

A substrate-less nanomesh receptor with meta-learning for rapid hand task recognition

Received: 8 March 2022

Accepted: 3 November 2022

Published online: 28 December 2022

Kyun Kyu Kim^{1,6}, Min Kim^{1,6}, Kyungrok Pyun³, Jin Kim⁴, Jinki Min³, Seunghun Koh², Samuel E. Root¹, Jaewon Kim³, Bao-Nguyen T. Nguyen¹, Yuya Nishio¹, Seonggeun Han³, Joonhwa Choi³, C-Yoon Kim⁵, Jeffrey B.-H. Tok¹, Sungho Jo^{1,2}✉, Seung Hwan Ko^{1,3}✉ & Zhenan Bao¹✉

 Check for updates

With the help of machine learning, electronic devices—including electronic gloves and electronic skins—can track the movement of human hands and perform tasks such as object and gesture recognition. However, such devices remain bulky and lack an ability to adapt to the curvature of the body. Furthermore, existing models for signal processing require large amounts of labelled data for recognizing individual tasks for every user. Here we report a substrate-less nanomesh receptor that is coupled with an unsupervised meta-learning framework and can provide user-independent, data-efficient recognition of different hand tasks. The nanomesh, which is made from biocompatible materials and can be directly printed on a person's hand, mimics human cutaneous receptors by translating electrical resistance changes from fine skin stretches into proprioception. A single nanomesh can simultaneously measure finger movements from multiple joints, providing a simple user implementation and low computational cost. We also develop a time-dependent contrastive learning algorithm that can differentiate between different unlabelled motion signals. This meta-learned information is then used to rapidly adapt to various users and tasks, including command recognition, keyboard typing and object recognition.

Humans can adapt to a diverse range of daily tasks with the aid of sensory feedback. Proprioception, in particular, provides an understanding of the real-time postural configuration of the hand and plays a key role in interactive tasks such as object recognition, manipulation and communication^{1–3}. Fundamental knowledge on proprioception is developed at an early age in children (the sensorimotor stage⁴) and involves correlating hand motions with the information relayed by cutaneous

receptors distributed throughout various locations of skin associated with skin stretching during the motion^{5–7}. This sensorimotor information serves as prior knowledge^{8,9}, which helps infants to quickly learn to perform new tasks with only a few trials. This process constitutes the basis of meta-learning^{10,11} (Fig. 1a).

Electronic devices can identify the movement and intended tasks of the human hand. For example, electromyography wrist bands

¹Department of Chemical Engineering, Stanford University, Stanford, CA, USA. ²School of Computing, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea. ³Department of Mechanical Engineering / Institute of Engineering Research, Seoul National University, Seoul, Republic of Korea. ⁴College of Veterinary Medicine, Seoul National University, Seoul, Republic of Korea. ⁵College of Veterinary Medicine, Konkuk University, Seoul, Republic of Korea. ⁶These authors contributed equally: Kyun Kyu Kim, Min Kim. ✉e-mail: shjo@kaist.ac.kr; maxko@snu.ac.kr; zbao@stanford.edu

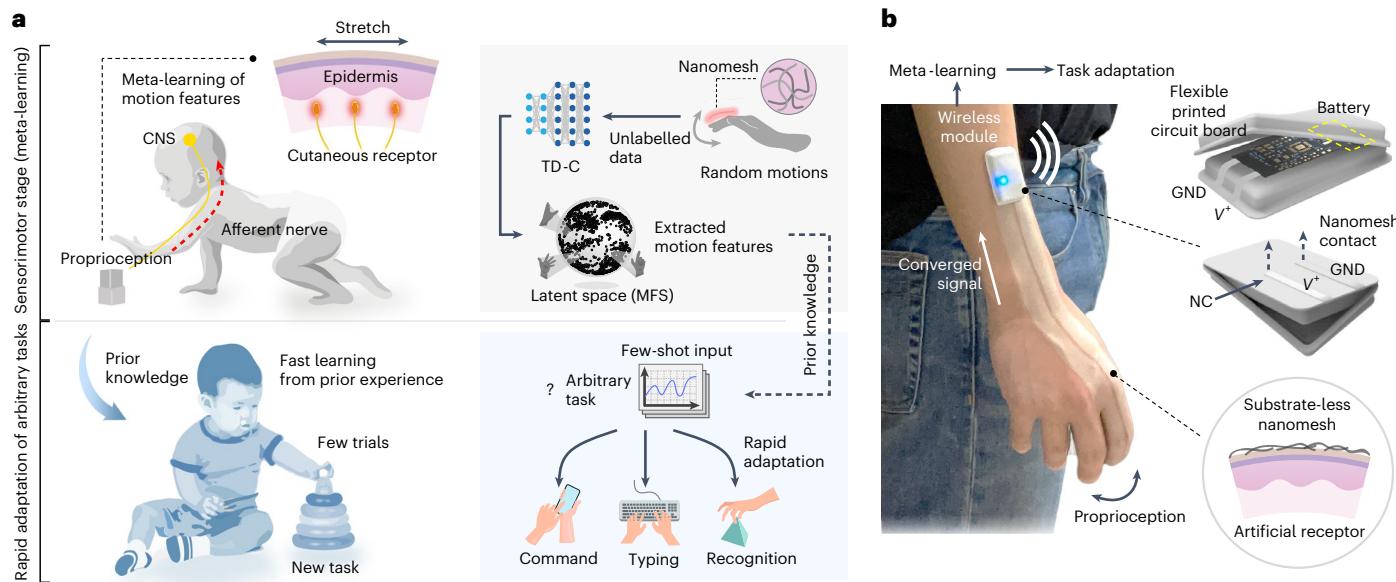


Fig. 1 | Artificial sensory intelligence system. **a**, Illustration of human sensorimotor stage that includes the meta-learning of motions through cutaneous receptors (proprioceptive information to the central nervous system (CNS)) and its rapid adaptation to unknown tasks. Resembling this nature, the first stage of our learning agent extracts the prior knowledge of human motion as MFS through unsupervised TD-C learning from random hand motion. Prior

knowledge is then transferred with few-shot labels that allows rapid adaptation to versatile human tasks. **b**, An artificial sensory intelligence system that consists of printed, biocompatible nanomesh cutaneous receptors directly connected with a wireless Bluetooth module through a nanomesh connector (NC), and is further trained through few-shot meta-learning.

and wearable electronic gloves can track hand movements. With the help of machine learning, these devices can perform complex tasks such as object interaction^{12–14}, translation of finger spelling^{15,16} and gesture recognition^{17,18}. However, their bulkiness and constraints on body movement limits their widespread adoption. Electronic skin sensors—such as artificial mechanoreceptors^{19,20}, ultrathin sensors²¹, stretchable sensors^{22,23} and nanomesh sensors^{24–26}—have rapidly developed in recent years, but they typically require multiple sensors and a high level of system complexity to pinpoint the motions from multiple joints²². Furthermore, the algorithms that have been used in such applications are based on supervised training methods that require large amounts of labelled data to perform individual tasks¹². Since the large variability of tasks and differences in individual body shapes generate different sensor signal patterns, these methods require intensive data collection for every single user and/or task^{27,28} (Extended Data Fig. 1).

In this Article, we report the development of a nanomesh artificial mechanoreceptor that is integrated with an unsupervised meta-learning scheme and can be used for the data-efficient, user-independent recognition of different hand tasks. The nanomesh is based on biocompatible materials and can be directly printed onto skin without an external substrate, which improves user comfort²⁵ and increases its sensitivity. The system can collect signal patterns from fine details of skin stretches and can be used to extract proprioception information analogous to the way cutaneous receptors provide signal patterns for hand motion recognition (Supplementary Note 1 and Supplementary Fig. 1). With this approach, complex proprioceptive signals can be decoded using information from a single sensor along the index finger, without the need for a multisensing array. Multijoint proprioceptive information can be reconstructed from low-dimensional data, reducing the computational processing time of our learning network (Supplementary Note 2 and Supplementary Fig. 2). When performing different tasks, signal patterns from various joint movements are transmitted using an attached wireless module placed on the wrist (Fig. 1b).

Similar to the learning process during an infant's sensorimotor stage, our unlabelled random finger motions provide prior motion representation knowledge. As a result, our learning framework does not require large amounts of data to be collected for each individual user. We developed time-dependent contrastive (TD-C) learning to provide an awareness of temporal continuity and to generate a motion feature space (MFS) representation of prior knowledge. This allows our system to learn prior knowledge using unsupervised contrastive learning from unlabelled signals collected from three different users to distinguish user-independent, task-specific sensor signal patterns from random hand motions. This prior knowledge can subsequently be transferred to other users with an accuracy of 80% within 20 transfer training epochs. We show that the pretrained model can quickly adapt to different daily tasks—motion command, keypad typing, two-handed keyboard typing and object recognition—using only a few hand signals.

Cutaneous nanomesh artificial mechanoreceptor

Proprioception—our body's ability to sense movement, action and location—relies on encoding mechanical signals collected by numerous cutaneous receptors into neural representations, that is, patterns of neural activities⁷. These cutaneous receptors are activated by the stretching of the skin and can detect various joint movements (Supplementary Note 1). Such a function can be emulated by employing a single two-terminal substrate-less nanomesh element along the index finger extending to the wrist. The integrated signals of the entire finger postures and movements can be collected (Fig. 2a). Due to the direct contact of the nanomesh with skin, it closely follows the topography of the skin and transforms even micro-movements into resistance variations with high sensitivity (Supplementary Fig. 5). Signal outputs corresponding to fine details of elongation and compression of the skin due to arbitrary hand movements are then collected and transmitted by a wireless module (Methods and Extended Data Fig. 3).

In addition to the ability to generate proprioception-like diverse sensing output patterns based on fine movements of finger and wrist, our nanomesh is biocompatible, breathable and mechanically

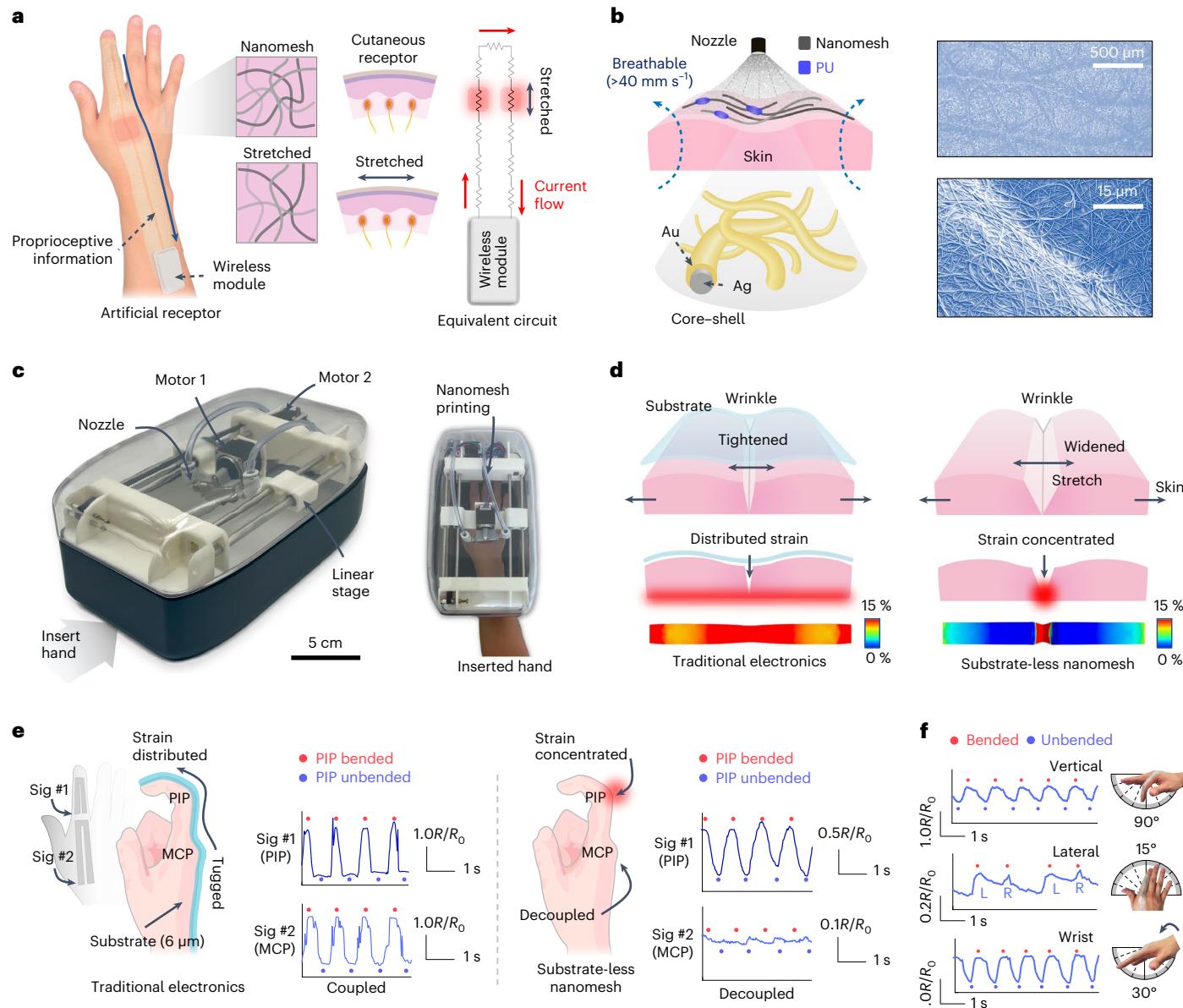


Fig. 2 | Nanomesh and device performance. **a**, Comparison between a human cutaneous receptor and our nanomesh receptor. Resistance variations generated by the nanomesh are measured through the wireless module. **b**, Reinforced nanomesh through consecutive printing of PU and core-shell Ag@Au structures. The nanomesh shows endurance against cyclic friction and maintains high breathability and biocompatibility. The image on the right shows the intimate contact of the nanomesh above the skin with a magnified view via the scanning electron microscopy (SEM) image. The images are representative of three

independent experiments. **c**, Photograph of the portable nanomesh printer. **d**, Traditional electronics and substrate-less nanomesh under 15% strain. The substrate constrains the wrinkles due to its lack of intimate contact (Methods). **e**, Nanomesh on the MCP area is activated by the PIP movement alone in traditional electronic design (substrate thickness, 6 μm ; modulus, 7 MPa), whereas no coupling was seen in our design. The nanomesh was printed with 16 cycles of spraying. **f**, Nanomesh response to both finger and wrist movements.

stable (Fig. 2b). The gold-coated Ag (core)/Au (shell) nanostructures of the nanomesh prevent the cytotoxic silver ions from direct contact with skin²⁹. The long dimension of the Ag (core) wires (length, ~80 μm) results in high mechanical stability³⁰ (Supplementary Fig. 5). No inflammation response of skin to the nanomesh was observed during in vitro and in vivo experiments (Supplementary Fig. 6). A polyurethane (PU) coating was sprayed onto the nanomesh to create a droplet-like porous structure to prevent the nanomesh from being easily scratched and retaining an air permeability of more than 40 mm s^{-1} (Supplementary Fig. 8). A scratch test was carried out in which the sample was printed on porcine skin to mimic human skin. The PU-reinforced nanomesh (Supplementary Fig. 10) outlasted the unprotected nanomesh (over 1,500 cycles) when subjected to

scratching from a silicon tip (Supplementary Fig. 11). These results show that the PU-protected nanomesh is suitable for daily activities (Supplementary Note 4 and Supplementary Figs. 3 and 13), but can still be removed as needed by rubbing during handwashing (Supplementary Fig. 12). Furthermore, the nanomesh was readily applied to skin using a custom-designed portable skin-printing device (Fig. 2g, Supplementary Fig. 14 and Supplementary Video 6). A subsequently attached silicone-encapsulated wearable wireless module further provided user comfort and ensured a self-contained system (Extended Data Fig. 3 and Supplementary Video 7).

The substrate-less feature of our artificial receptor is a marked improvement from previously reported substrate-based wearable electronics. Due to its ultraconformal nature (Supplementary

Note 3 and Supplementary Fig. 7), the substrate-less nanomesh enables capturing proprioceptive signals without losing its local information (Supplementary Fig. 21). Although ultrathin sensors (sub-micrometre) were recently demonstrated, challenges remain in terms of reducing motion artifact noises since even an extremely thin layer can still suffer from substantial information degradation²⁵. The benefit of reducing motion artifacts by directly printed sensors on the human body has been previously demonstrated³¹. The high conformability of our substrate-less interface is critical in the resistance of the sensor to motion artifacts. As illustrated in Fig. 2d, during the flexion of joints with a printed substrate-less nanomesh, the strain caused by the opening and closing of the wrinkles contributed to a majority of detected changes. However, the presence of a substrate reduces the conformability of the sensor and hinders the detection of changes associated with wrinkle movement, making it harder to detect intricate changes from different types of finger movement. In addition, we designed a control experiment in which a thin layer of PU substrate (6 µm) was applied before nanomesh printing and compared with two separate printed substrate-less nanomesh sensors, to gather signals from both proximal interphalangeal (PIP) and metacarpophalangeal (MCP) regions (Fig. 2e and Supplementary Fig. 9). During the isolated PIP movement under normal conditions, the stretching of the substrate placed on the top of MCP joints (tugging the MCP region; Fig. 2e) resulted in strong signal changes in the MCP (Sig #2) area. However, for the substrate-less nanomesh, most of the strain was centred on the PIP joint, activating only the nanomeshes on the PIP (Sig #1) area (resolution, 15 mm; Supplementary Fig. 9). The localized and decoupled signal properties of the substrate-less nanomesh enabled better learning performance (Extended Data Fig. 4). In contrast, performance degradation can be observed on applying a substrate. The nanomesh further differentiated various patterns of hand motions (Fig. 2f) and exhibited high durability on various environmental effects (Supplementary Note 4 and Supplementary Fig. 3). These overall characteristics of the substrate-less nanomesh rendered the measurement of multijoint proprioceptive information with a single sensor element possible (Supplementary Fig. 21). Importantly, our approach enabled the minimization of circuit complexity and computing resources by providing low-dimensional but highly informative proprioceptive information to the learning network (Supplementary Note 2).

Meta-learning and few-shot adaptation to a new user

Inspired by the development process of proprioception during the sensorimotor stage, we aim to create a general latent feature space, termed MFS, to represent prior knowledge of human finger motions and make it generalizable to different users and daily tasks. For arbitrary users with newly printed sensors, different signal patterns will be generated including changes in the signal amplitudes and frequencies due to distinctive hand shapes and postures (Supplementary Fig. 15). When a learning model tries to infer hand gestures from signals generated by users that were not included in the training dataset, the variabilities lead to substantial out-of-distribution and domain shift errors^{27,28}. Furthermore, considering the diversity of hand gestures and tasks performed in daily lives, it was necessary to both collect a labelled training dataset and modify the model architecture for each individual task when applying previous supervised learning models to general usage. To address these limitations, we set out to generate a separable MFS that can be used on signal patterns not shown in the training dataset. Although other stable methods exist for training neural networks to build feature spaces for later adaptations, these methods either deal with formalized data (tokenized words^{32,33} or images^{34,35}) or have restricted target users and tasks^{12,36}. Consequently, these methods are unable to use small amounts of random motion data to differentiate pattern differences caused by variations in both tasks and user. Therefore, instead of mapping sensor signals to specific motion

labels, we developed a learning model that utilizes unlabelled random motion data to meta-learn, allowing us to discriminate between different signals by projecting signals into a separable space. In short, after a new user provides a small set of actions, these signals are separated in our MFS to be compared with real-time user inputs, which allows our metric-based inference mechanism to correctly recognize gestures of the user even though the signal patterns are different from those of the users and tasks in the training data.

To generate MFS without labels, we adopt an unsupervised learning method called contrastive learning in which the model learns to discriminate different inputs by maximizing the similarities of positive pairs augmented from the same instances and minimizing the similarities of different instances. However, recently developed methods^{37,38} have been designed to encode image data and do not consider time correlation. Analogous to how the awareness of motion continuity in time helps infants to develop a stable perception³⁹, models without time consideration would inevitably omit consecutive sensor signals, resulting in an unstable motion space, which lowers the accuracy (Extended Data Fig. 5e). Furthermore, we need to carefully select data augmentation methods, such that the corresponding hand postures of the data-augmented signals remain consistent.

We, therefore, propose TD-C learning that uses temporal features to generate MFS from unlabelled random hand motions. Instead of providing specific labels to train our neural network, we generated positive pairs based on time-wise correlation and trained our model to minimize distances (based on cosine similarity) between the positively paired signals in our encoded latent space. First, we generated strong positive pairs through data augmentation. Given a sensor input grouped with a sliding time window (Supplementary Note 5 and Supplementary Fig. 4 show the model performance for various time-window sizes), we generated two augmented sensor signals through jittering data augmentation (Extended Data Fig. 7). Although other temporal signal augmentation methods exist⁴⁰, these methods altered the signal amplitude patterns and hindered the model from distinguishing different motions (Extended Data Fig. 7c-e and Methods). Originating from the same sensor signals, these two signals were considered as a strong positive pair since they represent the same motion. Second, we assigned consecutive augmented signals (distanced at most a half of the time window) as positive pairs. These consecutive time windows represent similar hand poses since our hand motions are continuous. Therefore, we assigned a connection strength between the signals based on their time differences and our model receives discounted positive rewards that are proportional to the connection strength for grouping consecutive sensor signals (Methods). A transformer encoder⁴¹ supported by the attention mechanism was adopted to encode long-term temporal signal patterns without iterative signal processing. Giving a tolerance for the model to map semantically similar sensor signals based on temporal correlation, the model could generate better quality features and showed stronger performances when it was transferred to different tasks. Extended Data Figs. 2 and 7a,b illustrate the experimental results of our model's ability to distinguish different sensor signals even without any labels.

To investigate the model's ability to extract useful motion features, we collected unlabelled random finger motions (bending and rotation) of the PIP, MCP and wrist motion signals through a single substrate-less nanomesh and then conveyed the information through the TD-C network (Extended Data Fig. 1a and Methods). The joint signals of PIP, MCP and wrist are clearly represented in the uniform manifold approximation and projection (UMAP)^{42–46} (Extended Data Fig. 1b), illustrating the ability of the TD-C model to extract useful information from coupled signals. The signals of the combined motion of PIP and MCP joints were located in between the PIP-only and MCP-only motions, and therefore, our nanomesh sensor signals contain all three joint movements and can be used to effectively translate skin stretches into multijoint proprioception. Moreover, to determine the use of resistive variations

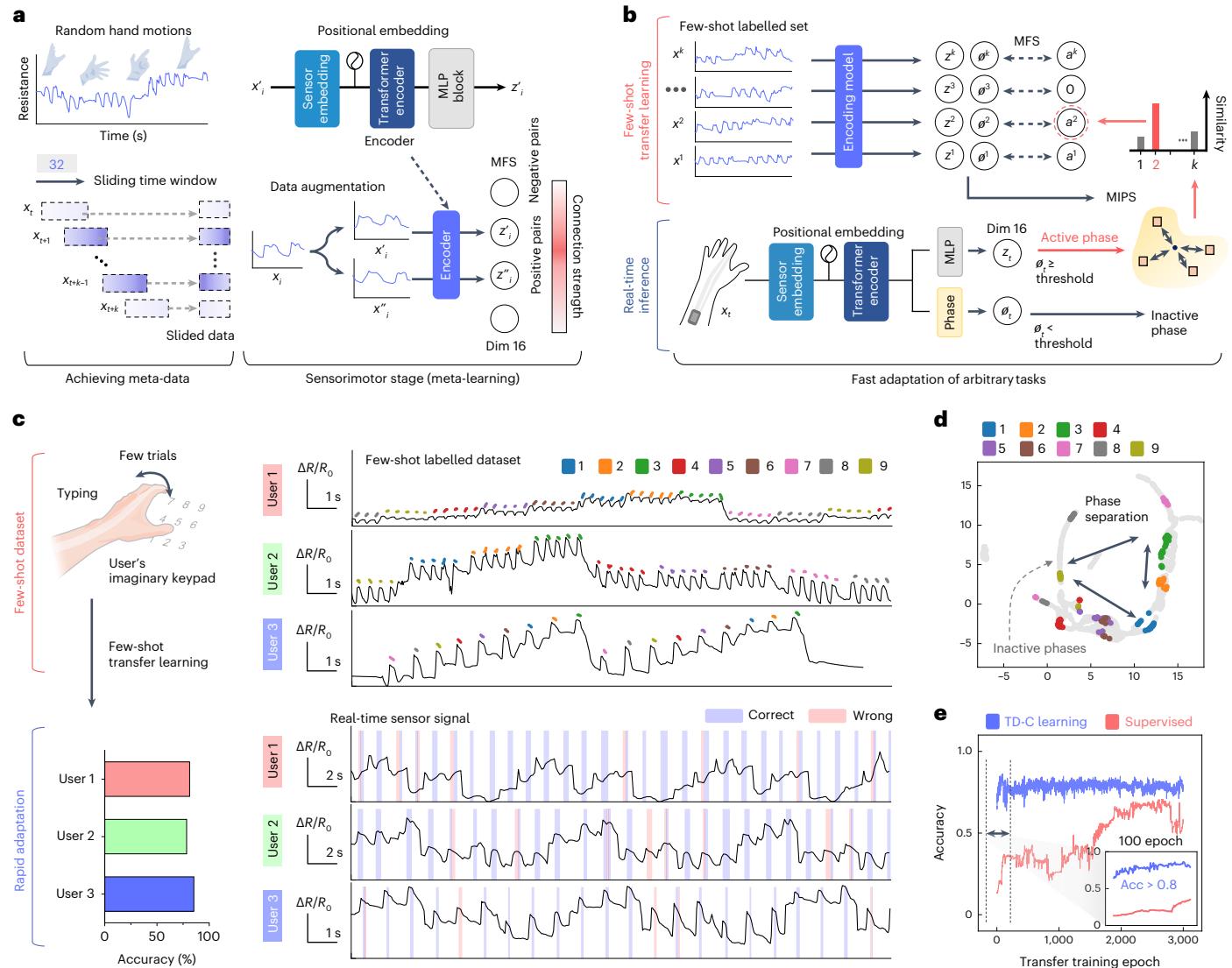


Fig. 3 | TD-Clearing principle and experimental results on learning performance. **a**, Sensor signal processing and unsupervised TD-C learning for learning the MFS. **b**, Transfer learning and metric-based real-time inference mechanism with provided few-shot labelled dataset gathered from each arbitrary user. Dim 16 and MIPS stand for dimension of 16 and maximum inner product search, respectively. **c**, Few-shot dataset and real-time sensor signal prediction

for different users typing nine different keys. **d**, UMAP projection of latent numpad typing vectors, where the grey dots indicate inactive phase signals and coloured dots indicate active phase signals. **e**, Inference accuracy trends for nine-class numpad typing with further transfer training epochs: model pretrained with TD-C learning (blue line) and the same model with last linear layer modification for classification pretrained with supervised learning (red line).

between joints for complex tasks, we utilized these signals for actual motion prediction. As shown in Supplementary Fig. 21, our system was able to determine the position and bending angle of motion (bending and rotational), as well as multimodal movements.

The extracted motion features were then used for few-shot adaptation to arbitrary tasks. To overcome domain shift issues, we adopted a metric-based inference mechanism to predict users' gestures in various daily tasks (Fig. 3b). The model was first fine-tuned to refine the MFS by additionally giving rewards for mapping the same-classified latent vectors to a closer feature space. The model performs maximum inner product search (MIPS) with a given few-shot labelled dataset to identify the current gesture. Comparing signals generated from the same user with the aid of the highly separable MFS, the model can avoid domain shift issues and utilize motion knowledge generated from TD-C learning. Details of the learning procedures are further described in Methods and those of the pseudocode are provided in Supplementary Fig. 17.

We note that in between two different gestures (active phase), there exists an intermediate period where users have no specific intention (inactive phase). Since inactive phases occur between active phases of motion, it is unavoidable for the model to project inactive phase signals near to active phase signals and considering temporal correlations (Fig. 3d). To avoid misclassification caused by neighbouring inactive phases, we additionally train a phase block in transfer learning to clearly delineate the active and inactive phase of gestures (Supplementary Fig. 16). Specifically, input signals are regarded as active phase signals if the corresponding phase variables generated by the phase block are higher than a predefined threshold. Therefore, in actual testing time, we performed MIPS only between the active phase signals and few-shot demonstrations annotated as active phases. Our model with the phase block clearly separated these entangled phases, and ablation studies on adding the phase block are shown in Extended Data Fig. 5f. The user-wise few-shot labelled dataset and the corresponding model predictions are illustrated in Fig. 3c. With transferable

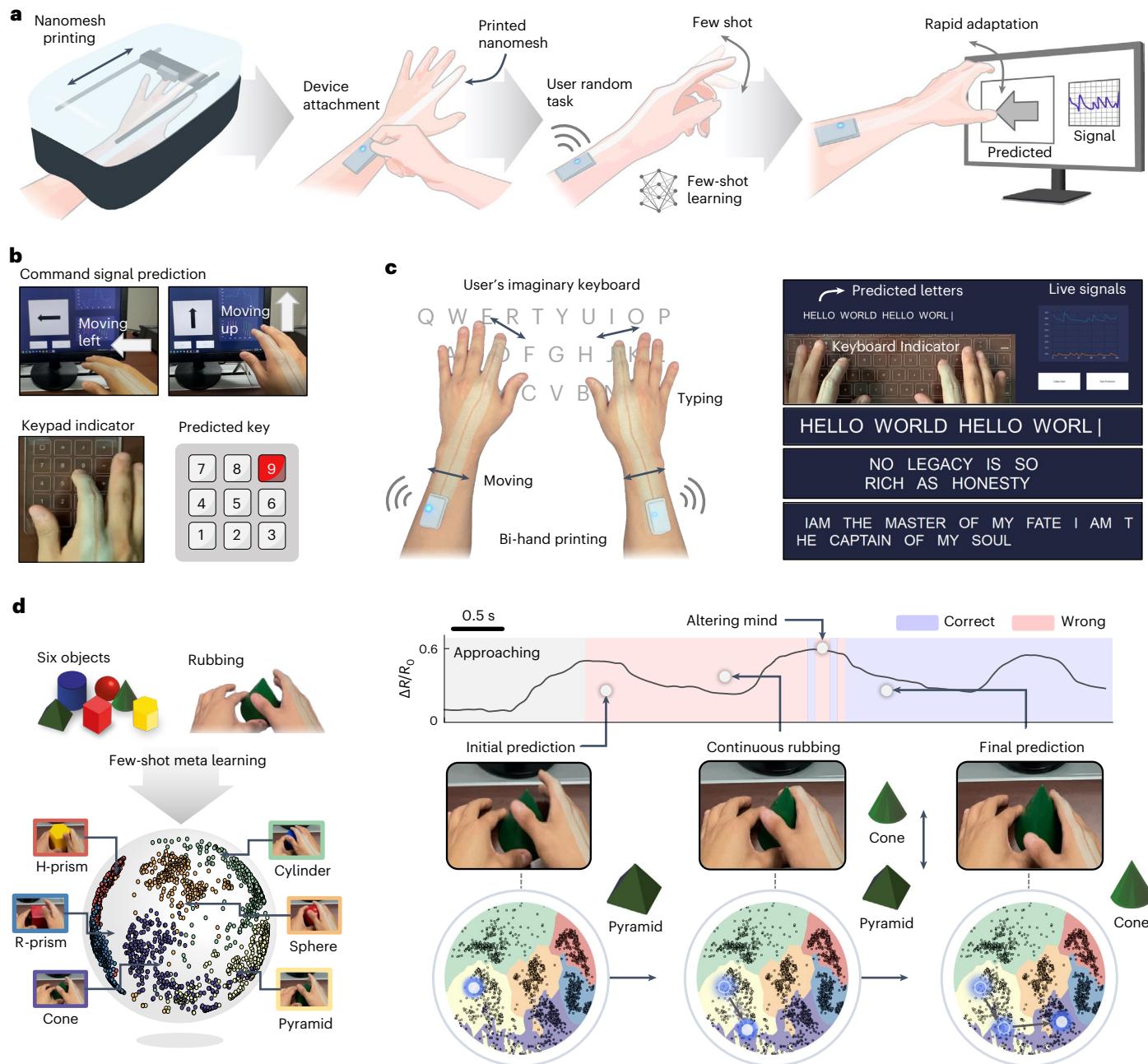


Fig. 4 | Demonstration of fast adaptation to various daily tasks. **a**, Illustration of the general process of nanomesh printing, wireless device attachment, few-shot inference and prediction. **b**, Command signal prediction and virtual keypad typing recognition. Four command signals are initially provided only for lateral hand motions. The model can further be generalized to predict both lateral and vertical hand motions as the user provides more demonstration on vertical hand signals. **c**, Two-handed QWERTY keyboard typing recognition

with nanomesh printed on both hands. The predicted letters appear on the user interface as a user consecutively types various sentences. The acrylic keyboard identifier is placed beneath to show the intention of the user (Supplementary Fig. 20). **d**, Group of recognition objects and UMAP projection of embedded vectors in a few-shot demonstration set. The sequential changes of position in the projected embedding space of real-time user signals as the user starts to interact with an object.

MFS and user-wise metric-based inference, our model can robustly predict hand actions from different users. In addition, our learning framework can handle variations in nanomesh density (Supplementary Fig. 18). Our model's ability to transfer knowledge to users with newly printed sensors compared with a traditional supervised learning framework was demonstrated in Fig. 3e. Although the model trained with supervised learning methods required more than 3,000 training epochs to adapt to the new user, the model trained with our developed learning framework showed more than 80% accuracy within 20 transfer training epochs. Extended Data Fig. 5d illustrates the UMAP projection

of embedded signal vectors in MFS into a two-dimensional space where feature vectors were discriminated into correlated feature spaces.

Fast adaptation to arbitrary tasks

Having demonstrated the ability of our learning framework to use unlabelled random motion data to learn MFS and make gesture inference for arbitrary users with few-shot demonstrations, several representative daily tasks were subsequently conducted, which include command signal recognition, one-handed numpad typing (Fig. 4b), two-handed sentence typing (Fig. 4c) and object recognition (Fig. 4d).

These applications demonstrate the potential of using our system in daily life, including applications such as human motion recognition, human device interactions and human object interactions. For each individual task, the user first printed the sensor by applying 16 cycles of nanomesh printing through the portable printing machine (Fig. 4a). A wireless module was then attached to the two terminals of the nanomesh. The user then provided a few-shot labelled dataset by performing each individual gesture five times. The generated sensor signals and the corresponding labels were transmitted to the receiver by the module. For grouping of the latent feature vectors based on a given task-specific dataset, we further trained the model for additional 20 transfer training epochs by providing positive rewards for mapping the same gestures into closer vectors.

We observed that our model can efficiently adapt to identify new gestures added to our few-shot labelled dataset without requiring any modifications to model structures or intensive training processes. Even though initially trained only for lateral finger motions, our model can be further enhanced to recognize vertical motions by further providing corresponding few-shot labels. After initial training for left- and right-hand gestures, our model could then recognize gestures for all directions (left, right, up and down) by additionally providing up- and down-hand gesture signals into the initially trained model (Fig. 4b and Supplementary Video 1). Furthermore, our model was able to distinguish fast and subtle movements of finger motions that move along the user's small imaginary keypad by discriminating between nine different numpad keys (Supplementary Video 2). Our model achieved 85% accuracy within 20 transfer epochs (Extended Data Fig. 5c). We used the numpad typing task as a major benchmark for comparing our methods with others, since it consists of nine classes of similar hand postures. To analyse the transfer capability of the training framework to the opposite hand, an additional sensor was printed on the left hand, allowing the user to type with two hands on the entire QWERTY keyboard (Fig. 4c). When given a sentence, an arbitrary user initially provided few-shot labels by typing each character five times. The pretrained model was further transferred to two different models to discriminate between the two-handed keyboard typing signals (Supplementary Fig. 19) of typed sentences, namely, 'Hello World' and 'Imagination' (Supplementary Video 3). Moreover, it also can be directly applied to predicted longer sentences, such as 'No legacy is so rich as honesty' (William Shakespeare) and 'I am the master of my fate I am the captain of my soul' (Invictus) (Supplementary Video 4). Minor typos may occur when typing longer sentences and can be readily modified through open-source word correction libraries for further practical usage (Supplementary Fig. 25). The above keyboard application demonstrated that our inference mechanism can accurately deal with asynchronously generated multiple sensor signals to decode wide finger motion ranging from the left and right ends of the keyboard. The full keyboard of the alphabet was predictable (Extended Data Fig. 8), where each hand took charge of the left half and right half of the keyboard.

In the same way as humans identify objects through gestural information during interactions¹, users with the nanomesh could continuously rub the surface of different objects and eventually recognize them. The sphere diagram (Fig. 4d) illustrates the UMAP projections of embedded labelled dataset onto the contour of a sphere, where different colours represent six different objects. Since the shape of the pyramid and cone are similar and thus hard to distinguish from each other, the corresponding embedded points (Fig. 4d, yellow and purple) were initially intermixed together. As a result, although the hand interacted with a pyramid, the model initially predicted a cone. Continuously rubbing twice allowed the model to eventually predict it as a pyramid. The embedded vectors continuously move from the yellow region to the boundary and then to the purple region (Fig. 4d). This is akin to humans taking time to recognize objects, and altering their minds during interactions with various sections of the object.

The model can classify the objects with 82.1% accuracy via 20 transfer epochs versus thousands of epochs using a supervised training method (Extended Data Figs. 4 and 5b and Supplementary Video 5).

Conclusions

We have reported a substrate-less nanomesh artificial mechanoreceptor equipped with meta-learning. The system mimics human sensory intelligence and exhibits high efficacy and rapid adaptation to a variety of human tasks. Similar to cutaneous receptors recognizing motion via skin elongation, our nanomesh receptor gathers hand proprioception signal patterns with a single sensing element. The substrate-less feature of the nanomesh receptor allows intricate signal patterns to be collected from many areas using a single sensor. With a highly separable MFS, our learning framework can effectively learn to distinguish different signals, and this knowledge can be used to robustly predict different user tasks with the aid of a metric-based inference mechanism. The robustness of our model allows quick adaptation to multiple users regardless of variations in density for printed nanomesh receptors. It is expected that increasing the number of nanomesh elements to five fingers or more will enable the recognition of more complex motions, allowing future applications in robotics, metaverse technologies and prosthetics.

Methods

Software and system design of wireless measuring module

The device consists of a miniaturized flexible printed circuit board that includes an analogue-to-digital converter sensing element, Bluetooth low-energy module, lithium polymer battery and anisotropic flexible connector (Extended Data Fig. 1). An embedded nanowire network at the end of the connector allows conformal and direct contact with the existing nanomesh electronics. The two terminals of the nanomesh are connected to the wireless module via direct contact with the embedded nanowire network; then, the translated digital signals from a voltage divider are further conveyed to the receiver through Bluetooth transmission at a data rate of 30 Hz. The compact wireless module can measure arbitrary hand motions and maintain conformal contact with the printed nanomesh. The wireless system is programmed through a system on chip (CC2650, Texas Instruments) with the Code Composer Studio software (version 9.3.0, Texas Instruments). The packet of the measured analogue sensor signals is transferred at a data rate of 30 Hz, where the receiving system on chip transforms the received packets into a universal asynchronous receiver-transmitter (UART) data transmission. A Python-integrated LabVIEW (version 3.6.8) system transfers these data into the learning network for task identification.

Mechanical simulation of substrate-less nanomesh

Strain distribution of traditional substrate-based and our substrate-less electronics is compared through the finite element method (version 5.6, COMSOL Multiphysics). The depth and width of the wrinkles are set as 500 and 200 μm , respectively. A thin layer (6 μm , measured by Bruker Dektak XT-A; Supplementary Fig. 22) of PU material (Alberdingk) is applied above the wrinkle to investigate the strain distribution of substrate-based electronics.

Portable printing system

The linear stage and nozzle are moved by the Nema 11 stepping motor, controlled by an L298N controlling driver (Supplementary Fig. 14). The spray nozzle (air-atomizing nozzles, Spraying Systems) is connected to 20 psi air pressure through a compact air compressor (Falcon Power Tools). The nozzle is controlled via a 5 V activated solenoid gas valve, and the entire nozzle is moved through the linear stage with 20 mm s^{-1} speed. A 5 mg ml^{-1} of Ag–Au nanomesh solution is prepared and sprayed for 16 cycles over the entire hand covered with a polydimethylsiloxane stencil mask. Then, three cycles of diluted (25 mg ml^{-1}) water-based PU (U4101, Alberdingk) were spray coated.

Biocompatible Ag@Au core–shell nanomesh synthesis

First, Ag nanowires (length, ~80 µm; diameter, ~80 nm) were synthesized by a modified one-pot polyol process. In 50 ml of ethylene glycol, 0.4 g of polyvinylpyrrolidone (M_w , ~360,000) and 0.5 g of silver nitrate were sequentially dissolved using a magnetic stirrer. Then, 600 µl FeCl₃ (2 mM) was rapidly injected into the mixture and stirred mildly. The stirrer was carefully removed from the mixture solution once all the chemicals were thoroughly mixed. Finally, the mixture solution was immersed in a preheated silicone oil bath (130 °C). After 3 h of nanowire growth, the resultant solution was cleaned using acetone and ethanol to remove the chemical residues along with centrifugation at 1,500×g for 10 min for three times. The purified Ag nanowires were redispersed in water for use. For preparation of the Au precursor solution, 30 mg chloroauric acid (formula weight, 339.79; Sigma-Aldrich), 17 mg sodium hydroxide (Samchun Chemicals) were dissolved in 70 ml distilled water. After 30 min, the hue of the solution turned from yellowish to transparent, and 33 mg sodium sulfite (Sigma-Aldrich) was added to the solution. For the Ag nanowires preparation solution, 800 mg of poly(vinylpyrrolidone) (M_w , 55,000; Sigma-Aldrich), 70 mg sodium hydroxide, 300 mg L-ascorbic acid (Samchun Chemicals) and 10 mg sodium sulfite were added to 100 ml of the previously synthesized Ag nanowire solution (Ag, 20 mg). Thereupon, the Au precursor solution was slowly poured into the Ag nanowire preparation solution for 2 min. After 30 min, the Ag@Au core–shell nanowires were fully synthesized and cleansed for three times through centrifugation and dispersed into water with 10 mg ml⁻¹ concentration and sprayed for nanomesh formation (Supplementary Fig. 23).

Nanomesh breathability measurement

The measurement was conducted using a custom-built acrylic air channel (Supplementary Fig. 8). An air pump was built at the back end of the air channel to create a consistent airflow, and the nanomesh was installed in the centre. The flow rate and pressure drop between the nanomesh were monitored via a flow meter and differential manometer. The pressure drop is measured at varying flow rates, and the air permeability of different samples is calculated through Darcy's law ($q = -\frac{kAP}{\mu L}$, where q , k , P , μ and L denote the flux, permeability, pressure, viscosity and channel length, respectively).

Cell toxicity

Cell toxicity is compared between the control, only PU, Ag nanomesh with PU and Ag@Au nanomesh with PU. L929 (KCLB), a mouse fibroblast cell, was cultured in Dulbecco's modified Eagle's medium (11885-084, Thermo Fisher Scientific) containing 10% foetal bovine serum (F2442, Sigma-Aldrich) and Anti-Anti (15240-062, Thermo Fisher Scientific) at 37 °C with 5% CO₂. Nanomeshes with different conditions (PU, Ag with PU and Ag–Au with PU) were prepared and used for in vitro toxicity evaluation. After attaching the sample to the bottom in a 1/10 size of the well area, 5 × 10⁵ cells per well were seeded on the six-well culture plate (3516; Costar). After incubation for 24 h, cell morphological changes were observed and photographed using a Nikon Eclipse TS100 microscope. To analyse the toxicity of cells, an MTT assay was performed using three wells for each sample. The cells were incubated in 0.5 mg ml⁻¹ MTT solution (M6494, Thermo Fisher Scientific) at 37 °C for 1 h; then, the solution was removed and dimethyl sulfoxide was used to dissolve MTT formazan. The absorbance was measured at an optical density of 540 nm using an Epoch Microplate spectrometer (BioTek Instruments) and normalized using a control (unpaired, two-tailed Student's *t*-test).

In vivo test (spraying)

Four-week-old Hos:HR-1 male mice were purchased from Central Laboratory Animal for spraying. All the experiments involving mice were performed with the approval of the Konkuk University Institutional Animal Care and Use Committee (KU21212). All the animals were maintained in

a 12 h light/dark cycle at 23 ± 1 °C and 50 ± 10% relative humidity with free access to food and water. Hos:HR-1 mice were anaesthetized by intraperitoneal administration of alfaxalone (100 mg kg⁻¹) and xylazine (10 mg kg⁻¹). After placing the anaesthetized mouse under the portable printing system, the nanomesh solution was sprayed. A surgical cloth with a window of about 1 × 4 cm² was covered so that the nanomesh could be applied only to the exposed area. To maintain the body temperature, a heating pad or an infrared lamp was used. After application, the mice were returned to the cage for recovery; 24 h later, CO₂ euthanasia was performed to obtain the skin sample. Fixed skin samples were dehydrated (70%, 80%, 90% and 100% ethanol), transferred to xylene for 2 h incubation and infiltrated with paraffin. Paraffin blocks were prepared using a Tissue-Tek TEC5 tissue-embedding console system (Sakura Fine-tek Japan), and sectioned using the Microm HM 340E microtome (ThermoScientific) at a thickness of 5 µm. Subdermal implant of the Ag–Au nanomesh is also carried out within seven days (Supplementary Fig. 6). The tissue slices were placed on a glass slide (Marienfeld) and stained with haematoxylin and eosin. The slides were photographed using an Olympus IX70 fluorescence microscope and a Nikon D2X instrument.

Graph analysis of nanomesh network

Graph theory is used to analyse the electrical properties of the nanomesh network (Supplementary Fig. 5). First, the adjacent matrix (A) is formed from the distributed random nanomesh network (a random nanowire (length, 100 µm) network is distributed in 500 × 500 µm²). Then, the incidence matrix (I) is formed to generate the graph network of the nanomesh, where the edge and node represents nanowires and intersections, respectively. According to Ohm's law, the current flow in each wire can be calculated by $i = Clx$, where C is the conductivity and lx represents the voltage difference of the nodes (x is the node matrix). Then, the current of each node can be expressed by $I^T Clx$ following Kirchhoff's law. Therefore, the voltage and current of the network can be related by Laplacian matrix $L = ClI^T$, with the relationship of $Vi = L$. Voltage and total resistance of the nanomesh can be derived using the boundary conditions of the input and output current flow of both ends of the edge ($i(o) = 1, i(N) = -1$). The resistance of the network under 20% strain and the sensitivity of the nanomesh under 15% strain are derived from averaging 100 simulation runs for each segment. The percolation threshold is found at six cycles of spraying, where the network density can be derived as 180#/500 µm. Therefore, the approximate nanomesh density per spraying cycle can be obtained.

Gesture recognition experiments

The nanomesh was applied with 16 cycles of spraying (linear stage with 20 mm s⁻¹ speed and 10 mg ml⁻¹ of Ag–Au nanomesh solution). Then, the wireless module is attached to the terminal of the nanomesh and three users are asked to perform data collections for three times for each application (total, 10 min). Each few-shot data collection lasted around 1 min (~1,800 data points), with 30 s of rest before the next trial. The data are saved to check whether they are effectively collected, and the adaptation performance is evaluated. This process took less than 5 min (the achieved raw data are shown in Supplementary Fig. 24). All the experiments were performed in strict compliance with the guidelines of the Institutional Review Board at Seoul National University (project title, Electrophysiological signal sensing by direct-printed electronicskin; IRB no. 2103/001-008). Informed consent was obtained from all the participants.

Dataset acquisition methods

Previous studies on gesture classification from sensor signals have focused on model prediction accuracies with a designed experimental setup where the users are instructed to perform a specific task having a fixed number of classes. Our model framework is designed to learn transferable information from unlabelled motion data of a limited number of users and be generalized to various daily tasks with only a few labelled sets given as guidance. Our general objective here is to

verify the ability of pretrained models to adapt to various daily tasks and compare their ability to create a normal supervised learning framework. There are three different types of dataset that are used in our experiment: pretrained dataset $X_{\text{pretrain}} = \{s_i\}$, few-shot labelled dataset $X_{\text{train}}^{\text{user}, j} = \{x^k, a^k\}$ for task adoption and task-wise testing dataset $X_{\text{test}}^{\text{user}, j} = \{x^k\}$. The pretrained dataset is used when pretraining our models and it is generated from random motions (900 s of random finger movements). The models that are used for task-wise adaptation (Supplementary Videos 1–7) are trained with unlabelled data that are collected from three users performing random hand motions when skin sensors are printed on their fingers. To quantitatively compare our training framework to a normal supervised learning framework, we additionally collected a labelled dataset as a user types the keyboard numpad. The keyboard inputs are regarded as data labels and we use these labels for the supervised pretrained model. Quantitative comparisons between our framework and supervised pretrained frameworks are based on the pretrained model using the numpad typing dataset, whereas labels are only used for supervised learning framework. We prepared four different applications, each representing real-life hand computer interaction cases (Fig. 4c–e). For each application, we collected a five-shot labelled dataset for transfer learning our model. In terms of the five-shot dataset, it means a user performs each gesture class for five times, for example, in the object interaction task, rubbing an object from the left end to the right end would be regarded as a single shot. The corresponding gesture labels are collected as the user is typing the keyboard keys that represents specific gesture labels. The collected few-shot labelled dataset is then used for retraining the data-embedding model. For reflecting real-life usage scenarios, the task-wise testing dataset is collected as a user naturally interacts with the system, for example, typing different numpad keys or rubbing random objects after retraining the model.

Details of signal preprocessing and data augmentation

To limit the sensor input domain, the sensor values are normalized through minmax normalization. The minimum and maximum values among each individual pretrained dataset and few-shot labelled dataset are used to normalize the corresponding dataset. For real-time user test scenarios, the user signal inputs are normalized based on the minimum and maximum values of a given user labelled dataset. Given a signal group collected for the same user $R = \{r_i, i \in N\}$, the corresponding normalized signal group $S = \{s_i \in [0, 1], i \in N\}$ is generated as

$$s_i = \frac{r_i - \min(R)}{\max(R) - \min(R)}, \forall r_i \in R.$$

The sensor signals are collected at 30 fps and signal s_t represents the normalized sensor signal collected in time frame $t \in N$ (s_t and s_{t+1} are 1/30 s apart). Consecutive 32 sensor signals are grouped into a single model input, so that the model can utilize not only the current sensor signal but also the temporal signal patterns to generate signal embeddings. A sliding time window of size 32 with stride 1 is used to group consecutive raw signal inputs to generate model inputs ($x_t = [s_{t-31}, s_{t-30}, \dots, s_{t-1}, s_t]$).

Two different types of signal data augmentation are used for unique purposes and generating MFSs. First, signal jittering is used to generate strong positive pairs for contrastively training our learning model. Given an input signal sequence x_t , a strong pair $[x'_t, x''_t]$ is generated as follows:

$$x'_t = [s'_{t-31}, s'_{t-30}, \dots, s'_{t-1}, s'_t], \text{ where } s'_i = s_i + z'_i, z'_i \sim N(0, 0.1),$$

$$x''_t = [s''_{t-31}, s''_{t-30}, \dots, s''_{t-1}, s''_t], \text{ where } s''_i = s_i + z''_i, z''_i \sim N(0, 0.1).$$

Generated from the same input signal, the strong pair $[x'_t, x''_t]$ will be regarded as positive pairs having positive strength of size 1. When

generating a strong positive pair with data augmentation, we need to carefully choose which data augmentations are used. We further demonstrate the experimental results on task transfer accuracy for different types of data temporal signal augmentation methods (Extended Data Fig. 7).

Although out-of-distribution issues are mitigated with data normalization that bounds input signal domains, we further use data-shifting augmentations to mitigate signal differences between different users and printed sensors. Given an input signal sequence x_t , another input signal sequence x_t is generated as follows.

$$x_t = \begin{cases} x_t + z, z \approx U[0.3, 0.5] & \text{for } \max(x_t) < 0.5 \\ x_t - z, z \approx U[0.3, 0.5] & \text{for } \min(x_t) \geq 0.5 \end{cases}$$

Unlike signals generated by jittering, which were regarded as positive pairs representing the same hand motions, we regard shifted signals as a completely new input. Since the amplitude of a sensor signal is correlated with the amount of sensor deformation, shifting signals in the y axis would result in new signal patterns representing different hand motions. At the same time, the model can learn how to embed sensor signals positioned in various input domains. This is the conventional way of data augmentation that is used to increase the amount of training datasets by providing more training examples.

Details of attention mechanism in a transformer encoding block

Given a signal input sequence with a size of 32, $x_t \in [0, 1]$, the model first embeds the sensor signals into high-dimensional vectors $x_{\text{enc}_t} \in R^{32 \times 32}$ with a sensor embedding block, f_{enc} , consisting of a single linear layer:

$$x_{\text{enc}_t} = [v_{t-31}, v_{t-30}, \dots, v_{t-1}, v_t], v_i = f_{\text{enc}}(s_i) \in \mathbb{R}^{32}, \forall i \in [t-31, t].$$

Before encoding the signals into an MFS with attention mechanism, we add positional embedding to the embedded vectors so that the model can understand the relative position of input sequences and encoding them in parallel. Positional embedding is one of the key features of transformer architecture⁴¹, which allows the model to avoid iterative computation for each time frame. The position-encoded input vector $x_{\text{enc_pos}_t}$ is generated as follows:

$$\begin{aligned} x_{\text{enc_pos}_t} &= x_{\text{enc}_t} + \text{pos} = [v_{t-31}^{\text{pos}}, v_{t-30}^{\text{pos}}, \dots, v_{t-1}^{\text{pos}}, v_t^{\text{pos}}] \\ &= [v_{t-31}, v_{t-30}, \dots, v_{t-1}, v_t] + [\text{pos}_1, \text{pos}_2, \dots, \text{pos}_{\text{window}_{\text{size}}}], \\ \text{pos}_i &:= \begin{cases} \sin(\omega_k, t), \text{ if } i = 2k \\ \cos(\omega_k, t), \text{ if } i = 2k + 1 \end{cases}, \omega_k = \frac{1}{10,000^{2k/32}} \end{aligned} .$$

This allows unique positional vectors to be added to different positions in a time window (Extended Data Fig. 6b).

Entire signal windows are encoded into latent vectors by using the transformer encoding layers that utilize multihead attention blocks. Given an embedded vector $x_{\text{enc_pos}_t} \in R^{32 \times 32}$ consisting of 32 vectors representing sensor signals for each time frame, the model first encodes the vector into three representative vectors called query, key, and value_t. For each query, the model compares its values with other keys to generate attention weights, and these weights are multiplied by their values to generate embedded vectors that have referred to entire time-window signals. This can be computed in parallel by matrix multiplication, which massively increases the model's encoding speed for sequential signals.

$$\text{query}_{t_j} = W_{qj}(v_i^{\text{pos}}), \text{key}_{t_j} = W_{kj}(v_i^{\text{pos}}), \text{value}_{t_j} = W_{vj}(v_i^{\text{pos}}), \\ i \in [t-31, t], j \in [0, 3]$$

where $W \in R^{32 \times (n_{\text{head}} \times 32)}$ indicates linear layers that project the embedded vector v_i^{pos} into multiple triplet heads comprising query, key and value. We note that instead of generating a single triplet (query, key

and value) for each embedded vector v_i^{pos} , we generate multiple (query, key and value) triplets that are called heads. Utilizing the ability to easily compute attention in parallel, the model is designed to simultaneously generate multiple attentions. In this work, we generated four heads in parallel, which is half the number of heads compared with the original language model for fast real-time computation.

Given the query and key value heads, namely, query _{t_{ij}} and key _{t_{kj}} , respectively, the model first computes attention as follows:

$$\text{attn}_{ikj} = \text{softmax} \left(\frac{\text{query}_{t_{ij}} \times \text{key}_{t_{kj}}^T}{\sqrt{32}} \right), i \in [t - 31, t], j \in [0, 3].$$

The generated attention vectors contain weights that determine the amount of information that the model gathers from different values. Therefore, for each time-frame vector, the model generates output vectors as

$$o_{t_{ikj}} = \text{attn}_{ikj} \times \text{value}_{t_{kj}}.$$

Therefore, the output vectors for the j th head are computed as

$$o_t = [o_{t_i}, \forall i \in [t - 31, t]],$$

where

$$o_{t_i} = \text{concat} \left(\sum_{k=t-31}^t o_{t_{ikj}}, \forall j \in [0, 3] \right).$$

We repeat the above multihead attention mechanism three times. Through stacked attention blocks, the model can encode temporal signal patterns by learning how to extract useful information from sequential signal inputs without iteratively processing every time frame.

After encoding temporal signal patterns, we further project output vectors with a position-wise feedforward layer. The projected vectors are concatenated to generate a latent vector representing the entire signal sequence as below:

$$q_t = \text{concat} (W_i o_{t_i} + o_{t_i}, \forall i \in [t - 31, t]).$$

This is a linear block applied to each time-frame output with a residual connection. Using integrated representation q_t , the model generates motion-feature vector and phase variables as below:

$$\begin{aligned} z_t &= f_z(q_t), z_t \in \mathbb{R}^{16}, \\ \emptyset_t &= \text{softmax} (\text{sigmoid}(f_\emptyset(q_t))), \emptyset_t \in \mathbb{R}^2, \end{aligned}$$

where f_z and f_\emptyset are two separate three-layered linear blocks with a Leaky ReLU activation function between the linear layers. At the end of the phase block, we apply the sigmoid function and softmax function so that the phase variables express the binary state of the input signal (active and inactive phases). We note that the phase block is not trained and used in the pretraining stage.

Structure and implementation of temporal augmentation contrastive feature learning

Given the latent motion feature vectors $Z = \{z_i\}$ encoded by the model, we apply timely discounted contrastive loss, which is a generalization of InfoNCE³⁸ by applying guided tolerance for mapping semantically similar signals to a closer space. For each latent motion feature vector, we have a time variable t that indicates the time that corresponding sensor signals are collected. Based on the time distance between two different latent features vectors, we subdivide them into positive pairs Z^+ and negative pairs Z^- as follows:

$$\begin{cases} (z_i, z_j) \in Z^+, \text{ if } |t_i - t_j| \leq D_w/\tau \\ (z_i, z_j) \in Z^-, \text{ if } |t_i - t_j| > D_w/\tau \end{cases},$$

where τ is a hyperparameter that determines the tolerance distance and D_w is the window size for the sliding time window.

We note that the latent motion features generated from the same motion signals through data augmentation would have zero distance since the time labels are the same. For each positive pair, we assign a time discount factor based on the distance between the two vectors as below.

$$\text{TD}_{ij} = \exp(-\alpha |t_i - t_j|^2)$$

Here α is a hyperparameter that determines the discount rate; in this work, we set it as 4. Therefore, applying the time discount factor, we can get a new loss function as below:

$$\text{TDC_Loss}(Z) =$$

$$-\mathbb{E}_{i,j, s(t_i, z_i) \in Z^+} \left[\log \frac{\text{TD}_{ij} \times \exp\left(\frac{\text{cosim}(z_i, z_j)}{\text{temp}}\right)}{\sum_{p \in Z^+} \text{TD}_{ij} \times \exp\left(\frac{\text{cosim}(p, z_j)}{\text{temp}}\right) + \sum_{p \in Z^-} \exp\left(\frac{\text{cosim}(p, z_j)}{\text{temp}}\right)} \right],$$

where temp refers to the temperature and it is set as 0.07. By giving a time discount factor and extending the boundary of positive pairs for signals correlated with their measurement time, we can avoid the model from pushing semantically similar signals apart.

Details of transfer learning with few-shot labelled set

For each task, such as keyboard typing or object recognition, we fine-tune the pretrained models with a few-shot labelled dataset given by a specific user. In transfer learning, we aim to further refine our MFS for more clearly discriminating the task gestures and simultaneously training the phase block so that the model can distinguish active and inactive phases for the current user.

Therefore, given a few-shot labelled dataset $X_{\text{train}}^{\text{user}} = \{x^k, a^k, t^k\}$, the model first encodes each signal input to MFS to form the labelled set $Z_{\text{train}}^{\text{user}} = \{z^k, \emptyset^k, a^k, t^k\}$, where z^k is the encoded latent motion feature vector of x^k and \emptyset^k is a predicted phase variable. The gesture label a^k represents one of the action gestures; in particular, 0 action would indicate the inactive phase where the user is not intending to perform any of the gestures in the task. Therefore, we additionally generate the phase label as

$$\begin{cases} a_{\emptyset}^k = 0, \text{ if } a^k = 0 \\ a_{\emptyset}^k = 1, \text{ if } a^k \neq 0 \end{cases}$$

The model is then fine-tuned with the loss function stated below:

$$\text{Loss}(Z_{\text{train}}^{\text{user}}) = \text{TDC_Loss}(\{z^k\}) + \alpha \text{NCELoss}(\{z^k, a^k\}) + \beta \text{BCELoss}(\{\emptyset^k, a_{\emptyset}^k\}).$$

where NCELoss indicates InfoNCE³⁸ where we regard signals with the same labels as positive pairs, and BCELoss is a binary cross-entropy loss. Hyperparameters α and β are assigned for controlling the ratio between different loss values.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The collected finger datasets for various daily tasks performed in this study are available via GitHub at https://github.com/meta-skin/metaskin_natelec. Further data that support the plots within this paper and other findings of this study are available from the corresponding authors upon reasonable request.

Code availability

The source code used for TD-C learning, rapid adaptation and results are available via GitHub at https://github.com/meta-skin/metaskin_natelec.

References

1. Bergquist, T. et al. Interactive object recognition using proprioceptive feedback. In *Proc. 2009 IROS Workshop: Semantic Perception for Robot Manipulation*, St. Louis, MO (2009).
2. Emmorey, K., Bosworth, R. & Kraljic, T. Visual feedback and self-monitoring of sign language. *J. Mem. Lang.* **61**, 398–411 (2009).
3. Proske, U. & Gandevia, S. C. The proprioceptive senses: their roles in signaling body shape, body position and movement, and muscle force. *Physiol. Rev.* **92**, 1651 (2012).
4. Piaget, J. & Cook, M. T. *The Origins of Intelligence in Children* (WW Norton, 1952).
5. Edin, B. B. Cutaneous afferents provide information about knee joint movements in humans. *J. Physiol.* **531**, 289–297 (2001).
6. Collins, D. F., Refshauge, K. M. & Gandevia, S. C. Sensory integration in the perception of movements at the human metacarpophalangeal joint. *J. Physiol.* **529**, 505–515 (2000).
7. Edin, B. B. & Abbs, J. H. Finger movement responses of cutaneous mechanoreceptors in the dorsal skin of the human hand. *J. Neurophysiol.* **65**, 657–670 (1991).
8. Liu, Y., Jiang, W., Bi, Y. & Wei, K. Sensorimotor knowledge from task-irrelevant feedback contributes to motor learning. *J. Neurophysiol.* **126**, 723–735 (2021).
9. Hadders-Algra, M. Early human motor development: from variation to the ability to vary and adapt. *Neurosci. Biobehav. Rev.* **90**, 411–427 (2018).
10. Altmann, G. T. & Dienes, Z. Rule learning by seven-month-old infants and neural networks. *Science* **284**, 875–875 (1999).
11. Wang, J. X. Meta-learning in natural and artificial intelligence. *Curr. Opin. Behav. Sci.* **38**, 90–95 (2021).
12. Sundaram, S. et al. Learning the signatures of the human grasp using a scalable tactile glove. *Nature* **569**, 698–702 (2019).
13. Luo, Y. et al. Learning human–environment interactions using conformal tactile textiles. *Nat. Electron.* **4**, 193–201 (2021).
14. Chun, S. et al. An artificial neural tactile sensing system. *Nat. Electron.* **4**, 429–438 (2021).
15. Caesarendra, W., Tjahjowidodo, T., Nico, Y., Wahyudati, S. & Nurhasanah, L. EMG finger movement classification based on ANFIS. *J. Phys. Conf. Ser.* **1007**, 012005 (2018).
16. Zhou, Z. et al. Sign-to-speech translation using machine-learning-assisted stretchable sensor arrays. *Nat. Electron.* **3**, 571–578 (2020).
17. Moin, A. et al. A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition. *Nat. Electron.* **4**, 54–63 (2021).
18. Kim, K. K. et al. A deep-learned skin sensor decoding the epicentral human motions. *Nat. Commun.* **11**, 2149 (2020).
19. Yan, Y. et al. Soft magnetic skin for super-resolution tactile sensing with force self-decoupling. *Sci. Robot.* **6**, eabc8801 (2021).
20. You, I. et al. Artificial multimodal receptors based on ion relaxation dynamics. *Science* **370**, 961–965 (2020).
21. Kaltenbrunner, M. et al. An ultra-lightweight design for imperceptible plastic electronics. *Nature* **499**, 458–463 (2013).
22. Tang, L., Shang, J. & Jiang, X. Multilayered electronic transfer tattoo that can enable the crease amplification effect. *Sci. Adv.* **7**, eabe3778 (2021).
23. Araromi, O. A. et al. Ultra-sensitive and resilient compliant strain gauges for soft machines. *Nature* **587**, 219–224 (2020).
24. Miyamoto, A. et al. Inflammation-free, gas-permeable, lightweight, stretchable on-skin electronics with nanomeshes. *Nat. Nanotechnol.* **12**, 907–913 (2017).
25. Lee, S. et al. Nanomesh pressure sensor for monitoring finger manipulation without sensory interference. *Science* **370**, 966–970 (2020).
26. Wang, Y. et al. A durable nanomesh on-skin strain gauge for natural skin motion monitoring with minimum mechanical constraints. *Sci. Adv.* **6**, eabb7043 (2020).
27. Hendrycks, D. & Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *Proc. Int. Conf. Learning Representations (ICLR)*, 2017).
28. Shimodaira, H. Improving predictive inference under covariate shift by weighting the log-likelihood function. *J. Stat. Plan. Inference* **90**, 227–244 (2000).
29. Choi, S. et al. Highly conductive, stretchable and biocompatible Ag–Au core–sheath nanowire composite for wearable and implantable bioelectronics. *Nat. Nanotechnol.* **13**, 1048–1056 (2018).
30. Kim, K. K. et al. Highly sensitive and stretchable multidimensional strain sensor with prestrained anisotropic metal nanowire percolation networks. *Nano Lett.* **15**, 5240–5247 (2015).
31. Ershad, F. et al. Ultra-conformal drawn-on-skin electronics for multifunctional motion artifact-free sensing and point-of-care treatment. *Nat. Commun.* **11**, 3823 (2020).
32. Radford, A., Narasimhan, K., Salimans, T. & Sutskever, I. Improving language understanding by generative pre-training. *OpenAI Blog* (2018).
33. Radford, A. et al. Language models are unsupervised multitask learners. *OpenAI Blog* (2019).
34. Wu, Z., Xiong, Y., Yu, S. X. & Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 3733–3742 (IEEE, 2018).
35. Hjelm, R. D. et al. Learning deep representations by mutual information estimation and maximization. In *Proc. Int. Conf. Learning Representations (ICLR)* (2019).
36. Kim, D., Kim, M., Kwon, J., Park, Y.-L. & Jo, S. Semi-supervised gait generation with two microfluidic soft sensors. *IEEE Robot. Autom. Lett.* **4**, 2501–2507 (2019).
37. He, K., Fan, H., Wu, Y., Xie, S. & Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 9729–9738 (IEEE, 2020).
38. Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. A simple framework for contrastive learning of visual representations. In *Proc. 37th International Conference on Machine Learning* 1597–1607 (PMLR, 2020).
39. Spelke, E. S., Katz, G., Purcell, S. E., Ehrlich, S. M. & Breinlinger, K. Early knowledge of object motion: continuity and inertia. *Cognition* **51**, 131–176 (1994).
40. Iwana, B. K. & Uchida, S. Time series data augmentation for neural networks by time warping with a discriminative teacher. In *2020 25th International Conference on Pattern Recognition (ICPR)* 3558–3565 (IEEE, 2020).
41. Vaswani, A. et al. Attention is all you need. In *Advances in Neural Information Processing Systems* 5998–6008 (Curran Associates, 2017).
42. McInnes, L., Healy, J., Saul, N. & Großberger, L. UMAP: uniform manifold approximation and projection. *J. Open Source Softw.* **3**, 861 (2018).
43. Mahmood, M. et al. Fully portable and wireless universal brain-machine interfaces enabled by flexible scalp electronics and deep learning algorithm. *Nat. Mach. Intell.* **1**, 412–422 (2019).
44. Kim, D., Kwon, J., Han, S., Park, Y. L. & Jo, S. Deep full-body motion network for a soft wearable motion sensing suit. *IEEE/ASME Trans. Mechatron.* **24**, 56–66 (2019).
45. Wen, F. et al. Machine learning glove using self-powered conductive superhydrophobic triboelectric textile for gesture recognition in VR/AR applications. *Adv. Sci.* **7**, 2000261 (2020).
46. Wang, M. et al. Gesture recognition using a bioinspired learning architecture that integrates visual data with somatosensory data from stretchable sensors. *Nat. Electron.* **3**, 563–570 (2020).

Acknowledgements

Part of this work was performed at the Stanford Nano Shared Facilities (SNSF), supported by the National Science Foundation under award ECCS-2026822. K.K.K. acknowledges support from the National Research Foundation of Korea (NRF) for Post-Doctoral Overseas Training (2021R1A6A3A14039127). This work is partially supported by the NRF Grants (2016R1A5A1938472 and 2021R1A2B5B03001691).

Author contributions

K.K.K., M.K., S.J., S.H.K. and Z.B. designed the study. K.K.K. and M.K. designed and performed the experiments. M.K. and K.K.K. developed the algorithms and analysed the data. B.T.N. and Y.N. conducted the experiments on the substrate's property. Jin Kim performed the biocompatibility tests. K.P. assisted in the sensor printing and setups. J.M., Jaewon Kim, S.H. and J.C. carried out the nanomesh synthesis. K.K.K., M.K., S.E.R., S.J., S.H.K., J.B.-H.T. and Z.B. wrote the paper and incorporated comments and edits from all the authors.

Competing interests

A US patent filing is in progress by Zhenan Bao and Kyun Kyu Kim.

Additional information

Extended data is available for this paper at

<https://doi.org/10.1038/s41928-022-00888-7>.

Supplementary information

The online version contains supplementary material available at <https://doi.org/10.1038/s41928-022-00888-7>.

Correspondence and requests for materials should be addressed to Sungho Jo, Seung Hwan Ko or Zhenan Bao.

Peer review information *Nature Electronics* thanks Nitish Thakor and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permissions information is available at www.nature.com/reprints.

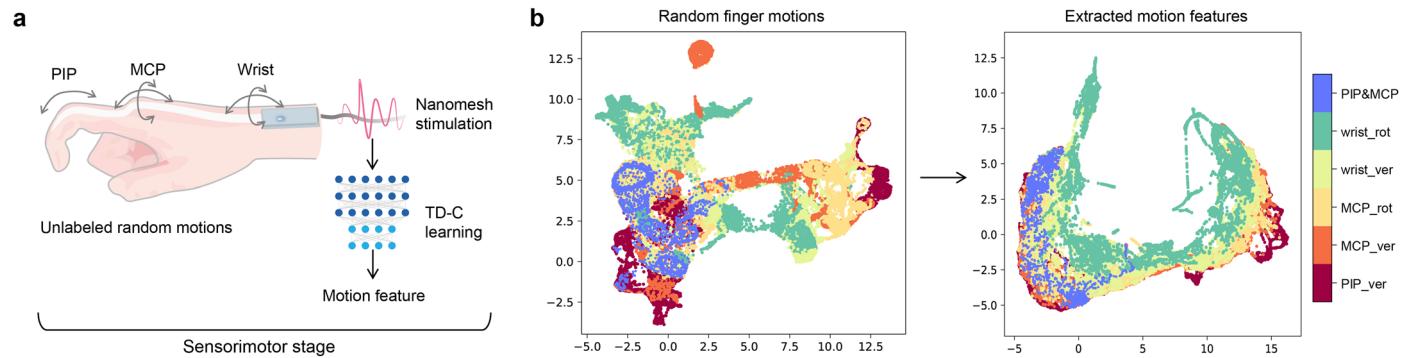
Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

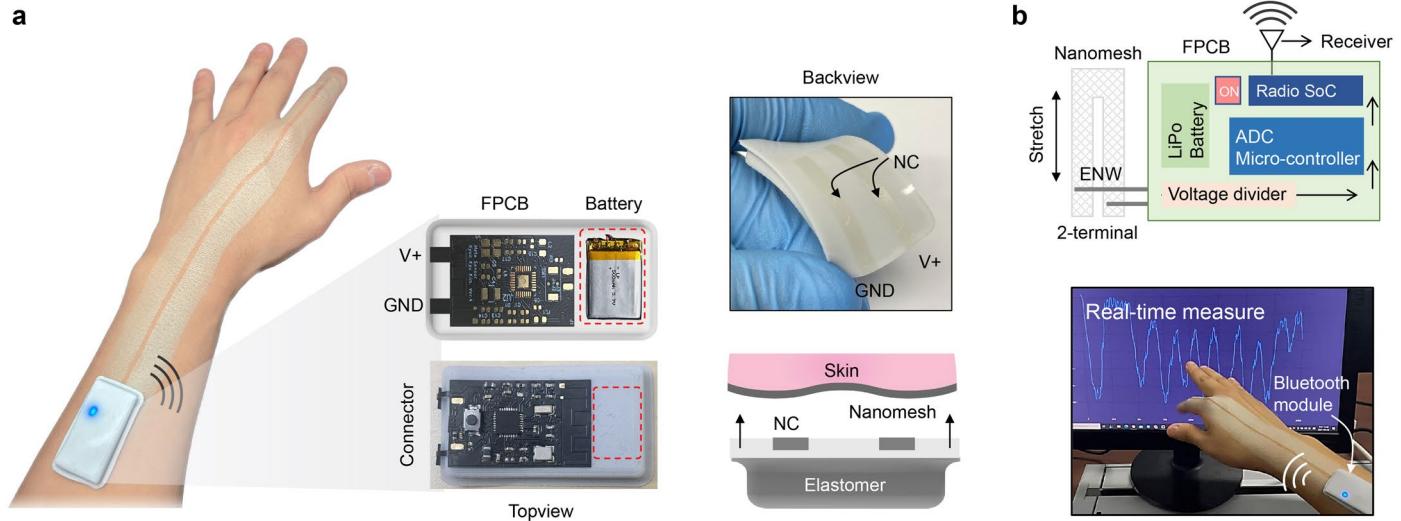
© The Author(s), under exclusive licence to Springer Nature Limited 2022

Ref.	This work	Wearable Glove [12]	Wearable Glove [16]	Wearable EMG [17]	Skin-like strain [18]	Skin-like EEG [43]	Soft strain sensor [23]	Tactile textiles [13]	Soft Strain sensor [44]	Textile Glove [45]	Vision and strain sensor [46]
Sensor structure	Direct printed substrate-less nanomesh	Glove + Conductive thread sensor	Glove + Conductive yarn	PET+ Screen printed silver ink	Polyimide+ Laser induced crack	Serpentine electrodes through micro-fabrication	Sleeve + laser-machined carbon fiber polymer composites	Glove + Conductive yarn	Wearable suit + liquid metal sensor	Glove + triboelectric textile	Hydrogel layer + carbon nanotube sensor
Sensing Target											
Function	Keyboard typing 4 Command 6 objects	26 objects	10 sign languages	13 Gestures	5 gestures	5 Command signals	3 gestures/hand opening	10 letter objects 10 poses 9 actions	Motion tracking for 3 motion flow	11 two hand gestures 3 gestures for different games	10 gesture comments
Learning Method	Unsupervised Meta learning	Supervised Learning	Supervised Learning	HD Computing	Supervised Learning	Supervised Learning	Supervised Learning	Supervised Learning	Supervised Learning	Supervised Learning	Supervised Learning
Data Samples	Unlabeled random motions (900 s) Few-shot transfer (1 ~ 5 shots)	Labeled 1,353 shots	Labeled 440 shots/class	Labeled 4 s window per gesture	Labeled 50 shots/class (train) 5 shots (transfer)	Labeled 450 shots/class	3 data collection session (a single 25 s session for testing data)	82,836 frames (Pose classification) 62,932 frames (Letter classification) 90,295 frames (Action classification)	7560 frames for each motion sequence	200 samples per class (two hand gestures) 120 samples per class (games)	2000 samples of hand image and somato-sensory pairs for training
Multi-user adaptation											
Multi-task adaptation											
No Layer modification											
Skin conformability											

Extended Data Fig. 1 | Soft sensors with intelligence. Taxonomy of augmented soft sensors combined with machine intelligence.



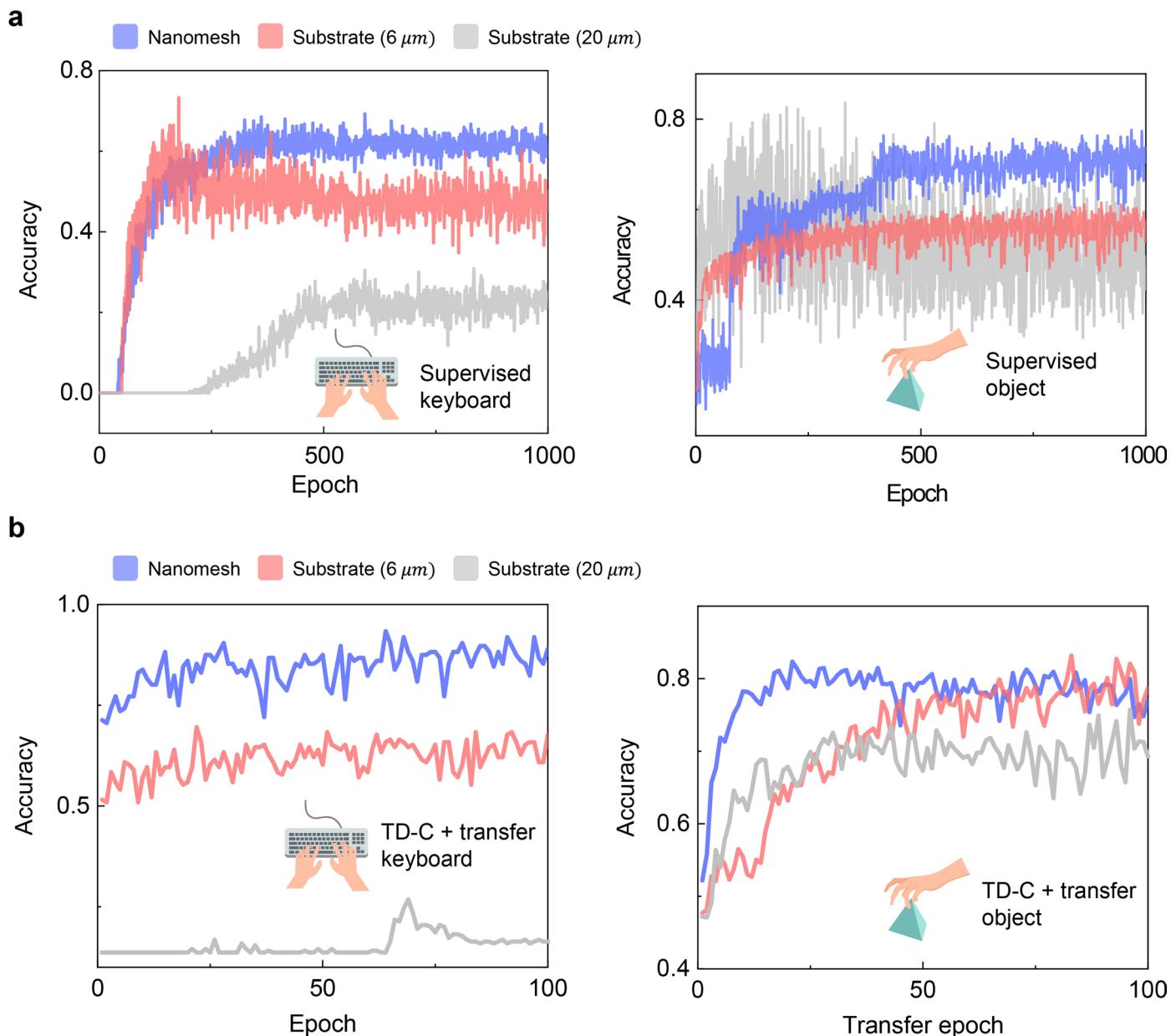
Extended Data Fig. 2 | Learning robust motion representation from unlabeled data. **a**, Schematic illustration of the wireless module that transfers multi-joint proprioceptive information. Random motions of PIP, MCP, and Wrist motions are collected. **b**, UMAP embedding of raw random finger motions and after motion extraction through TD-C learning.



Extended Data Fig. 3 | Wireless module for measuring changes of nanomesh.

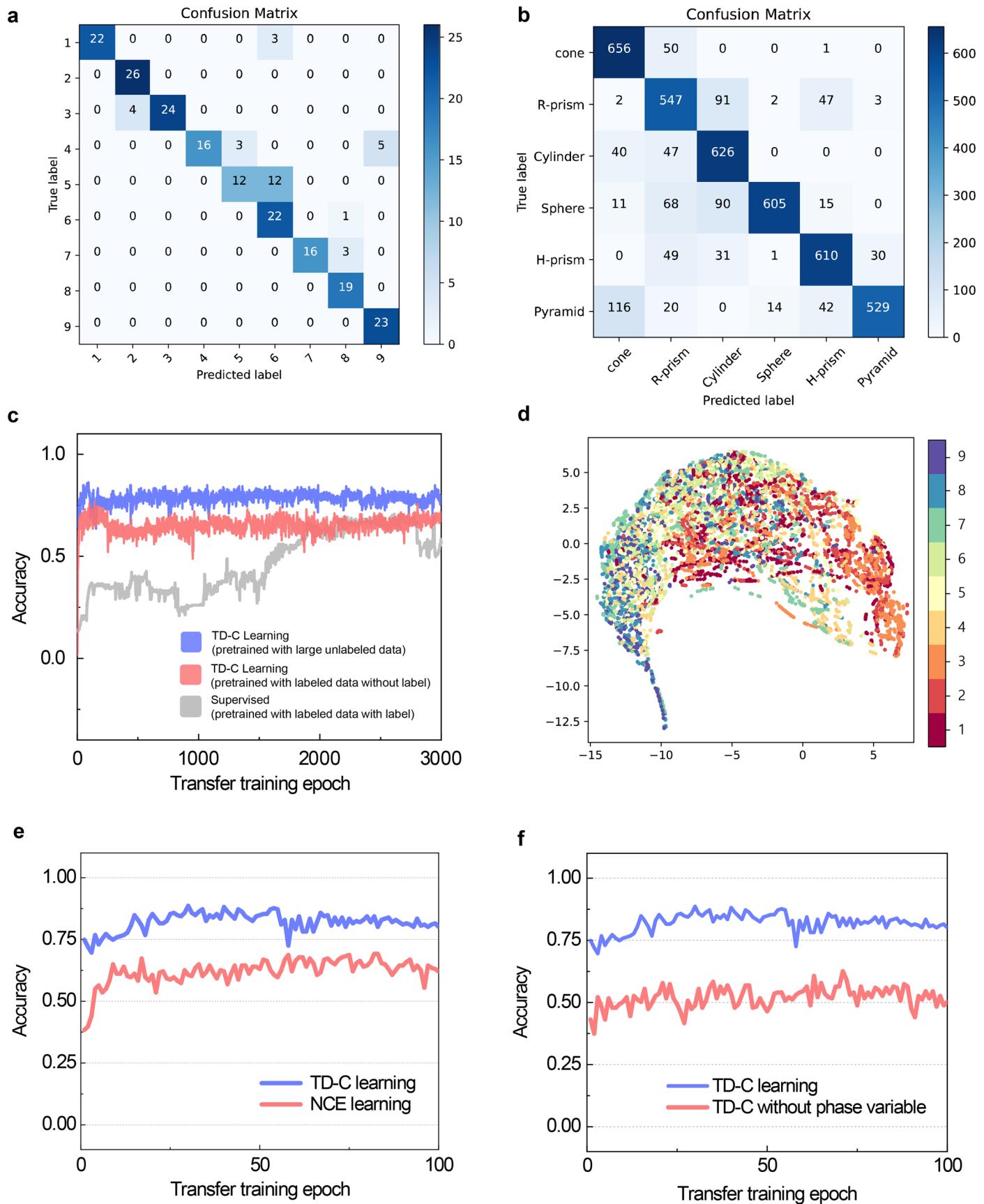
a, Schematic illustration of the wireless module that transfers proprioceptive information through simple attachment above the printed nanomesh. Illustration and image of the module is shown. Flexible printed circuit board (FPCB), lithium polymer battery, and connector is shown. Right image depicts

backview of the module. Nanomesh connector (NC) is applied, and electrical contact is made by simple attachment of the module to the printed nanomesh. **b**, Block diagram of the main components constituting the wireless module. Photograph shows real-time measurement through the module.



Extended Data Fig. 4 | Model validation accuracies and transfer learning accuracies for sensor signal with and without substrate. To investigate how substrate-less property contributes to the model discriminating different subtle hand motions, the same amount of sensor signals is collected while a user typing Numpad keys and interacting with 6 different objects. **a**, Collected dataset is divided into training and validation datasets with a ratio of 8:2 for normal

supervised learning. **b**, For transfer learning, we apply our TD-C learning with unlabeled random motion data to pretrain our learning model and use the first five-shot demonstrations to further transfer learning. Directly attached to the finger surface, nanomesh without substrate outperforms sensor with substrates in different tasks and training conditions.

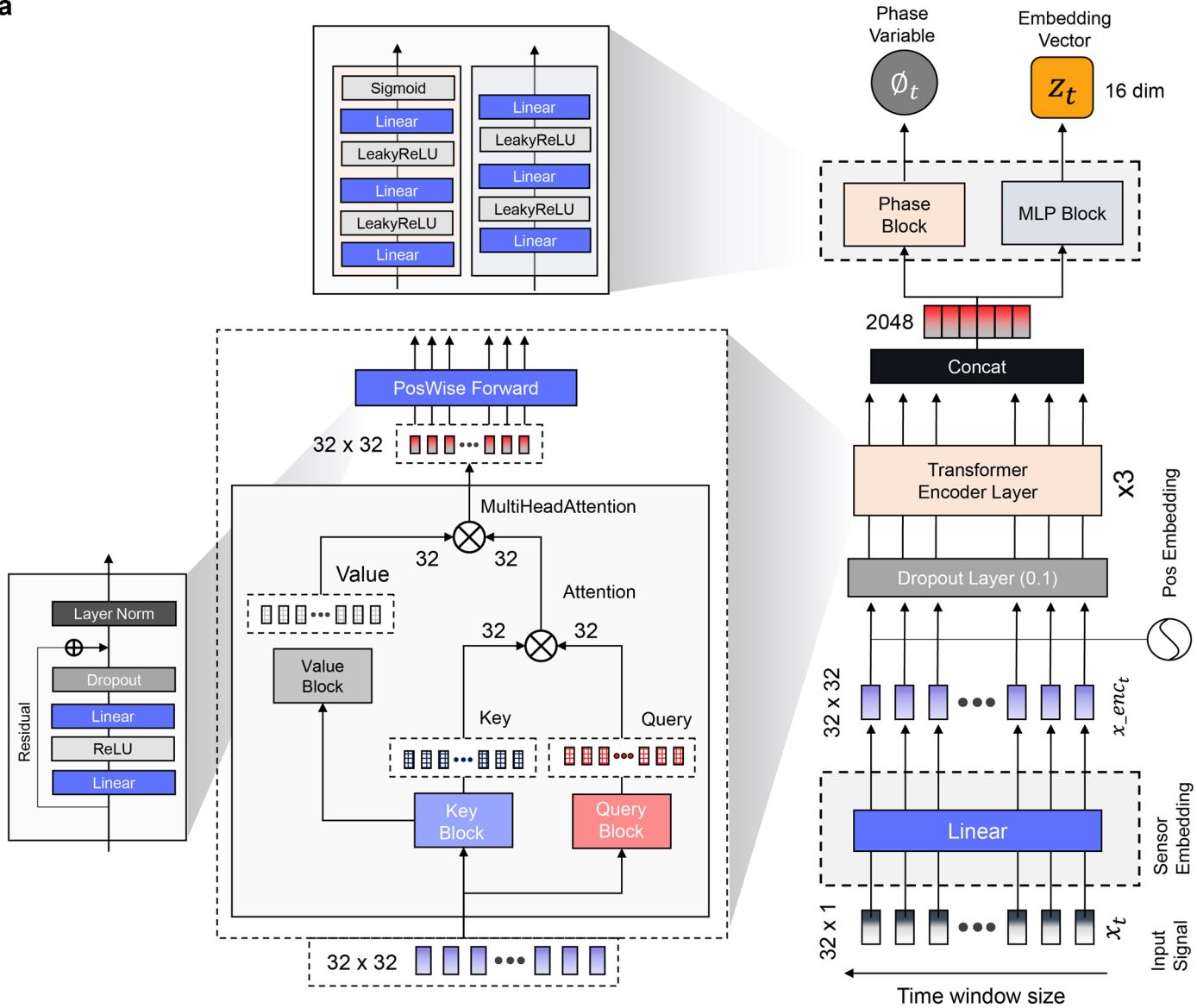


Extended Data Fig. 5 | See next page for caption.

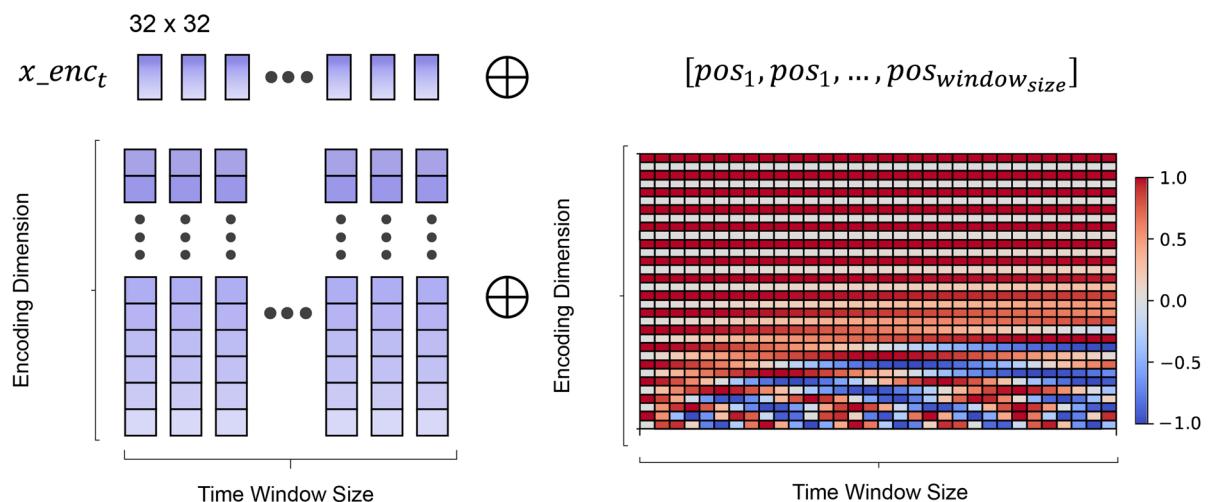
Extended Data Fig. 5 | Model performance analysis and ablation studies for components in our learning models. **a**, Confusion matrix for numpad typing data for each typing stroke after 20 transfer training. **b**, Confusion matrix for object recognition tasks for individual signal frame after 20 transfer training. **c**, More detailed comparison between TD-C Learning and supervised learning with last layer modification. For more precise comparison, we additionally trained TD-C learning model with labelled data used to train supervised model by removing labels. Even with the same number of training samples, our learning framework significantly outperform normal supervised learning when the model

is transferred to predict different tasks. With more easily collectable unlabeled training samples, TD-C learning model pretrained with large random motion data shows higher accuracies in all transfer training epoch than other models. **d**, UMAP projection of latent vectors of labelled keypad typing data projected by our model pretrained with TD-C learning method. **e**, Ablation study for transfer accuracy comparison between applying timewise dependency loss and original contrastive learning loss. **f**, Ablation study for applying phase variable by comparing transfer accuracy trends for models with and without phase discrimination when inferencing different gestures in MFS.

a



b

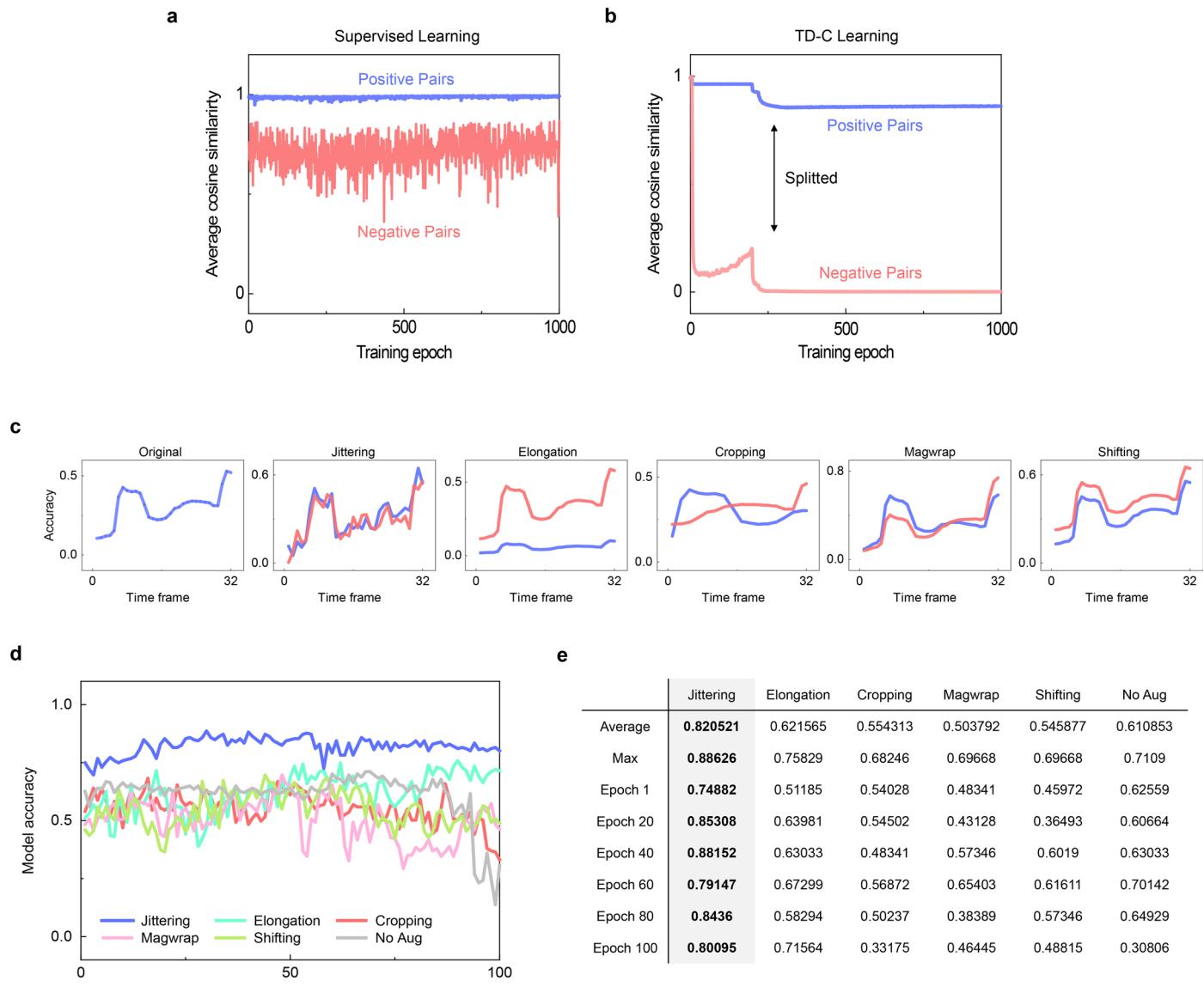


Extended Data Fig. 6 | See next page for caption.

Extended Data Fig. 6 | Details of the learning model architecture.

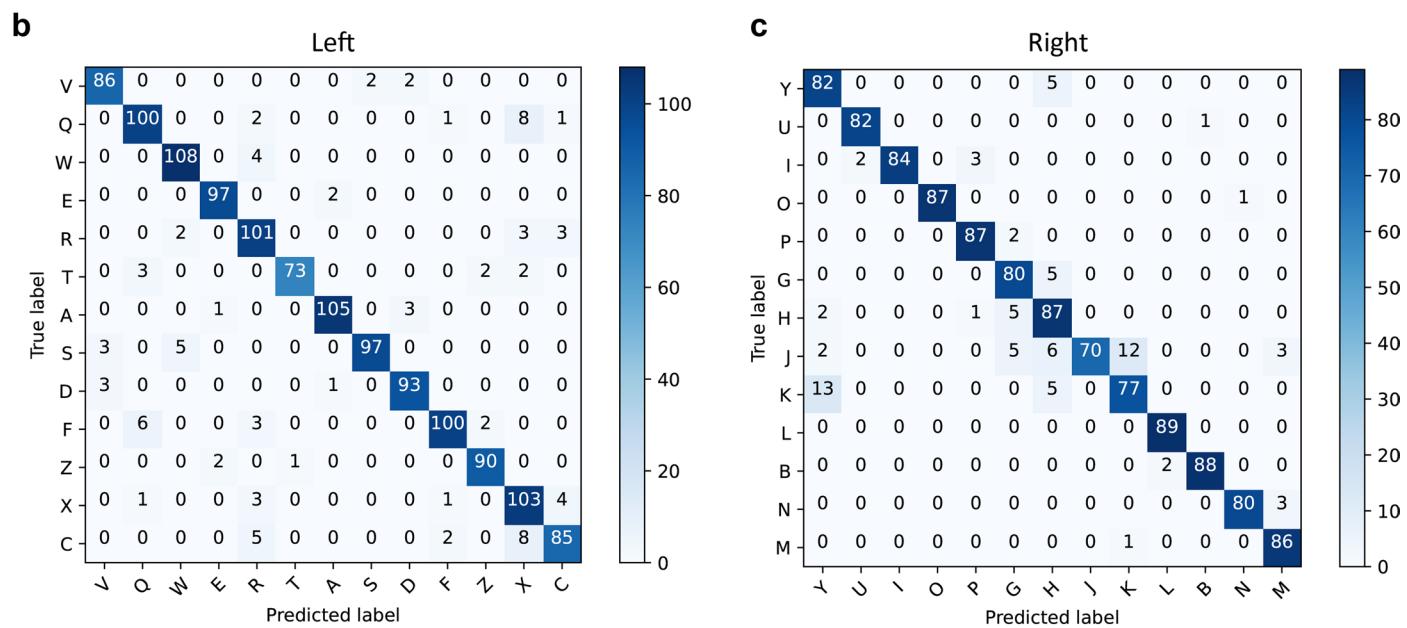
a, Illustration of detailed layer structure for signal encoding model. The temporal signal patterns are encoded through transformer encoders with the aid of attention mechanism. Following linear blocks, MLP block and phase block, utilize encoded latent vectors to generate embedding vectors in our feature space and

phase variables distinguishing active and inactive phases. **b**, Visualization of positional embedding used to advise model time-wise correlation between signal frames within a time window. Positional embedding allows the model to process temporal signal patterns in parallel using attention mechanisms enabling fast encoding of complex signal patterns for real time usages.



Extended Data Fig. 7 | Ablation Studies on different learning methods and different temporal signal data augmentations. **a.** Cosine similarity for supervised learning framework. **b.** Similarity based on TD-C learning. **c.** Examples of signal patterns before and after applying different data augmentations. **d.** Transfer accuracy comparison for learning models pretrained with different data augmentations predicting user numpad typing data. Jittering augmentation

that does not change signal amplitude or frequencies allows the model to generate more transferable feature spaces. **e.** Summary table of prediction accuracy for different data augmentations. Compared to models trained with different data augmentations, the model trained with jittering shows 20% higher accuracy in average.



Extended Data Fig. 8 | Prediction of full keyboard. **a**, Each hand taking charge for the left half and right half of the keyboard. **b**, Confusion matrix of left side of keyboard. **c**, Confusion matrix of right part of keyboard. (Five-shot demonstrations for each key for transfer training dataset, accuracy left: 93.1%, right 93.1%).

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see our [Editorial Policies](#) and the [Editorial Policy Checklist](#).

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a Confirmed

- The exact sample size (n) for each experimental group/condition, given as a discrete number and unit of measurement
- A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
- The statistical test(s) used AND whether they are one- or two-sided
Only common tests should be described solely by name; describe more complex techniques in the Methods section.
- A description of all covariates tested
- A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
- A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
- For null hypothesis testing, the test statistic (e.g. F , t , r) with confidence intervals, effect sizes, degrees of freedom and P value noted
Give P values as exact values whenever suitable.
- For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
- For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
- Estimates of effect sizes (e.g. Cohen's d , Pearson's r), indicating how they were calculated

Our web collection on [statistics for biologists](#) contains articles on many of the points above.

Software and code

Policy information about [availability of computer code](#)

Data collection	A custom built SoC (CC2650, Texas Instruments) was used for the data collection and the firmware was programmed through Code Composer Studio (CCS).
Data analysis	Data was analyzed using Python. The source code is available at https://github.com/meta-skin/metaskin_natelec

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research [guidelines for submitting code & software](#) for further information.

Data

Policy information about [availability of data](#)

All manuscripts must include a [data availability statement](#). This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

The data that support the plots within this paper and the other findings of this study is available at https://github.com/meta-skin/metaskin_natelec

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

Life sciences Behavioural & social sciences Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	Three users are asked to perform total of 10 min of data collections for each application. Each few-shot data collection lasted around 1 min (~1800 data points), with 30 sec of rest before the next trial. Data is saved to check whether the data is well collected, and the adaptation performance is evaluated. This process took less than 5 min. The sample size was chosen to fine-tune the previously-learned network and attain around 80% accuracy.
Data exclusions	No data were excluded from the analysis
Replication	Two users are asked to perform each of three different tasks and three different densities of nanomesh. All attempts at replication were successful.
Randomization	Participants were randomly chosen while the users perform identical tasks.
Blinding	Blinding was not relevant to our experiments since there were no subjective elements to effect users.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
<input checked="" type="checkbox"/>	Antibodies
<input type="checkbox"/>	Eukaryotic cell lines
<input checked="" type="checkbox"/>	Palaeontology and archaeology
<input type="checkbox"/>	Animals and other organisms
<input type="checkbox"/>	Human research participants
<input checked="" type="checkbox"/>	Clinical data
<input checked="" type="checkbox"/>	Dual use research of concern

Methods

n/a	Involved in the study
<input checked="" type="checkbox"/>	ChIP-seq
<input checked="" type="checkbox"/>	Flow cytometry
<input checked="" type="checkbox"/>	MRI-based neuroimaging

Eukaryotic cell lines

Policy information about [cell lines](#)

Cell line source(s)	Mouse fibroblast cells (L929) were purchased from Korean Cell Line Bank (KCLB), Seoul, South Korea.
Authentication	None of the cell lines used were authenticated
Mycoplasma contamination	All cell lines tested negative for mycoplasma contamination
Commonly misidentified lines (See ICLAC register)	None

Animals and other organisms

Policy information about [studies involving animals](#); [ARRIVE guidelines](#) recommended for reporting animal research

Laboratory animals	Four-week-old Hos:HR-1 male mice were used from Central Laboratory Animal Inc. (Seoul, South Korea)
Wild animals	No wild animals were used in the study
Field-collected samples	No field collected samples were used in the study

Ethics oversight

All experiments involving mice were performed with the approval of Konkuk University Institutional Animal Care and Use Committee (KU21212).

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Human research participants

Policy information about [studies involving human research participants](#)

Population characteristics

Three able-bodied male human subjects between 30 and 35 years old. The study population is a non-vulnerable highly-educated adult population.

Recruitment

A recruitment was announced by the bulletin board and made it clear that participation is voluntary and that non-participation will have no impact.

Ethics oversight

IRB at Seoul National University (Project title, Electrophysiological signal sensing by direct-printed electronic skin, IRB no., 2103/001-008). Informed consent was obtained from all participants.

Note that full information on the approval of the study protocol must also be provided in the manuscript.