reaching the top row, all the read analog values were saved as raw sensor values of each subelement at a certain moment and repeated from beginning in order to obtain sensor values for the next situation. The model proposed in this letter uses 25 raw sensor value array to calibrate the sensor. The current pressing coordinate and the pressure value are combined to generate a label, a high-resolution (20x20) pressure map. The CNC machine is coded to press all 400 (20x20) coordinates with 5 different pressing depths for 5 times each. Different pressures ranging from 30 kPa to 400 kPa are applied to each sensor coordinate.

To collect multiple ground truth pressures applied to the sensor surfaces, we manually hold the second small-sized load cell (Futek LCM300) to collect the second pressure. While the CNC milling machine is instructed to press a single point in the sensor subelement, we vertically press other sensor subelements by manually holding the small-sized load cell. Due to physical limitations that two load cells collide when pressing two consecutive sensor subelements, we could only collect multiple touch data that are pressing two subelements that are at least one subelements apart from each other. Note that while we tried to accurately press the correct position in z-direction, there could be human error where the pressed location is slightly shifted or the load cell is tilted when generating data.

III. Proposed Methods

A. Gaussian Pressure Map

Since the model is trained with single touch data and tested with multi touch data, single touch data and multi touch data should be labeled with the same data shape. Thus, a naive expression such as discrete coordinates would make it hard to express multi-touch data as the number of simultaneous touches in real life applications is unknown. In addition, to enable high-resolution calibration, labels should express high resolution pressures. Therefore, we use a Gaussian pressure map to label pressures applied to a certain area of a sensor. Supposing that a pressure $\mathbf{P}$ is applied onto (c_1, c_2) coordinates, the pressure map $\mathbf{M}$ for this pressure can be expressed with the equation below:

$$\mathbf{M} = \frac{P}{\max(X)} X, \forall X \sim \mathcal{N}_k(\mu, \sigma) \tag{1}$$

where $\mu = (c_1, c_2)$, $\sigma = \frac{\mathbf{r}}{\mathbf{2}}$ and r stands for the radius of the pressing tip. As a result, the pressure map labels express current pressure in a higher resolution while preserving its spatial information. When multiple simultaneous touches occur, the pressure map is the Gaussian mixture of individual pressure maps.

B. Local Network

The key design of our network is the assignment of individual subnetworks for each sensor subelement so that the input domain is consistent even when multiple pressures are applied to the sensor. As illustrated in Fig. 3, sensor input vectors are distributed to different subnetworks. Each subnetwork takes an assigned sensor value and a previous hidden vector as its input. These two inputs are used to generate a new hidden vector. This process is recurrently applied for each time step so that our network can utilize temporal behavior for corresponding sensor subelements.

Given a current sensor value $\mathbf{x} = (x_1, \ldots, x_{25})^T \in \mathcal{R}^{25}$ and a previous hidden state $\mathbf{h_t} = (h_{t_1}, \ldots, h_{t_{25}})^T \in \mathcal{R}^{25\times 128}$, subnetwork i will receive sensor value x_i and previous hidden vector h_{t_i}. We extract the current sensor state $h_{t_i}^{(1)}$ based on both current sensor input and previous hidden state as follows:

$$h_{t_i}^{(1)} = W_i(x_i, h_{t_i}), i \in [1, 25] \tag{2}$$

where W_i is the iterative function, which in our case, is a gated recurrent unit (GRU) [23]. To reduce latency caused by passing information through recurrent units, we used a single layered GRU for the iterative function. We assign distinct iterative functions for each subelement and sensor values will pass through different networks in parallel. To overcome hysteresis embedded in piezoresistive film, the model is designed to utilize temporal behavior of the sensor signals by recurrent structure.

C. Message Passing and Hidden Vector Update

Inspired by the powerful message passing style of RecGNN [24], a summation-based information aggregation method is selected as communication channels between different subnetworks. In particular, once every subnetwork generates a new hidden vector, hidden vectors of neighboring subelements are aggregated to update the hidden vector. Therefore, when pressure is applied in between two different subelements, the correlation between the current hidden vector and neighboring vectors is delivered through transmitted messages. Given a sensor subelement with its hidden vector $h_{t_i}^{(1)}$, $i \in [1, 25]$ and its neighboring subelements $\mathcal{N}(i)$, we calculate a message vector

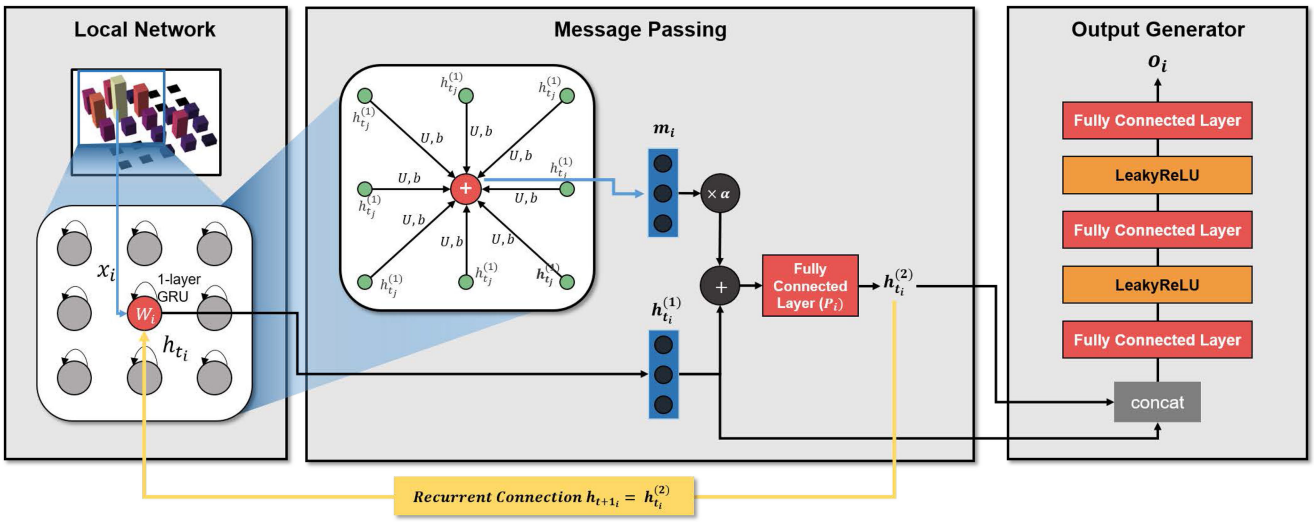

Fig. 3. The illustration of Local Message Passing Network (LoMP) structure consists of local network, message passing mechanishm and output generator.

m_i as below:

$$m_i = \sum_{\forall j \in \mathcal{N}(i)} \sigma(U h_{t_j}^{(1)} + b) \tag{3}$$

where U and b represent trainable parameters of a fully connected network and σ represents an activation function such as ReLU. The hidden values $h_{t_i}^{(2)}$ of each sensor subelement i is then updated using messages as below:

$$h_{t_i}^{(2)} = \sigma(P_i(h_{t_i}^{(1)} + \alpha m_i)), \forall i \in [1, 25] \tag{4}$$

where P_i denotes parameters of a fully connected network of subnetwork i, σ is the activation function such as a rectified linear unit (ReLU) [25]. A discount factor α is multiplied to the messages so that a neighboring information is not propagated to distant sensor subelements. Generated new hidden vectors are passed on to the output generators for predicting pressure map applied to the sensor.

D. Pressure Map Output Generator

Each individual network in LoMP generates a local pressure map. These maps are combined to generate an overall output pressure map. Our output model consists of 25 individuals 3-layered fully connected networks. Each network uses its corresponding hidden vector generated by the message passing scheme and generates a 4x4 pressure map. The output pressure map o_i is computed as below:

$$o_i = f(h_{t_i}^{(2)} || h_{t_i}^{(1)}) \forall i \in [1, 25] \tag{5}$$

where f denotes 3-layered fully connected network and $||$ represents concatenation between two hidden vectors. Generated 4x4 output pressure maps are concatenated while preserving their spatial arrangement within the sensor. At the same time, the new hidden vectors are passed to local network for the next time step as the recurrent connection.

$$h_{t+1_i} = h_{t_i}^{(2)} \tag{6}$$

E. Optimization Objectives and Training Strategy

The training process of the proposed method is to minimize mean squared errors between generated heatmaps and labels. The 4x4 pressure map generated by 25 individual sensor networks would be concatenated while preserving spatial configuration to generate a 20x20 pressure map. The loss values between the output vector $o = [o_1, \ldots, o_{25}]$ and label vector $l = [l_1, \ldots l_{25}]$ is expressed as below:

$$Loss = E_{\forall o,l}\left[\sum_{\forall i \in [1,25]} [(o_i - l_i)^2]\right] \tag{7}$$

We train the network using an Adam optimizer with a learning rate of 0.001.

IV. Experiment Analysis

A. Data Description

1) Single-Touch Dataset: The first dataset is a single touch dataset, which consists of 5×5 sensor inputs and high resolution 20x20 pressure maps as labels. By using a sliding time window of size 8 along the data sequence, 8 frames of consecutive sensor inputs were grouped as a single input so that models can utilize temporal sensor behavior. Therefore, the model can utilize temporal behaviors of sensor signals. The dataset was then divided into a train set, an evaluation set, and a test set with ratio 8:1:1. Our model was trained using only training set of the single touch dataset to demonstrate the expandability of our model.

2) Double-Touch Dataset: The second dataset was acquired by pressing two different sensor locations. To avoid collision between the two different load cells while collecting the data, two non-consecutive subelements are pressed. The double touch dataset was used only as a testing set; the models did not receive any multi touch information during training.

TABLE I
EXPERIMENT RESULTS OF SINGLE AND DOUBLE TOUCH DATASET AND COMPARISON OF THE OTHER CALIBRATION METHODS

Model	Single Touch RMSE (kPa)	Double Touch RMSE (kPa)
Linear Regression	8.68	16.91
Polynomial Regression	6.72	14.71
Decision Tree Regression	14.74	22.35
RBF-Net	5.02	23.25
CNN	2.41	26.21
VGG-16	2.18	1346.86
ConvLSTM	2.72	22.05
RNN	**2.16**	19.28
Ours (LoMP)	2.17	**7.49**

B. Experimental Details and Comparison Methods

To validate the effectiveness and the superiority of our proposed method, we compared our results with conventional regression models [8]–[10] and supervised learning models [15], [17]–[20] used to calibrate tactile sensors. In particular, we compare behaviors of LoMP with RNN [18], CNN [19], and ConvLSTM [15], three fundamental supervised learning architectures. All the hyperparameters are set as described on the previous researches except the modified final layer for matching the output dimension.

C. Results of Single Touch Pressure Calibration

In this section, we first use a single touch dataset to verify whether the proposed model could accurately calibrate a high-resolution single touch dataset when each subnetwork utilizes only local information. Table I shows the root mean squared errors of the single touch test dataset for different models. From this table, we can observe that there is a clear loss gap between conventional regression approaches and learning-based approaches. Underlying hysteresis and non-linearity in the sensor behavior make conventional regression approaches hard to map sensor values into correct pressure values.

With a strong ability to analyze temporal data behavior, learning-based approaches with deep neural networks could predict not only the amount of pressure applied to the sensor, but also pressure locations. While the sensor only has 25 discrete subelements, learning-based algorithms can accurately calibrate pressures applied to 400 different locations, generating high resolution pressure maps. Although the recurrent neural network showed the lowest loss values, the losses between the learning-based methods are relatively close as demonstrated in Fig. 4. However, when a high pressure is applied, networks with the convolutional neural network structure showed significant perturbation. Since each sensor subelements was showing different patterns, convolutional filters that share information across the entire input make the network more sensitive to sensor value changes. Furthermore, as each subnetwork of LoMP uses local information, the proposed model is still more sensitive to local sensor value changes than the RNN model. The conventional regression models and RBF-Net could not accurately predict pressures. With low model complexity, these models failed to accurately handle non-linearity and hysteresis entangled with multiple sensor subelements.

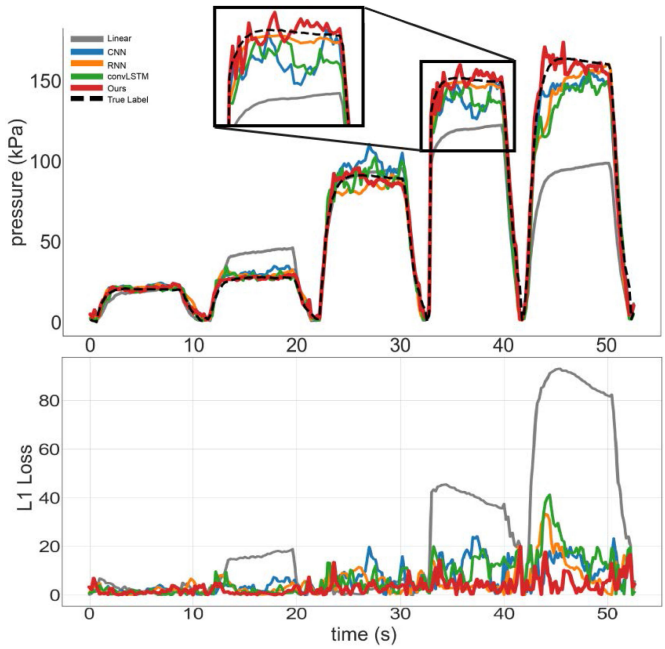

Fig. 4. Measured (black dashed lines) and estimated pressure for the single touch data and corresponding L1 Loss values.

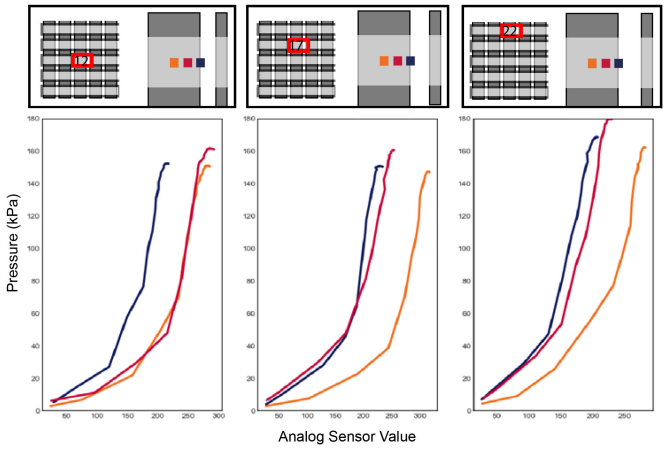

Fig. 5. Sensor behavior patterns when pressing different location (2.875 mm apart) within the same sensor subelement 12, 17 and 22.

As demonstrated in Fig. 5, the sensor values change even when the pressing location varies within the same sensor subelement. When the pressing location is far away from the center, the sensor values curve shifts to the left even when the same pressures are applied to the sensor. Therefore, simply estimating pressures with conventional regression methods is not enough for modeling different spatial sensor behaviors. In the proposed model, while each subnetwork only takes a single sensor cell value, our message passing mechanism allows it to gather information from neighboring subelements. Therefore, even if a pressure is applied in between two different sensor subelements, the networks can still correctly plot a shifted Gaussian pressure map for their

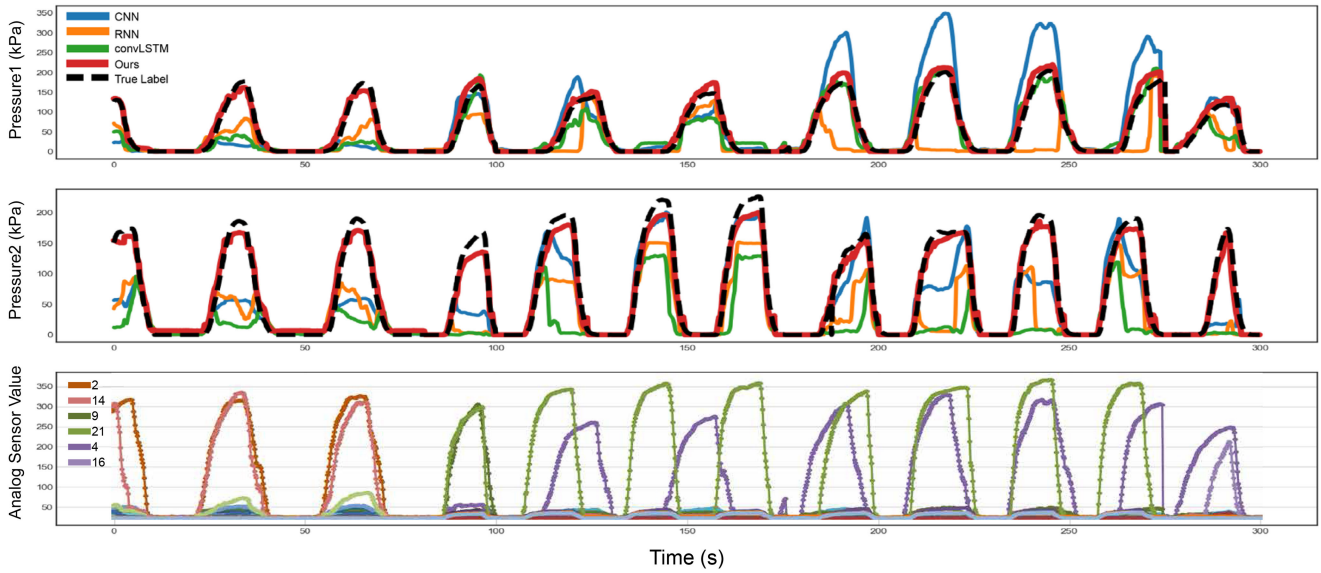

Fig. 6. Measured and estimated pressures when two simultaneous pressures are applied to two different sensor subelements. Top two graphs compare measured pressure and model predicted local maximums in pressed sub-region. Bottom row demonstrates corresponding sensor value changes. Label numbers represents subelement numbers.

TABLE II
AVERAGE INFERENCE TIME FOR CALIBRATING SINGLE-TOUCH TEST DATASET SAMPLES FOR DIFFERENT MODELS

	Inference Time			
Model	Linear Regression	RNN	CNN	LoMP
Time (ms)	0.96	1.82	1.33	1.45

designated area. Supplementary Video 1 shows more detailed calibration results for RNN, CNN and LoMP.

Since calibration normally works as a preliminary task before using sensors, it is important to have a model inference time that is short enough. To analyze inference time for our LoMP model, we measured average inference time while calibrating 1000 single touch test dataset samples. Different models, linear regression, CNN, RNN and LoMP are tested using the Nvidia GeForce GTX 1070, a performance-segment GPU. As listed in Table II our LoMP model shows reasonable inference time. While LoMP showed relatively higher inference time than those of linear regression and CNN models, it only takes 1.45 ms to calibrate a sensor input. Our LoMP model uses a 1-layered GRU to reduce time required to calibrate sensor inputs than the previous RNN model while showing similar performance for the single touch data.

D. Results of Multi Touch Pressure Calibration

The second dataset contains data of two simultaneous touches. As demonstrated in Table I, the loss values for RNN, CNN, convLSTM, VGG16 increase significantly compared to our proposed model. The calibrated results for two simultaneous pressure values are illustrated in Fig. 6. While LoMP could generally predict two different pressures correctly, other learning-based models were unable to predict pressure when multiple pressures were applied. Fig. 7 highlights the loss values for the different models when two pressures are applied to the

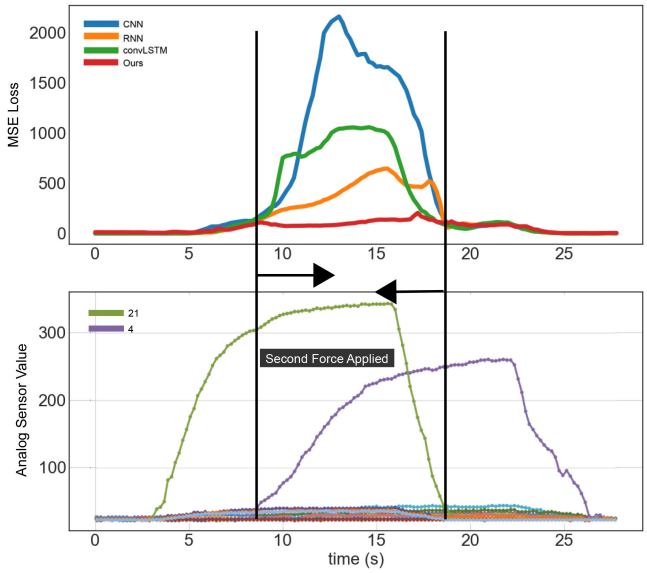

Fig. 7. Mean squared error loss values for different models when two pressures are applied simultaneously.

sensor. Except for our proposed model, the loss values drastically increase when a second pressure is applied to the sensor. This is due to the domain shift problem of the original learning-based approaches. Training only with single touch data, original supervised learning models are only generalized to map single pressure correctly. Therefore, when multiple pressures are applied to the sensors, the input sensor value is out of their domain, resulting in unexpected misestimations for the applied pressures. Interestingly, conventional regression models show relatively low double touch losses. Since regression function assign a simple individual weight to each sensor value, the loss value only doubles when a second pressure is applied to the sensor. In contrast, with high model complexity, VGG16 is more overfitted

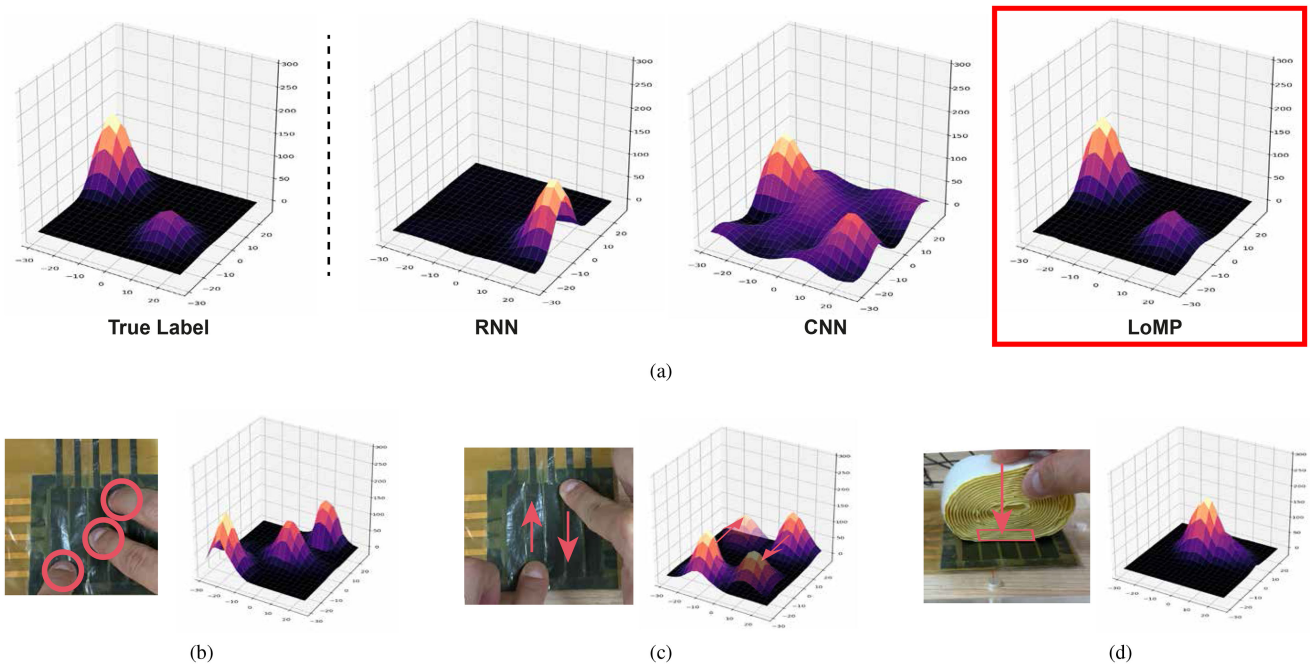

Fig. 8. 3D illustration of generated pressure maps. (a) Comparison between RNN, CNN and proposed LoMP model when two simultaneous forces are applied to the sensor. (b) LoMP-estimated pressure map when triple pressures are applied at the same time. (c) LoMP-estimated pressure map for two continuous dragging actions. (d) Generated pressure map when a rolled tape contacts with multiple neighboring subelements.

to the single data than other learning-based approaches. While the RNN model showed lower loss values than the CNN and convLSTM models, CNN and convLSTM showed better predictions in Fig. 6. This can be clearly explained with 3D illustration of the estimated pressure maps in Fig. 8(a). While the RNN network showed good performance in calibrating single touch cases, it was overfitted to the single touch domain and could not distinguish two different pressures applied to the sensor. CNN, with its shared filter mechanisms, could generate two local maximums at the correct positions. However, as the shared filters were applied to all input values, positions where no force was applied were also affected by its filters. As a result, CNN-based models exhibited higher loss values than the RNN network. As demonstrated in the figure, only LoMP could calibrate multiple points of pressure without falling into a domain shift issue.

Since LoMP consists of different local networks that only take local information, the domain taken by each individual network is hardly affected by a second pressure. When an additional force is applied, the sensor inputs only affect their corresponding local networks. The information in each network's point of view is bounded to the sensor signals applied to its own cell and neighboring cells. Therefore, these results demonstrate that the proposed model framework avoids domain shift issues.

E. Different Multi Press Cases

To further demonstrate the effectiveness of our model, we demonstrated three different multi-press cases. As illustrated in Fig. 8(b), (c), (d) and Supplementary video 1, our model can calibrate different multi-touch cases. The model could not only detect more than two simultaneous touches, but also two simultaneous sliding motions across different sensor cells. Furthermore, as demonstrated in Fig. 8(d), when we press multiple neighboring cell with a rolled velcro tape, the pressure map deforms as the contact area between the tape and the sensor increases. The generated left-tilted pressure map clearly reflects the pressing hand motion in which the thumb is slightly lifting the right side of the object while the index finger is pressing the center. Therefore, above pressure estimations reveal that the proposed learning framework can also be expanded to calibrate different complex pressing patterns.

V. Conclusion

To overcome drawbacks of soft tactile sensors and achieve a calibration method that enables high-resolution multi-touch calibration, this article proposed an efficient learning model called LoMP. By assigning different subnetworks to different sensor subelements, we avoid domain shift problems when the domain for training and testing is different. To achieve the high-resolution calibration, a message passing mechanism is applied so that each subnetwork can obtain neighboring cells' information through their messages. Meanwhile, Gaussian pressure maps are used to label inputs to not only guide different models to learn the spatial arrangement of pressure values but also express different pressure patterns in a single map. Testing loss values for both single touch dataset and double touch dataset were compared between our proposed model and previous calibration models. LoMP could calibrate pressures applied to one of 400 pressing locations within a 5×5 array-like tactile sensor. Furthermore, the results indicate that LoMP could achieve multi-touch calibration even trained with single touch data, an

achievement infeasible for previous learning-based models. Different multi-touch calibration scenarios such as triple touches, two simultaneous sliding touches, and area pressing were also illustrated. The results show that LoMP can be applied to various multi-press scenarios.

While we tested double touch scenarios for LoMP, we could not collect ground truth pressing data for two consecutive pressing points due to collision between two load cells. Such physical difficulties in gathering labeled data are critical for designing learning-based calibration methods, and we believe our proposed method can be a good solution for it. Furthermore, although LoMP addresses the domain shift issues due to different number of pressure points, the domain shift issues for different pressing speeds, materials and patterns are also likely to cause errors in data-driven learning-based approaches, and fixing these issues have not yet been explored. Therefore, in future work, we would like to improve our model to be robust to various differences between training conditions and testing conditions.

REFERENCES

[1] L. Wang and Y. Li, "A review for conductive polymer piezoresistive composites and a development of a compliant pressure transducer," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 2, pp. 495–502, Feb. 2013.

[2] M. Shimojo, A. Namiki, M. Ishikawa, R. Makino, and K. Mabuchi, "A tactile sensor sheet using pressure conductive rubber with electrical-wires stitched method," *IEEE Sensors J.*, vol. 4, no. 5, pp. 589–596, Oct. 2004.

[3] L. Wang, T. Ding, and P. Wang, "Thin flexible pressure sensor array based on carbon black/silicone rubber nanocomposite," *IEEE Sensors J.*, vol. 9, no. 9, pp. 1130–1135, Sep. 2009.

[4] Y. Kervran, O. De Sagazan, S. Crand, N. Coulon, T. Mohammed-Brahim, and O. Brel, "Microcrystalline silicon: Strain gauge and sensor arrays on flexible substrate for the measurement of high deformations," *Sensors Actuators A: Phys.*, vol. 236, pp. 273–280, 2015.

[5] P. Bahrami, N. Yamamoto, Y. Chen, and H. Manohara, "Capacitance-based damage detection sensing for aerospace structural composites," in *Sensors Smart Struct. Technol. Civil, Mech., Aerospace Syst.*, 2014, Art. no. 90612M.

[6] L. Wang, Y. Han, C. Wu, and Y. Huang, "A solution to reduce the time dependence of the output resistance of a viscoelastic and piezoresistive element," *Smart materials structures*, vol. 22, no. 7, 2013, Art. no. 075021.

[7] O. Atalay and W. R. Kennon, "Knitted strain sensors: Impact of design parameters on sensing properties," *Sensors*, vol. 14, no. 3, pp. 4712–4730, 2014.

[8] O. Al-Mai, M. Ahmadi, and J. Albert, "Design, development and calibration of a lightweight, compliant six-axis optical force/torque sensor," *IEEE Sensors J.*, vol. 18, no. 17, pp. 7005–7014, Sep. 2018.

[9] Y. Noh *et al.*, "A 2-piece six-axis force/torque sensor capable of measuring loads applied to tools of complex shapes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 7976–7981.

[10] W. Li, A. Alomainy, I. Vitanov, Y. Noh, P. Qi, and K. Althoefer, "F-touch sensor: Concurrent geometry perception and multi-axis force measurement," *IEEE Sensors J.*, vol. 21, no. 4, pp. 4300–4309, Feb. 2021

[11] D. Kim, M. Kim, J. Kwon, Y.-L. Park, and S. Jo, "Semi-supervised gait generation with two microfluidic soft sensors," *IEEE Robot. Automat. Lett.*, vol. 4, no. 3, pp. 2501–2507, Jul. 2019.

[12] 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 Trans. Mechatronics*, vol. 24, no. 1, pp. 56–66, 2018.

[13] M. Liu *et al.*, "Large-area all-textile pressure sensors for monitoring human motion and physiological signals," *Adv. Mater.*, vol. 29, no. 41, 2017, Art. no. 1703700.

[14] K. K. Kim *et al.*, "A deep-learned skin sensor decoding the epicentral human motions," *Nat. Commun.*, vol. 11, no. 1, pp. 1–8, 2020.

[15] B. S. Zapata-Impata, P. Gil, and F. Torres, "Learning spatio temporal tactile features with a convlstm for the direction of slip detection," *Sensors*, vol. 19, no. 3, p. 523, 2019.

[16] S. Sundaram, P. Kellnhofer, Y. Li, J.-Y. Zhu, A. Torralba, and W. Ma-tusik, "Learning the signatures of the human grasp using a scalable tactile glove," *Nature*, vol. 569, no. 7758, pp. 698–702, 2019.

[17] J. Ye, Z. Lin, S. Huang, Y. Zhang, and H. Wu, "A flexible touch-sensing sensor based on the theory of non-uniform gradient potential distribution," *IEEE Sensors J.*, vol. 20, no. 7, pp. 3396–3405, Apr. 2019.

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

[19] C. Trueeb, C. Sferrazza, and R. D'Andrea, "Towards vision-based robotic skins: A data-driven, multi-camera tactile sensor," in *Proc. 3rd IEEE Int. Conf. Soft Robot.*, 2020, pp. 333–338.

[20] 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.

[21] D. A. Valle-Lopera, A. F. Castaño-Franco, J. Gallego-Londoño, and A. M. Hernández-Valdivieso, "Test and fabrication of piezoresistive sensors for contact pressure measurement," Revista Facultad de Ingeniería Universidad de Antioquia, no. 82, pp. 47–52, 2017.

[22] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domainadaptation," 2015. *arXiv:1511.05547*.

[23] K. Cho *et al.*, "Learning phrase representations using Rnn encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[24] A. Sperduti and A. Starita, "Supervised neural networks for the classification of structures," *IEEE Trans. Neural Netw.*, vol. 8, no. 3, pp. 714–735, May 1997.

[25] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, 2010.