

# Adaptive Calibration of Soft Sensors Using Optimal Transportation Transfer Learning for Mass Production and Long-Term Usage

DongWook Kim, Junghan Kwon, Byungjun Jeon, and Yong-Lae Park\*

Soft sensors suffer from high manufacturing tolerances and signal drift from long-term usage, which degrades their practicality. Although deep learning has recently been proposed to address these issues, it is expensive in terms of data collection and processing. Therefore, an adaptive calibration method is proposed for soft sensors, suitable for mass production and long-term usage. In addition to maintaining the original benefits of deep learning characterization, this method enables fast and accurate calibration by capturing the change in the characteristics of the sensor through domain adaptation, using optimal transportation. An offline calibration method is first described, which is for alleviating the difficulty in calibrating every single unit from mass produced soft sensors. The main advantage is that identically manufactured soft sensors in a large volume with variations can be calibrated with reduced time and effort for collecting and processing data. Online calibration is then discussed, which compensates for the parameter changes when a soft sensor is continuously used for an extended period of time. For a single sensor, even though the sensor shows signal drift from the long-term usage, the calibrated network weights can be quickly adjusted online. Finally, the proposed adaptive calibration is experimentally evaluated using actual soft sensors.

a tremendous growth in the field of soft robotics for the past decade focusing on creating unique mechanisms of grasping, manipulation, and locomotion, by means of high compliance of soft materials.<sup>[5–7]</sup> Especially, sensors made of soft materials (i.e., soft sensors) have been developed for applications, including wearables or haptics.<sup>[8–10]</sup> For example, room-temperature liquid metal or carbon nanoparticles have been combined with silicone rubber to create soft sensors that show different electrical resistance values with varied stresses or strains. Another example includes electronic components, such as a metal oxide semiconductor (MOS) transistors, or light-emitting diodes (LEDs), which have been integrated with elastomer structures to be stretchable.<sup>[8,11,12]</sup> In particular, soft sensors made of eutectic gallium indium (eGaIn)<sup>[13]</sup> embedded in silicone matrices have recently shown dramatic reduction in fabrication time by applying 3D printing technologies combined with pneumatic dispensing or ink jetting techniques.<sup>[14,15]</sup>

## 1. Introduction


Soft robots made of highly deformable materials have shown their applications in a wide range, such as wearables, rehabilitation devices, and physically interactive robots.<sup>[1–4]</sup> There has been

Despite the advantages of the variety of soft sensors, soft sensors in general show large nonlinearities and hysteresis in their responses, which makes it difficult to use a simple calibration process by regression. To address this issue, different efforts have been made not only in a low level, such as modification of physical designs, but also in a high level, such as implementation of deep learning in characterization and control.<sup>[16–26]</sup> Deep learning can be used to provide a characteristic curve of a soft sensor that even contains the sensor's hysteresis behavior, as deep learning algorithms consider hysteresis as an inherent characteristic of the sensor and do not attempt to eliminate or reduce it. Instead, they use learning to model and fit the hysteresis curve itself.<sup>[25,26]</sup>

Although various types of soft tactile sensors can show better performances and higher accuracies in virtue of deep learning, there are still two major limitations in practical implementation. The first limitation is relatively high-performance variations of the sensors from manufacturing tolerances. Sensors even from the same batch can show different characteristics due to multiple reasons, such as variations in the properties of the polymer material, variations in the curing conditions, inhomogeneity of the polymer mixture, and manufacturing errors, as the performance of soft sensors is mostly dominated by the properties and the production

D. W. Kim, J. Kwon, B. Jeon, Prof. Y.-L. Park  
Department of Mechanical Engineering  
Institute of Advanced Machines and Design  
Seoul National University  
1 Gwanak-ro, Gwanak-Gu, Seoul 08826, Republic of Korea  
E-mail: ylpark@snu.ac.kr

D. W. Kim, J. Kwon, B. Jeon, Prof. Y.-L. Park  
Institute of Engineering Research  
Seoul National University  
1 Gwanak-ro, Gwanak-Gu, Seoul 08826, Republic of Korea

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/aisy.201900178>.

© 2020 The Authors. Published by WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.201900178

conditions of the host polymer materials.<sup>[27,28]</sup> These problems not only lead to different offsets in the initial sensor outputs but also cause distribution of operating ranges. Therefore, to ensure the accuracy, the user needs to collect the training data of each sensor and then train its individual neural network. This step is expensive and cumbersome. It is also highly inefficient to use an individual neural network with different weights for each sensor when many sensors are simultaneously used. Imagine a situation in which 20 neural nets are used in parallel. It will be significantly slow, and the kernel may even shut down. The second limitation is the assumption that the characterization of the sensor can be completed by one-time calibration. However, if we use the same sensor for an extended period of time, the sensor calibration using deep learning will become useless, as the sensor output will drift due to several reasons, such as polymer aging and changes in sensor positions in case of applications in wearables.<sup>[29–33]</sup> This means the sensor needs to be periodically recalibrated and repositioned, which is tedious and time consuming.

In this article, we propose to address these two limitations by solving a domain adaptation problem.<sup>[34,35]</sup> Both limitations are caused by domain shifting, in which a set  $X$  shifts given a mapping  $f: X \rightarrow Y$ , but a task set  $Y$  remains intact. In both cases,  $X$  is the set of possible sensor values, and  $Y$  is the set of admissible reference stress or strain levels, depending on the purpose of the soft sensor. For characterizing a large volume of soft sensors, if a single sensor is calibrated, there exists a neural network mapping  $f_\theta: X_s \rightarrow Y$  with a weight  $\theta$  (Figure 1a). Rather than making another neural network  $g_{i,\phi_i}: X_{T_i} \rightarrow Y$ , we introduce the mappings  $\gamma_i^*: X_{T_i} \rightarrow X_s, i = 1, 2, \dots, n$  that connect each target domain to the source domain. Then, the neural network  $g_{i,\phi_i}$  can be replaced with a composite  $g_{i,\theta}: X_{T_i} \rightarrow Y = f_\theta \circ \gamma_i^*$ . Finally, if we find suitable mappings for  $\gamma_i^*$ , we can represent multiple sensor mappings using only a single neural network. For using a single soft sensor in a long term, the source should be the initial calibration with a neural network mapping  $f_\theta: X_s \rightarrow Y$ , and the target domain will shift over time. We can introduce a mapping  $\gamma_t^*: X_t \rightarrow X_s$ , where  $t$  is the evolution time (Figure 1b). Finding  $\gamma_i^*$  and  $\gamma_t^*$  is called an optimal transportation problem.

In this work, we use microfluidic soft sensors to implement and experimentally evaluate the performance of our proposed

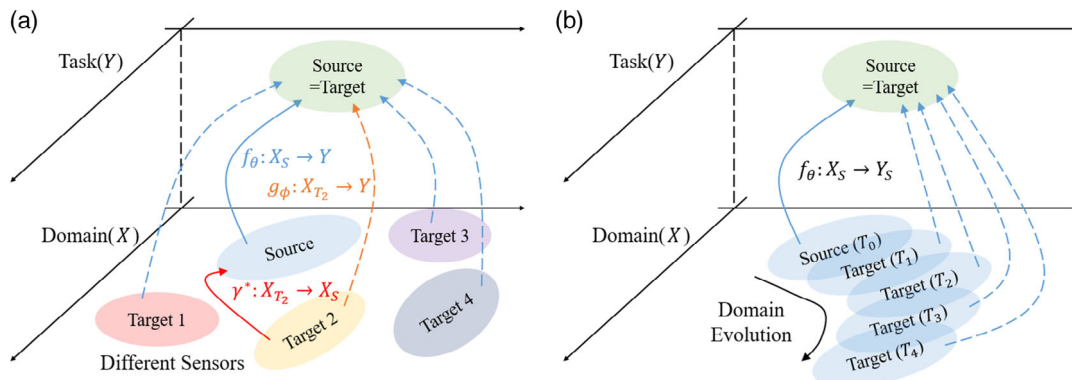
method. The sensor is composed of a silicone matrix and an embedded microchannel filled with eGaIn. Any mechanical pressure or strain applied to the sensor changes the geometry of the embedded microchannel and consequently its electrical resistance. Our contribution is the use of optimal transportation transfer learning to solve two common problems in calibration of soft sensors 1) when a large volume of sensors is produced with manufacturing tolerances and 2) when a single sensor is used for a prolonged period of time. We validate solutions to these two problems through two different experiments: 1) calibration of seven identically manufactured sensors with variations in their characteristics and 2) calibration of a single sensor attached to a human body for gait cycle measurement. The result shows that the proposed method yielded a similar level of errors compared with previous studies even with less training time.<sup>[25,36]</sup>

The remainder of this article is structured as follows. Section 2 explains the concept of a time-delay artificial neural network (TDANN) to compensate for nonlinearity and hysteresis of soft sensors and then discuss the optimal transportation problem to describe transfer learning and the algorithm used in this work. Experimental validation of the proposed method using microfluidic soft sensors is shown in Section 3 and 4 for the two cases: large-volume characterization and a single sensor calibration for long-term usage. Fabrication of the soft sensors and the data collection process for transfer learning are also described in these sections. Section 5 finally presents the results, followed by the discussion in Section 6 and the conclusion in Section 7.

## 2. Theoretical Preliminaries and Hyperparameter Settings

### 2.1. TDANN for Calibration

Silicone rubber was used as the main material for the soft sensors in this work, and it shows hysteresis with time dependency due to the viscoelastic behavior. Therefore, using a recurrent neural network (RNN) to model hysteresis may be suitable, as it refers to past signals for learning. However, the goal of this article is to perform fast sensor calibrations and online estimations by transfer learning,



**Figure 1.** Mass calibration and long-term usage problem can be redesigned as solving an optimal transportation problem (i.e., finding mappings  $\gamma_i^*$ ). a) Each sensor has a different sensor signal pattern, originated from a different domain, and the mappings  $\gamma^*$  connect each target domain to the source domain. b) Due to long-term usage, polymer material of soft sensor experiences aging, which leads to domain evolution (shifting), and the domains are connected via mappings  $\gamma^*$ .

and we decided to use a TDANN instead of an RNN. As we are interested in estimating the current state value rather than extracting the latent feature from the sensor signal or predicting the future sequence, there is no reason for a TDANN to fall behind an RNN. Implementation details for the structure of the TDANN and back-propagation steps are specified in Supporting Information.

## 2.2. Transfer Learning and Domain Adaptation

When designing a function that maps the relationship between the input and output through machine learning, it is most likely to be a process of decorating the neural networks or kernels by supervised learning. However, a necessary condition for such supervised learning is that the training dataset and the testing dataset must be extracted from the same probability distribution. If not, the network should be redesigned, or the weights should be tuned by additional work. For example, imagine a situation in which a photographer trains a convolutional neural network (CNN) that enables classifying pictures of dogs and cats. He takes pictures of dogs and cats in the park and then trains the network. In the next day, he goes to the park and takes pictures of dogs and cats again and then put them into the learned network. If the network is appropriately trained, it can readily classify the two classes with high accuracy. However, if he takes pictures of dogs and cats in a dark café and put them into the network, they cannot be adequately categorized due to the brightness difference and surrounding conditions. This is one of the most common drawbacks of machine learning because the network may need to be redesigned to accommodate the new distribution if the distribution of the sampling dataset changes. However, domain adaptation transfer learning used in our work makes this process easy.

Let us define the source, the target, the domain, and the task through the aforementioned example. The source is the pictures of dogs and cats in the park, which describes the population of the sampled training datasets. The target is a population of the pictures of dogs and cats in a café, which is a sampling population of a testing dataset. The domain and the task are the concepts existing both in the source and in the target. We can think of a domain as an input of the CNN (the pixel value of the pictures) and the task as an output label.

Transfer learning can be divided into an inductive, an unsupervised, and a transductive process according to the conditions of the source, the target, the domain, and the task. We focus on transductive transfer learning in this article. Transductive transfer learning is a transition learning method that solves a problem when the source and the target domain distributions are different. For example, the distribution of the output values of a sensor may vary, but the distribution of the reference force or strain remains intact. The key feature of transductive transfer learning is that task labeling of the target is not necessary but only the target domain values are required. For the picture classification problem, pictures taken in the park should be labeled, but pictures taken in the café need not be labeled. Instead, the relationship between the distributions of the source and the target domains is studied. The mathematical definition of transductive transfer learning is shown as follows.

Let the source domain  $D_S$  and its corresponding task  $T_S$ , the target domain  $D_T$ , and then, the corresponding task  $T_T$  exists.

Transductive transfer learning finds the mapping  $f_T: D_T \rightarrow T_T$  under the condition  $D_S \neq D_T$ , and  $T_S = T_T$  by only using the information of  $D_S$ ,  $D_T$ , and  $T_S$ .

One way to solve the transductive transfer learning problem is to find a mapping which connects  $D_T$  and  $D_S$ , i.e., finding  $g: D_S \rightarrow D_T$ , and finding the mapping is often called domain adaptation. In this article, we utilize this domain adaptation using the optimal transportation theory.

## 2.3. Domain Adaptation by the Optimal Transportation Theory

In this section, we review the concept of the optimal transportation theory to solve the domain adaptation problem for implementing it to our soft sensor application. The optimal transportation theory first starts with the Monge problem, which attempts to find a mapping that minimizes the target cost function while maintaining the image measure condition.<sup>[37]</sup> Let  $\Omega_S, \Omega_T \in R^n$  be a compact, measurable space of dimension  $n$ , and there exists a probability distribution  $\mu_S \in P(\Omega_S)$  and  $\mu_T \in P(\Omega_T)$ , and the cost function  $c: \Omega_S \times \Omega_T \rightarrow R^+$ . Then, the Monge problem seeks to obtain a mapping  $T: \Omega_S \rightarrow \Omega_T$  which satisfies

$$\inf_{T \# \mu_S = \mu_T} \int_{\Omega_S} c(x, T(x)) \mu_S(x) dx \quad (1)$$

where  $T \# \mu_S(X) = \mu_T(T^{-1}(X))$ ,  $\forall X \subset \Omega_T$ .

However, as this optimization problem is nonconvex, the solution may not exist. Thus, Kantorovich established a relaxed problem that finds a mapping in the joint distribution space  $\gamma \in P(\Omega_S \times \Omega_T)$  satisfying

$$\inf_{\gamma \in P(\Omega_S \times \Omega_T)} \int_{\Omega_S \times \Omega_T} c(x, y) \gamma(x, y) dx dy \quad (2)$$

such that  $\gamma \in P(\gamma \geq 0, \int_{\Omega_T} \gamma(x, y) dy = \mu_S, \int_{\Omega_S} \gamma(x, y) dx = \mu_T)$ .<sup>[38]</sup> This problem is a linear program formulation that always has a unique solution. The Kantorovich problem can be described as a Wasserstein metric formula, as shown as

$$\begin{aligned} W_p(\mu_S, \mu_T) &= \inf_{\gamma \in P(\Omega_S \times \Omega_T)} \int_{\Omega_S \times \Omega_T} c(x, y) \gamma(x, y) dx dy \\ &= E_{(x, y) \sim \gamma} [c(x, y)] \end{aligned} \quad (3)$$

where  $E[\cdot]$  is the expectation operator. The Wasserstein metric is often used to measure the distance between two probabilistic distributions. Therefore, the optimal cost for the optimal transportation mapping is equivalent to the 1-Wasserstein metric.<sup>[39]</sup>

For the strictly convex cost function  $c(x, y) = \frac{1}{2} \|x - y\|^2$ , the dual Kantorovich problem can be written as maximizing the following equation

$$\sup \left[ \int_{\Omega_S} \varphi(x) \mu_S(x) dx + \int_{\Omega_T} \phi(y) \mu_T(y) dy \right] \quad (4)$$

where  $\varphi: \Omega_S \rightarrow R, \phi: \Omega_T \rightarrow R$  is a bounded, continuous function satisfying  $\varphi(x) + \phi(y) \geq xy$ . Chartrand et al. proved that if the pair  $(\varphi, \phi)$  are convex conjugate functions,  $T = \nabla \varphi$  solves the Monge problem.<sup>[40]</sup> Other techniques, such as gradient descent, flow minimization, linear programming, or entropy regularization, exist as well.

## 2.4. Optimal Transportation to Sensor Calibration

We use a TDANN to calibrate the sensor output values and their corresponding reference values. The sensor changes its electrical resistance due to the changes in the geometry of the embedded microchannel, and the resistance value can be measured by a voltage divider with a static resistor connected in series. The reference values for the sensors are the force and the elongation length for the stress and the strain, respectively. The input and the output of the source are defined as a vector  $x_i^s \in X_S$  and a scalar  $y_i^s \in Y_S$ , respectively, where  $s$  is the source and  $i$  is the data number from an empirical perspective. Similarly, the input and the output of the target are  $x_j^t \in X_T$ ,  $y_j^t \in Y_T$ . Then, our goal is to find a function that maps from the target domain to the target task  $f: X_T \rightarrow Y_T$  and the optimal transportation map  $\gamma^*$  that satisfies the following equation

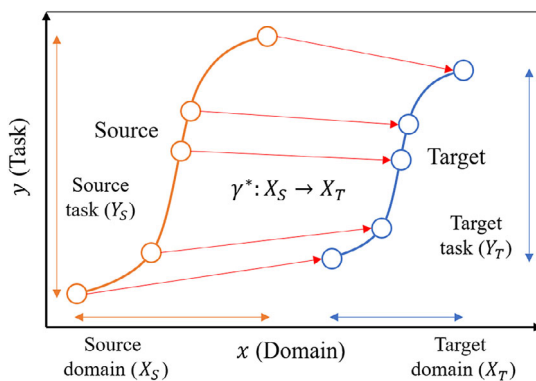
$$\min_{\gamma, f} \int_{X_S \times Y_S \times X_T \times Y_T} c(x_S, y_S, x_T, f(x_T)) d\gamma(x_S, y_S, x_T) \quad (5)$$

Courty et al. solved this minimization problem using a block coordinate descent (BCD) algorithm.<sup>[39]</sup> We define another mapping  $\gamma^*: X_T \rightarrow X_S$ . Then, by iteratively performing a stochastic gradient descent of  $\gamma^*$  and  $f$ , we can finally solve the transfer learning problem (Figure 2). As the source domain and the task data are transported to the target domain by  $\gamma$ , and the transformed source data acts as a training dataset of the target domain, the number of the target domain data may not be as large as the source domain data.

## 3. Offline Adaptive Calibration for Mass Production

### 3.1. Materials and Manufacturing Methods

As mentioned earlier, we use a soft sensor made of silicone elastomer with an embedded microchannel filled with liquid metal (eGaIn). Park et al. have estimated the resistance change ( $\Delta R$ ) when a strain ( $\epsilon$ ) or stress ( $p$ ) is applied



**Figure 2.** Optimal transportation problem attempts to find a mapping between source domain and target domain.

$$\Delta R = \frac{\rho L}{wh} \left\{ \frac{(1 + 2\nu)\epsilon - \nu^2 \epsilon^2}{(1 - \nu\epsilon)^2} \right\},$$

$$\Delta R = \frac{\rho L}{wh} \left\{ \frac{1}{1 - 2(1 - \nu^2)wp/Eh} - 1 \right\} \quad (6)$$

where  $\rho, \nu, L, w$ , and  $h$  are the resistivity of eGaIn, the Poisson's ratio of the elastomer, the length, the width, and the height of the microchannel, respectively.<sup>[8]</sup> For fabrication, a laser cutter has been used to create a microchannel on a silicone rubber block.<sup>[41]</sup> Similarly, silicone rubber has been cured in a 3D-printed mold with microchannel patterns.<sup>[8]</sup> However, we used a pneumatic dispenser integrated with a motorized  $x$ - $y$ - $z$  stage to directly print liquid-metal microchannel patterns on a silicone substrate and then cover the top with another layer of silicone to fully encapsulate the microchannel patterns.<sup>[42]</sup>

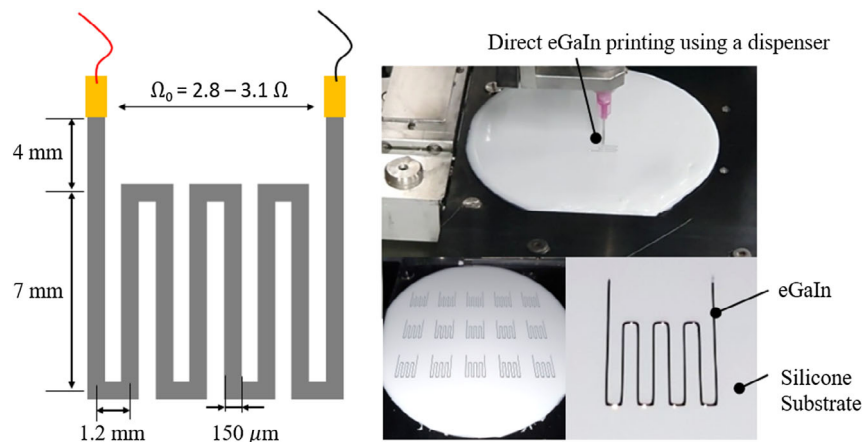
Details on the fabrication process are as follows. A room-temperature-vulcanizing (RTV) silicone elastomer (Ecoflex-0030, Smooth-On) was mixed using a centrifugal mixer (ARE-310, Thinky) with a weight ratio of 10:10:1 for parts A and B with an addition of a white pigment (Silc Pig, Smooth-On). The mixture was poured on a silicon wafer for spin coating at 200 rpm for 30 s. Once the elastomer was fully spread on the wafer, it was cured in an oven of 60 °C for 20 min. The cured silicone substrate on the wafer was then transferred to the printing stage, and serpentine patterns were directly printed using eGaIn (gallium 75.5% and indium 24.5% by weight). For printing, a motorized  $x$ - $y$ - $z$  stage (Shotmaster 300ΩX, Musashi) and a pneumatic dispensing system (Supersigma CMIII V2, Musashi) were used with a laser distance sensor (LK-G32, Keyence) (Figure 3). The printed trace was then covered with another layer of uncured elastomer, spin coated at 200 rpm for 30 s and cured in the oven. Finally, copper wires were plugged into the microchannel to make interconnections with the eGaIn channel at both ends for each sensor, and silicone epoxy (Sil-Poxy, Smooth-On) was applied to fix them in place.

### 3.2. Adaptive Calibration Procedure

We printed seven identical liquid-metal patterns (i.e., sensors) with the same geometry and printing conditions on a cured silicone substrate (Figure 3). However, the initial resistance value of the seven channels were slightly different due to the manual wiring process and the nonuniform curing conditions. Without the help of transfer learning, we have to calibrate all seven sensors individually, which is a tedious and time-consuming task. To calibrate a sensor using our method, we first need to install a commercial load cell with one sensor and receive reference data that need to be mapped with the sensor data with synchronization. We then apply force to the sensor by compressing with different rates and magnitudes using a motorized stage and gather the output data. After collecting all the data, we preprocess them to match the 15-time sequence data of the sensor output voltage and a single reference force value (i.e.,  $[X_{t-14:t}, Y_t], \forall t \geq 15$ ). Finally, a TDANN is trained by the minibatch gradient descent.

Instead of gathering the input-output tuple of all seven sensors individually and training them all, we use domain adaptation transfer learning here. After finishing the calibration of the first sensor that is used as a source distribution, the remaining six sensors are compressed with the motorized stage, and only the





**Figure 3.** Direct liquid printing was implemented to fabricate soft sensors. Each sensor has a different initial resistance value due to manufacturing tolerance.

output values, which are used as the target domain data, are collected. In this stage, we do not need the full signal tuples. Even if we use only a portion of the signal values, we can still configure the optimal transportation map without considerable loss of accuracy. After all, as the source domain and the task data are transferred to the target region, it is the source data that learns the target input–output relation. In this problem, the source domain ( $X_S$  in Equation (5)) is the sensor output which is completely calibrated, and the source task ( $Y_S$ ) is the corresponding force output. The target domain ( $X_T$ ) is the sensor output from the remaining six sensors, and the target task ( $Y_T$ ) is the force output of the other six sensors. The influence of the size ratio between the source and the target is inspected in the result. The target domain data contribute only to formulation of the transportation mapping. This property means that the effort to calibrate the remaining sensors is significantly reduced compared with the process of repeatedly training the individual sensors with their input–output signal tuples. Finally, the optimal transportation mapping and the input–output relation mapping of the target sensor are found.

We evaluated the result by calculating the root-mean-squared error (RMSE) between the estimated force by the TDANN and the reference from the load cell by applying forces to the seven sensors with the same test pressure inputs. Then, we evaluated the performance of the transfer learning by comparing the RMSE from individual training and the transfer learning together with the time and the computation costs. Finally, we investigated the influence of the source–target domain data ratio. If the performance does not decrease even though the size of the target domain data is far smaller than the source domain data, we can find the optimal ratio which consists of the time and the computation costs and the actual performance.

## 4. Online Adaptive Calibration for Permanent Usage

### 4.1. Materials and Experimental Settings

In this section, we discuss the calibration issue in long-term usage of a single soft sensor. The goal is to remove the contamination of the calibrated TDANN induced by the sensor drift

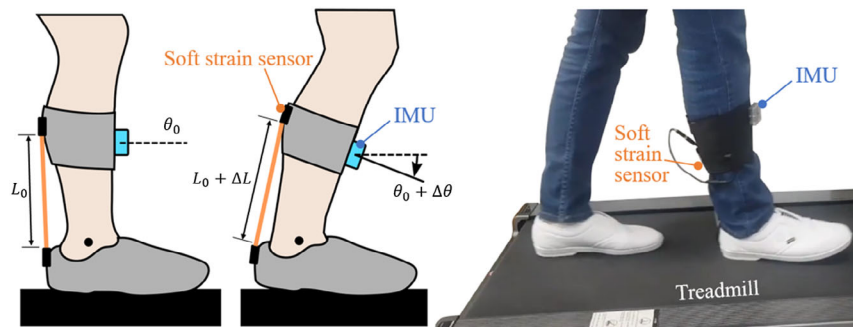
online. Rather than doing a simple repeatability test with controlled conditions, we consider a more undefined and unidentified system, which can also be applied to human wearables for motion sensing. If a soft strain sensor is attached between the heel and the calf of the leg, the roll angle of the ankle can be estimated, referring to the soft sensor signal patterns. To obtain the reference data, an inertial measurement unit (IMU) was attached to the front of the knee and acquires roll angle. Our objective was to estimate the ankle angle without the aid of the IMU, only with the soft strain sensor (**Figure 4**). The experiments involving human subject have been performed with the full and informed consent of the volunteer.

A gait measurement module was developed for collecting the strain data from the soft sensor in real time. The resistance change of the sensor was measured by a voltage divider circuit and an analog-to-digital converter (ADC) included in a microcontroller (Arduino MKR-1000) powered by a battery and remotely operated. The sampling frequency was 50 Hz. An IMU (EBMotion V4, E2BOX) was additionally attached to the module to measure the reference pitch angles with a sampling rate of 50 Hz. A digital low-pass filter with a cutoff frequency of 96 Hz was applied for the accelerometer and the gyro sensors in the IMU module. The pitch angles were calculated by the IMU's own algorithm based on the sensor output from the accelerometer and the gyro sensors. All the measured sensing data were wirelessly transferred to a data acquisition module (myRIO-1900, National Instrument) and recorded with synchronization.

### 4.2. Online Adaptation and Learning

In this problem, the source domain ( $X_S$  in Equation (5)) is the sensor output at the initial state, and the source task ( $Y_S$ ) is the corresponding IMU angle. The target domain ( $X_T$ ) is the sensor output when the time has passed, and the target task ( $Y_T$ ) is the corresponding IMU angle.

The process of the experiment is as follows. First, a human subject wears the soft sensor and the IMU, then starts to walk on a treadmill, and signals from both the soft sensor and the IMU are collected. The signal from the soft sensor is the source domain and that from the IMU is the task value. Then, if a



**Figure 4.** Semigait phase detection is possible by observing the ankle heading angle. The angle value can be captured by the soft strain sensor attached between the calf and the heel.

sufficient amount of data is collected, the TDANN starts to learn from the gathered data, whereas only upcoming signals from the soft sensor are mustered in the queue. The IMU signals are not queued here. When learning of the TDANN is complete, the queued soft sensor signals act as the target domain signal to update the TDANN weights using the domain adaptation, and then all the data in queue are cleared. However, the domain adaptation process is in progress, the incoming sensor signals are now accumulated in the queue again until the updating process ends and repeats to update the TDANN by the domain adaptation when the previous updating process is finished. Meanwhile, when the TDANN first succeed to learn the data, it can produce the estimation of the knee angle without the aid of the IMU. In summary, updating the weights by transfer learning and the state estimation can be performed simultaneously online, reflecting the signal drift of the soft sensor by the domain adaptation.

We can improve this process by also updating the source data using the queued soft sensor signals and the corresponding TDANN estimation values. Optimal transportation works powerfully when the Wasserstein distance between the source and the target domains is small. To reduce the gap between the source and target domains, the source is updated by the best estimation of the previous target domain data. After adapting the transportation by the queued soft sensor signals, we estimate the ankle angle using the best-updated TDANN ever and use their input-output tuple as new source data. This process is graphically shown in Figure 5.

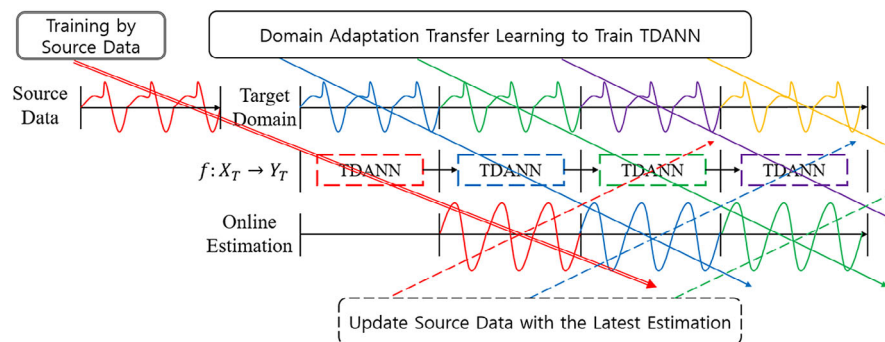
We evaluated the performance of this method by comparing the RMSE values in both cases in which transfer learning was and was not performed, and with the improved method. Moreover, we show the reference and the estimated values of each method and show that the performance gradually improved while the transfer learning was adapted.

## 5. Results

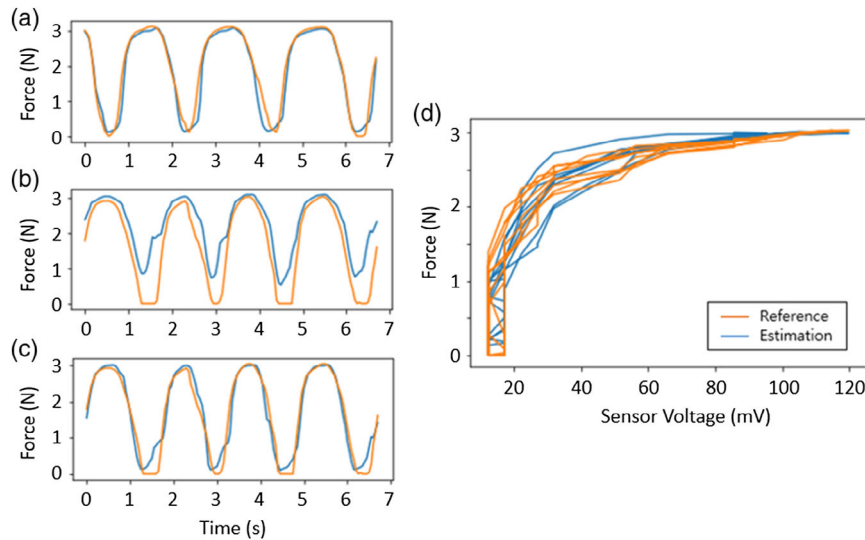
### 5.1. Offline Adaptive Calibration for Mass Production

The size of the training set for the source sensor was 25 000, and that of the testing dataset was 2500. Optimal transportation was implemented by solving Equation (5), for which the size of the target domain was 1250–25 000.

Figure 6 shows the adaptive calibration result of two soft sensors, which are the source and the target sensors. The source sensor was first trained with TDANN and tested with the testing dataset (Figure 6a) estimating the applied force level, for which its RMSE was 0.27 N. Then, optimal transportation was implemented to find the mapping from the target domain to the source domain, and the size of the target domain dataset was 25% of the source domain dataset. Figure 6b shows the result when optimal transportation was not implemented, and Figure 6c shows the result after optimal transportation was implemented. The RMSE improved from 0.70 to 0.29 N with implementation of the optimal transportation, and the result is comparable with



**Figure 5.** Online mapping improvement by capturing previous domain data. TDANN is implemented using python Tensorflow environment, and domain adaptation is implemented by python optimal transportation function library.



**Figure 6.** a) Calibration result of the source sensor, b) calibration result of the target sensor before domain adaptation, c) calibration result of the target sensor after domain adaptation, and d) input–output hysteresis curve of the target sensor. The target sensor does not directly estimate the force value by only using the source sensor’s neural network, but it is possible to estimate the force value by the aid of the domain adaptation.

its own calibration result (RMSE 0.27 N), training from the sensor’s own input and output. The RMSE results of the other sensors are shown in **Table 1**. The results from source learning and domain adaptation were indicated with the labels “Before domain adaptation” and “After domain adaptation,” respectively, in the table, and “Own calibration” means the own input–output learning result, which is for comparison with domain adaptation. It can be noticed that the calibration was improved after domain adaptation was implemented.

The influence of the size of the target domain dataset is shown in **Table 2**. The result shows that the correlation between the size of the target domain dataset and the RMSE value is weak from the observation that the RMSE value does not increase when the size of the dataset decreases. However, the target domain dataset should cover at least all of the admissible regions of the domain set to perform effective optimal transportation. Finally, the

**Table 1.** RMSE results for the mass calibration of soft sensors.

Calibration RMSE [N]	Sensor 1 (Source)	Sensor 2 (Target)	Sensor 3 (Target)	Sensor 4 (Target)	Sensor 5 (Target)	Sensor 6 (Target)
Before domain adaptation	–	0.57	0.70	0.76	1.35	4.32
After domain adaptation	–	0.35	0.29	0.39	0.40	0.41
Own calibration	0.27	0.31	0.27	0.35	0.39	0.38

**Table 2.** The influence of the size of the target domain dataset to the RMSE value. The original size of the dataset is 25 000.

Dataset size [%] (actual size)	100 (25 000)	75 (18 750)	50 (12 500)	25 (6250)	10 (2500)	5 (1250)
RMSE [N]	0.29	0.30	0.28	0.29	0.29	0.27

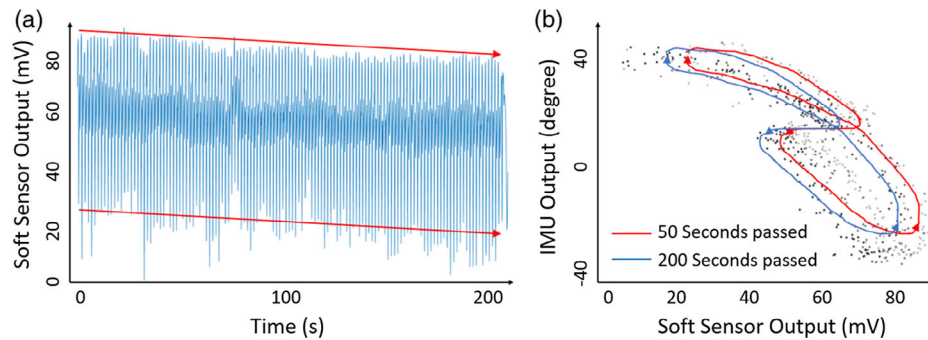
training time and the operating frequency before and after the domain adaptation are shown in **Table 3**. The result shows that the training of a single neural network required 86 s, but the optimal transport mapping required only 2.6 s to train. This is a dramatic reduction in calibration time. However, as the target sensors need to go through two mappings (optimal transportation mapping and the original neural network), the operating frequency was lowered from 40 to 25 Hz. Comparing to previous works that had conducted deep learning techniques or simple transfer learning, we maintained a similar level of error in force estimation. For example, the normalized-root-mean-squared error (NRMSE) of our result is between 8–11%, which is slightly higher than the previous work (8.38%<sup>[25]</sup>, 6.5–10%<sup>[36]</sup>).

## 5.2. Online Adaptive Calibration for Permanent Usage

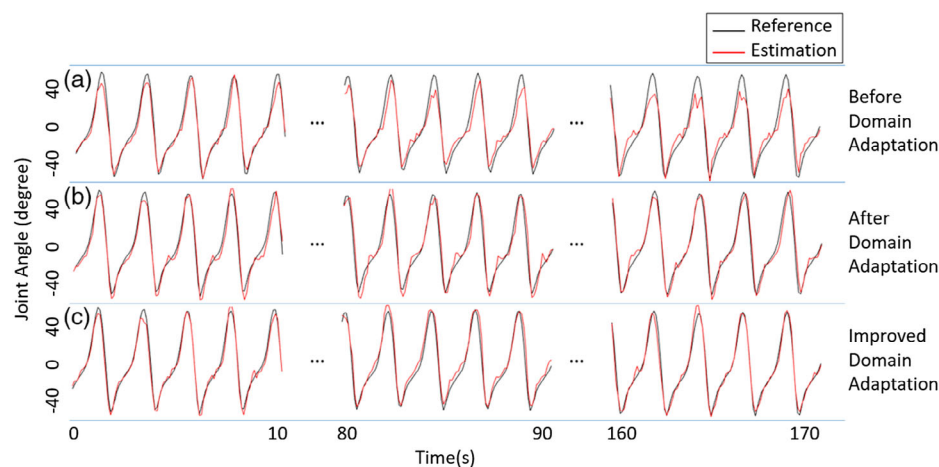
The raw data of the signals observed by the soft strain sensor were plotted to screen the distribution change (**Figure 7**). From this figure, we can conclude that a sensor drift exists when the soft sensor is used for a prolonged period of time from the observation of the input–output relation, as the IMU signal distribution remains intact whereas the soft sensor signal distribution changes over time. Comparison of the three cases, original, with domain adaptation, and with improved domain adaptation, is shown by evaluating the reference and the estimation sequence at each time interval and the evolution of the RMSE value as time passes

**Table 3.** The time cost and the operating speed before and after domain adaptation.

Test criteria	Training time [s]	Testing frequency [Hz]
Before domain adaptation	86	40
After domain adaptation	2.6	25

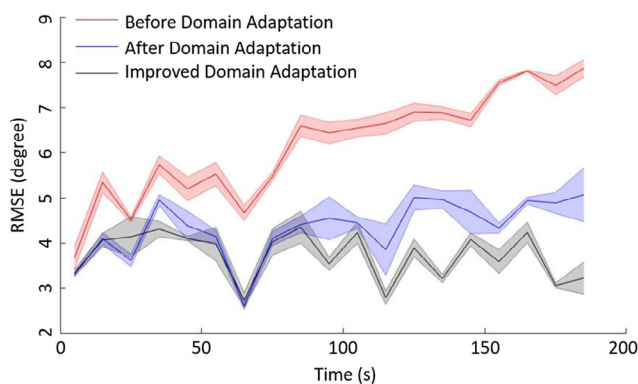


**Figure 7.** Sensor signal drift is visualized by fitting the raw data. a) Soft sensor experiences aging and its output value range is shifted. b) The IMU signals remained intact, but the sensor output changed.



**Figure 8.** Sensor calibration shows poor result when a) before domain adaptation, but b) after domain adaptation and using c) the improved domain adaptation, sensor signal was significantly improved.

(Figure 8 and 9). We can observe that the RMSE value increased when the sensor was used for a long time, but it stayed almost the same with transfer learning. These results strongly suggest that use of transfer learning (notably the improved algorithm) can prevent performance loss caused by sensor drift.



**Figure 9.** The evolution of estimation error remained the same level after the domain adaptation was implemented.

## 6. Discussion

The experimental results show that optimal transportation mapping is possible by analyzing the sampled dataset from the source domain and the target domain distributions. In mass calibration, the size of the target domain dataset was less than 25% of that of the source domain dataset. Adaptive calibration was still possible with a small RMSE value. However, the user must be cautious and ensure that the source task and the domain task distributions are identical; if not, it should work at least within the same scope. In the case of long-term usage, the RMSE value stayed at the level of the initial calibration even with the time lapse. Although the reference corresponding to the actual value of the sensor (i.e., IMU signals) is not known in the online testing situation, it is a significant advantage to use only the domain signals to compensate the calibration by the aid of optimal transportation as the task space is intact. Finally, other types of neural networks, such as long-term-short-term memory (LSTM) networks can be also used in our work.

There are variations in the techniques of transfer learning, from simple fine tuning to meta learning in artificial intelligence. One of possible alternative ways to calibrate soft sensors is



multidomain learning, where a single neural network handles data from a multidomain space and then provides results. It will be possible to apply this method to mass calibration, but online adaptation will not be possible as the multidomain learning requires a full training dataset from both domain and task data.

## 7. Conclusions

In this article, we solved the optimal transportation problem to address limitations raised in practical applications of soft sensors, for which it was successful to handle both mass calibration and long-term usage of soft sensors. First, avoiding generating multiple neural networks was possible by introducing a mapping  $\gamma_i^*$  that connects the source and the target domains. Second, the problem of signal drift of a soft sensor, when the sensor is used for a long period of time, was solved by updating the transportation mapping  $\gamma_i^*$  in real time. The experimental results showed that introducing optimal transportation reduced the error between the estimation and the reference values. Our method is memory efficient in terms of reducing the number of neural networks. Moreover, fast calibration makes online implementation possible. Finally, task reference data were not required. Future work will include the integration of the proposed methods with soft robot applications in a system level, such as wearable sensing suits for motion sensing, assistance, and rehabilitation. We believe this research will benefit research related to soft robotics where hysteresis and signal drift are major limiting factors for real-world applications. The implementation source code of the mass calibration problem is available at [https://github.com/mochacoco/AIS\\_Transfer\\_Learning](https://github.com/mochacoco/AIS_Transfer_Learning).

## Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

## Acknowledgements

This work was supported in part by the National Research Foundation (grant no.: NRF-2016R1A5A1938472) and in part by the Technology Innovation Program (grant no.: 2017-10069072) funded by the Korean government agencies, MSIT and MOTIE, respectively.

## Conflict of Interest

The authors declare no conflict of interest.

## Keywords

deep learning, optimal transportations, soft robotics, soft sensor calibrations, transfer learning

Received: December 16, 2019

Revised: January 26, 2020

Published online:

- [1] H. In, B. B. Kang, M. Sin, K.-J. Cho, *IEEE Robot. Autom. Mag.* **2015**, 22, 97.
- [2] M. Cianchetti, M. Manti, V. Cauciolo, C. Laschi, *IEEE Robot. Autom. Mag.* **2016**, 23, 93.
- [3] P. Polygerinos, Z. Wang, K. C. Galloway, R. J. Wood, *Rob. Auton. Syst.* **2015**, 73, 135.
- [4] R. Filippini, S. Sen, A. Bicchi, *IEEE Robot. Autom. Mag.* **2008**, 15, 31.
- [5] J. W. Booth, D. Shah, J. C. Case, E. L. White, M. C. Yuen, O. Cyr-Choiniere, R. Kramer-Bottiglio, *Sci. Robot.* **2018**, 3, eaat1853.
- [6] D. Kim, J. I. Kim, Y.-L. Park, *IEEE Robot. Autom. Lett.* **2019**, 4, 2289.
- [7] N. N. Goldberg, X. Huang, C. Majidi, A. Novelia, O. M. O'Reilly, D. A. Paley, W. L. Scott, *Soft Robot.* **2019**.
- [8] Y.-L. Park, B. Chen, R. J. Wood, *IEEE Sensors J.* **2012**, 12, 2711.
- [9] Y. Lee, M. Kim, Y. Lee, J. Kwon, Y.-L. Park, D. Lee, *IEEE/ASME Trans. Mechatronics* **2019**, 24, 67.
- [10] S. Xu, D. M. Vogt, W. Hsu, J. Osborne, T. Walsh, J. R. Foster, S. K. Sullivan, V. C. Smith, A. W. Rousing, E. C. Goldfield, R. J. Wood, *Adv. Funct. Mater.* **2019**, 29, 1807058.
- [11] K. Takei, T. Takahashi, J. C. Ho, H. Ko, A. G. Gillies, P. W. Leu, R. S. Fearing, A. Javey, *Nat. Mater.* **2010**, 9, 821.
- [12] C. To, T. L. Hellebrekers, J. Jung, S. J. Yoon, Y.-L. Park, *IEEE Robot. Autom. Lett.* **2018**, 3, 3216.
- [13] M. D. Dickey, R. C. Chiechi, R. J. Larsen, E. A. Weiss, D. A. Weitz, G. M. Whitesides, *Adv. Funct. Mater.* **2008**, 18, 7, 1097.
- [14] Y. Yoon, S. Kim, D. Kim, S. K. Kauh, J. Lee, *Adv. Mater. Technol.* **2018**, 1800379.
- [15] J. W. Boley, E. L. White, G. T.-C. Chiu, R. K. Kramer, *Adv. Funct. Mater.* **2014**, 24, 3501.
- [16] Y.-L. Park, D. Tepayotl-Ramirez, R. J. Wood, C. Majidi, *Appl. Phys. Lett.* **2012**, 101, 191904.
- [17] H.-S. Shin, J. Ryu, C. Majidi, Y.-L. Park, *J. Micromech. Microeng.* **2016**, 26, 025011.
- [18] Z. Kiu, C. Wu, M. Koishi, *Comput. Methods Appl. Mech. Eng.* **2019**, 345, 1138.
- [19] A. Mendizabal, P. Marquez-Neila, S. Cotin, *arXiv preprint* **2019**.
- [20] X. Tan, J. S. Baras, *Automatica* **2004**, 40, 1469.
- [21] M. Al Janaideh, S. Rakheja, C.-Y. Su, *IEEE/ASME Trans. Mechatronics* **2011**, 16, 734.
- [22] C. Ru, L. Chen, B. Shao, W. Rong, L. Sun, *Control Eng. Pract.* **2009**, 17, 1107.
- [23] M. Rakotondrabe, in American Control Conf. (ACC), Montreal, Canada June **2012**, p. 1646.
- [24] M. Ismail, F. Ikhouane, J. Rodellar, *Arch. Comput. Method. E.* **2009**, 16, 161.
- [25] S. Han, T. Kim, D. Kim, Y.-L. Park, S. Jo, *IEEE Robot. Autom. Lett.* **2018**, 3, 873.
- [26] D. Kim, Y.-L. Park, in *Int. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, Madrid, Spain **2018**, p. 7480.
- [27] M. Lacasse, V. Duchaine, C. Gosselin, in *IEEE Int. Conf. on Robotics and Automation*, ICRA, Anchorage, AK **2010**, p. 4842.
- [28] J. Jung, M. Park, D. Kim, Y.-L. Park, *IEEE Robot. Autom. Lett.* **2020**, 5, 2333.
- [29] Y. Menguc, Y.-L. Park, H. Pei, D. Vogt, P. Aubin, L. Fluke, E. Winchell, L. Stirling, R. J. Wood, C. J. Walsh, *Int. J. Robot. Res.* **2014**, 33, 1748.
- [30] D. Kim, J. Kwon, S. Han, Y.-L. Park, S. Jo, *IEEE-ASME T. Mech.* **2019**, 24, 56.
- [31] *Handbook of Smart Textiles* (Ed: X. Tao), Springer, Singapore **2015**.
- [32] J. M. Hutchinson, *Prog. Polym. Sci.* **1995**, 20, 703.
- [33] J. R. White, C. R. Chim, **2006**, 9, 1396.
- [34] K. Weiss, T. M. Khoshgoftar, D. Wang, *J. Big Data* **2016**, 3, 9.
- [35] Z. Zhang, M. Wang, A. Nehorai, *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**.

- [36] C. Sferrazza, R. D. Andrea, *IROS*, **2019**.
- [37] Y. Li, W. Li, G. Cao, *Inverse Probl. Imag.* **2019**, 13, 805.
- [38] S. Angenent, S. Haker, A. Tannenbaum, *Siam J. Math. Anal.* **2003**, 35, 61.
- [39] N. Courty, R. Flamary, A. Habrard, A. Rakotomamonjy, in *31st Conf. on Neural Information Processing Systems*, NIPS, Long Beach, CA **2017**, p. 3730.
- [40] R. Chartrand, B. Wohlberg, K. Vixie, E. Boltt, *Appl. Math. Sci.* **2009**, 3, 1071.
- [41] J.-B. Chossat, H.-S. Shin, Y.-L. Park, V. Duchaine, *ASME J. Mech. Robot.* **2015**, 7, 021008.
- [42] G. Shin, B. Jeon, Y.-L. Park, *J. Micromech. Microeng.* **2020**, 30, 034001.