AI-On-Skin: Towards Enabling Fast and Scalable On-body AI Inference for Wearable On-Skin Interfaces

ANANTA NARAYANAN BALAJI, Department of Electrical and Computer Engineering, National University of Singapore, Singapore
LI-SHIUAN PEH, Department of Computer Science, National University of Singapore, Singapore

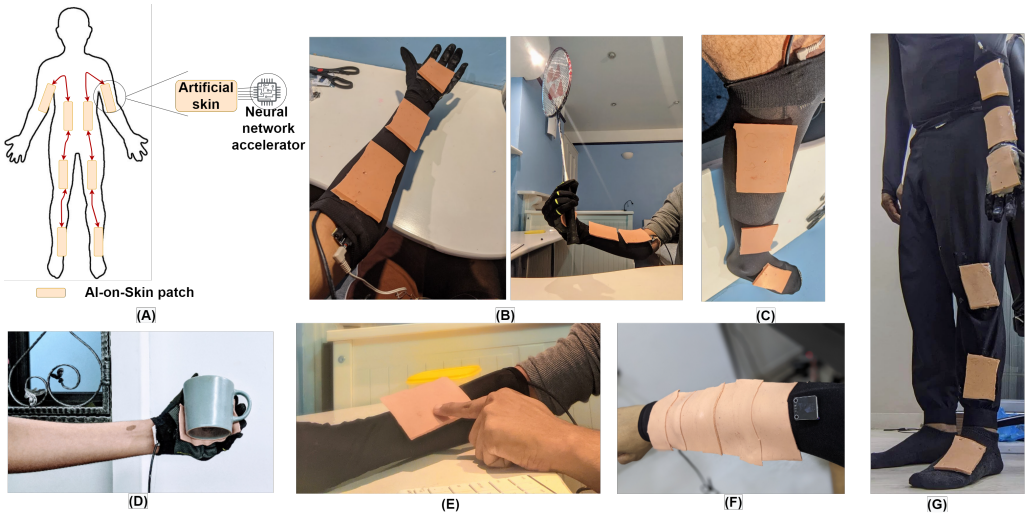

Fig. 1. (A) Overview of AI-on-skin architecture for on-body distributed AI compute (B) 3-patch hand-worn AI-on-skin prototype for badminton coaching with near-instant feedback (C) 3-patch leg-worn AI-on-skin system for real-time corrective feedback during stroke rehabilitation exercises (D) AI-on-skin powered shape sensing gloves to identify objects (E) Emotional communication with a virtual embodied agent through AI-on-skin enabled MUCA skin interface (F) A 7-patch AI-on-skin enabled handwritten word recognition system and (G) A 6 patch AI-on-skin enabled leg and arm worn real-time exercise feedback.

Existing artificial skin interfaces lack on-skin AI compute that can provide fast neural network inference for time-critical applications. In this paper, we propose AI-on-skin - a wearable artificial skin interface integrated with a neural network hardware accelerator that can be reconfigured to run diverse neural network models and applications. AI-on-skin is designed to scale to the entire body, comprising tiny, low-power, accelerators distributed across the body. We built 7 AI-on-Skin application prototypes and our user trials show AI-On-Skin achieving 20X and 50X speedup over off-body inference via bluetooth and on-body centralized microprocessor based inference approach respectively. We also project the power-performance of AI-on-skin

This research was partially funded by Singapore National Research Foundation: NRF-RSS2016-005.
Authors' addresses: Ananta Narayanan Balaji, Department of Electrical and Computer Engineering, National University of Singapore, Singapore, ananta@comp.nus.edu.sg; Li-Shiuan Peh, Department of Computer Science, National University of Singapore, Singapore, peh@nus.edu.sg.

2573-0142/2022/4-ART $15.00
https://doi.org/XXXXXXX.XXXXXXX

with our accelerator fabricated as silicon chips instead of emulated on FPGAs and show 10X further power savings. To the best of our knowledge, AI-on-Skin is the first ever wearable prototype to demonstrate skin interfaces with on-body AI inference.

CCS Concepts: • **Human-centered computing → Haptic devices**; **Ubiquitous and mobile computing systems and tools**; • **Hardware → Haptic devices**; • **Computer systems organization → Neural networks**.

Additional Key Words and Phrases: Epidermal computing, Artificial skin interfaces, Neural network accelerators, Smart-textiles

ACM Reference Format:
Ananta Narayanan Balaji and Li-Shiuan Peh. 2022. AI-On-Skin: Towards Enabling Fast and Scalable On-body AI Inference for Wearable On-Skin Interfaces. *Proc. ACM Hum.-Comput. Interact.* 1, 1 (April 2022), 34 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Skin is the most user-friendly biological interface for sensing and communicating with the outside world. In recent years, there has been increasing interest in developing skin interfaces for user-friendly interactions [1][2][3]. Skin-on interfaces mimic human skin and augment our skin surface with artificial skin to provide useful interactions like gesture recognition, haptic feedback etc.

Artificial skin sensors have been applied to health monitoring and rehabilitation. Artificial ionic skin [4] made with a low cost, biocompatible hydrogel can stick firmly to skin and potentially drive the development of skin-like smart watches that can measure various health indicators. They can also measure minute finger bending movements to assist patients with hand rehabilitation [5]. Artificial skin consisting of soft sensors and actuators [6] that conform to the wearer's skin can provide haptic feedback with pressure and vibration, monitoring rehabilitation in parents with neurological conditions. Artificial skins have recently been developed that can sense touch much faster than the human nervous system [7], enabling high-functioning prosthetic limbs for disabled individuals.

Artificial skins have also been developed for gaming and virtual reality applications. Synthetic skin composed of an array of actuators [8] can produce pressure-based tactile sensations that provide haptic feedback for VR headsets. iSkin [9] comprises thin stretchable skin stickers capable of capacitive and resistive touch sensing. It can serve as an on-body input for various end-user applications like gaming, messaging etc. SkinMorph [10] is an on-skin interface that can tune its texture to serve as a wearable skin input.

All these devices provide immense amount of real time sensor data that is typically processed with AI for inferring useful information from the electronic skin. In robotics, where artificial skins have been applied to provide robots with human-like sensing capabilities [11], neural networks have been used on sensor data from on-skin interfaces to infer human touch [12], human-like grasp [13], bodily awareness [14], motion tracking [15] etc. There have also been recent works that used neural networks to identify hand gestures. [16], assess fitness [17], and track finger motion [18] using sensory data from wearable artificial skin interfaces.

However, due to the lack of on-skin compute, current skin sensors have to rely on off-body computers such as phones and cloud servers, for compute. This causes a significant slowdown in real time response. While most existing wearable artificial skins can detect continuous touch, they cannot perform practical real-time recognition tasks that require **not just sensing, but also computing and responding**, because they lack on-skin compute. For instance, for a person with a prosthetic arm, on-skin neural network compute can enable data from the skin-on interface integrated with his/her haptic arm to respond to touch, pressure, heat etc in real-time. We see a

clear need for a fast, low-power on-skin AI compute engine (neural network accelerator) that is integrated with current artificial skins.

In this paper, we demonstrate AI-on-skin - a wearable prototype comprising artificial skin sensor patches integrated with a low power, software-reconfigurable, scalable neural network accelerator. AI-on-skin's compute engine is highly configurable (can be realized on FPGA or as a silicon chip), so it can run diverse neural networks, and support multiple applications. It scales across the entire body - thereby allowing for faster communication of neural network inferences across multiple artificial skin patches to achieve AI powered full-body artificial skin. We choose a spiking neural networks (SNN) based accelerator as SNN spikes can capture touch events from artificial skin efficiently. Compared to resource hungry deep learning models, converting a deep learning model to an SNN model leads to advantages of low latency and ultra-low-power consumption necessary for on-body deployment.

To summarize, the primary contributions of our work are as follows:

- We present AI-on-skin - the first ever wearable artificial skin based prototype integrated with an on-skin neural network accelerator for low-power, fast real-time compute and response which can empower future interaction and health sensing applications. We detail our engineering efforts in integrating on-body AI accelerators onto artificial skin sensors.
- We demonstrate the generality of AI-on-skin, programming it for 6 user applications that demand real-time response, namely (i) Badminton coaching with real-time, near-instant feedback; (ii) Leg stroke rehabilitation exercises; (iii) Real-time feedback for improving exercises involving hands and legs; (iv) Gloves that can detect objects as the user grasps them; (v) Proof of concept gloves for prosthetic arms to detect textures; (vi) Artificial skin that allows users to communicate emotions with a virtual agent via touches on the skin; (vii) Artificial skin worn on a user's arm that can correct typos and spelling mistakes as he writes. We also plan to release AI-on-Skin as an open source makers platform (see Section 8) and hope the research community can leverage the platform to prototype even more exciting interactive on-skin applications.
- Through 'live' user trials, we show how our AI-on-skin can achieve fast, accurate neural network inference for these applications, achieving upto 35X and 65X speedup over conventional off-body, BLE-based neural network inference and on-body centralized microprocessor based neural network inference respectively. AI-on-skin achieves such speedup because it does sensing, compute and response entirely on the body, thus no wireless communications off the body is needed.
- We also show AI-on-skin realized with silicon chips can obtain power savings of 11X and 15X compared to BLE based neural network inference and microprocessor based neural network inference respectively.
- We conducted a short user study with 12 participants to understand the wearability factor and initial user perception of our current AI-on-skin prototypes.
- Finally, we discuss potential use-cases in health, virtual reality/augmented reality as well as gaming applications where AI-on-skin's faster neural network inference capabilities could be used to empower end users.

2 Background and Related Works

Artificial skin sensors mimic the properties of human skin to create interesting skin-like on-body interactions. This paper proposes the integration of AI compute directly on artificial skins, focusing on the **computing** rather than the sensing aspects. It is thus orthogonal to the specific artificial skin sensor used and can interface to any artificial skin sensor [1-11]. In this section, we first summarize state-of-the-art artificial skin sensors and survey the artificial skin applications that have used AI

to process touch data from artificial skin sensors. All rely on an off-body external computer for AI inference, leading to delays in response time. We then delve into the alternative ways of supporting AI compute for skin sensing applications today, highlighting the deficiencies, and motivating the need for AI-on-skin.

2.1 Artificial skin sensors

DuoSkin [19] presents a simpler fabrication process to create customized functional stickers capable of customized touch detection. SkinWire [20] demonstrates an on-hand gestural interface composed of IMUs placed on various locations on the hand connected via skin conformable wiring to a small central microprocessor. SkinWire uses only a 27mAh rechargeable battery to detect gestures in real-time. ACES [7] is an electronic skin capable of detecting touch in 60ns - 1000 times faster than the human nervous system. ACES supports real-time sensing (60ns for touch detection) at low-power (7mW) and can scale over the entire body. SkinMorph [10] is an on-skin interface that can tune its texture to serve as a wearable skin input. SkinMorph is real-time and scalable. Multi-Touch Skin [2] is a thin, flexible sensor capable of detecting high resolution multi-touch to provide on-skin interactions. It is real-time (100 touch detections per second) and scalable across a large skin surface. SkinMarks [3] are skin-worn, highly conformable I/O devices composed of touch-sensitive electrodes. We observe that state-of-the-art artificial skin sensors can provide continuous streams of skin sensed data within 60ns to 100ms. This is just the sensing latency, not accounting for the communications and compute latency experienced by interactive applications.

2.2 Artificial skin applications that rely on off-body AI compute

2.2.1 ***Robotics*** *:* Many artificial skin sensors have targeted robotics applications, providing robots with the human sense of touch. Larson et al. [12] makes use of convolutional neural networks to identify skin deformations on a soft skin interface, thereby enabling robots to interpret human touch. Their system transfers data from 25 sensing electrodes via BLE to a laptop for inference. Subramanian et al. [13] uses convolutional neural networks on the frames obtained from a tactile glove that covers the full hand (consisting of 548 sensors) to learn the signatures of human grasp. The gloves transmit 8 continuous frames of 32X32 sensor data using a wired interface to the laptop/PC and a ResNet-18 based architecture was used to perform neural network inference. Recently, a combination of stacked convolutional autoencoders and recurrent neural networks was able to provide bodily awareness to robots [14] fusing touch data obtained from soft body bend sensors and visual input from cameras. Their system is not demonstrated in real-time and the evaluations were made using dataset collected from a sensing suit covering the robot with multiple soft sensors. The measured data is transferred to an external computer for inference through a USB interface.

2.2.2 ***Wearables*** *:* Besides robotics, there have also been recent research into wearable applications with smart artificial skins worn on humans processed by AI. Liu et al. [16] uses LSTM on the data obtained from a low-cost PDMS based soft and stretchable smart electronic skin to recognize four types of hand gestures. An Arduino Uno transfers the 5X7 touch data to a PC via USB host interface for neural network inference. An artificial skin vision (ASV) device [17] which consists of an array of optical sensors to detect biomarkers from sweat have been proposed. It uses a LSTM RNN based ASV-bodyNET on the data from four independent ASVs placed on different skin locations to predict fitness assessments. The system is not demonstrated in real-time and uses data collected from the device to perform inference evaluations. Recently, an ultrasensitive electronic skin capable of detecting minute skin deformations [18] powered with a deep neural network is demonstrated to identify five finger motions with an average accuracy of 96%. Although real-time inference latencies

were not reported, they made use of a USB interface to perform neural network inference with a PC. U-net convolutional neural networks [21] have also been proposed for detection of multiple touch points from an electrical impedance tomography based artificial skin sensor. Their system was not demonstrated in real-time and used data collected offline from the device to perform inference evaluations.

All the above applications have to transfer artificial skin sensor data to an off-body computing device for AI inference. Relying on off-body external compute devices implies that the complete cycle from sensing the touch on the artificial skin, to communicating the touch data to an external computer, to computing on this sensed data, then communicating the results back to the body and activating response on the artificial skin needs to be within the human reaction time of 150ms [22, 23] in order to realize practical, real-time interactive applications on artificial skin. State-of-the-art skin sensors have a data sensing acquisition delay (($t_{acquisition}$) ranging from 60ns [7] to 10ms[24]. However, data acquisition is only part of the total application response time. Once touch data is acquired, a pre-processing delay is incurred to compute the features/spikes of the neural network from the touch data (($t_{pre-process}$). Next, the communication delay for transferring the pre-processed touch data to the AI engine is incurred (($t_{Skin-to-AI}$), before the neural network can compute the inference ($t_{inference}$). Finally, there will be a communication delay to transfer the inference back to the artificial skin sensor interface ($t_{AI-to-Skin}$). Thus, the total delay t for real-time feedback is:

$$t = t_{acquisition} + t_{pre-process} + t_{Skin-to-AI} + t_{inference} + t_{AI-to-Skin} \quad (1)$$

Figure 2 shows a timeline of these five delays. As communicating off the body is slow and power-hungry, we believe that on-body compute is the most viable option for realizing real-time artificial skin response.

Architecture	Real-time response	Power consumption	Scalability
On-body wired compute	Limited by slower neural network inference	Higher power incurred during neural network inference computation	Hard to scale as many long wires would be needed to connect multiple artificial skin sensor patches
Off-body BLE compute	Limited by BLE communications delay	Higher power incurred during bluetooth transmission and reception	Scalability limited by the number of BLE connections
AI-on-skin	**Faster neural network inference and feedback**	**Lower power consumption**	**Highly scalable**

Table 1. Comparison of AI-on-skin against alternative on-body compute architectures

2.3 Alternative AI compute architectures for artificial skin interfaces

Here, we explore alternative ways of realizing off-body and on-body AI compute for artificial skin, and discuss their tradeoffs in real-time response, scalability, communications and power at the high level. Our experimental results in Section 4 present measurements from our prototypes of these alternative baselines and AI-on-skin.

2.3.1 ***Off-body compute via bluetooth*** Most smart skin sensors today rely on an external computer such as the phone to process the sensed data [7, 16, 17]. A wireless bluetooth chip is interfaced to each artificial skin sensor patch and the touch data from each skin sensor can be sent readily to an edge device where off-body neural network compute occurs and the inference is wirelessly sent back to the body for response. Figure 2(A) shows an off-body compute architecture in which each artificial skin sensor is equipped with bluetooth chip via which the touch data is sent to an edge device for neural network inference and the inference is wirelessly sent back. This architecture is the most common since edge devices like mobile phones, tablets or laptops are pervasive and can perform fast neural network inference. However, wireless communication is much slower relative to computation time. Bluetooth transmission and reception delays depend on

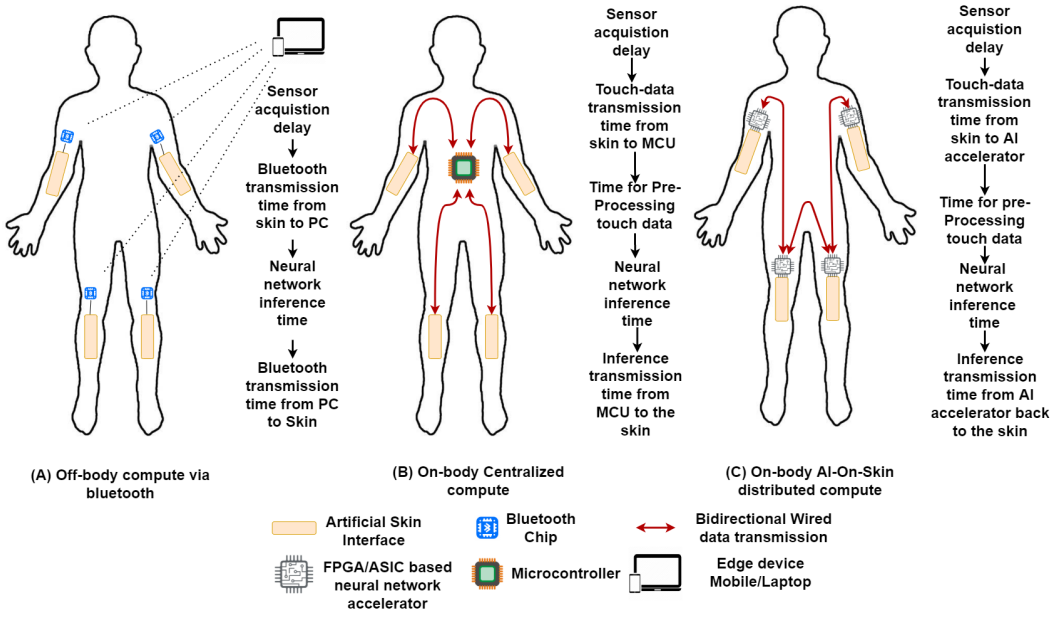

Fig. 2. Alternative AI compute architectures and their corresponding delays

the connection interval setting; The recommended low-power connection intervals for bluetooth is 100ms [25] and hence the device has to wait 100ms to receive the inference feedback after transmission, which is already close to human touch reaction time of 150ms, even before any computation. Shrinking the connection interval speeds up bluetooth, but increases power – A connection interval of 7.5ms consumes 10X higher power than a 100ms connection interval [26]. Wireless interfaces also take up substantially more power than processors; even the newest bluetooth-low-energy 5 (BLE 5) chipsets consume 8-32mW[1], which is significantly higher than the power consumption of low-power processors[2]. Besides, most edge devices support a limited number of concurrent connections: A mobile phone can only stay connected with 7 bluetooth devices at a time due to the inherent limitations in the buffering capacity [27]. This would limit the number of artificial skin sensor patches to 7, affecting system scalability. Even if the edge device is modified to handle more than 7 concurrent connections, the data transfer will still occur sequentially in BLE with the edge device being the central device obtaining data from each AI-on-skin patch in turn. In addition, BLE (both BLE 4 and 5) suffers from higher packet loss under lower connection intervals and concurrent connections [28] making them unsuitable for artificial skin applications demanding critical real-time feedback.

2.3.2 ***On-body centralized compute*** When we think of adding on-body compute capabilities to artificial skin sensors, the first thing that comes to mind is the addition of an on-body processor to perform neural network inference. Figure 2(B) shows an architecture in which multiple electronic skins connect via long wires to a centralized on-body microprocessor which performs neural network inference. Wearable on-body compute can only support a lightweight battery and so, an energy-efficient microprocessor is a more suitable choice than higher-end mobile processors. This

[1]https://www.silabs.com/wireless/bluetooth/efr32bg22-series-2-socs#
[2]https://www.arm.com/products/silicon-ip-cpu/cortex-m/cortex-m4

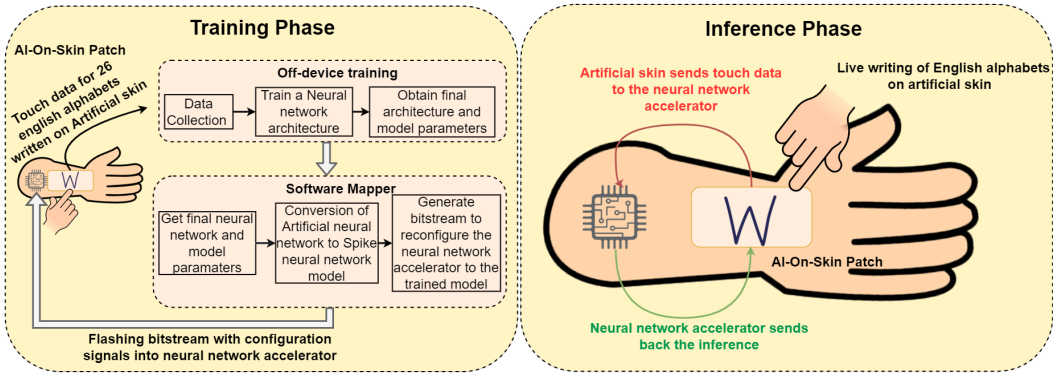

Fig. 3. AI-On-Skin workflow for a simple handwritten English alphabet recognition

architecture requires multiple I/Os on the microprocessor depending on the number of artificial skin patches to be interfaced, which would make this hard to scale to an entire body. Also, long wires from multiple artificial skins will have to interface with the central microprocessor. This will cause I/O delay as well as higher power consumption, not to mention wiring congestion scaling to many patches across the body.

2.3.3 ***On-body AI-On-Skin distributed compute*** Both the previous architectures suffer from the following disadvantages: (1) They cannot achieve real-time touch response close to human reaction time, (2) They have higher power consumption which reduces the battery life of the entire wearable artificial skin, (3) They are not particularly suited for scaling to cover the entire body. In order to tackle the aforementioned challenges in integrating on-device compute capabilities to artificial skin interfaces, we propose AI-on-skin - a distributed on-body architecture, which sprinkles ultra-low-power AI accelerator chips across the body, each attached to a skin sensor, and computing AI inference across the body. Figure 2 shows our on-body AI-on-skin architecture where each artificial skin sensor patch is connected to a local neural network accelerator chip computing neural network inference and returning the inference results to skin sensor for fast response. In this architecture, the AI compute is distributed since each artificial skin sensor patch has its own neural network accelerator and adjacent sensor patches can communicate easily via their neural network accelerator. This helps to achieve a highly scalable and distributed wearable full-body worn artificial skin with multiple artificial skin sensor patches communicating across each other. Since each artificial skin sensor patch is only connected to its nearest neighbors, this architecture requires only short wires which reduce the transmission power. We build on the prior work [29] - where the authors introduced the idea of on-body distributed compute for artificial skin sensors.

Custom-designed AI accelerator chips, termed NPUs by Apple [30], are now commonly used in mobile phones and laptops for speeding up neural network inference. For AI-on-skin, in order to achieve the ultra low power required for a wearable, we chose an open-source spiking neural network (SNN) based neural network accelerator [31], instead of the artificial neural network (ANN)-based accelerators that are more common in industry which are targeted for video and camera inputs. SNNs process discrete spikes and are well suited to process touch event data sent by artificial skin sensors. They offer very low latency, low power consumption and high temporal resolution.

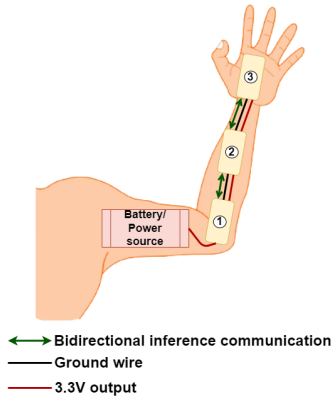

Fig. 4. AI-on-skin workflow for a real-time stroke rehabilitation application

3 Training, configuring and using AI-on-skin

We first begin with a high-level overview of how a single artificial skin sensor patch (AI-on-skin patch) can be trained to run a neural network model before extending it to our full-body worn AI-on-skin architecture consisting of multiple AI-on-skin patches. For ease of understanding, Figure 3 walks through a very simple handwritten alphabet recognition application running on AI-on-skin. There are two phases – the *Training phase* where the neural network model for handwritten English alphabet recognition is trained and AI-on-skin is configured to fit the specific neural network model (layers, neurons, synapses), and loaded with the trained weights; and the *Inference phase* when AI-on-skin performs real-time inference of the handwritten alphabets.

During the training phase, we collect multiple labelled samples (sufficient for training the neural network model) of handwritten English alphabets from the artificial skin sensor. The labelled data is used to train a neural network architecture on a PC or server using well-known deep learning libraries such as keras, PyTorch to obtain the trained model parameters. The trained model is then fed into our AI-on-skin software mapper which configures the neural network accelerator hardware to run the model. Design of our neural network accelerator is detailed in Section 4.3. Our software mapper automates the following operations: (1) It takes in the final model parameters and converts it into a spike neural network (SNN) model parameters as our neural network accelerator is based on SNNs. (2) Generates the ASIC/FPGA configuration bitstream from the SNN model parameters. The bitstream is then flashed into the neural network accelerator. AI-on-skin is now configured and ready for use.

During actual use of AI-on-skin, a user can write letters on the AI-on-skin patch, which will lead to touch data being sensed and input to the AI-on-skin accelerator. The preconfigured accelerator will now run in inference mode, computing the neural network given the trained weight and incoming touch data spikes, to infer the specific handwritten alphabets written on the artificial skin sensor. The inference never leaves the body; instead, it triggers a response to the user through the AI-on-skin patch, such as a vibration buzz to indicate a typo for this simple application.

We next illustrate how multiple AI-on-skin patches can be readily scaled up to distribute across the body. Figure 4 shows a full arm worn AI-on-skin system consisting of 3 AI-on-skin patches (labelled 1,2 and 3) for stroke rehabilitation. During each exercise made by the user, the system identifies the type of exercise and provides near-instant real-time feedback for correcting mistakes made during the exercise. Each AI-on-skin patch runs its own neural network as configured using the procedure explained earlier. AI-on-skin patch 1 will classify elbow movements and pass its inference output to AI-on-skin patch 2. Patch 2 will identify wrist movements taking in inference

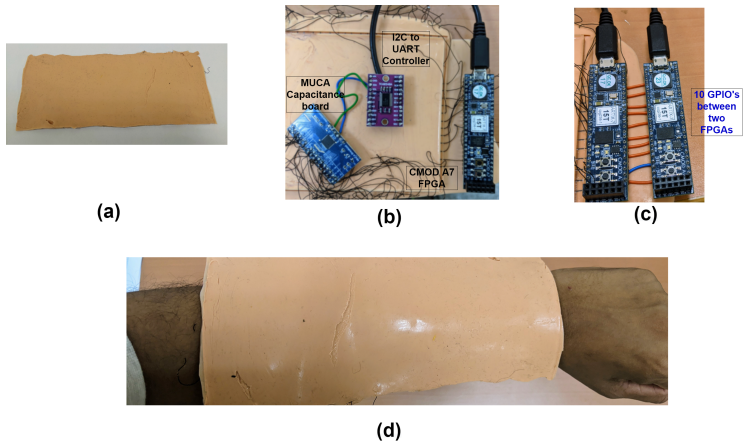

Fig. 5. Components of our AI-on-Skin prototype - (a) MUCA artificial skin interface equipped with a MUCA capacitance board (b) I2C to UART interface to send data from the MUCA capacitance board to the CMOD A7 FPGA (c) GPIO interface between two patches' FPGAs and (d) AI-on-Skin prototype worn on hand with all the hardware components housed beneath the skin

from patch 1 and sensor data from patch 2 as inputs and pass its inference output to AI-on-skin patch 3. Patch 3 will take in both inferences from patch 1 and patch 2 along with sensor data from patch 3 as input and classify the exercise type and provide real-time audio corrections like move left, move down, move diagonally right etc. A similar prototype realized with AI-on-skin is demonstrated in Section 5.5 of this paper to respond within 6.3ms.

4 AI-on-skin Prototype design

The key design challenges of designing AI-on-skin lie in first, realizing a large skin sensor that can scale to cover the entire body, and second, designing an ultra-low-power AI hardware accelerator that can be distributed across multiple patches on the body and yet readily programmed as a single, coherent AI model for the application while achieving high accuracy. In this section, we will delve into our design choices for the sensor, as well as the accelerator, and show how they achieve our design goals.

Figure 5 shows photos of our prototype where each AI-on-skin patch comprises (a) MUCA [24] artificial skin-on interface equipped with a MUCA capacitance board, (b) I2C to UART interface to send data from the MUCA capacitance board to the CMOD A7 FPGA,(c) GPIO interface between two patches' FPGAs and (d) showing how our prototype is being worn. In this prototype, we have used one of the smallest FPGA development boards, CMOD A7 [32]. Our prototype with multiple AI-on-skin patches is powered by a single battery source, with the battery connected to just the first patch, and subsequent patches daisy chained and powered by adjacent patches. This results in a lightweight, practical prototype that scales readily.

Our AI-on-skin's MUCA artificial skin interface comprises of an array of 12X21 touch points - each of the touch points are placed 8mm apart. The entire artificial skin interface is approximately 17cm X 10cm, though each patch can be readily scaled to larger dimensions. Multiple of these skin interfaces can be connected to cover the entire body.

4.1 MUCA Skin-on-interface

MUCA [24] is a maker platform that allows us to quickly produce multi-touch skin interfaces. It consists of a matrix of electrodes realized using flexible conductive wires, capable of sensing mutual

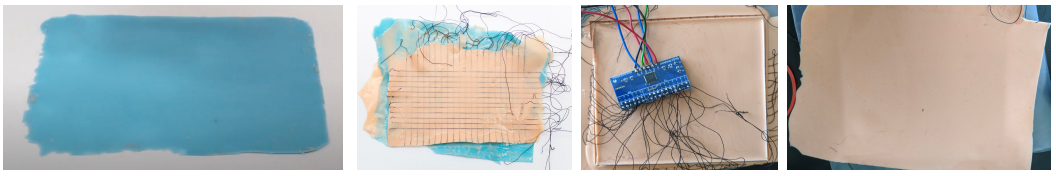

Fig. 6. Steps illustrating how we fabricated the MUCA electronic skin following [24]

capacitance. Each electrode in the sensing matrix corresponds to a touch point. The steps involved in fabricating the MUCA artificial skin interface (see figure 6) is explained below.

(1) First, the epidermis layer of skin is replicated by pouring DragonSkin silicone mixed with skin color pigments on a smooth textured mold. The thickness of this layer is around 0.6-0.7mm.
(2) Second, the epidermis layer is placed upside down on a cardboard. The carboard is provided with guide holes spaced 8mm on the all the sides. Using the guide holes, data stretch conductive threads are sewed in a matrix layout with each electrode separated by 8mm. Once the electrode layout is secured, another layer of silicone is poured to secure the electrodes in place. The total interface is of thickness close to 1.5mm. Our MUCA skin interface consists of 12X21 electrodes each spaced by 8mm and the dimensions is approximately 17x10cm.
(3) Third, a rectangular mould is made and placed on top of the electrode layer. We pour silicone ecoflex gel to achieve the expected human hypodermis skin thickness. In our prototype, this thickness is approximately 14mm. The electrodes are soldered to MUCA capacitance breakout board [24] which receives the capacitance information from electrodes and can be interfaced to any microcontroller or processing board via I2C.
(4) Lastly, the shape of the skin interface can be improved by trimming extra silicone on the sides or folding them to the sides and gluing them with silicone glue.

MUCA sends data via I2C at a frequency of 100KHz. Using a standard low-power I/O interface like I2C enables ready integration of AI-on-skin's accelerator at low power, a critical step towards ensuring that the entire AI-on-skin system is power efficient, from sensor acquisition, to communications, compute and response. AI-on-skin's current prototype only has a sensor array of 12X21 sensors as this is the upper limit supported by MUCA capacitance boards.

4.2 Sensor to AI accelerator interface

Figure 5(b) shows the MUCA capacitance breakout board connected to the CMOD A7 FPGA via I2C to UART interface (I2C) interface. The FPGA does the following pre-processing steps on the data obtained on the MUCA skin interface. It first receives the capacitance reading of dimension 12X21 - corresponding to the electrode layout of MUCA skin interface. Thresholding is next performed on the capacitance readings to remove background noise. Thereafter, a 12X21 matrix which indicates which specific electrodes were touched is generated. For example, if the electrode at location (1,2) is touched, the corresponding value at location (1,2) in the matrix is set to 1. Finally, the 12X21 matrix is serialized into a 252 length 1D buffer and fed to the accelerator, which performs the neural network inference, as will next be explained.

4.3 AI accelerator: Shenjing spike neural network accelerator

With AI-on-skin being an on-body AI accelerator, real-time response and ultra-low-power are the key design goals, since AI computation needs to keep up with interactive sensing and provide real-time response, and batteries take up substantial weight and bulk. In recent years, there have

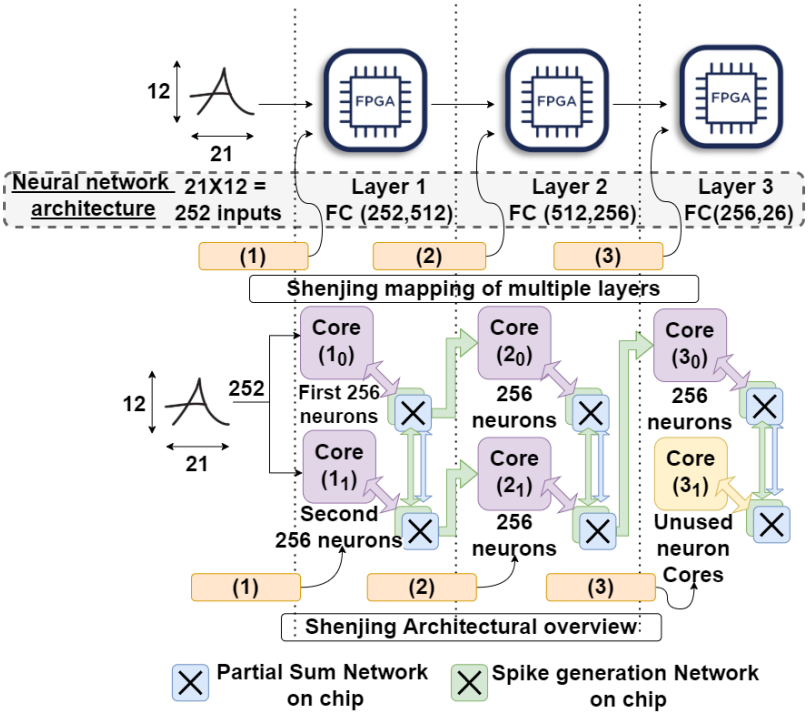

Fig. 7. Shenjing architecture for MLP with 12x21 inputs

been numerous AI hardware deep neural network accelerators released in the market, targeting mobile edge devices, ranging from GPUs [30], TPUs [33], and NPUs [34].

Our AI accelerator is based on spiking neural networks (SNNs). SNNs encode the input data into a spike train whereas a normal artificial neural network (ANN) would just process input data in its original form. In an SNN, the data that passes through the neural network layers is binary and the weighted sum calculation can be replaced with only additions instead of multiplications. This makes SNNs more energy efficient than ANNs.

We chose Shenjing [35], an open-source spiking neural network (SNN) based hardware accelerator, for AI-on-skin. Unlike other SNN accelerators, Shenjing supports loss-less weight summation across neuron cores, so a large neural network model can be simply mapped across multiple cores without loss of accuracy. Shenjing operates based on the leaky integrate and fire neuron model. This allows AI-on-skin's FPGA prototype to realize applications, across multiple skin patches, at high accuracy. The open source release of Shenjing's RTL also enables us to prototype it readily on FPGA, and fabricate it on silicon.

4.3.1 ***Walkthrough of character recognition application mapped onto Shenjing AI accelerator*** - We will use the same simplistic example of English alphabet recognition shown in Figure 3 for the walkthrough of how we use the Shenjing accelerator. The neural network model is trained offline on a PC. Our software mapper takes the model as input and generates the bitstream for configuring the Shenjing neural network accelerator. Figure 7 shows the high level architectural overview of Shenjing neural network accelerator along with the corresponding neural network models for English alphabet recognition.

The Shenjing architecture comprises multiple identical cores, that can be housed across multiple chips. Each chip consists of a fixed number of neuron cores, depending on the resources available

(transistor count on FPGA/ASIC). The software mapper maps any neural network model onto the minimum number of cores required, depending on the number of layers, neurons, synapses of the neural network model, and the number of cores per chip. In our implementation, we use a FPGA to emulate a chip since FPGAs are faster to prototype[3]. Since we use CMOD A7 FPGA in our implementation, we can only fit 2 neuron cores due to FPGA resource constraints. Each neuron core consists of 256 neurons and hence can take in 256 inputs and produce 256 output spikes. In Figure 7, three AI-on-skin patches are connected to three FPGAs. We first begin by explaining with a single AI-on-skin patch (which is labelled (1) in Figure 7). The Shenjing AI accelerator of AI-on-skin patch(1) requires 3 FPGA chips to emulate the neural network model .The first, second and third layers in the neural network model are mapped to FPGAs 1, 2 and 3 respectively as each layer requires 2 cores each.

Next, for multiple AI-on-skin patches, we walk through the following two cases:

(1) ***AI-on-skin patches (1) and (2) share the same neural network model*** - When patches (1) and (2) use the same neural network model for learning, then the touch data from patch (2) connected to FPGA 2 will be re-routed to FPGA 1 and the data flow will follow the same steps as explained earlier for skin patch (1).
(2) ***AI-on-skin patches (2) and (3) run different neural network models*** - When patch (3) uses a different neural network model, then the neural network layer mapping can start from the unused core 3_1 in FPGA 3 and then the remaining layers of the network can be implemented on another FPGA chip. This flexible mapping of neural network models across multiple skin patches maximizes the use of the limited on-body compute resources, enabling more sophisticated AI applications to be realized on AI-on-skin, without being limited by each skin patch's accelerator's resources.

4.3.2 ***Shenjing Software Mapping tool*** The software mapper for Shenjing performs the following steps :

(1) The final trained ANN model is taken as input and converted to spiking neural network (SNN) model along with constraint on the number of cores available per FPGA.
(2) Each layer is mapped onto logical cores in the FPGA.
(3) It schedules output spikes from the layer to pass to cores mapping the next layer. Based on the number of cores estimated, the number of FPGAs required will also be calculated.
(4) It then performs the physical mapping where the logical cores are mapped to cores available in FPGA chips and the routing between layers is scheduled.
(5) It generates the configuration bitstream that sets the FPGA/ASIC Shenjing chip to match the software specified SNN model.

The Shenjing software mapper can map fully connected layers, convolutional layers, and residual network layers.

4.4 Emulating AI-on-skin using FPGAs

As explained earlier, we use FPGAs to emulate Shenjing neural network accelerator as they are faster to prototype. The specific FPGA used is the CMOD A7 board which consists of a Artix 7 FPGA co-located with a 512 KB SRAM. It is 2.75in × 0.7in, and thus feasible for wearable skin interfaces.

We encounter two major challenges when implementing the open source Shenjing's hardware design (in Verilog) onto FPGA – FPGAs have limited computational resources and I/Os. Each FPGA can only accommodate a small number of cores depending on its computational resources. CMOD

[3]We have fabricated a 12-core Shenjing chip and will incorporate that into an AI-on-skin prototype in the future.

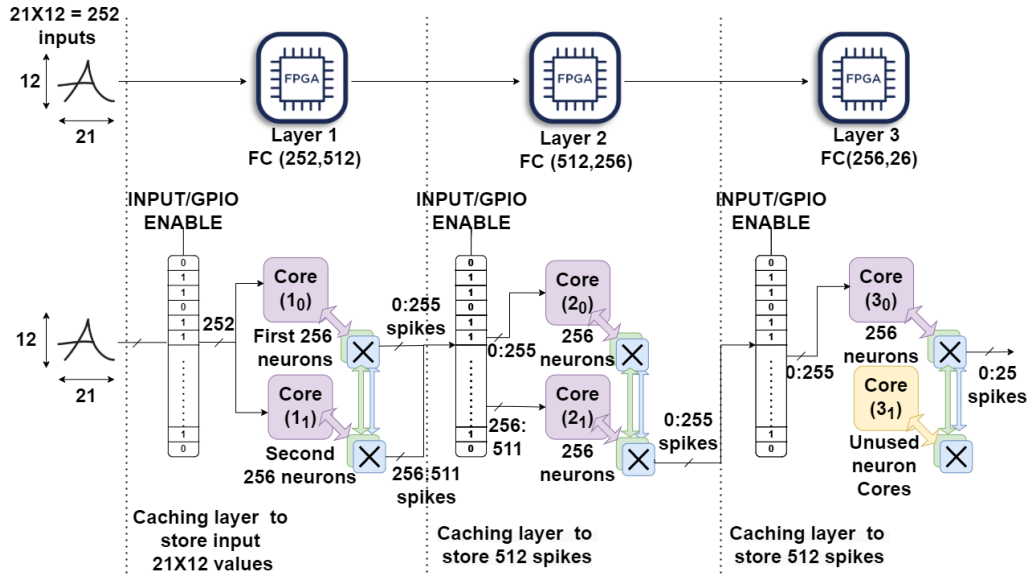

Fig. 8. A MLP inferring hand-written english alphabet recognition (from figure 7) with an artificial skin interface of 21x12 sensors is mapped to 3 FPGAs: In the first FPGA, the input serializing layer is used since it is attached directly to the artficial skin interface. In the subsequent FPGAs the spike serializing layer is enabled. Also, complete layers are implemented on a single FPGA itself - FC0 on FPGA 1, FC1 on FPGA 2 and FC3 on FPGA 3.

A7 FPGA has 225KB BRAM, 41600 flipflops and 20800 LUTs [32]. Since each neuron core stores 256X256 5-bit weight parameters, the resources suffice to only fit 2 neuron cores. The following design changes were made to overcome these challenges:

(1) *Restricting a layer to be implemented on a single FPGA* - Our AI-on-skin software mapping tool maps a layer to cores on the same FPGA only. A layer cannot be implemented across multiple FPGAs since it will incur heavy intra-layer communication cost . We only send 8 bit output spikes between FPGAs.
(2) *Serializing communications* - FPGAs have limited I/Os for communication. In our AI-on-skin prototype, we only use 10 digital I/O pins - in order to send 512 1-bit spikes from one FPGA to another, we have to serialize and send it over 51 cycles. GPIOs typically toggle at 70KHz [36]. So, we have to send 512 spikes for 512 spikes/10 IOs = 52 clock cycles = 0.1 μs*52= 5.2 μs. In the receiving FPGA, We introduce a serializing buffer to hold all the 512 spikes received from the previous FPGA. Inference proceeds only when all 512 spikes are received.

Figure 8 shows a detailed implementation of a MLP neural network on our accelerator. Each FPGA consists of an input serializing layer as well as a spike serializing layer which is configurable via software using the INPUT/GPIO ENABLE signal. Our AI-on-skin patch consists of a MUCA artificial skin interface comprising 12X21 electrodes = 252 electrodes/inputs capable of capacitive touch sensing. We receive this live touch data and cache it in an input serializing layer of size 252 bits. Each location in the input serializing layer corresponds to the particular electrode location on the MUCA artificial skin interface. The spike serializing layer used in our AI-on-skin prototype is 512 bits wide, since each FPGA can only send a maximum of 256 spikes from each core × 2 cores per CMOD A7 FPGA = 512 spikes.

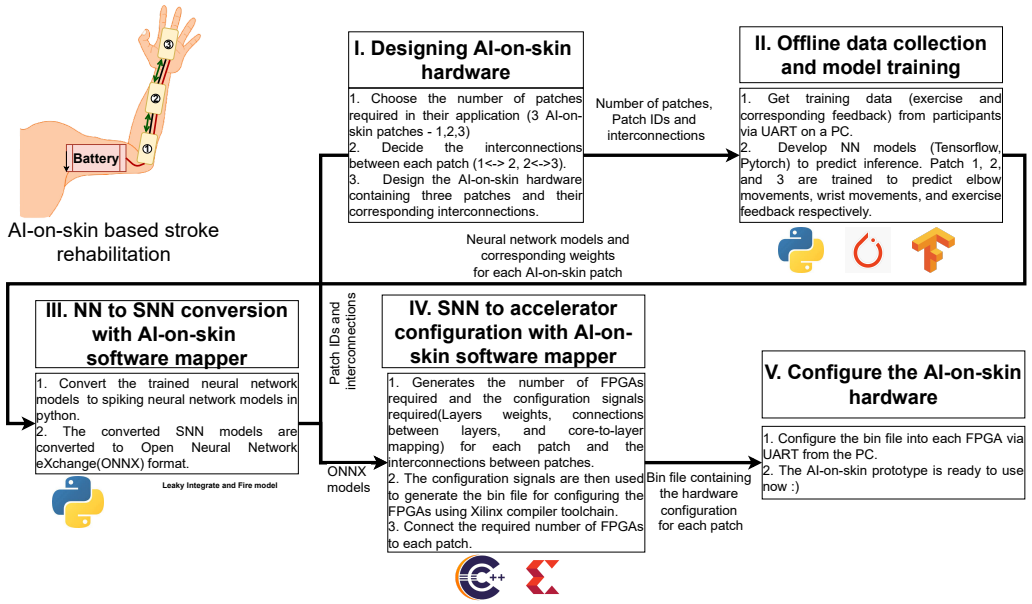

Fig. 9. AI-on-skin's software flow for configuring a three patch AI-on-skin prototype for stroke rehabilitation.

4.5 AI-on-Skin's software flow

Here, we summarize the steps involved in programming and configuring a multi-patch distributed AI-on-skin application using our AI-on-skin software mapper.
(I) The developers decide the number of AI-on-skin patches to be used for their application and how they are interconnected, assigning a unique ID to each skin patch.
(II) During training, data is streamed via UART to a PC for offline training of the models. The developers design the corresponding neural network models and train them in either TensorFlow or PyTorch. Currently, AI-on-skin supports fully connected layers, convolution (with filter sizes of 3X3,5X5 and maximum number of filters = 64), pooling (max and average pooling) and residual layers. We support ReLU activations in our current AI-on-skin implementation.
(III) The trained models, the number of AI-on-skin patches and the adjacent connection information are then passed as input to AI-on-skin software mapper's NN to SNN conversion module written in python which converts each trained model into Open Neural Network Exchange (ONNX) representation. The ONNX representations are then further converted into spiking neural network representations based on the leaky integrate and fire neuron model. Our current AI-on-skin software mapper supports both TensorFlow and PyTorch models since they are the most widely used libraries for neural networks.
(IV) Our C++ based SNN to accelerator configuration module in AI-on-skin software mapper takes in the converted ONNX representation of the SNN model and generates the number of FPGAs required and the configuration signals for CMOD A7 FPGA such as layer-to-core mappings, weights of each layer, partial sum configurations, and inter-layer connections.
(V) The generated configuration signals are fed into the Xilinx compiler toolchain to set configuration parameters in the AI-on-skin accelerator which is in SystemVerilog. The resulting bin file generated using the Xilinx toolchain is flashed onto the FPGAs via UART through a PC.

Figure 9 shows the software flow required to configure a 3-patch AI-on-skin prototype for exercise feedback in real-time stroke rehabilitation.

5 AI-on-skin application prototypes and evaluation

5.1 Experimental methodology

In this section, we demonstrate seven application use cases of AI-on-skin. For each application, we compare the latency and power measurements of our AI-on-skin prototypes against six alternative compute architectures. The same MUCA skin sensor (12x21 sensor array) feeds AI-on-skin and the six baselines, sampled at 100KHz.

1. Off-body bluetooth compute baseline - For the off-body BLE compute baseline, we interfaced our MUCA artificial skin interface to a bluno BLE beetle[37] equipped with a TI CC2540 chip. The touch data from the skin interface is sent via BLE to a laptop with 16GB RAM to perform neural network inference and the inference values are then sent back to the skin via BLE. We set the BLE connection interval to 7.5 ms to enable faster communications, trading off the higher power consumption.

2. On-body bluetooth compute(via smartphone) baseline - We simulated an on-body bluetooth compute using a Google Pixel 3 XL smartphone being attached on the participant's chest with the help of a chest band. The implementation is similar to our off-body BLE compute baseline, except that the touch data is sent via BLE to the smartphone. We used TFLite library to implement the neural network inference in Pixel 3 XL. We found out that the BLE communication delay reduced by 15% whereas the inference computation time increased by 70% because of the compute capacity of the smartphone processor.

3. On-body centralized compute baseline - For the on-body microprocessor based compute architecture, we used a 100KHz Raspberry Pi Zero 2 W which is an ARM Cortex A53 processor board. Each artificial skin patch is connected to the processor via I2C, with two SCL and SDA wires. Since its I2C ports are limited, we used an I2C expander board to connect multiple artificial skin patches to the processor. For our prototype with 3 patches, we measured I2C communications taking 97 ms on average, as each AI-on-skin needs to send multiple frames of 12X21 touch data, multiplexed through the I2C expander, to the centralized Raspberry Pi. A larger artificial skin patch with more electrodes will increase the I2C communications delay further. This contrasts with the 1.8ms communications delay for AI-on-skin, highlighting the wiring contention overhead of a centralized vs distributed design.

4. On-body centralized compute with multi-hop routing - One of the other alternatives to improving the communication delay in centralized compute will be connecting each artificial skin patch with the adjacent artificial skin patch and sending the inputs through multi-hop from one patch to another and finally to the centralized processor. Each artificial skin patch was interfaced to a Teensy 4.0 board(via I2c) to facilitate data acquisition. We then connected each artificial skin patch to the nearest skin patch forming a network of artificial skin patches via GPIO pins in Teensy 4.0. This facilitates multi-hop routing where each skin patch sends input to the centralized processor via closest skin patches and forming a shortest path of skin patches for hopping input to the centralized compute. We found that a multi-hop architecture implementation contributes to even higher communication delays. Although the artificial skin patch closest to the centralized processor transfers input faster, the artificial skin patch farther in distance from the centralized processor experiences higher communication delays owing to the fact that it needs to undergo multiple hops between adjacent skin patches.

5. On-body centralized Shenjing accelerator compute baseline - In order to evaluate the performance of a centralized accelerator compute, we made use of a single Shenjing accelerator implemented using CMOD A7 FPGAs for neural network inference. Each artificial skin patch is interfaced to the FPGA using I2C interface and the FPGA selects each artificial skin patch input using a TCA9548A I2C multiplexer for neural network inference. However, this architecture also

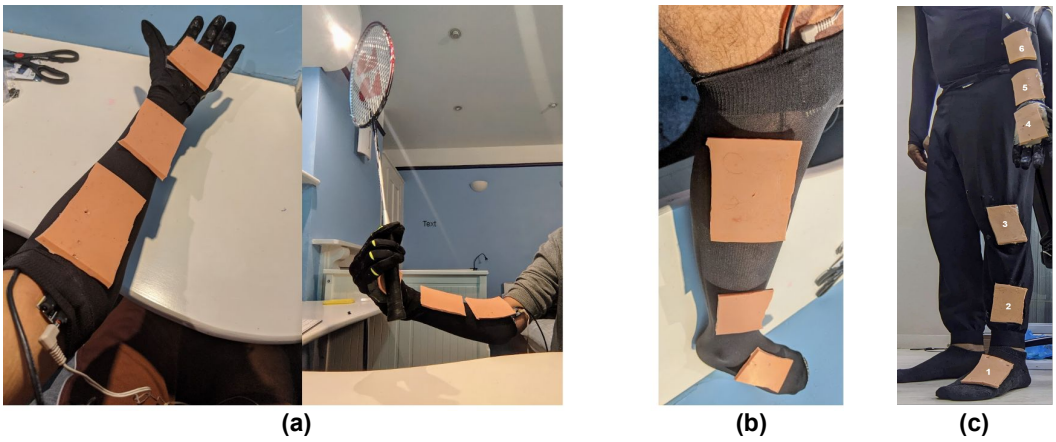

Fig. 10. (a)Hand worn three patch AI-on-skin enabled badminton training system for real-time training of badminton shots (b) Leg worn three patch AI-on-skin system for stroke rehabilitation exercises and (c)A 6-patch AI-on-skin enabled leg and arm worn prototype providing real-time exercise feedback

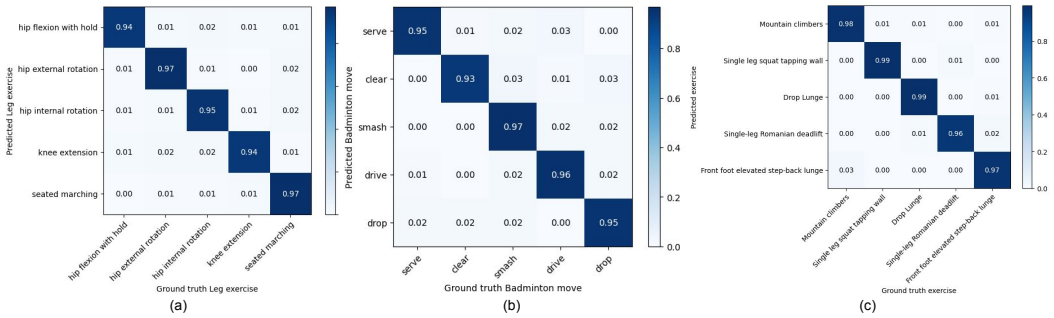

Fig. 11. Confusion matrices of our AI-on-skin predictions for - (a)Hand worn three patch AI-on-skin enabled badminton training system for real-time training of badminton shots (b) Leg worn three patch AI-on-skin system for stroke rehabilitation exercises and (c)A 6-patch AI-on-skin enabled leg and arm worn prototype providing real-time exercise feedback.

suffers from higher I2C communications delay as well as compute delay arising from sequential computation of neural network inference on the data from each skin patch. This demonstrates that a distributed neural network compute architecture offers faster communication as well as inference computation time.

6. On-body distributed non-shenjing accelerator compute baseline - To evaluate the performance of our shenjing based spike neural network accelerator against commercially available accelerator boards, we replicated our on-body distributed AI-on-skin accelerator compute by replacing our Shenjing FPGA accelerator with the state-of-the-art commercial MAX78000FTHR[4] CNN accelerator board running at 60MHz. We interfaced each skin patch to a MAX78000FTHR board via I2C and each skin patch can communicate inferences with adjacent skin patches through GPIOs. We show that Shenjing outperforms MAX78000FTHR by 10X in latency due to the sparse computation associated with it.

[4]https://www.maximintegrated.com/en/products/microcontrollers/MAX78000FTHR.html

Model details	Input (12X21) (253) (254) Convolutional layer (3X3) Convolutional layer (3X3) Average Pooling (2X2) Fully connected layer (256) Fully connected layer (5)(5)(50 neurons)	Input (12X21) (253) (254) Convolutional layer (3X3) Convolutional layer (3X3) Average Pooling (2X2) Convolutional layer (3X3) Convolutional layer (3X3) Average Pooling (2X2) Fully connected layer (256) Fully connected layer (5)(5)(50 neurons)	Input (12X21) (253) (254)(253)(254)(255) Convolutional layer (3X3) Convolutional layer (X3) Average Pooling (2X2) Convolutional layer (3X3) Convolutional layer (3X3) Average Pooling (2X2) Convolutional layer (3X3) Convolutional layer (3X3) Average Pooling (2X2) Fully connected layer (128) Fully connected layer (5)(5)(50)(5)(5)(50 neurons)
Training - samples per user	100 samples per exercise * 5 exercises = 500 samples	50 samples per shot * 5 shots = 250 samples	50 samples per exercise * 5 exercises = 250 samples
Total training samples	500 samples per user * 5 user = 2500 samples	250 samples per user * 7 user = 1750 samples	250 samples per user * 5 user = 1250 samples
Testing - samples per user	30 samples per exercise * 5 exercises = 150 samples	20 samples per shot * 5 shots = 100 samples	20 samples per exercise * 5 exercises = 100 samples
Total testing samples	150 samples per user * 5 user = 750 samples	100 samples per user * 6 user = 600 samples	100 samples per user * 7 users = 700 samples
ANN accuracy (%) [obtained from PC]	96.3	96.5	97.9
Off-body BLE compute inference time (ms)	170	206	434
On-body BLE compute inference time (ms)	223	297	563
On-body centralized compute inference time (ms)	257	298	574
On-body centralized compute with multi-hop routing inference time (ms)	378	389	658
On-body centralized shenjing compute inference time (ms)	198	214	387
On-body distributed non-shenjing accelerator compute inference time (ms)	163	181	278
Off-body BLE compute Power (mW)	38.35	39.45	76.8
On-body BLE compute Power (mW)	38.35	39.45	76.8
On-body centralized compute Power (mW)	52.5	54.25	72.35
On-body centralized compute with multi-hop routing Power (mW)	59.5	63.75	87.35
On-body centralized shenjing compute Power (mW)	39	40.8	63.7
On-body distributed non-shenjing accelerator Power (mW)	51.2	54.5	85.8
AI-on-skin FPGA Power (mW)	143	148	205

Table 2. Neural network model details and AI-on-skin power-performance for user trials of 3 prototype applications- (a) Real-time feedback for leg stroke rehabilitation (b) Real-time feedback for badminton coaching and (c) Real-time feedback for exercises involving legs and hands.

Timing. We measure the inference latencies for AI-on-skin and the baselines as follows. AI-on-skin's inference latency = time taken to send data from SRAM to FPGA + inference time taken by Shenjing AI accelerator emulated by FPGAs + time taken to send inference back from FPGA to SRAM (All occuring within the CMOD A7 board). The off-body/on-body BLE baseline's inference latency = BLE transmission time of the input touch data from Bluno to PC + inference time taken on PC + BLE transmission time of the inference from PC to Bluno. The on-body centralized compute(without/with multi-hop) baseline's inference latency = time taken for sending touch data to the centralized compute device+ time taken for the neural network inference by TfLite + time

taken for sending the inference back. We did not take into account the data acquisition delay and the pre-processing time for all prototypes, since it is identical for AI-on-skin as well as the baselines.

Power. We measured the power consumption of the baselines using a Monsoon power monitor. The power consumption from our AI-on-skin FPGA prototypes were also measured using a Monsoon power monitor. As explained earlier, we used FPGAs to emulate our AI accelerator as they can enable rapid prototyping and hence our prototypes result in relatively high power consumption as shown in Tables 2 and 3. As the CMOD A7 FPGA board used in our prototype consists of many unused chips such as SRAMs, FTDI chips and many IOs which consume power, the power consumption of our FPGA prototype is high. For instance, just the FTDI chip in CMOD A7 consumes an average current of 30 mA. Future deployments of AI-on-skin will use fabricated ASIC chips instead of FPGAs. We fabricated a 12-core Shenjing AI accelerator chip on a commercial 40nm process. Using circuit simulations with Synopsys Design Compiler, we obtained the power and performance of the fabricated AI-on-skin silicon chip for each of the demonstrated applications at the targeted clock frequency of 10MHz. Timing and power numbers are obtained for each neural network model with actual touch data inputs, based on the exact number of neuron cores used, since unused neuron cores are clock and power gated.

Tables 2 and 3 summarize the experimental setup and measurements we obtained from user trials carried out for 7 applications. Each application was deployed on 5 to 7 users for training, and then tested on 5 to 6 different users. We can clearly see that AI-on-skin outperforms both the on-body microcontroller compute and off-body BLE compute baselines by **50X,100X and 20X,11X** respectively in latency and power on average. Figures 11 and 12 depict the confusion matrices of our AI-on-skin predictions against ground truth labels for each of the 7 applications demonstrated below.

The neural networks used in our 7 applications range from 5 to 9 layers (Table 3), and are not that complex computationally. The reason the applications perform poorly in off-body implementations is because of the amount of sensor data that was wirelessly transmitted via BLE. As our results (Tables 2 and 3) show, off-body computation took only 5-20ms, as the laptop runs much faster at a 4GHz clock whereas the Bluetooth data transfer took around 50-350ms. Depending on the application, if the sensor resolution could be optimized to send less data to the off-body compute, communications overheads can be reduced. NailO [38] illustrates this, using just 9 sensors on the finger nail for tracking finger gestures. However, this trades off sensor resolution, as compared to our 22x21 skin sensor resolution.

We will discuss the high level accuracy and performance of each application next in Sections 5.2 to 5.5, before doing a detailed comparison of the various baseline architectures against AI-on-skin's on-body, distributed architecture with AI accelerator in Section 5.6.

5.2 Leg and arm sleeves for stroke rehabilitation and sports coaching applications

5.2.1 AI-on-skin for stroke rehabilitation: Research studies have shown that rehabilitation is critical for patients recovering from strokes. Neuroplasticity needs thousands of hours of very focused and concentrated training [39] to induce functional changes in the brain during stroke rehabilitation. During rehabilitation exercises, patients require accurate and near-instant real time feedback to correct the patient's hand or leg movement. Figure 10(a) shows a 3-patch AI-on-skin based prototype worn on leg for providing real-time feedback on patient's leg movements during rehabilitation exercise. During exercising, our AI-on-skin system is able to identify the type of exercise and also provide 10 corrective feedback - move left, move right, move down, move up, stretch front, stretch back, move diagonally up, move diagonally down, move diagonally left, move diagonally right. In our prototype, patch 1 (on the foot) identifies foot movement, and passes to patch 2 on

the leg which identifies leg movements which passes to patch 3 to identify the exercise and its corresponding corrective feedback. The corrective feedback is sent from patch 3 on the knee to a PMODAMP3 audio amplifier for audio output, to nudge the user towards the correct exercise motion. Our AI-on-skin prototype is trained on five different leg exercises - hip flexion with hold, hip external rotation, hip internal rotation, knee extension and seated marching. Five healthy users were recruited for training our 3-patch AI-on-skin prototype and were asked to practice each exercise 100 times - resulting in a total of 100 times X 5 exercises = 500 training samples per user collected.During collection of training data, the healthy users were instructed to make mistakes during exercising and corresponding real-time feedback was logged manually. AI-on-skin is highly responsive with a feedback latency of 6.3ms. Future designers working on developing smart suits for time-critical health sensing applications can use AI-on-skin to achieve their goals.

5.2.2 AI-on-skin for Sports training: Another important use case for on-body AI compute will be providing real-time feedback for athletes during sports training. To demonstrate the rapid response needed in such sports coaching, we developed a hand-worn 3-patch AI-on-skin system (Figure 10(b)) to identify five different badminton strokes (serves, clears, smashes, drives, drops) and provide real-time corrective feedback. Our AI-on-skin system can provide ten corrective feedbacks as explained earlier in the stroke rehabilitation application. In our prototype, patch 1 (on the arm) identifies knee movement, and passes to patch 2 on the wrist which identifies wrist movements, then communicates to patch 3 which identifies the specific badminton stroke and its corresponding corrective feedback. The corrective feedback is sent from patch 3 on the knee to a PMODAMP3 audio amplifier for audio feedback. Seven users were recruited for training our 3-patch AI-on-skin prototype and were asked to practice each shot 50 times - resulting in a total of 50 times X 7 users X 5 exercises = 1750 training samples collected by a laptop. We recruited six new users for live badminton play with our hand worn AI-on-skin prototype. AI-on-skin achieved an inference accuracy of 95.8% with test data of 20 times X 5 exercises X 6 users = 600 samples. AI-on-skin provided a highly responsive corrective feedback within 6.3ms whereas the same application on the off-body BLE baseline took 170 ms. AI-on-skin is a highly suitable choice for real-time athlete training which demands accurate corrective feedback. Since each sport requires a coordination of multiple regions of the body, artificial skin sensors can be readily integrated into the desired locations on the athletes' clothing to improve the real-time performance of an athlete without the need for a human trainer.

5.2.3 AI-on-skin for real-time exercise feedback: Studies [40] show that prompt and real-time feedback while exercising helps to improve sports performance in athletes. Traditionally, athletes rely on a human trainer to receive timely corrective feedback while exercising to improve their performance. In recent years, there have been attempts to provide users with an AI enabled virtual trainer with VR devices such as Kinect [41] to provide real-time exercise feedback. With the advent of smart textiles integrated with artificial skin sensors, athletes can benefit from real-time exercise feedback with the sensory inputs provided from the artificial skin sensors. To demonstrate the ultra-fast response time provided by AI-on-skin in providing real-time exercise feedbacks for five exercises - (a) Mountain climbers, (b) Single leg squat tapping wall, (c) Drop Lunge, (d) Single-leg Romanian deadlift and (e) Front foot elevated step-back lunge , we developed a leg and arm worn AI-on-skin prototype consisting of 6 AI-on-skin patches with 3 AI-on-skin patches on the left leg and 3 AI-on-skin patches on the left arm as seen in Figure 10(c). Our AI-on-skin system can provide ten corrective feed backs to adjust arm and leg positions(joint angle, positions etc.) during exercising. In our prototype, patch (1) on the foot starts inferring the exercise feedback and the inference is being sent in a daisy chain fashion from patch (1) to (2) to (3) .. to (6) where the exercise and its corresponding corrective feedback(if required) will be identified. The corrective feedback is

sent from patch 6 on the knee to a PMODAMP3 audio amplifier for audio feedback. 5 users were recruited for training our 6-patch AI-on-skin prototype and were asked to practice each exercise 50 times - resulting in a total of 50 times X 5 users X 5 exercises = 1250 training samples collected by a laptop. We recruited seven new users for live exercise feedbacks with our full-body worn AI-on-skin prototype. AI-on-skin achieved an inference accuracy of 95.8% with test data of 20 times X 5 exercises X 7 users = 700 samples. AI-on-skin provided a highly responsive corrective feedback within 24ms whereas the same application on the off-body BLE baseline took 434 ms. This demonstrates the scalability and the performance guarantee provided by AI-on-skin since all the other baselines fail to provide a corrective feedback within the human reaction time of 150ms. Even as the number of AI-on-skin patches increases considerably, the fully distributed design of AI-on-skin ensures it will still deliver fast response.

To explore if a neural network is necessary for our real-time exercise feedback, and whether other learning models can be used instead, we replaced our neural network model with an SVM model (rbf kernel and kernel coefficient = 1/252 running on a Raspberry Pi) to provide real-time feedback for exercises involving legs and hands and found that it still took 438ms (whereas our neural network model just consumed 434ms).

These three applications showcase how AI-on-skin can deliver much faster inference than the state-of-the-art, truly realizing the real-time sensing, compute and response needed for these applications. From Table 2, we find that AI-on-skin achieves 27X, 19X and 18X faster inference against off-body BLE compute baseline in leg stroke rehabilitation, badminton coaching and real-time exercise feedback respectively. It also demonstrates how the distributed nature of AI-on-skin enables multiple skin patches across the body to work together. AI-on-skin's embedding of SNN hardware accelerators on the artificial skin managed to realize high inference accuracy close to that of software ANNs with a negligible accuracy degradation of 1%.

5.3 Gloves for shape sensing and texture sensing for prosthetic arms

One of the major use cases for artificial skin sensors is to provide rich tactile sensing abilities for prosthetic arms and thereby enable such arms to sense texture, shape etc. of object via touch. Figure 13(B) shows that a regular textile glove can be augmented with on-skin AI compute capacities by simply overlaying a layer of AI-on-skin patch over it. Figure 13(B) shows our AI-on-skin enabled glove sensing prototype. We made use of the glove prototype to sense the shape of 5 objects - (a) Coffee container (b) TV remote (c) Laptop (d) A toy gun and (e) cup (as seen in figure 13(C)). The neural network model used for recognizing object shapes with AI-on-skin gloves is shown in Table 3. The shape sensing gloves were able to infer the shapes of each object with 98% accuracy in 1.5ms. Recently, Tasbolat et al. [42] showed that touch data from artificial skin and visual information can be fused together to enable shape recognition of 36 types of containers with an accuracy of 80% with CNN based neural network model. Their design consists of a single skin patch sensor, wired to a off-body Loihi AI accelerator to perform neural network inference with a latency of 1.3ms. As Loihi is not designed for battery-powered wearables, it consumes a high power consumption of 1.34W for neural network inference. On the other hand, our AI-on-skin gloves can provide low latency of 1.5ms for neural network inference. Even with our FPGA implementation, AI-on-skin consumes a total power of 260 mW which is 5X more power efficient than Loihi. With AI-on-skin fabricated as silicon chips, the power consumption can be further reduced by 10X vs. the AI-on-skin FPGA implementation. Although AI-on-skin is primarily designed to serve applications with multiple skin patches, we also show that it achieves better performance even in the case of a single patch AI-on-skin application. This is largely due to the faster compute provided by the SNN accelerator being used in AI-on-skin. Besides, AI-on-skin can scale to multiple skin patches across the body, enabling fusion of sensing data from multiple prosthetic limbs, with AI computed

Model details	Input layer(252) Convolutional layer(3X3) Pooling layer(2X2) Fully connected layer (128) Fully connected layer (64) Fully connected layer (5)	Input layer(252) Fully connected layer (256) Fully connected layer (128) Fully connected layer (64) Fully connected layer (5)	Input layer (107) Convolutional layer (3X3) Convolutional layer (3X3) Convolutional layer (3X3) Pooling layer (2X2) (Average) Fully Connected layer(128) Fully connected layer(2)	Input (12X21) Convolutional layer (3X3) Convolutional layer (3X3) Average Pooling (2X2) Convolutional layer (3X3) Convolutional layer (3X3) Average Pooling (2X2) Fully connected layer (128 neurons) Fully connected layer (7 neurons)
Number of FPGAs (Each FPGA can map 4 cores)	2	2	2	3
Training - samples per user	50 samples per object * 5 objects = 250 samples	50 samples per user * 5 textures = 250 samples	50 samples per word * 50 words = 2500 samples	100 samples per gesture * 7 gestures = 700 samples
Total training samples	250 samples per user * 5 user = 1250 samples	250 samples per user * 5 users = 1250 samples	2500 samples per user * 5 user = 12500 samples	700 samples per user * 5 user = 3500 samples
Testing - samples per user	20 samples per object * 5 object = 100 samples	20 samples per user * 5 textures = 100 samples	5 samples per user * 50 words = 250 samples	50 samples per gesture * 7 gestures = 350 samples
Total testing samples	100 samples per user * 5 user = 500 samples	100 samples per user * 5 user = 500 samples	250 samples per user * 5 user = 1250 samples	350 samples per user * 5 user = 1750 samples
ANN accuracy (%) [obtained from PC]	98	97.5	97.5	95.9
Off-body BLE compute inference time (ms)	52	63	254	62
On-body BLE compute inference time (ms)	61	77	403	69
On-body centralized compute inference time (ms)	98	123	258	102
On-body centralized compute with multi-hop routing inference time (ms)	98	123	381	102
On-body centralized shenjing compute inference time (ms)	1.5	2.2	304	3.5
On-body distributed non-shenjing accelerator compute inference time (ms)	68	76	318	74
Off-body BLE compute Power (mW)	12.5	12.5	68.5	14.25
On-body BLE compute Power (mW)	12.5	12.5	68.5	14.25
On-body centralized compute Power (mW)	18.75	19.1	108.5	21.5
On-body centralized compute with multi-hop routing Power (mW)	18.75	19.1	127.5	21.5
On-body centralized shenjing compute Power (mW)	0.97	0.84	92.5	1.02
On-body distributed non-shenjing accelerator Power (mW)	21.3	20.6	121.5	26.7
AI-on-skin FPGA Power (mW)	45	43	214	47

Table 3. Neural network model details and AI-on-skin power-performance for user trials of 4 prototype applications.- (a) Shape sensing with AI-on-skin enabled gloves (b) Texture sensing with prosthetic arms (c) Spell correction enabled handwritten word recognition and (d) Emotional communication with virtual embodied agent.

on these sensor streams directly on the body, at low power. In addition, AI-on-skin is also highly accurate (98%) which would be a critical requirement for any prosthetic limb to ensure smooth day-to-day activities.

Similarly, we can use gloves to identify the texture of touch surfaces and reproduce a similar tactile perception on the elbow for users with prosthetic arms. Figure 13(b) shows our AI-on-skin glove which can detect textures, with vibrators on the user's elbow producing a unique vibratory pattern for each texture. In future, the vibrators on user's elbow can be replaced with skin sensors such as Haptic Skin[8], Soft actuators[6], Springlets[43], Tacttoo[44] etc when they become commercially available. Our glove prototype was trained to identify five touch textures - soft, hard, silky, slimy,

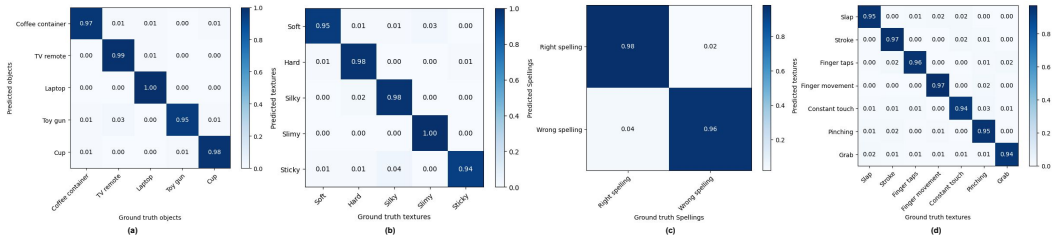

Fig. 12. Confusion matrices of our AI-on-Skin predictions - (a) Shape sensing with AI-on-skin enabled gloves (b) Texture sensing with prosthetic arms (c) Spell correction enabled handwritten word recognition and (d) Emotional communication with virtual embodied agent.

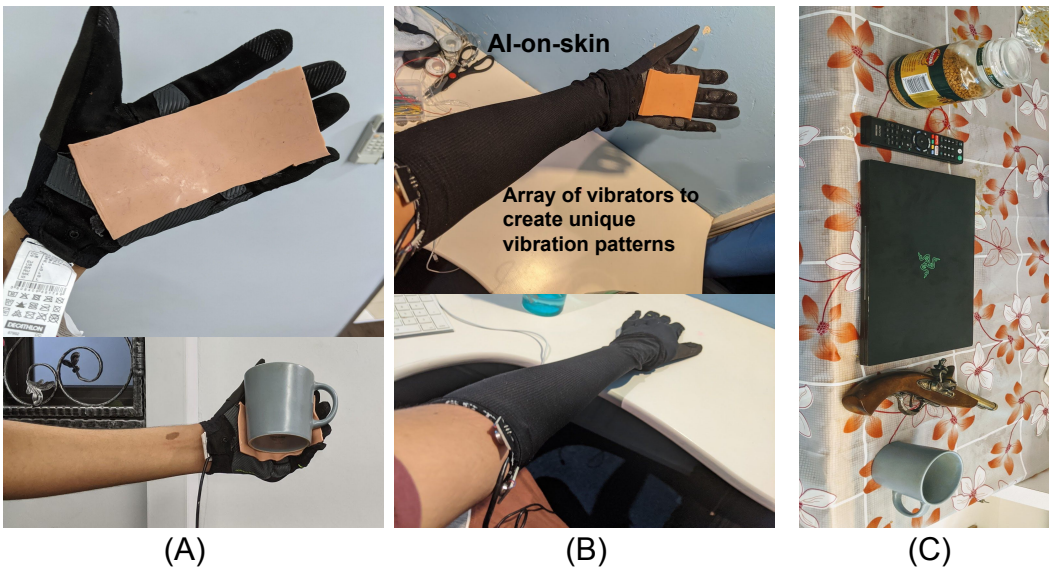

Fig. 13. (A) AI-on-skin enabled shape sensing gloves for recognizing object shapes via grasp and the user is wearing AI-on-skin enabled gloves to recognize a cup via it's shape (B) AI-on-skin enabled texture sensing for prosthetic gloves to reproduce tactile outputs at the arm. (C) Objects used for shape recognition were a coffee container, TV remote, laptop, toy gun and cup

sticky. cotton pillow, blanket and soft clothes were used for soft textures. For hard textures, floor, rock and wood were used. We used silk fabric, jelly and sticky silicone rubber for silky, slimy and sticky textures respectively. AI-on-skin's texture sensing gloves were able to infer the texture of each surface at 97.3% accuracy in 2.2ms. This rapid response from AI-on-skin can benefit the community of researchers and developers working on such tactile perception applications, enabling not just single touch applications but applications that rely on near-instant response to sequence of touches.

5.4 Emotional communication with artificial skin interfaces

Figure 14 shows our AI-on-skin prototype communicating with the on-body display of a virtual embodied agent which can react to human touch gestures mimicking human-to-human interactions. For example, people can express happiness by a series of repeated taps, sympathy by means of

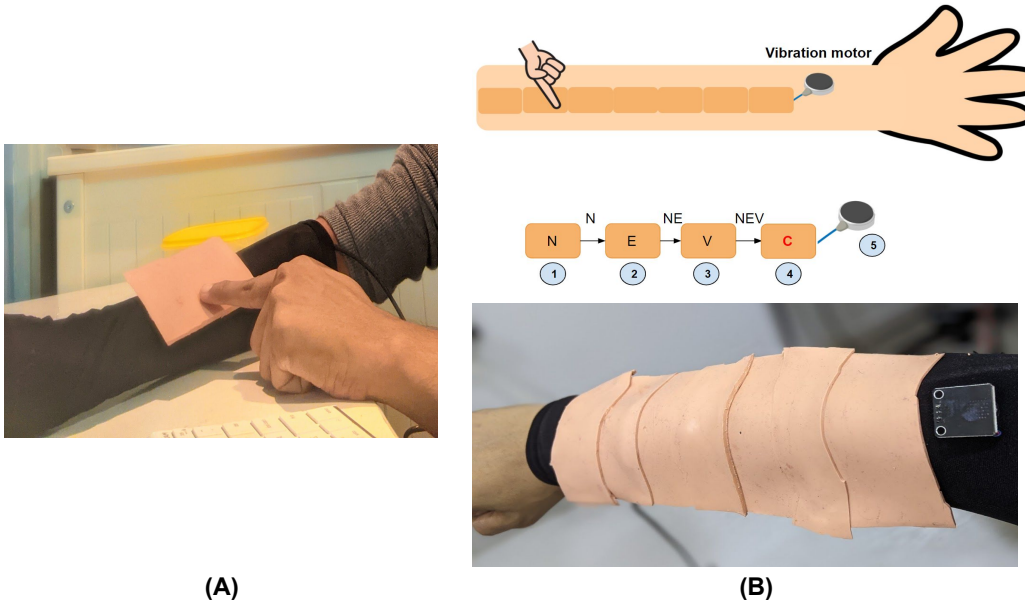

Fig. 14. (A)AI-on-skin for emotional communication with embodied conversational agent in a wearable display (left image) and user interacts with a AI-on-skin patch to communicate with a virtual embodied agent. (B) Our 7-patch AI-on-skin prototype with an integrated vibration motor (bottom) to denote spelling errors made while writing. Users write one alphabet per AI-on-skin patch, with outputs from previous patches passed to the next patch to enable spell checking. E.g., a user tries to write the word NEVER - (1) "N" is written on first patch, recognized and passed to second patch (2) "E" is written on the second patch, recognized and "NE" passed to third patch (3) "V" is written on the third patch, recognized and "NEV" passed to fourth patch (4) Here, the user makes a mistake, writing C instead of V. This prompts detection of a spell error in the fourth patch, triggering the vibration motor (top)

.

a soft stroke, anger using pinching etc. Users can also make a light touch on the skin to make sure that he/she is listening to the embodied conversational agent (ECA). Such scenarios require a real-time ECA that can react to human touch swiftly, with no time lag.

We program our AI-on-skin prototype to identify representative touch gestures and its corresponding human emotion to be conveyed to the embodied conversation agent. Our AI-on-skin can identify the following touch gestures: (1) Slap (anger emotion) (2) Stroke (sympathy/comfort) (3) Finger taps (to seek for attention) (4) Finger movement (for tickling) (5) Constant touch (comforting the embodied agent) (6) Pinching (upset) and (7) Grab (anger emotion). The touch gestures made on AI-on-skin are recognized using our corresponding 6-layer neural network model shown in Table 3. Our setup for AI-on-skin communication with ECA is shown in Figure 14. The user makes emotional communication with the AI-on-skin prototype worn on his wrist and the touch emotions identified using our neural network model are communicated to an ECA displayed on the laptop via UART. In future, we can support ECA on smartwatch's display, or wearable displays integrated with e-textiles.

Our measurements show AI-on-skin can perform live inferences at 95.9% accuracy, communicating the inference to the embodied conversational agent within 3.5ms, well within human reaction time. In future, the tactile emotion perceived by AI-on-skin can also be recreated by neural network

and sent to the receiving user. The receiving user can then experience tactile emotions from the sender via tactile feedback sensors. AI-on-skin remained highly accurate in perceiving tactile emotions made by the user to ensure a smooth emotional communication with the virtual ECA. Since AI-on-skin can be readily integrated into any form-factor to provide real-time response, future developers can also make use of AI-on-skin to develop off-the-shelf emotional communication devices for phones, watches, toys, household appliances/furniture etc.

5.5 Hand-worn spell correction enabled handwritten word recognition

We demonstrate a simple 7 AI-on-skin patch based spell correction enabled handwriting word recognition system (see Figure 14(B)) that can be used by students in their learning. The prototype is trained to recognize a fixed set of 50 seven letter words. Each AI-on-skin patch is connected to a single FPGA which runs neural network inference for handwritten uppercase English alphabet recognition. We enable spelling correction into this system by passing outputs from each AI-on-skin patch to the next patch. Figure 14(B) shows a use-case in which the user tries to write a word, and AI-on-skin will trigger a vibration in real-time when spell check fails.

During training, each AI on skin patch is trained on a fixed set of 50 words. For example, if the word is Jacuzzi, then the input-output pairs for neural network model will be all the correct letter combinations like (("<Empty>","J"),"correct"), (("J","a"),"correct"), (("Ja","c"),"correct"), (("Jac","c"),"correct") etc. for all 50 seven letter words. We also create training pairs for mis-spelt words, such as (("J","b"),"incorrect"),...,(("J","z"),"incorrect") for invalid combinations in our word set. In such a way, the training for correct and incorrect spellings are made in the system. The prototype may be worn by a student to practice word spellings within a particular seven lettered word vocabulary. For training the system, we collected handwritten data of 7 lettered words from 5 users. From the collected handwritten data, the training pairs were constructed as explained earlier and the neural network model shown in Table 3 was trained. The trained parameters are then fed into AI-on-skin software mapper which maps the trained parameters into configuration signals for our FPGA neural network accelerator.

During testing, the users were asked to intentionally make many mistakes in order to trigger spell error identification. AI-on-skin obtained an average spell correction accuracy of 97% while the same test data (collected from the users during test) had an average accuracy of 97.5% when run through the corresponding ANN on a PC. This demonstrates the negligible drop in accuracy with the ANN to SNN conversion of AI-on-skin. Each AI-on-skin patch performed neural network inference within 3.6 ms. Even with 7 AI-on-skin patches being used in this prototype, AI-on-skin still delivers high responsiveness that is much faster than the human reaction time. This reaffirms the scalability of AI-on-skin, making it an apt choice for full body skin sensing applications.

5.6 Why AI-on-skin is faster compared to alternative AI-compute baselines?

Here, we delve into our experimental results which demonstrate AI-on-skin providing faster, real-time response compared to alternative AI-compute baseline architectures. The speedup realized by AI-on-skin is primarily due to two factors - (1) AI-on-skin's Shenjing SNN accelerator provides much faster real-time inference compared to traditional CPUs and microcontrollers, and (2) The distributed architecture of AI-on-skin ensures that the neural network computation required for each AI-on-skin patch is done locally, thus removing the high sensor data communication cost incurred in the baseline compute architectures.

To demonstrate the above factors in play, we consider the real-time exercise feedback application (6 AI-on-skin patches) demonstrated in Section 5.2.3. Similar trends persist for the other applications, as shown in the results tables. Figure 15 depicts the complete breakdown of the delay components for real-time exercise feedback across the 6 baseline compute architectures and AI-on-skin. We will

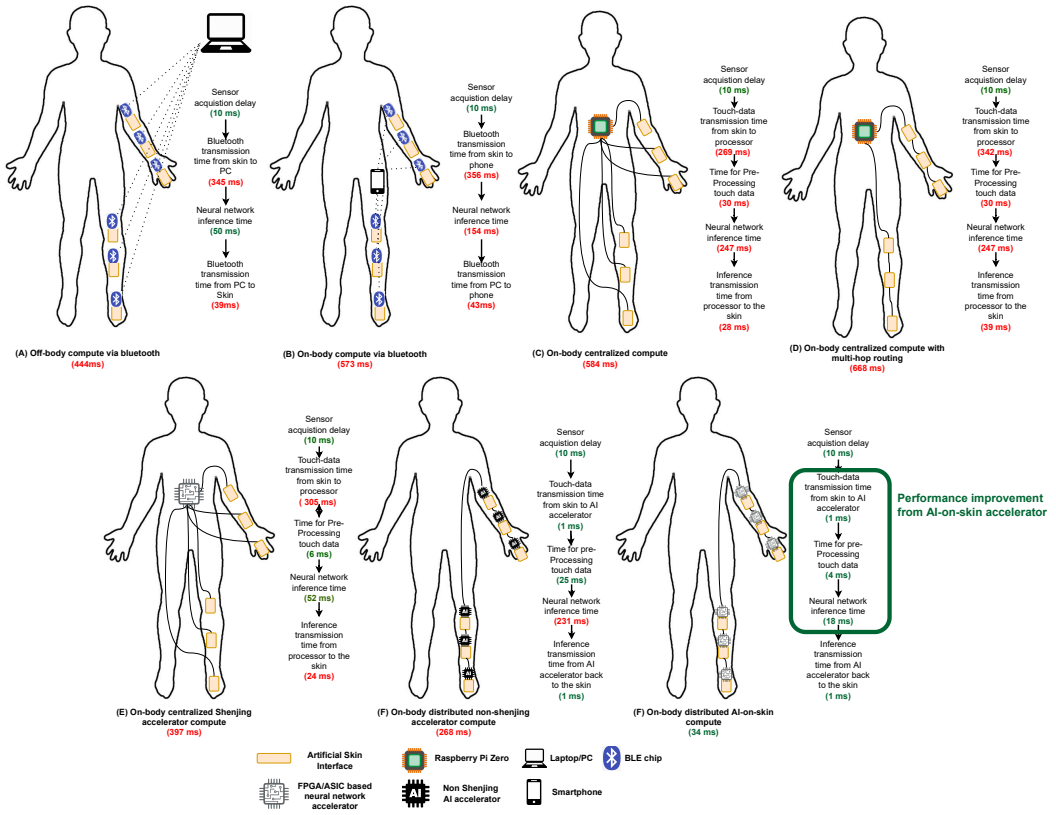

Fig. 15. Breakdown of the individual component delays experienced in three AI compute architectures for real-time exercise feedback.

ignore the sensor data acquisition delay across all architectures since it is the same irrespective of the architecture.

1. Off-body/On-body BLE compute - In an off-body BLE compute architecture, all 6 artificial skin patches transmit data via BLE to an edge device (a powerful laptop) for performing the neural network based real-time feedback and the inferences are communicated back to the prototype. The BLE connection intervals were set to 7.5ms. The total time taken for the feedback was 434ms. There are three significant delay components - (1) Bluetooth transmission time from the artificial skin sensor to the PC = 345ms, (2) Neural network inference time at the PC = 50ms and (3) Bluetooth transmission time from PC to the skin sensor = 39ms. It is clear that more than 80% of the delay is from the bluetooth communications. This also implies that the delay will be worse when the number of artificial skin sensor patches increases further. Optimizing the communications while keeping the compute centralized, with skin patches wirelessly sending via BLE to a centralized smartphone, takes 563ms. As shown in Figure 15, an on-body BLE compute via a smartphone held on the pant pocket incurs a high communication cost (356ms) and compute cost (154ms). The compute time is higher in an on-body BLE setup since smartphone processors are lower performance than the processors within laptops. Overall, this design fails to deliver the required response within the human reaction time of 150ms.

2. On-body centralized compute - In an on-body centralized compute architecture, all 6 artificial skin patches transmit data via direct wires (I2C) to a centralized Raspberry pi Zero board for performing the neural network based real-time feedback and the inferences are communicated back to the prototype via I2C. The total time taken for the feedback was 574ms. There are four significant delay components - (1) the time taken to send touch data from all six artificial skin patches to the centralized compute via I2C = 269ms, (2) time taken for prepossessing the touch data = 30ms, (3) Neural network inference time at the Raspberry pi zero = 247ms and (4) Inference transmission time from raspberry pi zero to the skin sensor = 28ms. We again see that more than 60% of the delay is from the I2C communications - thereby failing to provide feedback within the human reaction time of 150ms. I2C communications takes significant time, as each AI-on-skin needs to send multiple frames of 12X21 touch data, multiplexed through the I2C expander, to the centralized Raspberry Pi zero.

Optimizing this on-body communication delay by doing multi-hop routing across several short wires to the centralized Raspberry Pi processor further increases the response time to 658ms, as on-body communication delay increases from 269ms to 342ms. Even replacing the centralized Raspberry Pi processor with a faster processor does not suffice: Replacing Raspberry Pi with the Shenjing accelerator leads to a faster response time of 397ms. It is still not fast enough for interactive human perception since the communications delay was still 305ms.

All these architectural exploration results show that a centralized compute architecture is not suited for on-body computing.

3. On-body distributed AI-on-skin compute - AI-on-skin gets rid of the communication cost involved in transferring touch data to the off-body/on-body compute since each artificial skin patch is interfaced to its own localized AI-on-skin accelerator for performing localized neural network inferences. This fully distributed architecture of AI-on-skin removes the need for the communication of touch data between adjacent AI-on-skin patches. Since AI-on-skin only communicates 8 bit inferences between adjacent patches, the communication cost being incurred is negligible. Overall, AI-on-skin took just ***24ms*** for real-time exercise feedback. The communication time was only 2ms in AI-on-skin (Touch-data transmission time from artificial skin to AI accelerator + inference transmission time from AI-accelerator to artificial skin sensor), while the total time for computing the feedback was 22ms. The total time taken for neural network inference was just 18ms due to the fast SNN accelerator. When the AI-on-skin distributed architecture was redesigned by replacing the Shenjing SNN accelerator with a commercially available neural network accelerator (MAX78000FTHR4 CNN accelerator board), the total delay was 258 ms with a high neural network inference computation time of 231ms. This shows that AI-on-skin's leveraging of the inherent sparsity of SNNs led to a magnitude of improvement in the inference computation time.

6 Qualitative wearability study

In this short user study, we tried to understand the wearability factor of our currently developed AI-on-skin prototype. We aim to uncover the user reaction towards AI-on-skin based prototypes developed from our current fabrication approach which would be embedded into smart textiles in future. Based on previous studies [45], we chose the following aspects for our user evaluations:

- Mechanical durability - Do our AI-on-skin prototype remain adhered well to our skin/textile under long-term wear?
- Electrical Functionality - Do our AI-on-skin based skin interfaces remain electrically functional while being worn on the skin?
- Comfort - Does the wearer feel comfortable while wearing our AI-on-skin prototypes?

- Social perception - Does the wearer feel comfortable wearing our AI-on-skin based prototypes in public while being seen by others?
- Perceived speed - How fast does the user receive real-time feedback from AI-on-skin?

Procedure. We used the AI-on-Skin prototype for providing real-time exercise feedback as shown in Figure 10(c). We recruited 12 participants (8 Male and 4 Female) with ages between 20 to 31. The participants were asked to wear our prototype for 45 minutes as they carried on with their exercising activities. After the end of 45 minutes, the participants reported their experience via a post-interview study. Figure 16 shows the results of our post-interview study.

Results.

1. Mechanical Durability. For all the participants, the device remained attached on participants' body for entire 45 minutes. For one participant, one of the skin patch came off slightly at the edges due to a tear in the sewing. All participants confirmed that the device remained very well attached during exercising, with a median score of 6.25 on the likert scale.

2. Electrical Functionality. All the participants agreed that the device (median = 6.75) remained electrically functional throughout the 45 minutes of exercising. One participant experienced a wire disconnection due to the fact that his arm was bigger for the prototype. We then customized the prototype with longer wires to fit the participant arm and fixed the wire disconnection. The electrical functionality was scored by measuring the number of real-time feedback misses that occurred and converted the success ratio to the 7 point likert score.

3. Comfort. Participants agreed that the prototype was fairly comfortable to wear during exercising with a median score of 5.5. However, all the participants reported that the prototype felt slightly heavy. This is due to the fact that the currently used commercially available DIY MUCA in a slightly heavier artificial skin interface owing to its fabrication process. With other thinner artificial skin interfaces [2, 19], our prototypes can be easily made lighter.

4. Social Perception. Participants rated that they would wear our prototypes in public with a median score of 5.25. 6 participants reported that they would not feel comfortable to wear the prototype in public since the color of our current prototype is very bright and does not match their skin color or dressing styles. In future, we can customize the color of the artificial skin patches to go well with each user's dressing style or skin color.

5. Perceived speed. All the participants strongly agreed that AI-on-skin provided ultra-fast feedback while exercising with a median score of 6.75. Only one participant reported a score of 6 because of the wire disconnection issue between his earphones and our audio amplifier which resulted in the loss of certain real-time exercise feedbacks.

Previous works [46] have studied the social acceptability of on-body skin sensor locations. Although the primary focus of our work is to demonstrate the power and performance benefits of computing AI-on-skin, AI-on-skin can readily support the desired on-body locations chosen in earlier works, such as arms, thighs, legs, chest, shoulder-back etc.. As the AI-on-skin accelerator can be fabricated as a tiny ASIC chip (2cmX2cm) and integrated into future smart textiles, it can potentially provide better comfort, social acceptance and all-day wearability.

7 Future use-cases for AI-on-skin

AI-on-Skin can be used for many applications in health, user interfaces and gaming industry. In this section, we present two main use-case scenarios of AI-on-Skin in real-world applications.

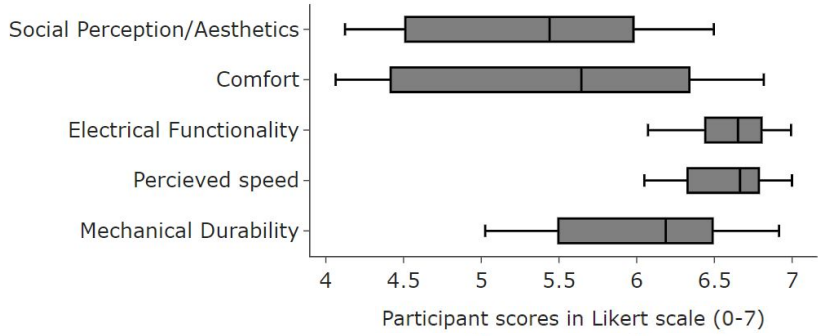

Fig. 16. Box plot of our participant scores (in likert scale) during our user trials.

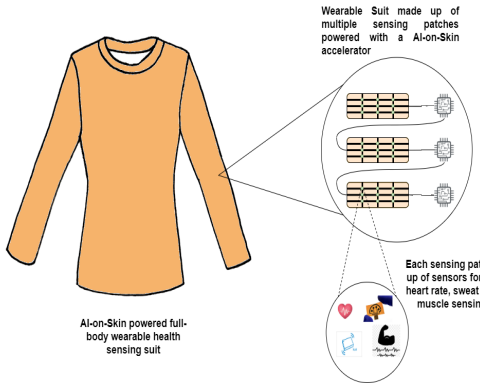

Fig. 17. Full body wearable health sensing suit powered with AI-on-Skin

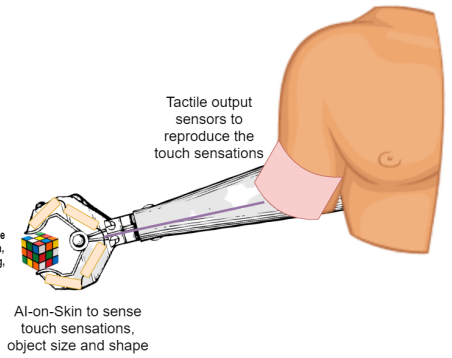

Fig. 18. AI-on-Skin to assist tactile output sensors for haptic arms

7.1 Full body suit with AI-on-Skin

Recently, autonomous humanoid robots fully covered with artificial skin to sense the entire body have been demonstrated [47]. In recent years, electronic skin based fabrics have garnered increasing attention owing to the intelligent functionalities they can provide to the user. An all-fabric pressure sensor [48] was realized with a wireless battery-free monitoring system. Wicaksano et al. [49] developed a tailored electronic textile suit to perform temperature, heart rate and respiration in vivo and aid in the diagnosis of seismocardiac events. A low-cost, stretchable electronic fabric to sense pressure, stress and flexion induced by non-contact finger proximity was also demonstrated [50]. This could open up spaces for applications like voice recognition, non-contact airflow monitoring and finger/wrist/muscle movement monitoring etc. There have also been increasing interest in real-time, non-invasive monitoring of sweat analytes such as glucose [51], sodium [52], pH [53] etc aiding in everyday molecular monitoring providing a clear picture of your health. We believe that in future, readily available fully worn electronic fabrics (embedded with artificial skin sensors) will be commonplace and AI would be the key driving factor to process data from those electronic fabric sensors in real-time. Since all these applications sense your health and fitness, real-time feedback from AI engine would be critically needed. For instance, ReStore [54] is a battery powered, very soft exo-suit to assist in lower limb disability caused by stroke. Exo-suits completely covered with artificial skin sensors could potentially drive real-time feedback for patients to make the right limb

movement in a split second. However all the above-mentioned applications require long battery life and extremely fast real-time feedback from the AI compute engine. AI-on-Skin presents a suitable option that is scalable, ultra-low-power and capable of fast inference.

Figure 17 shows a full-body worn textile suit embedded with artificial skin sensors for health and environment sensing ranging from heart rate sensors, EMG sensors for muscle movement monitoring, IMU sensors for activity monitoring to sweat sensors for molecular monitoring. This wearable suit would be a potential use case for AI-on-Skin since each artificial skin sensing patch on the suit can be powered with AI-on-Skin to produce rapid inferences on our health in real-time. For example, it could help prevent stroke or paralysis or anxiety attacks with real-time inferences on your health every second and can potentially predict an onset of adverse health conditions. It could also be helpful in paralysis or stroke for rehabilitation to monitor muscle movements or minute motions made by patients and make real-time feedback or recommendations.

7.2 AI-on-Skin for prosthetic arms

When a person loses his arm or leg during an accident, an artificial prosthetic arm or leg can be used to replace it. However, when he/she touches or holds something with his prosthetic arms, a haptic feedback would be required to make the person feel the sense of touch. In recent years, there has been increasing interest in developing haptic feedback sensors to recreate the feeling of being touched. Some of the recent works include (but not limited to) - Haptic Skin [8], Soft Actutators [6], Springlets [43] etc. Currently, most of these haptic feedback sensors are targeted for gaming/virtual reality applications. With AI-on-Skin integrated into prosthetic arms, we can enable these haptic feedback sensors to recreate an accurate feeling of touch and help people with prosthetic arms experience touch.

Figure 18 shows a prosthetic arm covered with AI-on-Skin which in turn powers a tactile output/haptic feedback sensor. The prosthetic arm covered with distributed, touch-sensitive AI-on-Skin sensor patches can be trained to sense various feelings of touch like heat, cold, texture of the object being touched etc. Since AI-on-Skin can run much faster than the human reaction time of less than 150 ms[23], the inferred sense of touch from AI-on-Skin can be used to generate configuration signals to enable the haptic feedback sensors to recreate an accurate feeling of touch like force of touch, texture being touched etc. Similarly, AI-on-Skin can also be used with robotic arms to help robotic arms produce the appropriate force and pressure to hold an object, to learn the shape of the object etc.

8 AI-on-skin as a makers platform

We plan to release an AI-on-skin makers platform as shown in Figure 19, which consists of an AI-on-skin accelerator and associated AI-on-skin software mapper tool. As already mentioned, current artificial skin sensors had to rely on off-body compute for neural network inference which introduces time lag and makes highly interactive applications infeasible. However, with our AI-on-skin makers platform, the community of researchers, engineers and designers working on artificial skin based prototyping can easily integrate rapid on-body neural network inference abilities into their system. We are working on fabricating a small flexible AI-on-skin accelerator PCB (of size 2cmX2cm) that consists of a programmable 12-core AI-on-skin accelerator and I/Os like USB to UART interface, I2C, SPIO, GPIO etc. As shown in Figure 19, a developer can easily integrate any artificial skin sensor with the AI-on-skin accelerator for interaction/sensing applications.

Firstly, the developer can collect data from the artificial skin sensor via UART and use the collected data to train a neural network model via well-known neural network libraries like TensorFlow and PyTorch. The trained Tensorflow/PyTorch models can be passed to our AI-on-skin GUI software which converts the neural network model to configuration signals for programming the neural

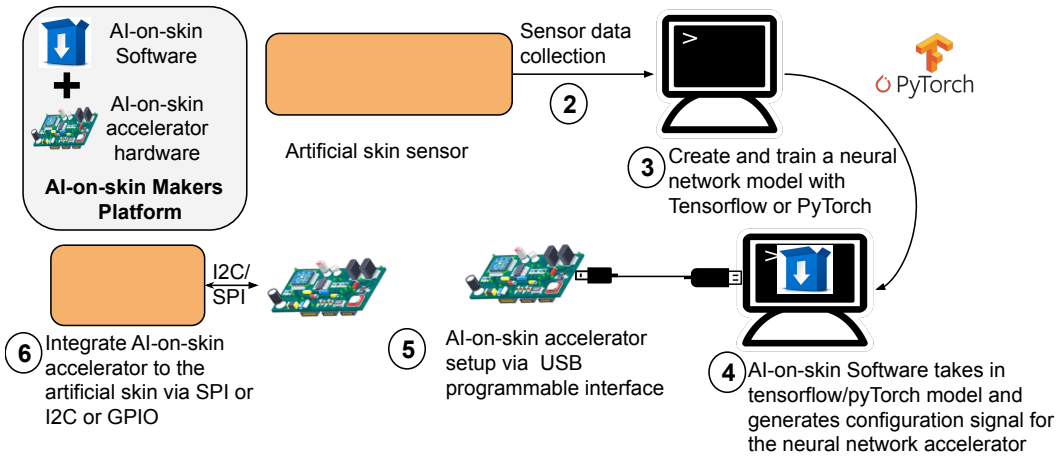

Fig. 19. AI-on-skin makers platform and step-by-step procedure involved in integrating AI-on-skin accelerator to artificial skin sensors

network accelerator. AI-on-skin GUI software will then flash the generated configuration signals into our AI-on-skin accelerator board through a USB programming interface. Finally, our AI-on-skin accelerator configured for a particular neural network model can be integrated into our AI-on-skin accelerator board via I2C/SPI. AI-on-skin accelerator board will also have a programmable ATmega328P arduino microcontroller which allows user to do light on-chip pre-processing on the touch data received from the artificial skin sensor. We believe this would greatly benefit engineers working on wearable artificial skin based applications to introduce on-body AI compute at a relatively low cost.

9 Discussion and Future work

AI-on-skin's benefit to future developers. Future developers can benefit from faster response times for various end user applications such as wearable suits for stroke rehabilitation, neural rehabilitation, identification of touch responses for future prosthetic arms etc. As explained earlier, We also plan to release our AI-on-skin makers board (which is under fabrication) and AI-on-skin software mapper with end-to-end tutorials on how to program multi-patch AI-on-skin applications using our demonstrated applications for easier access to developers/designers in the epidermal computing space.

Support for larger neural network models. AI-on-skin can support neural networks with any number of layers. However, the number of FPGA/ASIC chips used to implement a larger network will increase. The distributed architecture of AI-on-skin scales readily by decomposing larger neural networks into smaller neural network models that do localized neural network inferences and communicate inferences with adjacent AI-on-skin patches.

Relevant scenarios for distributed AI-on-skin AI-on-skin is designed for artificial skin sensors distributed around the body, such as smart suits that demand fast reaction times. For eg., the physiological reaction time for badminton athletes is 80-130ms. If we have an on-body distributed AI-on-skin suit for real-time feedback, the athlete would be able to dynamically adjust their strokes; Off-body computing will not be fast enough. As our experiments show, AI-on-skin outperforms off-body computing via bluetooth and readily achieves neural network feedback within 11ms (19X speedup) (see Table 2. All our demonstrated multi patch distributed applications shows the

superiority of AI-on-skin against off-body compute which fails to achieve a human reaction time of 150ms.

As shown in Tables 2 and 3, AI-on-skin performs very well in applications where multiple AI-on-skin patches (3 to 7 patches) are distributed and off-body compute fails to respond within human reaction time. However, AI-on-skin can also work well for single patch applications to obtain considerable speedup (20X) and low power. We acknowledge that AI-on-skin's distributed architecture is only suitable when the chosen application supports individual AI-on-skin patches to perform localized inferences and communicate only inferences with adjacent AI-on-skin patches.

Support for other artificial skin sensors. AI-on-skin can be integrated with other artificial skin interfaces with just changes in the IO interface. We used MUCA artificial skin interface as it is the only commercially available DIY artificial skin. AI-on-skin can interface with most prior artificial skins without any additional hardware: iSkin[9], PhysioSkin[55] and DuoSkin[19] through the FPGA's analog pins, and Multi-touch skin[2] and SkinMarks[3] through the I2C digital pins. ACES[7] will require additional decoding logic implemented within the FPGA to decode touch data from the single serialized analog signal.

The current AI-on-skin prototype is limited by the sensing modalities supported by the MUCA skin sensor. Interfacing with other skin sensors that embed force, pressure and bio sensing electrodes will broaden the scope of AI-on-skin.

Bluetooth connection interval tradeoff. In our evaluations, we set a connection interval of 7.5ms for off-body computation via bluetooth baseline. However, the recommended low-power connection intervals for bluetooth is 100ms[25], which is 8X more power efficient than 7.5ms connection intervals. Yet, if 100ms connection interval is used, the BLE communication alone will take more than 150ms, unable to keep up with the human reaction time of 150ms. We thus chose a 7.5 ms connection interval so the off-body baseline via BLE is still viable, albeit at a higher power overhead.

Support for residual layers in AI-on-skin. In this work, we have only utilized convolutional layers, fully connected layers and pooling layers in our neural network architecture. However, Shenjing AI accelerator can also support residual layers. Since our architectures were comparatively smaller, we did not require the use of residual layers in our architecture. Currently, Shenjing does not support LSTMs and RNNs. Future research will need to be done to extend its architecture to those networks.

AI-on-skin with other accelerator chips. AI accelerators such as GPUs, NPUs, are now commonplace in many products such as laptops, phones, and many custom AI accelerator chips have been designed by industry and academic research groups. For instance, instead of the microcontroller or Raspberry Pi for on-body centralized compute, a Google edge TPU Coral board can be used [56]. However, the Coral board takes up 2W for inference, which is too high for wearables, and will still face the I/O congestion issue.

Also, most AI accelerators in consumer products target artificial neural networks (ANN). AI-on-skin chose a SNN accelerator chip instead of an ANN accelerator because of its ultra low power budget - Battery life of a wearable should last at least a day before requiring recharging. Amongst SNN accelerators, there have also been several chip prototypes from industry and academia, such as the IBM TrueNorth[57] and Intel Loihi[58]. IBM TrueNorth is one of the first SNN accelerator chips from industry but it suffers from a downside which is especially critical for AI-on-skin - When a model cannot fit within a single neuron core and is naively partitioned across multiple cores, inference accuracy will drop significantly as weight summations and spike generation have to be done at each core. Since our CMOD A7 FPGA has such limited resources, and can fit only 4 neuron cores, AI-on-skin's FPGA prototype will suffer significant accuracy degradation if we use TrueNorth as the accompanying accelerator for each patch without extensive neural network

model engineering. Intel Loihi is also not suited for our AI-on-skin, as each Loihi chip includes a conventional CPU core, leading to substantial power consumption. Powering multiple such Loihi chips on a full-body worn AI-on-skin will be infeasible. This led us to choose the open-source Shenjing SNN accelerator as our AI-on-skin compute engine.

Switching between diverse applications. Reconfiguring an AI-on-skin platform for different applications will be one of the key requirements of future wearable artificial skin interfaces. We measured the time taken to reconfigure AI-on-skin across two different applications, the 3-patch stroke rehabilitation and badminton coaching applications, through a USB to UART programming interface, and it took 45 seconds, essentially the time to load the bitstream onto a FPGA. Future AI-on-skin hardware prototypes can be modified to enable wireless reconfiguration using a master device such as a phone or smart watch, further easing reconfiguration.

10 Conclusion

In this paper, we have motivated the need for on-body AI-on-skin - AI computing that is integrated with artificial skins to achieve skin sensing, compute and response in real time, at ultra-low power. In the future, with developments in wearable displays on glasses, wristbands, sensors embedded into textiles, AI-on-skin's AI accelerator chips can be mounted directly on artificial skins, connected across the entire body through conductive threads. We envision AI-on-skin opening up the possibilities of wearable skins, enabling continuous monitoring of our health and wellness, augmenting and interfacing touch, one of our most critical senses, with computing.

References

[1] Chris Harrison, Desney Tan, and Dan Morris. "Skinput: appropriating the body as an input surface". In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2010, pp. 453–462.

[2] Aditya Shekhar Nittala et al. "Multi-touch skin: A thin and flexible multi-touch sensor for on-skin input". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018, pp. 1–12.

[3] Martin Weigel et al. "Skinmarks: Enabling interactions on body landmarks using conformal skin electronics". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2017, pp. 3095–3105.

[4] Binbin Ying et al. "An ambient-stable and stretchable ionic skin with multimodal sensation". In: *Materials Horizons* 7.2 (2020), pp. 477–488.

[5] *Skin-like sensors bring a human touch to wearable tech*. Jan. 2020. URL: https://tectales.com/wearables-sensors/skin-like-sensors-bring-a-human-touch-to-wearable-tech.html.

[6] Harshal A Sonar et al. "Closed-loop haptic feedback control using a self-sensing soft pneumatic actuator skin". In: *Soft robotics* 7.1 (2020), pp. 22–29.

[7] Wang Wei Lee et al. "A neuro-inspired artificial peripheral nervous system for scalable electronic skins". In: *Science Robotics* 4.32 (2019), eaax2198.

[8] Xinge Yu et al. "Skin-integrated wireless haptic interfaces for virtual and augmented reality". In: *Nature* 575.7783 (2019), pp. 473–479.

[9] Martin Weigel et al. "Iskin: flexible, stretchable and visually customizable on-body touch sensors for mobile computing". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2015, pp. 2991–3000.

[10] Hsin-Liu Kao et al. "Skinmorph: texture-tunable on-skin interface through thin, programmable gel". In: *Proceedings of the 2018 ACM International Symposium on Wearable Computers*. 2018, pp. 196–203.

[11] AK Dąbrowska et al. "Materials used to simulate physical properties of human skin". In: *Skin Research and Technology* 22.1 (2016), pp. 3–14.

[12] Chris Larson et al. "A deformable interface for human touch recognition using stretchable carbon nanotube dielectric elastomer sensors and deep neural networks". In: *Soft robotics* 6.5 (2019), pp. 611–620.

[13] Subramanian Sundaram et al. "Learning the signatures of the human grasp using a scalable tactile glove". In: *Nature* 569.7758 (2019), pp. 698–702.

[14] Gabor Soter et al. "Bodily aware soft robots: integration of proprioceptive and exteroceptive sensors". In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 2448–2453.

[15] Dooyoung Kim et al. "Deep full-body motion network for a soft wearable motion sensing suit". In: *IEEE/ASME Transactions on Mechatronics* 24.1 (2018), pp. 56–66.

[16] GY Liu et al. "Smart electronic skin having gesture recognition function by LSTM neural network". In: *Applied Physics Letters* 113.8 (2018), p. 084102.

[17] Yujia Zhang and Tiger H Tao. "A Bioinspired Wireless Epidermal Photoreceptor for Artificial Skin Vision". In: *Advanced Functional Materials* 30.22 (2020), p. 2000381.

[18] Kyun Kyu Kim et al. "A deep-learned skin sensor decoding the epicentral human motions". In: *Nature Communications* 11.1 (2020), pp. 1–8.

[19] Hsin-Liu Kao et al. "DuoSkin: rapidly prototyping on-skin user interfaces using skin-friendly materials". In: *Proceedings of the 2016 ACM International Symposium on Wearable Computers*. 2016, pp. 16–23.

[20] Hsin-Liu Cindy Kao, Abdelkareem Bedri, and Kent Lyons. "SkinWire: Fabricating a Self-Contained On-Skin PCB for the Hand". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.3 (2018), pp. 1–23.

[21] Xi Duan, Sebastien Taurand, and Manuchehr Soleimani. "Artificial skin through super-sensing method and electrical impedance data from conductive fabric with aid of deep learning". In: *Scientific reports* 9.1 (2019), pp. 1–11.

[22] Keneth Iceland Kasozi et al. "A study on visual, audio and tactile reaction time among medical students at Kampala International University in Uganda". In: *African health sciences* 18.3 (2018), pp. 828–836.

[23] *How Fast is Realtime? Human Perception and Technology*. Feb. 2015. URL: https://www.pubnub.com/blog/how-fast-is-realtime-human-perception-and-technology/.

[24] Marc Teyssier et al. "Skin-On Interfaces: A Bio-Driven Approach for Artificial Skin Design to Cover Interactive Devices". In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. UIST '19. New Orleans, LA, USA: Association for Computing Machinery, 2019, pp. 307–322. ISBN: 9781450368162. DOI: 10.1145/3332165.3347943. URL: https://doi-org.libproxy1.nus.edu.sg/10.1145/3332165.3347943.

[25] *CC2650 sensortag user's guide - texas instruments wiki*. May 2014. URL: https://processors.wiki.ti.com/index.php/CC2650_SensorTag_User's_Guide.

[26] Eduardo Garcia-Espinosa et al. "Power Consumption Analysis of Bluetooth Low Energy Commercial Products and Their Implications for IoT Applications". In: *Electronics* 7.12 (2018), p. 386.

[27] *Piconet*. Aug. 2020. URL: https://en.wikipedia.org/wiki/Piconet.

[28] Vishal Varun Tipparaju et al. "Mitigation of Data Packet Loss in Bluetooth Low Energy-Based Wearable Healthcare Ecosystem". In: *Biosensors* 11.10 (2021), p. 350.

[29] Ananta Narayanan Balaji and Li-Shiuan Peh. "AI-on-Skin: Enabling On-Body AI Inference for Wearable Artificial Skin Interfaces". In: *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems* ***(Late Breaking work)***. CHI EA '21. Yokohama, Japan: Association for Computing Machinery, 2021. ISBN: 9781450380959. DOI: 10.1145/3411763.3451689. URL: https://doi.org/10.1145/3411763.3451689.

[30] *Apple A12 bionic GPU*. Sept. 2018. URL: https://www.notebookcheck.net/Apple-A12-Bionic-GPU.331520.0.html.

[31] Bo Wang et al. "Shenjing: A low power reconfigurable neuromorphic accelerator with partial-sum and spike networks-on-chip". In: *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE. 2020, pp. 240–245.

[32] David Fergenson et al. *Cmod A7: Breadboardable Artix-7 FPGA Module*. 2018. URL: https://store.digilentinc.com/cmod-a7-breadboardable-artix-7-fpga-module/.

[33] Norman P Jouppi et al. "In-datacenter performance analysis of a tensor processing unit". In: *Proceedings of the 44th Annual International Symposium on Computer Architecture*. 2017, pp. 1–12.

[34] *Apple A12*. Oct. 2020. URL: https://en.wikipedia.org/wiki/Apple_A12.

[35] *Shenjing RTL for AI acceleration*. Feb. 2020. URL: https://github.com/Angela-WangBo/Shenjing-RTL.

[36] *GPIO toggling speeds*. Feb. 2015. URL: https://codeandlife.com/2012/07/03/benchmarking-raspberry-pi-gpio-speed/.

[37] *DFRobot Beetle BLE - The Smallest Board Based on Arduino Uno with Bluetooth 4.0*. Dec. 2016. URL: https://www.dfrobot.com/product-1259.html.

[38] Hsin-Liu Kao et al. "NailO: fingernails as an input surface". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2015, pp. 3015–3018.

[39] Patrice Voss et al. "Dynamic brains and the changing rules of neuroplasticity: implications for learning and recovery". In: *Frontiers in psychology* 8 (2017), p. 1657.

[40] Kyle Wilson et al. "Real-time quantitative performance feedback during strength exercise improves motivation, competitiveness, mood, and performance". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 61 (Sept. 2017), pp. 1546–1550. DOI: 10.1177/1541931213601750.

[41] Pradeep Kumar et al. "Virtual trainer with real-time feedback using kinect sensor". In: July 2017, pp. 1–5. DOI: 10.1109/TENCONSpring.2017.8070063.

[42] Tasbolat Taunyazoz et al. "Event-Driven Visual-Tactile Sensing and Learning for Robots". In: *Proceedings of Robotics: Science and Systems*. July 2020.

[43] Nur Al-huda Hamdan et al. "Springlets: Expressive, Flexible and Silent On-Skin Tactile Interfaces". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for

Computing Machinery, 2019, pp. 1–14. ISBN: 9781450359702. DOI: 10.1145/3290605.3300718. URL: https://doi.org/10.1145/3290605.3300718.

[44] Anusha Withana, Daniel Groeger, and Jürgen Steimle. "Tacttoo: A Thin and Feel-Through Tattoo for On-Skin Tactile Output". In: *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. UIST '18. Berlin, Germany: Association for Computing Machinery, 2018, pp. 365–378. ISBN: 9781450359481. DOI: 10.1145/3242587.3242645. URL: https://doi.org/10.1145/3242587.3242645.

[45] Ruojia Sun et al. "Weaving a Second Skin: Exploring Opportunities for Crafting On-Skin Interfaces Through Weaving". In: *Proceedings of the 2020 ACM Designing Interactive Systems Conference*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 365–377. ISBN: 9781450369749. URL: https://doi.org/10.1145/3357236.3395548.

[46] Clint Zeagler. "Where to Wear It: Functional, Technical, and Social Considerations in on-Body Location for Wearable Technology 20 Years of Designing for Wearability". In: *Proceedings of the 2017 ACM International Symposium on Wearable Computers*. ISWC '17. Maui, Hawaii: Association for Computing Machinery, 2017, pp. 150–157. ISBN: 9781450351881. DOI: 10.1145/3123021.3123042. URL: https://doi.org/10.1145/3123021.3123042.

[47] Gordon Cheng et al. "A Comprehensive Realization of Robot Skin: Sensors, Sensing, Control, and Applications". In: *Proceedings of the IEEE* PP (Aug. 2019), pp. 1–18. DOI: 10.1109/JPROC.2019.2933348.

[48] Ronghui Wu et al. "All-textile electronic skin enabled by highly elastic spacer fabric and conductive fibers". In: *ACS applied materials & interfaces* 11.36 (2019), pp. 33336–33346.

[49] Irmandy Wicaksono et al. "A tailored, electronic textile conformable suit for large-scale spatiotemporal physiological sensing in vivo". In: *npj Flexible Electronics* 4.1 (2020), pp. 1–13.

[50] Sungwoo Chun et al. "Bioinspired Hairy Skin Electronics for Detecting the Direction and Incident Angle of Airflow". In: *ACS applied materials & interfaces* 11.14 (2019), pp. 13608–13615.

[51] Hyunjae Lee et al. "Wearable/disposable sweat-based glucose monitoring device with multistage transdermal drug delivery module". In: *Science advances* 3.3 (2017), e1601314.

[52] V.A.T. Dam, M.A.G. Zevenbergen, and R. van Schaijk. "Flexible Chloride Sensor for Sweat Analysis". In: *Procedia Engineering* 120 (2015). Eurosensors 2015, pp. 237–240. ISSN: 1877-7058. DOI: https://doi.org/10.1016/j.proeng.2015.08.588. URL: http://www.sciencedirect.com/science/article/pii/S1877705815022511.

[53] E Scarpa et al. "Wearable piezoelectric mass sensor based on pH sensitive hydrogels for sweat pH monitoring". In: *Scientific reports* 10.1 (2020), pp. 1–10.

[54] *The ReStore™ Soft Exo-Suit*. Jan. 2020. URL: https://rewalk.com/restore-exo-suit/.

[55] Aditya Shekhar Nittala et al. "PhysioSkin: Rapid Fabrication of Skin-Conformal Physiological Interfaces". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1–10. ISBN: 9781450367080. URL: https://doi.org/10.1145/3313831.3376366.

[56] Amir Yazdanbakhsh et al. "An Evaluation of Edge TPU Accelerators for Convolutional Neural Networks". In: *CoRR* abs/2102.10423 (2021). arXiv: 2102.10423. URL: https://arxiv.org/abs/2102.10423.

[57] Dharmendra S Modha. "Introducing a brain-inspired computer". In: ().

[58] Mike Davies et al. "Loihi: A neuromorphic manycore processor with on-chip learning". In: *IEEE Micro* 38.1 (2018), pp. 82–99.