# ArmPad: Transforming Forearms into Interaction Interfaces with Smartwatches

Zhenyu Yang, Qiang Yang, Zhidan Liu, Zhenjiang Li, Yongpan Zou, and Kaishun Wu, *Fellow, IEEE*

***Abstract*—With the rapid development of new smart devices, such as smart home appliances and VR/AR equipment, there is an increasing demand for novel interaction methods. However, many existing interaction methods require external devices, are unintuitive, and demand substantial user learning effort. To fill this gap, we propose ArmPad, a system that leverages the smartwatch's built-in IMU to enable *multidimensional input* on the user's forearm. Methodologically, ArmPad is explicitly designed to address three core research challenges in forearm-based interaction. First, to resolve the inherent feature conflicts between discrete gesture recognition and continuous distance estimation, we propose a multi-task learning framework with a dynamic gating mechanism for cross-task synergy. Second, to tackle the physical limitation of rapid vibration attenuation across the forearm, we introduce a cross-device guidance strategy that incorporates high-fidelity fingertip knowledge during the training phase. Finally, to ensure robust generalization across diverse populations, we develop task-specific data augmentation and a lightweight user registration mechanism to effectively mitigate physiological variances. Experiments on 20 subjects demonstrate that ArmPad achieves an accuracy of 92.51% on nine gestures and a Mean Absolute Error of 1.53 $cm$ for sliding distance in cross-user settings. Extensive robustness evaluations and case studies further confirm the system's stability and usability under diverse real-world conditions.**

***Index Terms*—Inertial measurement unit, multi-task learning, gesture recognition, on-skin interaction.**

## I. INTRODUCTION

IN recent years, the rapid advancement of smart devices, such as smart home appliances, VR/AR systems, and motion-sensing games, has significantly enriched interaction modalities. For instance, smartwatches are typically operated via small touchscreens [1], VR/AR systems utilize handheld controllers [2], and smart home appliances are often controlled through remote controls or voice commands [3]. Nevertheless, these interfaces often disrupt the continuity of interaction due to physical encumbrances (e.g., handheld devices) or environmental dependencies (e.g., quiet spaces), failing to meet the demand for ubiquitous and intuitive user experiences. To address these limitations, extensive research has explored gesture recognition based on computer vision [4], dedicated sensors (e.g., sEMG) [5], and wireless signals [6]. While promising, these approaches are often hampered by high computational demands, susceptibility to environmental interference (e.g., lighting, occlusion), or the requirement for specialized, costly hardware.

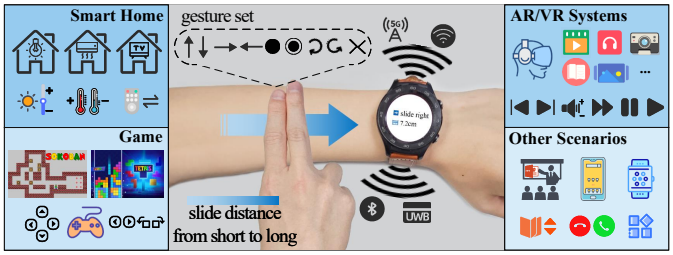



Fig. 1: Core concepts and application scenarios of ArmPad.

Recent works have explored gesture recognition using commercial off-the-shelf (COTS) smartwatches, which eliminates the need for additional hardware and brings notable advantages such as convenience and easy deployment. Some prior work leverages cameras [7], [8] or microphones [9], [10] on smartwatches for gesture recognition; however, these approaches raise privacy concerns and are sensitive to environmental conditions, such as lighting and noise. ViBand [11] and Taprint [12] rely on custom hardware modifications on smartwatches to enable higher accelerometer sampling rates for gesture recognition. Some work [13], [14], [15] recognizes tapping gestures at predefined locations, which introduces a learning and memorization burden. More importantly, they predominantly focus on recognizing discrete, simple gestures and critically lack the capability for continuous, fine-grained control necessary for nuanced interactions (e.g., controlling intensity or scale).

To address these limitations, we propose ArmPad (Fig. 1), a novel, non-intrusive smartwatch-based system that simultaneously supports both gesture classification and continuous distance regression, thereby enabling a *multidimensional input* modality and flexible user interactions. Specifically, ArmPad leverages the built-in IMU to capture vibration signals from finger-arm friction, modeling their spatiotemporal dynamics

This work was supported in part by National Natural Science Foundations of China under Grant 62572416 and the Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things under Grant 2023B1212010007. (*Corresponding author: Zhidan Liu.)*

Zhenyu Yang and Zhidan Liu are with INTR Thrust, System Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China, 510000. (E-mails: zyang483@connect.hkust-gz.edu.cn, zhidanliu@hkust-gz.edu.cn)

Qiang Yang is with the Department of Computer Science and Technology, University of Cambridge, CB2 1TN Cambridge, U.K. (e-mail: qiang.yang@cl.cam.ac.uk)

Zhenjiang Li is with Department of Computer Science, City University of Hong Kong, Hong Kong, China, 999077. (e-mail: zhenjiang.li@cityu.edu.hk)

Yongpan Zou is with College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China, 518060. (E-mail:yongpan@szu.edu.cn)

Kaishun Wu is with DSA Thrust and IoT Thrust, Information Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China, 510000. (E-mail: wuks@hkust-gz.edu.cn)

to achieve concurrent gesture recognition and sliding distance estimation. ArmPad can thus enable a variety of novel applications, including: (1) Smart Home, for seamless device control; (2) VR/AR and Gaming, for immersive media and game control; and (3) General UI, for daily productivity (e.g., answering calls or flipping slides). Taking Smart Home as an example, a single rightward slide can be mapped to a specific command, such as increasing brightness, where the sliding distance concurrently determines the precise adjustment scale. By reusing ubiquitous hardware, ArmPad provides a spontaneous adjustment channel while circumventing the discomfort associated with additional device layers.

However, realizing ArmPad as a robust interaction interface requires addressing three research questions. Firstly, ArmPad must simultaneously perform gesture classification and sliding distance regression. These two tasks impose divergent requirements on feature extraction: while classification relies on discrete and high-level representations, regression necessitates high-fidelity temporal features. This leads to the first research question (RQ1): *How can a unified model effectively balance these distinct feature requirements to achieve task synergy?* To address this, we design a multi-expert architecture with a dynamic gating mechanism that reconciles feature conflicts and ensures positive synergy between tasks (see Section IV-C). Beyond these feature-level conflicts, the physical sensing environment poses a vibration attenuation bottleneck; signals attenuate rapidly across the forearm, making it difficult for distal sensors to capture high-fidelity signals. Thus, we investigate RQ2: *How can we extract and model these subtle signals from noisy IMU data to achieve precise distance estimation?* In response, we adopt a dual-stream teacher-student network that leverages high-fidelity finger IMU data to guide the smartwatch's representation learning (see Section IV-D). Finally, the system must bridge the physiological and behavioral domain gap inherent in real-world deployment. Individual differences in skin properties and interaction habits cause significant distribution shifts in vibration signals; such shifts typically lead to performance degradation when a pre-trained model encounters a new user. This prompts the third research question (RQ3): *How can the system maintain high performance across diverse users with minimal overhead?* We mitigate this with a task-specific data augmentation strategy and a lightweight registration process to enhance cross-user personalization and robustness (see Section IV-E and IV-F).

The ArmPad system was implemented on two commercial smartwatches and evaluated through a user study with 20 participants of diverse ages and genders. Comprehensive experiments validated the effectiveness of the system across different scenarios. We further assessed resource consumption and conducted three types of user studies to evaluate usability and user experience. The main contributions of this work are summarized as follows:

- We propose ArmPad, a natural and intuitive HCI system that leverages the IMU embedded in COTS smartwatches to offer multidimensional input while posing no privacy risks and requiring no additional memorization.
- To address key implementation challenges, we introduced four innovative designs, including a multi-expert framework with dynamic gating to resolve feature conflicts, a dual-stream teacher-student module to stabilize weak signals, a two-level task-specific data augmentation strategy and registration mechanism to enhance robustness.
- Evaluation shows ArmPad achieves 92.51% accuracy for nine gesture classes and a 1.53 $cm$ error in distance estimation under cross-user settings. With minimal new-user calibration, accuracy rises to 96.99%. We also conducted extensive experiments and user studies, which confirmed the system's robustness and user-friendliness.

## II. RELATED WORK

### A. Hand Gesture Recognition

**Vision-based methods** typically use cameras to capture gesture execution through various visual modalities. For instance, FingerTrak [16] utilizes wrist-mounted thermal cameras for continuous 3-D finger tracking, and FaceSight [4] integrates an infrared camera on AR glasses for recognizing gestures near the face. Nevertheless, such approaches are often constrained by privacy issues, environmental dependency, and computational burdens that challenge real-time performance on edge devices. **Sensor-based methods** rely on specialized hardware to capture detailed physiological or physical signals, encompassing techniques like surface electromyography [5], force myography [17], sonomyography [18], and specialized wearable devices such as sensorized gloves [19]. For instance, Kim et al. [20] leveraged an RCE-DTW hybrid approach for high-precision, real-time 3D handwriting recognition. Zhang et al. [21] utilize body-part correlations and inertial sensing to facilitate training-free tracking of multi-scale gestures. However, these solutions often suffer from high cost, poor user comfort, and complex deployment, severely hindering their practicality and widespread adoption. Researchers have also explored **wireless sensing-based methods** for device-free gesture recognition. RF-based methods (e.g., Wi-Fi [22] and mmWave [23]) recognize gestures by analyzing signal features such as CSI and Doppler effects, whereas ultrasound-based approaches leverage acoustic reflections [6]. However, wireless solutions are sensitive to environmental interference, often require fixed deployments, and offer limited flexibility.

### B. Novel Interactions with Smartwatches

With the rapid evolution of wearable technology, COTS smartwatches have established a foundation for all-day, multi-scenario gesture interaction systems. Several studies have investigated the use of tapping gestures [13], [14], [12]. For instance, TapSkin [13] combines inertial sensors and a microphone to identify up to 11 tapping gestures on the hand dorsum. iDial [14] utilizes IMU and microphone signals to enable virtual keypad input through tapping on the back of the hand, while ViWatch [15] employs an unsupervised Siamese adversarial approach to facilitate robust finger knuckles interaction across diverse deployment environments. These studies typically divide the valid input interface into multiple smaller predefined areas, with each area corresponding to a specific logical operation or character input. This approach necessitates users to memorize and visually track these landmarks, while

TABLE I: Compared with existing smartwatch-based interaction methods, ArmPad offers several key advantages across technical implementation, user experience, and performance.

| System Name | Sensors Usage | No Privacy Risks | No Hardware Modification | Interaction Area | Landmark Free | **Sliding Distance Estimation** | Accuracy (%) Cross-User* |
|---|---|---|---|---|---|---|---|
| IPAND [9] | Mic | × | ✓ | hand-back | ✓ | × | 82.75 |
| Jannat et al. [7] | Camera | × | ✓ | mid-air | ✓ | × | 93.27 |
| Mudra [8] | IMU+Camera | × | ✓ | single-hand | ✓ | × | - |
| iDial [14] | IMU+Mic | × | ✓ | hand-back | × | × | - |
| GestEar [10] | IMU+Mic | × | ✓ | hand | ✓ | × | 97.2 |
| TapSkin [13] | IMU+Mic | × | ✓ | hand-back | × | × | 81.26 |
| Serendipity [24] | IMU | ✓ | ✓ | single-hand | ✓ | × | - |
| ViWatch [15] | IMU | ✓ | ✓ | hand-back/single-hand | × | × | 90 |
| ViBand [11] | IMU | ✓ | × | forearm/hand | ✓ | × | - |
| **ArmPad (Ours)** | IMU | ✓ | ✓ | forearm | ✓ | ✓ | 92.51 |

* All accuracies are reported under the **zero-shot cross-user** setting; "-" indicates that cross-user results were not provided in the original work.

the constrained size of the input interface limits the number and diversity of gestures. In contrast, ArmPad removes the cognitive and visual demands associated with predefined landmarks and provides a broader, more intuitive range of gestures, enabling a more natural and flexible interaction experience.

Other studies based on smartwatches have investigated alternative sensors and methods for gesture recognition [25], Serendipity [24] utilizes the built-in motion sensors to recognize five one-handed gestures, achieving an F1 score of 87%. GestEar [10] fuses audio and motion data to recognize sound-producing gestures like snaps, taps, and claps. ViBand [11] leverages high-frequency accelerometer sampling at 4 kHz on smartwatches to support applications such as gesture recognition and object detection. Jannat et al. [7] employ a camera on a COTS smartwatch to recognize gestures both in mid-air and on the back of the hand, but this approach is restricted by high power consumption and environmental conditions such as lighting. IPAND [9] leverages passive acoustic sensing to support four types of multi-finger gestures. However, it remains vulnerable to noise interference. Another line of research extends the sensing capability by employing multiple smart devices. Lu et al. [26] utilize dual wrist-worn commercial devices to recognize fourteen distinct bimanual gestures. SmartPoser [27] utilizes UWB-IMU fusion across a smartphone and smartwatch to provide robust, high-fidelity arm pose estimation for consumer applications. However, these multi-device approaches impose a higher hardware burden on users and require complex cross-device synchronization.

As shown in Table I, ArmPad significantly advances smartwatch interaction by utilizing only the IMU without modification, thus eliminating the power overhead and privacy risks of camera/mic-based systems. Unlike prior works requiring landmark memorization, our approach enhances flexibility and convenience by enabling intuitive multidimensional input through natural sliding motions. Furthermore, the cross-user without fine-tuning accuracy of ArmPad achieves performance comparable to or exceeding existing interactive systems.

## III. PRELIMINARY

### A. Multidimensional Gesture Design

In human-computer interaction systems, gestures with specific meanings are often performed to trigger corresponding logical operations. To align with users' habitual interactions and reduce cognitive load, we designed a set of gestures, as illustrated in Fig. 2. This design is informed by common tasks and shortcuts in contexts such as smart homes and AR/VR [28], while also considering user acceptance of on-skin gestures [29]. The gesture set comprises nine gestures, including tapping, sliding, and pinching, that enable users to execute a wide range of logical operations efficiently.

More importantly, we introduce the concept of *multidimensional gesture*. ArmPad not only recognizes various gesture categories but also estimates the length of the finger movement on the arm, i.e., sliding distance. This supplementary information expands interaction by allowing gestures to convey varying levels of operation intensity. By combining multiple information of gesture input, ArmPad supports more expressive and fine-grained interaction, enhancing intuitiveness across diverse applications.

### B. Fundamental Principle

The core principle of ArmPad is to convert on-skin gestures, such as sliding or tapping, into vibration signals generated by finger-arm friction, which propagate through the skin and tissue and are ultimately sensed by the IMU embedded in the smartwatch. These vibration patterns vary with gesture type and direction [30]. Specifically, the directional frictional force $\mathbf{F_f}$ produced by a gesture is given by:

$$\mathbf{F_f} = -\hat{\mathbf{v}}\mu N, \tag{1}$$

where $\hat{\mathbf{v}}$ is the direction of the gesture, $\mu$ is the coefficient of friction, and $N$ is the normal force applied by the finger. This force creates a vibration displacement $s(t)$, which can be modeled as a spring-damping system [31] with directional components, i.e.,

$$m\frac{d^2\mathbf{s(t)}}{dt^2} + c\frac{d\mathbf{s(t)}}{dt} + k\mathbf{s(t)} = \mathbf{F_f}, \tag{2}$$

where $\mathbf{s}(t) = [x(t), y(t), z(t)]^\top$ represents the vibration displacement in 3D space, $m$ is the effective mass of the tissue, $c$ is the damping coefficient, and $k$ is the stiffness of the skin-tissue system. Therefore, the gestures on the arm in different directions result in distinct vibration patterns along

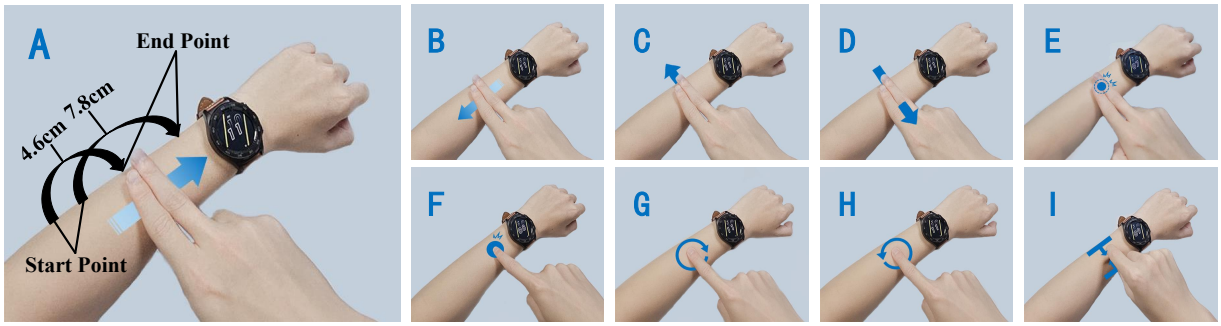
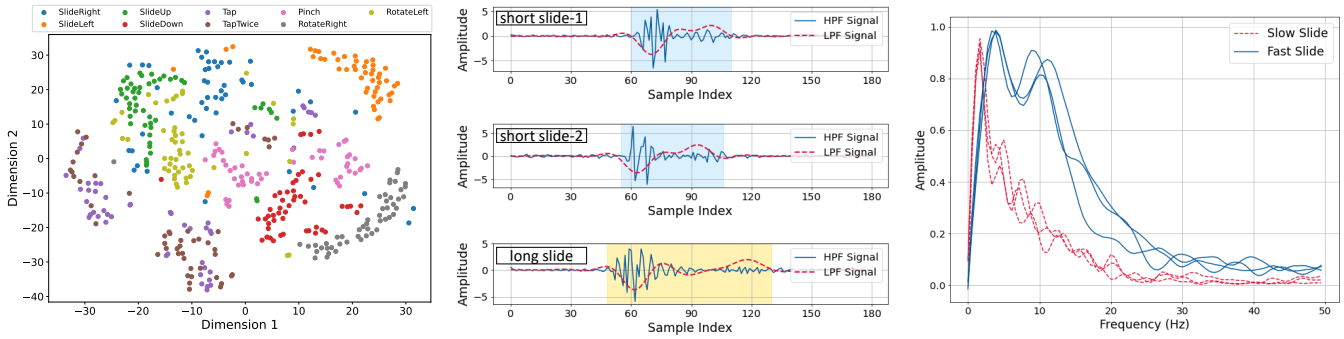



Fig. 2: On-arm gesture set involved in this study: (A) Rightward slide, (B) Leftward slide, (C) Upward slide, (D) Downward slide, (E) Tap twice, (F) Tap, (G) Clockwise slide, (H) Counterclockwise slide, (I) Pinch. The concept of sliding distance is also shown in (A), and it is worth noting that the sliding distance is meaningful only in the context of Leftward and Rightward slide gestures.




(a) (b) (c)

Fig. 3: The illustration of IMU signals from different gestures and sliding distances, where (a) is the t-SNE visualization of raw IMU signals for 9 gestures, (b) shows the IMU signal responses in the time domain, with blue highlighted intervals representing two short leftward slide samples exhibiting similar energy durations and a yellow highlighted interval corresponding to a long leftward slide sample with a longer energy duration, and (c) shows the IMU signal responses in the frequency domain for two different sliding speeds.

the $x, y, z$ axes, whose accelerations and rotations can be captured by IMU in the smartwatch. Based on this principle, ArmPad captures the spatiotemporal dynamics of gesture-induced vibrations and uses deep learning to recognize gesture categories and estimate sliding distances.

### C. Feasibility Studies

To assess gesture separability, we conducted a pilot study using a smartwatch collecting six-axis IMU data at 100 Hz. A predefined gesture set (Section III-A) was used, with 50 samples per gesture. The signals were denoised, gravity-corrected, and normalized, as detailed in Section IV-B. We applied t-SNE for dimensionality reduction and visualized the data in 2D space. As shown in Fig. 3a, different gestures exhibit clustering patterns, indicating the potential of six-axis IMU signals for gesture recognition. While non-negligible overlap between clusters is observed, this highlights the challenge of inter-class confusion and the need for robust models.

We further analyzed sliding distance distinguishability under two conditions. First, time-domain accelerometer signals from two sliding distances within the same gesture showed clear differences in signal duration (see Fig. 3b). We then explored speed effects by comparing frequency-domain energy for different speeds at the same distance (see Fig. 3c); faster gestures produced stronger high-frequency components. These results show that six-axis IMU signals can distinguish sliding distances, confirming the feasibility of distance estimation.

## IV. SYSTEM DESIGN

### A. System Overview

ArmPad utilizes the smartwatch's built-in IMU to record accelerometer and gyroscope signals during gesture execution, aiming to simultaneously perform gesture classification and sliding distance regression. As illustrated in Fig. 4, the system pipeline begins with preprocessing of the raw six-axis IMU signals. This phase involves a workflow that includes signal segmentation, noise reduction, gravity component removal, and data normalization. These steps mitigate signal drift and reduce various noise sources. Detailed preprocessing techniques are discussed in Section IV-B.

Following preprocessing, the data enter the feature extraction stage. We design a gated multi-expert architecture to extract high-dimensional feature representations for multi-task learning, alleviating negative mutual interference be-

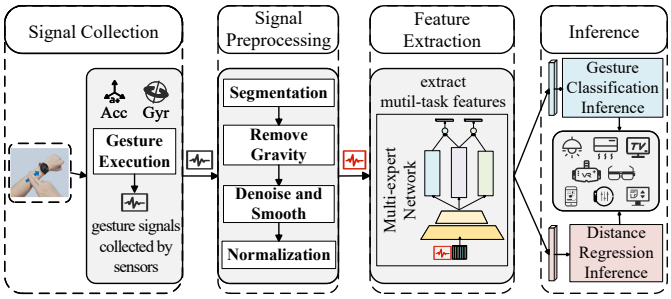



Fig. 4: The system overview.

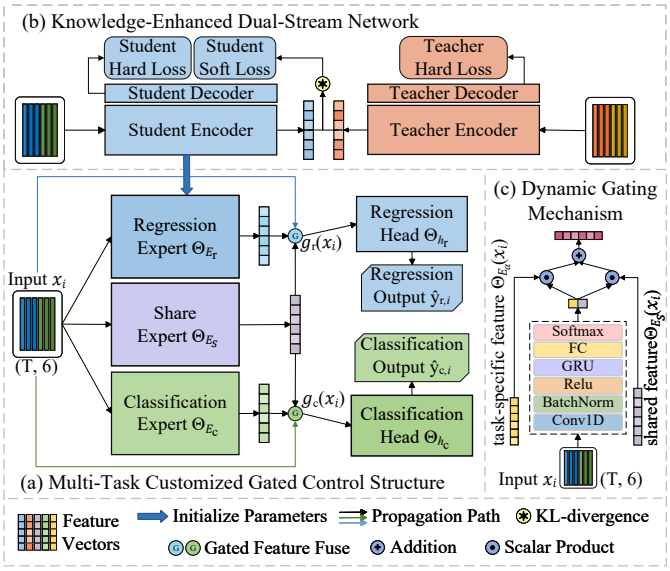



Fig. 5: Overview of the model.

tween the two objective tasks. Meanwhile, we introduce a knowledge-enhanced dual-stream network that utilizes multiple data sources to guide the model in learning effective representations. In addition, we adopt a task-specific two-level data augmentation strategy to increase data diversity and mitigate the generalization decline caused by variations in user behavior. We will discuss these module designs in detail in the following subsections.

Finally, the high-dimensional feature vectors extracted by the multi-expert extractor are fused via a gating mechanism, generating two intermediate vectors. The two vectors are directed to separate task-specific heads for gesture classification and distance regression.

### B. Signal Preprocessing

We perform data segmentation using a sliding window approach combined with dynamic thresholding based on signal energy. Specifically, the window length is set to 1.8 seconds with a step size of 125 ms to accommodate various gesture durations. The threshold, calculated as Threshold $= \mu + \delta \cdot \sigma^2$, where $\mu$ and $\sigma^2$ are the mean and variance, and $\delta$ controls sensitivity. This adaptive strategy helps filter out ambiguous gestures or borderline gestures. Once the threshold is exceeded, the peak of the absolute signal is located, and a window length segment centered on this peak is extracted.

To isolate the dynamic IMU signals associated with gesture-induced vibrations, we first apply a third-order Butterworth high-pass filter with 1 Hz cutoff frequency to remove the gravity component [32]. Additionally, the primary resonances of the hand-arm system were identified in the 10-40 Hz and 80-150 Hz frequency ranges [33]. As evidenced by our frequency domain analysis (see Fig. 3c), components above 40 Hz contain negligible information about gesture kinematics while introducing noise. This observation guides our implementation of a Butterworth low-pass filter with a 42 Hz cutoff frequency to eliminate high-frequency noise unrelated to target gesture motion. To further refine the signal and reduce small-scale noise, the data are then smoothed with a Savitzky-Golay filter [34], which has a window length of 7 and a polynomial order of 2. Finally, to eliminate inter-sample scale differences, we apply Z-score normalization to the denoised signals, transforming the data to have a mean of 0 and a standard deviation of 1. This step enhances model robustness across different users and conditions.

### C. Gated Multi-expert Network: Dynamic Multi-Task Learning

To address the conflict between discrete classification and continuous regression features as formulated in RQ1, we propose a Gated Multi-expert Network. This architecture is specifically designed to reconcile these distinct feature requirements by enabling dynamic task-specific representation learning. After signal preprocessing, the data are fed into the model for training with dimensions $T \times 6$, where $T$ is the length of the time dimension, set to 180, consistent with the window size.

**Multi-Expert Network.** When handling gesture recognition and distance estimation tasks simultaneously, a shared parameter set often struggles to adapt to the characteristics of each task, resulting in poor performance balance. By introducing specialized experts, each expert can independently focus on its related task while sharing common knowledge, improving the model's performance in complex tasks or scenarios with diverse input features. Based on this, we drew inspiration from the Customized Gate Control (CGC) model [35] to design a gated multi-expert network. As shown in Fig. 5, the multi-expert network comprises three expert modules: a gesture classification expert, a distance regression expert, and a knowledge-sharing expert. Each task-specific expert module focuses on its respective task, thereby mitigating interference among tasks, while a knowledge-sharing expert retains shared knowledge. As illustrated in Fig. 6a, we employ a hybrid CNN-GRU backbone in each module for feature extraction: the CNN captures local signal features, while the GRU excels at modeling temporal dependencies, facilitating simultaneous classification and regression. Internally, the convolutional layers comprise two one-dimensional temporal convolutional modules, each followed by batch normalization, a ReLU activation, and max-pooling to reduce feature dimensions, alleviate computational cost, and enhance generalization. To prevent overfitting, a 0.2 dropout is applied before the single-

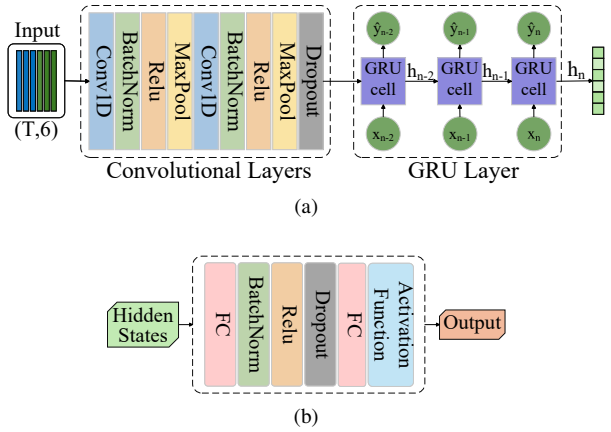



Fig. 6: Backbone architecture for the ArmPad, where (a) is feature extractor based on CNN-GRU hybrid model, and (b) is task head based on fully connected layers.

layer, unidirectional GRU with a 128-dimensional hidden state, whose final hidden vector serves as each module's output.

**Dynamic Gating Mechanism.** Each task-specific expert output is selectively integrated with the knowledge-sharing expert output through an input-based gating mechanism. This approach enables experts to focus on task-specific knowledge without interference from other tasks, while preserving shared information across tasks. As shown in Fig. 5c, the gating mechanism employs a hybrid model design, similar to the experts but more lightweight, ensuring accurate dynamic integration while maintaining computational efficiency. Specifically, the output after the gating mechanism is:

$$g_\alpha(x) = w_\alpha(x)\, s_\alpha(x), \tag{3}$$

where $x$ is the input, and $\alpha \in \{c, r\}$ represents the two task branches in ArmPad, i.e. classification and regression. $w_\alpha(x)$ is the weight matrix calculated through linear transformation and nonlinear activation functions, and $s_\alpha(x)$ is a selected matrix composed of the outputs from both the shared expert and the task-specific expert for task $\alpha$, defined as follows:

$$w_\alpha(x) = Softmax\left(\Theta_{w_\alpha}(x)\right) \tag{4}$$

$$s_\alpha(x) = stack\left[\Theta_{E_\alpha}(x), \Theta_{E_s}(x)\right] \tag{5}$$

Here, $\Theta_{w_\alpha}$ represents the gating network, where $\Theta_{E_\alpha}$ and $\Theta_{E_s}$ denote the task-specific expert and shared expert networks, respectively. Finally, the prediction result for task $\alpha$ is:

$$y_\alpha(x) = \Theta_{h_\alpha}\left(g_\alpha(x)\right), \tag{6}$$

where $\Theta_{h_\alpha}$ represents the task head network for task $\alpha$. As shown in Fig. 6b, each task head consists of two stacked fully connected layers, with a batch normalization layer, ReLU activation function, and dropout layer (p = 0.3) inserted between each pair of fully connected layers to enhance the model's nonlinear expressive capability and generalization ability.

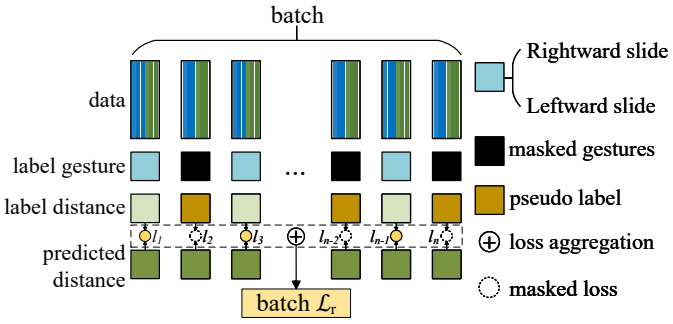



Fig. 7: Loss masking for irrelevant gesture samples.

**Loss Masking for Irrelevant Gestures.** During the training phase, Since training batches may contain gestures unrelated to the regression task, i.e., non-sliding gestures, we apply loss masking based on gesture labels to prevent interference. As shown in Fig. 7, pseudo labels are assigned to non-sliding gestures to match the input dimension, but their regression losses are masked during loss aggregation in each training batch. This ensures that only valid sliding gestures contribute to updates in the regression branch.

**Joint Classification and Regression Loss.** In our study, we employ Cross-Entropy and Mean Squared Error (MSE) as the target loss functions for classification and regression tasks, respectively. Given $N_1$ samples of input data $\mathcal{X} = \{x_i\}_{i=1}^{N_1}$ and task-specific labels $\mathcal{Y}_c = \{y_{c,i}\}_{i=1}^{N_1}$ and $\mathcal{Y}_r = \{y_{r,i}\}_{i=1}^{N_1}$, we mask out samples without distance information during the regression loss computation, as previously described. This results in $N_1'$ valid samples for the regression branch, where $N_1' \leq N_1$. The target equations for the loss functions are defined as follows:

$$\mathcal{L}_c(X, Y_c) = -\frac{1}{N_1}\sum_{i=1}^{N_1}\sum_{j=1}^{C} y_{c,i}\log(\hat{y}_{c,(i,j)}) \tag{7}$$

$$\mathcal{L}_r(X, Y_r) = \frac{1}{N_1'}\sum_{i=1}^{N_1'}\left(\hat{y}_{r,i} - y_{r,i}\right)^2 \tag{8}$$

Finally, the total loss $\mathcal{L}$ is a weighted sum of the two task losses, expressed as:

$$\mathcal{L}(X, Y_{c:r}) = \lambda\mathcal{L}_c(X, Y_c) + (1-\lambda)\,\mathcal{L}_r(X, Y_r), \tag{9}$$

where $\lambda$ is the weight hyperparameter, which we empirically set to 0.3.

### *D. Knowledge-Enhanced Dual-Stream Network: Distilling Effective Representations*

Skin vibrations attenuate faster than bone-conducted signals [36], making sliding gestures particularly susceptible to sensor noise. This signal decay directly leads to the modeling challenge formulated in RQ2. Unlike discrete tapping signals, sliding gestures exhibit continuous non-linear temporal patterns, posing greater challenges in signal modeling and distance estimation due to their dynamic strength and directional variations. Therefore, we introduce a high-fidelity

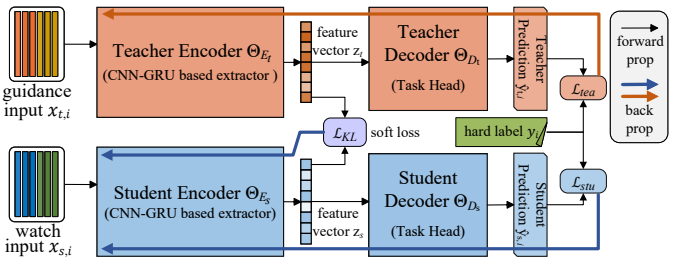



Fig. 8: The structure of knowledge-enhanced dual-stream network.

data source by mounting a dedicated IMU on the fingertip of the executing finger. Positioned closer to the point of contact, this sensor captures stronger, less attenuated signals with a higher signal-to-noise ratio and more comprehensive motion characteristics, offering richer supervision for learning effective representations. Crucially, the finger IMU data are used only during the training of the dual-stream network.

To fully leverage the finger IMU signals, we adopt a dual-stream architecture (Fig. 8), where the teacher branch learns discriminative representations of sliding distance from finger data and the student branch processes noisier smartwatch data. Within this teacher–student framework, the teacher branch transfers its robust distance-discrimination capability to the student branch via feature alignment, thereby improving modeling accuracy and generalization on smartwatch data [37]. To ensure alignment, we keep sampling details and preprocessing consistent for both finger and smartwatch data. We also design branch encoders with an identical structure, creating a shared feature space and matching the requirements of the regression expert. This design allows the student encoder parameters to be seamlessly transferred into the subsequent multi-task learning model after the knowledge-enhanced training stage. As a result, the knowledge distilled from finger data bolsters the stability and accuracy of the student branch when handling smartwatch data.

Training employs an efficient, parallel, end-to-end online approach, where both branches are trained synchronously from scratch. This strategy ensures simultaneous updates to both branches, leading to faster convergence and more efficient resource utilization [38]. Given $N_2$ samples from the teacher dataset $\mathcal{X}_T = \{x_{t,i}\}_{i=1}^{N_2}$ and the student dataset $\mathcal{X}_S = \{x_{s,i}\}_{i=1}^{N_2}$, with corresponding label sets denoted as $\mathcal{Y} = \{y_i\}_{i=1}^{N_2}$, we define the networks of the teacher network and student network as $\Theta_t$ and $\Theta_s$, respectively. For the regression task, the objective function for training networks $\Theta_t$ and $\Theta_s$ is defined as MSE between the predicted results and the true labels.

$$\mathcal{L}_{R_\beta} = \frac{1}{N_2}\sum_{i=1}^{N_2}(\hat{y}_{\beta,i} - y_i)^2, \tag{10}$$

where $\beta \in \{t, s\}$ represents the teacher branch or the student branch, and $\hat{y}_{\beta,i}$ is the predicted value, which is the output of $\Theta_\beta(x_{\beta,i})$.

As mentioned earlier, to enhance the recognition performance and generalization capability of the student branch, we leverage additional knowledge and training experience provided by the teacher branch through a posterior probability based on the intermediate vector. Specifically, for any given sample $x_\beta$, we apply the Softmax function to convert the intermediate hidden vector $\boldsymbol{z_\beta}$ generated by the encoder of network $\Theta_{E_\beta}$ into a probability $\boldsymbol{p_\beta}$. The computation is defined as follows:

$$p_{\beta,k}(x_\beta) = \frac{\exp\left(\boldsymbol{z_{\beta,k}}/T\right)}{\sum_{d=1}^{D}\exp\left(\boldsymbol{z_{\beta,d}}/T\right)}, \tag{11}$$

where $T$ represents a temperature parameter, which is set to 5 in our study. $D$ represents the length of the intermediate vector, set to 128 in our study. To quantify the degree of alignment between the outputs $\boldsymbol{p_s}$ and $\boldsymbol{p_t}$ from the two networks, we utilize the Kullback-Leibler (KL) divergence [39], defined as:

$$D_{KL}(\boldsymbol{p_s}||\boldsymbol{p_t}) = \sum_{i=1}^{N_2} p_s(x_{s,i}) \log \frac{p_s(x_{s,i})}{p_t(x_{t,i})} \tag{12}$$

Thus, the overall loss functions $\mathcal{L}_{tea}$, and $\mathcal{L}_{stu}$, for networks $\Theta_t$ and $\Theta_s$ are defined as:

$$L_{tea} = L_{R_t}, \tag{13}$$

$$\mathcal{L}_{stu} = \omega\mathcal{L}_{R_s} + (1-\omega)\, D_{KL}\left(\boldsymbol{p_s}||\boldsymbol{p_t}\right)\cdot T^2, \tag{14}$$

where $\omega$ is the weight for the loss function, and we empirically set it to 0.6.

After this stage, we save the parameters of the student branch encoder and use them to initialize the subsequent multi-task learning model. It should be emphasized that this auxiliary fingertip IMU is exclusive to the offline data-collection and training stage.

### *E. Task-Specific Data Augmentation*

Gesture execution varies across users due to individual physical characteristics (e.g., muscle mass, bone density) [40] and behavioral patterns (e.g., speed, force), causing distributional shifts that hinder generalization to unseen users. Motivated by the goal of ensuring cross-user robustness in RQ3, we propose a two-level task-specific data augmentation strategy, extending established methods for inertial data [41]. The first level includes four methods: (1) **jittering**, by injecting Gaussian noise $\mu \sim \mathcal{N}(0, 0.2^2)$; (2) **masking**, by randomly masking a continuous segment of $[15, 30]$ time steps; (3) **time shifting**, by shifting the signal $[5, 25]$ steps and padding with zeros; and (4) **random dropping**, by setting 20% of data points to zero. These methods introduce variability while preserving gesture semantics, thereby improving robustness to signal-level inconsistencies. They are applied uniformly across all gesture categories.

The second level simulates variations in execution strength and speed, encompassing two approaches: (1) **magnitude-warping**, achieved by three interpolation knots and a randomly generated scaling factor $m \sim \mathcal{N}(1, 0.3^2)$, where $m \in [0, 2]$, and (2) **Time-warping**, achieved by sampling segment-wise speed ratios from $\mathcal{U}(0.5, 2)$ over three intervals defined by

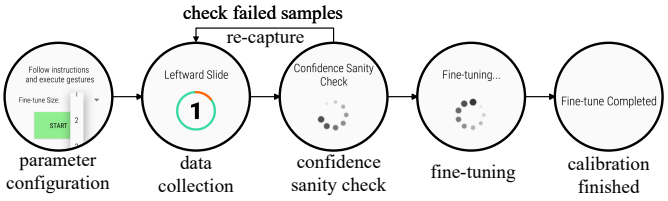



Fig. 9: New user registration and calibration workflow.

two interpolation knots. To maintain physical consistency for regression, Leftward and Rightward slide are excluded from augmentation to avoid disrupting critical semantic information. This two-level strategy improves robustness while maintaining consistency for distance estimation.

### F. New-user Registration and Model Calibration

Despite data augmentation, individual variances in gesture execution and physiology still induce data distribution shifts. To fulfill the objective of RQ3, we introduce a registration process that personalizes the model, enabling new users to achieve higher accuracy. During the registration phase (see Fig. 9), on-screen prompts guide users to perform each gesture category $n$ times. The value of $n$ reflects a trade-off between personalization and registration overhead.

After data collection, the model evaluates each gesture by computing a confidence score based on its current recognition capability. Gestures with scores above a 0.5 threshold are retained, while those below trigger re-capture prompts to ensure consistency. Valid samples are then used to fine-tune the model. As acquiring accurate distance ground truth for regression tasks typically requires specialized instrumentation, ArmPad in its current form emphasizes gesture recognition, where reliable supervision can be more readily obtained. Training details are provided in Section VI-D

## V. EXPERIMENTAL METHODOLOGY

### A. Implementation

We developed a dedicated Android application for ArmPad and deployed it on two smartwatches (HUAWEI Watch 2 and Samsung Galaxy Watch 7) for data collection and validation. Six-axis IMU data were recorded at 100 Hz using the Wear OS API, balancing compatibility and efficiency, as many wearables cap sampling at this frequency. Due to computational limitations, model training is performed offline. The model, implemented in PyTorch with 306K parameters, supports efficient deployment on smartwatches.

Training is conducted using the AdamW optimizer with a ReduceLROnPlateau scheduler over 100 epochs. The regression expert is initialized from the trained student encoder in the dual-stream network, while other modules use default initialization. A learning rate of 0.0001 is applied to the regression expert and 0.001 to the rest. Training runs on a workstation with 128 GB RAM, an AMD Ryzen 9 3900X (3.80 GHz), and an NVIDIA RTX 2070 GPU.

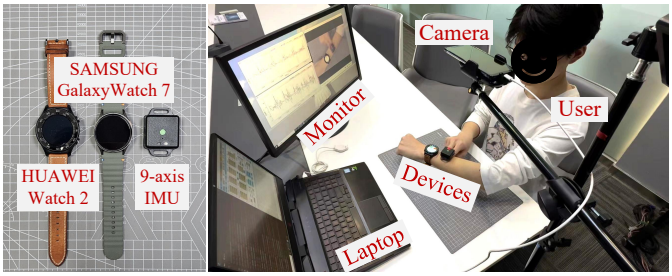



Fig. 10: Experimental equipment and environment.

### B. Data Collection

We recruited 20 participants (11 male, 9 female), ranging in age from 17 to 51 years (mean: 25.7) and in BMI from 16.6 to 29.1. All experimental procedures were approved by the Institutional Review Board.

The experimental setup is shown in Fig. 10. Participants first spent 10 minutes familiarizing themselves with the procedure, then wore the smartwatch on their preferred wrist with comfortable tightness and a natural hanging posture. Data were collected across three sessions, with 5–8 minute breaks to mitigate fatigue. Each session included at least 40 repetitions of Leftward and Rightward slide (covering various distances) and 10 repetitions of each remaining gesture, yielding 9,231 samples in total. A lightweight IMU sensor (9 $g$, ring-like, with no reported interference to movement) was attached to the executing finger. A vision-based algorithm using an overhead camera provided ground-truth distance labels.

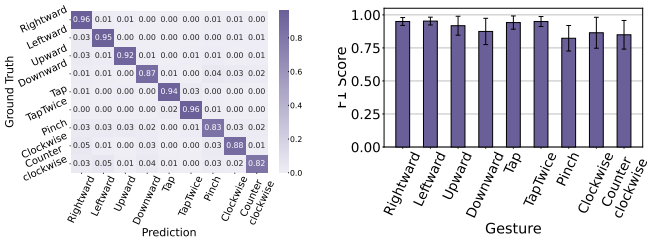
## VI. EVALUATION

This section presents a comprehensive evaluation of overall performance, ablation, fine-tuning results, robustness, and resource consumption. Metrics include accuracy and F1 score for classification, and MAE, RMSE, and MAPE for regression.

### A. Overall Performance

To evaluate the system's usability in realistic scenarios, particularly with previously unseen users, we adopt leave-one-out cross-validation (LOO-CV), using cross-user data as the test set in each round.

*1) Gesture Recognition Performance:* As shown in Fig. 11, our model achieved an average accuracy of 92.51% and an F1 score of 90.26%. The "Counterclockwise" gesture exhibited the lowest accuracy (82.45%) due to its prolonged execution and high variability, which caused confusion with "Leftward" slides. Individual results (Fig. 12) ranged from 98.1% (participant 19) to 86.1% (participant 8); for participant 8, intermittent skin contact during execution led to incomplete feature representations. Overall, ArmPad demonstrates robust zero-shot classification across nine gestures. To address challenges in complex gestures and cross-user variance, a calibration strategy using minimal new-user data is further detailed in Section VI-D.




(a) (b)

Fig. 11: Gesture classification results, where (a) is confusion matrix representing the average performance across 20 participants, and (b) is mean Macro-F1 scores for different gesture categories with error bars, indicating standard deviation.

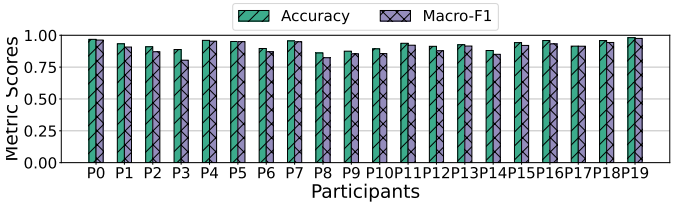



Fig. 12: Participant gesture classification performance.

*2) Distance Regression Performance:* As shown in Fig. 13, our model achieved an average MAE of 1.53 $cm$ and MAPE of 0.2 $cm$. Individual performance varied from 1.14 $cm$ (participant 0) to the maximum error in participant 6; video analysis attributed participant 6's lower accuracy to execution stuttering, which disrupted the model's temporal feature extraction. Regression scatters (Fig. 14) indicate that rightward slides generally yielded lower errors due to higher movement consistency, though individual preferences influenced direction-specific accuracy (see Section VII). Furthermore, errors scaled with sliding distance, likely due to noise accumulation during extended physical execution.

### B. Baseline Comparison

To evaluate ArmPad's effectiveness, we benchmarked it against ViBand [11] and ViWatch [15], representing traditional hand-crafted feature approaches and modern neural-network architectures, respectively. All models were evaluated under a consistent cross-user zero-shot protocol using identical datasets and preprocessing procedures.

As shown in Fig. 15, our method outperforms all baselines across all metrics. Notably, ViBand exhibits the lowest performance (e.g., 46.3% accuracy) as its SVM-based framework fails to extract discriminative features from standard 100 Hz signals, compared to its original 4 kHz requirement. Similarly, ViWatch shows a significant degradation in both tasks, primarily due to its inability to resolve feature conflicts between discrete recognition and continuous regression. The superior performance of our system can be attributed to its Gated Multi-expert Network, which effectively decouples tasks, and a Knowledge-Enhanced Dual-Stream Network that extracts robust representations from low-SNR skin vibrations.

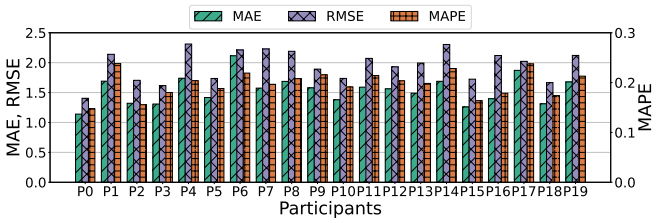
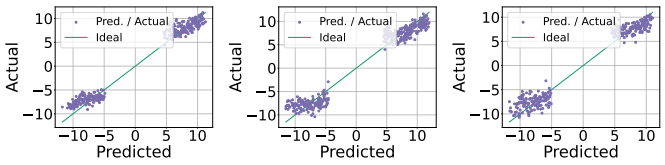



Fig. 13: Distance regression performance for each participant.




(a) (b) (c)

Fig. 14: Scatter plots of predicted vs. actual labels for the distance regression task. (a)–(c) present visualizations for participants 0, 3, and 18, respectively. Positive and negative values denote rightward and leftward sliding samples.

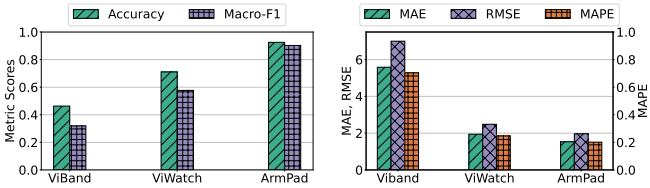
### C. Ablation Study

We conducted ablation studies to quantify the contribution of each module. All ablation variants are built on the backbone architecture shown in Fig. 6. Training is conducted using the AdamW optimizer with a ReduceLROnPlateau scheduler over 100 epochs. A learning rate of 0.0001 is applied to the regression expert and 0.001 to the rest. All variants were trained with identical hyperparameters to ensure fair comparison.

As shown in Table II, the *Full model* achieved the highest classification accuracy and the lowest distance regression error. Removing the data augmentation module (i.e., *w/o Aug*) leads to a decline in all indicators, highlighting its contribution to noise robustness and sample diversity. Notably, excluding the dual-stream component (i.e., *w/o DualS*) leads to a significant performance degradation in regression tasks, with MAE and MAPE increasing by 4.91% and 5.91%, respectively. Compared to the Full Model and the varient *w/o Aug+DualS*, the *Naive Model* (rely on a single shared backbon) shows the largest performance drop, a 2.42% decrease in accuracy and a 7.81% increase in MAE, highlighting the necessity of expert decoupling for multi-task learning.

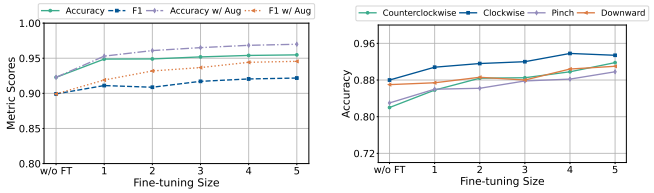
### D. Evaluation of Registration with Few-shot User Data

In practical use, individual differences can cause feature distributions to deviate from the training data. To improve adaptability to new users, we apply transfer learning by fine-tuning the trained model using a small set of user-specific samples, combined with data augmentation. Fine-tuning is performed offline for 30 epochs at a fixed learning rate of 0.0001 to balance efficiency and performance. To evaluate its effectiveness, we vary the number of samples per class (1–5) and compare results with a baseline model trained without fine-tuning (i.e., w/o FT).




(a) (b)

Fig. 15: Performance comparison between our proposed ArmPad and representative baselines.

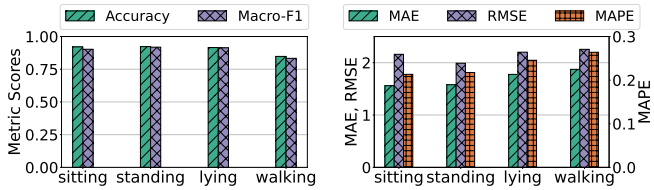



(a) (b)

Fig. 16: Fine-tuning results with varying data sizes. (a) Average performance across all gestures. (b) Gesture-level accuracy for representative challenging categories.

TABLE II: Results of the ablation study. ↓ indicates performance deterioration relative to the Full Model.

| Variants | Classification | | Regression | | |
|---|---|---|---|---|---|
| | Accuracy | F1 | MAE | RMSE | MAPE |
| **Full Model** | **0.9251** | **0.9026** | **1.5311** | **1.9632** | **0.2015** |
| w/o Aug | ↓ 1.11% | ↓ 1.60% | ↓ 1.35% | ↓ 7.10% | ↓ 2.13% |
| w/o DualS | ↓ 0.58% | ↓ 0.66% | ↓ 4.91% | ↓ 7.70% | ↓ 5.91% |
| w/o Aug+DualS | ↓ 1.78% | ↓ 2.08% | ↓ 5.15% | ↓ 10.24% | ↓ 7.64% |
| Naive Model | ↓ 2.42% | ↓ 2.45% | ↓ 7.81% | ↓ 12.85% | ↓ 8.88% |




(a) (b)

Fig. 17: Performance of two tasks across states.

As illustrated in Fig. 16a, the fine-tuned model demonstrates significant performance improvements over the baseline. Notably, with only three fine-tuning samples per category, the classification accuracy increases by 4.12% (reaching 96.51%), while the F1 score improves by 3.6% (reaching 93.67%). Even without the use of data augmentation, the accuracy and F1 score still achieve gains of 2.84% and 1.06%, respectively. Furthermore, Fig. 16b visualizes the performance evolution of four relatively challenging gesture categories, i.e., Counterclockwise, Pinch, Clockwise, and Downward. The results indicate a consistent and robust upward trend in accuracy as fine-tuning samples increases. For example, the Counterclockwise gesture increases by 7.9% with only three samples per class, and this gain further expands to 12.0% (reaching an accuracy of 91.8%) when the sample size is increased to five. These results highlight the system's sustained optimization potential, allowing users to effectively enhance recognition performance through personalized data enrollment. For the sliding distance regression task, we do not specifically fine-tune the regression branch, as obtaining actual sliding distances as ground-truth in real-world scenarios is impractical for new users.

### E. System Robustness

To evaluate the stability of ArmPad across various real-world usage scenarios, we designed and conducted a series of experiments. All the experimental results in this section were obtained using the model under a cross-user setting, i.e., without calibration on new user data. A total of 6 participants were involved in the robustness experiments, with each participant performing 20 repetitions of each gesture under each type of experiment in a single scenario.

*1) Impact of Different Body States:* As shown in Fig. 17, while sitting state yielded peak performance, walking induced a 7.86% drop in accuracy and a 0.3 $cm$ increase in MAE. These fluctuations likely result from gesture variability during dynamic movement. Nevertheless, the system maintained robust performance across all states, facilitated by preprocessing that mitigates noise from gravity and arm motion.

*2) Impact of Different Strengths:* Unlike touchscreens, on-skin interactions can vary substantially in applied force, influenced by factors such as emotional arousal. We classified gestures into three force levels, i.e., light, moderate, and heavy. As shown in Fig. 18, light gestures reduced classification accuracy by 4.3% and increased the MAE by 0.13 $cm$. In contrast, heavy gestures produced clearer feature patterns, maintaining a high classification accuracy of 93.11% with only a slight 0.03 $cm$ increase in MAE.

*3) Impact of Different Speeds:* We further examined the role of speed by instructing participants to perform gestures at slow, moderate, and fast speeds. As shown in Fig. 19, classification accuracy remained high at 91.72% under fast speeds, this could be because gestures are more clearer and coherent. However, the MAE in the regression increased by 0.26 $cm$, which may be attributed to signal instability. By contrast, slower gestures were more stable, producing a lower MAE of 1.59 $cm$. These results demonstrate that the system maintains robust performance across different execution speeds.

*4) Longitudinal Study:* To evaluate long-term robustness, we collected additional data over a four-week period (at intervals of 3 days, 1, 2, 3, and 4 weeks). As shown in Fig. 20, classification accuracy remained consistently high, while regression performance showed a slight decline after three weeks of disuse. Despite this, the system maintained an average accuracy of 91.04% and an MAE of 1.63 $cm$ at the four-week mark, demonstrating its stability under realistic conditions. Future work could employ continuous learning to further adapt to evolving user patterns [42].

*5) Impact of Sampling Rate:* We evaluated how the IMU sampling rate affects ArmPad's performance. We downsam-

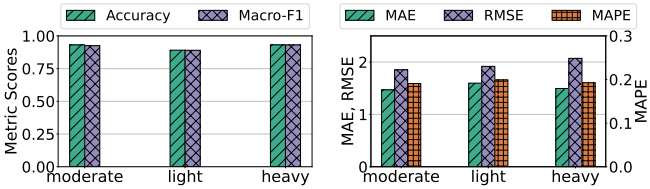



Fig. 18: Performance of two tasks across strengths.

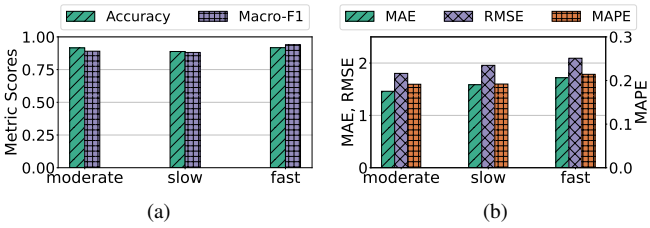



Fig. 19: Performance of two tasks across speeds.

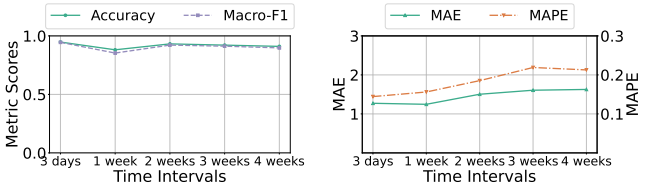



Fig. 20: Robustness evaluation of ArmPad across different time intervals in two tasks.

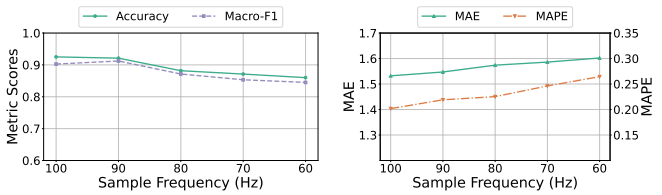



Fig. 21: Performance of two tasks across sampling rates

pled the input from 100 Hz to 60 Hz, maintaining consistent training configurations. As shown in Fig. 21, the performance of both tasks gradually degraded as the sampling rate decreased, with a more noticeable drop at 60 Hz relative to the 100 Hz baseline. Nonetheless, moderate reductions (e.g., 80–90 Hz) still provided acceptable recognition while lowering sensing and processing costs.

*6) Impact of Different Devices:* To evaluate generalizability across devices with different hardware configurations, we conducted additional testing on the Samsung Galaxy Watch 7 alongside the original HUAWEI Watch 2, using identical data collection protocols and processing pipelines. The results show that ArmPad achieved a gesture classification accuracy of 93.13% on the Galaxy Watch 7, with a comparable MAE of 1.61 $cm$. These results validate ArmPad's robustness and adaptability across different smartwatch platforms.

*7) Impact of Clothing Occlusion:* We further evaluated ArmPad under clothing occlusion by covering the forearm with cotton and polyester fabrics. Accuracy dropped by 2.4% and 2.6%, while MAE increased by 0.24 $cm$ and 0.26 $cm$, respectively. These findings suggest that classification remains robust by capturing global patterns, whereas regression is more vulnerable to clothing-induced damping and waveguide effects due to its reliance on fine-grained vibration modeling. We outline potential approaches to tackle the issue in Section VIII.

### F. Resource Consumption

To evaluate ArmPad's on-device performance, we conducted comprehensive measurements across two commercial smartwatches: HUAWEI Watch 2 (Snapdragon Wear 2100, 768MB RAM) and Samsung Galaxy Watch 7 (Exynos W1000, 2GB RAM). Our evaluation focused on three key metrics using Android Studio Profiler [43] and Battery Historian [44], all experiments executed the complete workflow on the smartwatch to reflect real-world operational conditions.

**(1) CPU and Memory Usage.** We measured ArmPad's resource consumption with all non-essential background processes disabled. During continuous operation, the two devices exhibited an average CPU usage increase of approximately 45% and 27%, while memory usage remained stable at 37MB (4.8%) and 54MB (2.6%). **(2) Power Consumption.** Additionally, we measured the time required to deplete 1% of the battery under two conditions: (i) a baseline "screen on only" state, and (ii) while running ArmPad continuously. According to Battery Historian, ArmPad reduced the time to deplete 1% battery by 157 s compared to the baseline, indicating the system's operational power consumption is merely 102 $mW$. **(3) System Latency Evaluation.** To evaluate ArmPad's latency, we measured its end-to-end response time on the Samsung Galaxy Watch 7, representative of recent smartwatch generations with improved processor optimization for deep learning. The response process consists of two main stages, data preprocessing and model inference, which average 18 $ms$ and 85 $ms$, respectively.

These results indicate that ArmPad operates efficiently without imposing significant load, preserving overall device performance and stability.

## VII. USER STUDY

To evaluate ArmPad's usability, we conducted a mixed-method study with 20 participants. The assessment included: (1) a gesture-wise assessment of user preferences and subjective ratings, (2) a System Usability Scale (SUS) evaluation, and (3) a workload evaluation based on the NASA Task Load Index (NASA-TLX). The questionnaires were administered following the completion of the entire interaction study, ensuring that participant ratings were based on their full experience with the watch-only system.

**Gesture-wise Preference Analysis.** We assessed user experience across five dimensions, e.g., intuitiveness, learnability,

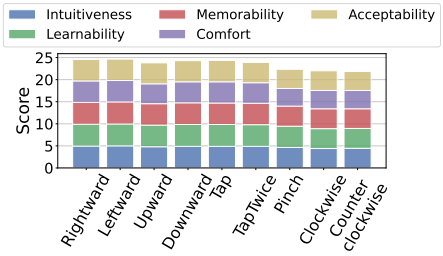



Fig. 22: User feedback scores for nine gestures. Each score is a composite of five evaluation aspects.

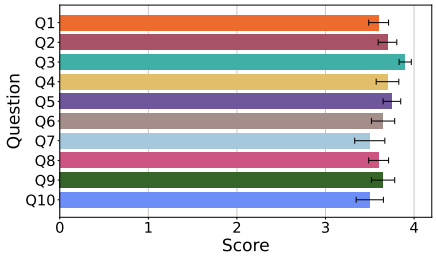



Fig. 23: Ten question scores of SUS evaluation, error bars represent standard error.

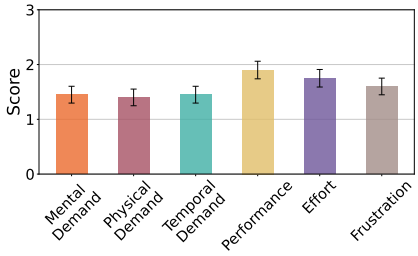



Fig. 24: Six dimensions scores of NASA-TLX evaluation, error bars represent standard error.

memorability, comfort, and social acceptability, via a 5-point Likert scale. As shown in Fig. 22, most gestures scored highly due to their intuitive mapping to physical actions. Although Pinch and rotation gestures (Clockwise/Counterclockwise) had slightly steeper learning curves, the overall feedback remained overwhelmingly positive (mean $23.5/25$), confirming the system's strong usability and social acceptance.

**SUS.** SUS is a widely adopted 10-item questionnaire used to evaluate a system's usability and user experience, with the full set of questions detailed in [45]. As shown in Fig. 23, ArmPad achieved a high SUS score of 91.38 (scored out of 100, computed as the sum of item scores multiplied by 2.5), indicating strong user satisfaction. Responses to Q2 $(3.65 \pm 0.13)$, Q5 $(3.65 \pm 0.13)$, and Q6 $(3.75 \pm 0.10)$ showed that users found the gesture design natural and appropriately simple. Items addressing ease of use and learnability (Q3: $3.60 \pm 0.11$, Q8: $3.90 \pm 0.07$, Q10: $3.60 \pm 0.11$) confirmed ArmPad as user-friendly and easy to learn, with participants also showing confidence in their ability to master the system (Q7: $3.70 \pm 0.13$). The slightly lower rating for Q4 $(3.50 \pm 0.17)$ suggested some users perceived a minor need for additional guidance or equipment during data collection.

**NASA-TLX.** We used a simplified NASA-TLX questionnaire [46] to measure perceived workload across six dimensions (Mental, Physical, Temporal Demands, Effort, Performance, and Frustration) on a 5-point scale (1=very low, 5=very high). ArmPad achieved a low overall workload rating (mean $1.6/5$), with notably low mental $(1.45 \pm 0.15)$, physical $(1.40 \pm 0.15)$, and temporal $(1.45 \pm 0.15)$ demands, confirming that gestures felt intuitive and straightforward. Effort $(1.75 \pm 0.16)$ and frustration $(1.60 \pm 0.15)$ ratings were slightly higher due to the initial adaptation required for novel on-skin gestures, which also likely contributed to the relatively higher performance score $(1.90 \pm 0.15$, inversed).

While these results confirm the system's comfort for short-term use, we acknowledge that long-term, high-frequency usage may require further evaluation, including fatigue and subjective load measures over extended periods.

## VIII. DISCUSSION AND FUTURE WORK

Our experimental results demonstrate that the system maintains stable recognition performance under various conditions. Nevertheless, several issues require further investigation.

*a) Wearing Thick Clothing:* ArmPad is primarily designed for indoor use (e.g., smart homes, AR/VR), and evaluation with thin clothing (see Section VI-E7) showed satisfactory performance. However, thick clothing may significantly degrade recognition accuracy due to stronger damping of skin vibrations. To mitigate this, future work will explore integrating friction-induced acoustic signals as a complementary modality, though this may increase computational cost. Additionally, the performance drop reflects a domain gap in training data; domain generalization techniques [47] offer a scalable solution by learning fabric-invariant gesture representations, enabling robust performance under unseen occlusion conditions.

*b) Gesture Set Expansion:* This study focused on nine touchscreen-like gestures designed to support basic interactions, which are sufficient for key tasks such as control, confirmation, and parameter adjustment. However, certain complex and path-varying gestures (e.g., counterclockwise slides) show slightly lower accuracy, primarily due to variability in user execution styles (e.g., differences in trajectory, movement amplitude, and alignment). Additionally, real-world gestures often involve more diverse directions and trajectories (e.g., oblique, curved, zigzag) that are not yet supported by the current system. As such, this work does not aim to cover all possible high-complexity gestures.

To address these limitations, we propose several potential extensions, including: 1) employing adversarial training to encourage the model to learn style-invariant features, making it more robust to individual execution styles, 2) implementing a weighted loss function that prioritizes high-variability 'hard samples' to better handle ambiguous and unstable gestures, and 3) applying style-targeted data augmentation during training to expose the model to more varied gesture trajectories. Additionally, one-shot learning [48] offers a promising path for efficient personalization and gesture set expansion.

*c) Potential of IMU Foundation Models:* Large-scale IMU foundation models offer promising improvements in feature extraction and data generation. Pre-trained models capture transferable motion patterns (e.g., acceleration decay, spectral features), enhancing robustness to noise [49]. Generative models can further synthesize IMU sequences from modalities like text or video [50]. However, their large size and classification focus pose deployment and adaptation challenges. Future work will explore integrating these models to boost system flexibility and generalization.

## IX. CONCLUSION

To enable intuitive and efficient gesture-based interaction, we propose ArmPad, a gesture input system based on a COTS smartwatch that uses the forearm as interaction interface. ArmPad employs a gated multi-expert architecture for feature extraction, enabling concurrent gesture recognition and sliding distance estimation. Experimental results indicate that ArmPad achieves 92.5% gesture recognition accuracy and 1.53 $cm$ MAE in distance estimation under cross-user settings. After registration with a small amount of data from the new user, the specialized model's accuracy can achieve higher performance. Participants' feedback shows that ArmPad largely aligns with users' everyday gesture habits and intuitive requirements, exhibiting high user acceptance.

## REFERENCES


[1] M. Dehghani and K. J. Kim, "The effects of design, size, and uniqueness of smartwatches: perspectives from current versus potential users," *Behaviour & Information Technology*, vol. 38, no. 11, pp. 1143–1153, 2019.

[2] C. Anthes, R. J. García-Hernández, M. Wiedemann, and D. Kranzlmüller, "State of the art of virtual reality technology," in *2016 IEEE aerospace conference*. IEEE, 2016, pp. 1–19.

[3] S. Guamán, A. Calvopiña, P. Orta, F. Tapia, and S. G. Yoo, "Device control system for a smart home using voice commands: A practical case," in *Proceedings of the 2018 10th International Conference on Information Management and Engineering*, 2018, pp. 86–89.

[4] Y. Weng, C. Yu, Y. Shi, Y. Zhao, Y. Yan, and Y. Shi, "Facesight: Enabling hand-to-face gesture interaction on ar glasses with a downward-facing camera vision," in *Proc. CHI*, 2021, pp. 1–14.

[5] Y. Liu, S. Zhang, and M. Gowda, "Neuropose: 3d hand pose tracking using emg wearables," in *Proceedings of the Web Conference 2021*, 2021, pp. 1471–1482.

[6] C.-J. Lee, R. Zhang, D. Agarwal, T. C. Yu, V. Gunda, O. Lopez, J. Kim, S. Yin, B. Dong, K. Li *et al.*, "Echowrist: Continuous hand pose tracking and hand-object interaction recognition using low-power active acoustic sensing on a wristband," in *Proc. CHI*, 2024, pp. 1–21.

[7] Marium-E-Jannat, X.-D. Yang, and K. Hasan, "Around-device finger input on commodity smartwatches with learning guidance through discoverability," *International Journal of Human-Computer Studies*, vol. 179, p. 103105, 2023.

[8] K. Guo, H. Zhou, Y. Tian, W. Zhou, Y. Ji, and X.-Y. Li, "Mudra: A multi-modal smartwatch interactive system with hand gesture recognition and user identification," in *Proc. INFOCOM*, 2022, pp. 100–109.

[9] S. Cao, X. He, P. Zhu, M. Chen, X. Li, and P. Yang, "Ipand: accurate gesture input with ambient acoustic sensing on hand," in *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 2018, pp. 1–8.

[10] V. Becker, L. Fessler, and G. Sörös, "Gestear: combining audio and motion sensing for gesture recognition on smartwatches," in *Proceedings of the 2019 ACM International Symposium on Wearable Computers*, 2019, pp. 10–19.

[11] G. Laput, R. Xiao, and C. Harrison, "Viband: High-fidelity bio-acoustic sensing using commodity smartwatch accelerometers," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 2016, pp. 321–333.

[12] W. Chen, L. Chen, Y. Huang, X. Zhang, L. Wang, R. Ruby, and K. Wu, "Taprint: Secure text input for commodity smart wristbands," in *The 25th Annual International Conference on Mobile Computing and Networking*, 2019, pp. 1–16.

[13] C. Zhang, A. Bedri, G. Reyes, B. Bercik, O. T. Inan, T. E. Starner, and G. D. Abowd, "Tapskin: Recognizing on-skin input for smartwatches," in *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces*, 2016, pp. 13–22.

[14] M. Zhang, Q. Dai, P. Yang, J. Xiong, C. Tian, and C. Xiang, "idial: Enabling a virtual dial plate on the hand back for around-device interaction," *Proc. ACM IMWUT*, vol. 2, no. 1, pp. 1–20, 2018.

[15] W. Chen, Z. Wang, P. Quan, Z. Peng, S. Lin, M. Srivastava, W. Matusik, and J. Stankovic, "Robust finger interactions with cots smartwatches via unsupervised siamese adaptation," in *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023, pp. 1–14.

[16] F. Hu, P. He, S. Xu, Y. Li, and C. Zhang, "Fingertrak: Continuous 3d hand pose tracking by deep learning hand silhouettes captured by miniature thermal cameras on wrist," *Proc. ACM IMWUT*, vol. 4, no. 2, pp. 1–24, 2020.

[17] J. Chapman, A. Dwivedi, and M. Liarokapis, "A wearable, open-source, lightweight forcemyography armband: On intuitive, robust muscle-machine interfaces," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4138–4143.

[18] J. Yan, X. Yang, X. Sun, Z. Chen, and H. Liu, "A lightweight ultrasound probe for wearable human–machine interfaces," *IEEE Sensors Journal*, vol. 19, no. 14, pp. 5895–5903, 2019.

[19] C.-M. Chiu, S.-W. Chen, Y.-P. Pao, M.-Z. Huang, S.-W. Chan, and Z.-H. Lin, "A smart glove with integrated triboelectric nanogenerator for self-powered gesture recognition and language expression," *Science and technology of advanced materials*, vol. 20, no. 1, pp. 964–971, 2019.

[20] M. Kim, J. Cho, S. Lee, and Y. Jung, "Imu sensor-based hand gesture recognition for human-machine interfaces," *Sensors*, vol. 19, no. 18, p. 3827, 2019.

[21] D. Zhang, Z. Liao, W. Xie, X. Wu, H. Xie, J. Xiao, and L. Jiang, "Fine-grained and real-time gesture recognition by using imu sensors," *IEEE Transactions on Mobile Computing*, vol. 22, no. 4, pp. 2177–2189, 2021.

[22] C. Li, M. Liu, and Z. Cao, "Wihf: Enable user identified gesture recognition with wifi," in *Proc. INFOCOM*, 2020, pp. 586–595.

[23] R. Hajika, T. S. Gunasekaran, C. D. S. Y. Haigh, Y. S. Pai, E. Hayashi, J. Lien, D. Lottridge, and M. Billinghurst, "Radarhand: A wrist-worn radar for on-skin touch-based proprioceptive gestures," *ACM Transactions on Computer-Human Interaction*, vol. 31, no. 2, pp. 1–36, 2024.

[24] H. Wen, J. Ramos Rojas, and A. K. Dey, "Serendipity: Finger gesture recognition using an off-the-shelf smartwatch," in *Proc. CHI*, 2016, pp. 3847–3851.

[25] C. Zhang, J. Yang, C. Southern, T. E. Starner, and G. D. Abowd, "Watchout: extending interactions on a smartwatch with inertial sensing," in *Proceedings of the 2016 ACM International Symposium on Wearable Computers*, 2016, pp. 136–143.

[26] Y. Lu, B. Huang, C. Yu, G. Liu, and Y. Shi, "Designing and evaluating hand-to-hand gestures with dual commodity wrist-worn devices," *Proc. ACM IMWUT*, vol. 4, no. 1, pp. 1–27, 2020.

[27] N. DeVrio, V. Mollyn, and C. Harrison, "Smartposer: Arm pose estimation with a smartphone and smartwatch using uwb and imu data," in *Proceedings of the 36th annual ACM symposium on user interface software and technology*, 2023, pp. 1–11.

[28] M. Hosseini, H. Mueller, and S. Boll, "Controlling the rooms: how people prefer using gestures to control their smart homes," in *Proc. CHI*, 2024, pp. 1–18.

[29] M. Weigel, V. Mehta, and J. Steimle, "More than touch: understanding how people use skin as an input surface for mobile computing," in *Proc. CHI*, 2014, pp. 179–188.

[30] S. Pan, C. G. Ramirez, M. Mirshekari, J. Fagert, A. J. Chung, C. C. Hu, J. P. Shen, H. Y. Noh, and P. Zhang, "Surfacevibe: vibration-based tap & swipe tracking on ubiquitous surfaces," in *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2017, pp. 197–208.

[31] S. S. Rao and F. F. Yap, *Mechanical vibrations*. Addison-Wesley New York, 1995, vol. 4.

[32] J. Fahrenberg, F. Foerster, M. Smeja, and W. Müller, "Assessment of posture and motion by multichannel piezoresistive accelerometer recordings," *Psychophysiology*, vol. 34, no. 5, pp. 607–612, 1997.

[33] T. Cherian, S. Rakheja, and R. Bhat, "An analytical investigation of an energy flow divider to attenuate hand-transmitted vibration," *International Journal of Industrial Ergonomics*, vol. 17, no. 6, pp. 455–467, 1996.

[34] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures." *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[35] H. Tang, J. Liu, M. Zhao, and X. Gong, "Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations," in *Proceedings of the 14th ACM Conference on Recommender Systems*, 2020, pp. 269–278.

[36] C. Harrison, D. Tan, and D. Morris, "Skinput: appropriating the body as an input surface," in *Proc. CHI*, 2010, pp. 453–462.

[37] G. Hinton, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

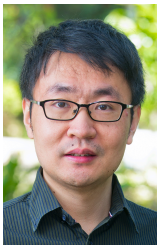
[38] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.
[39] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.
[40] W. Siri, "The gross composition of the body," *Advances in biological and medical physics/Academic Press*, 1956.
[41] B. K. Iwana and S. Uchida, "An empirical survey of data augmentation for time series classification with neural networks," *Plos one*, vol. 16, no. 7, p. e0254841, 2021.
[42] B. Liu, "Lifelong machine learning: a paradigm for continuous learning," *Frontiers of Computer Science*, vol. 11, no. 3, pp. 359–361, 2017.
[43] T. Hagos and T. Hagos, "Android studio profiler," *Android Studio IDE Quick Reference: A Pocket Guide to Android Studio Development*, pp. 73–82, 2019.
[44] Google Inc., "Profile battery usage with Batterystats and Battery Historian," accessed: March 5, 2026. [Online]. Available: https://developer.android.com/topic/performance/power/setup-battery-historian
[45] J. Brooke *et al.*, "Sus-a quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.
[46] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in *Advances in psychology*. Elsevier, 1988, vol. 52, pp. 139–183.
[47] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, and P. S. Yu, "Generalizing to unseen domains: A survey on domain generalization," *IEEE transactions on knowledge and data engineering*, vol. 35, no. 8, pp. 8052–8072, 2022.
[48] R. Xiao, J. Liu, J. Han, and K. Ren, "Onefi: One-shot recognition for unseen gesture via cots wifi," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 206–219.
[49] H. Xu, P. Zhou, R. Tan, M. Li, and G. Shen, "Limu-bert: Unleashing the potential of unlabeled data for imu sensing applications," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 220–233.
[50] Z. Leng, A. Bhattacharjee, H. Rajasekhar, L. Zhang, E. Bruda, H. Kwon, and T. Plötz, "Imugpt 2.0: Language-based cross modality transfer for sensor-based human activity recognition," *Proc. ACM IMWUT*, vol. 8, no. 3, pp. 1–32, 2024.

**Zhenyu Yang** received the BS degree from Henan University, in 2022, and the master's degree from Shenzhen University, in 2025. He is currently working toward the PhD degree with The Hong Kong University of Science and Technology (Guangzhou). His research interests include mobile computing and deep learning.

**Qiang Yang** (Member, IEEE) is an Assistant Research Professor in the Department of Computer Science and Technology at the University of Cambridge. He was previously a Postdoctoral Research Associate in the same department from 2023 to 2026. He received his Ph.D. degree in Computer Science from The Hong Kong Polytechnic University in 2023. His research interests lie in mobile health, Mobile computing, ubiquitous computing, and the Internet of Things.

**Zhidan Liu** received the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, China, in 2014. After that, he worked as a Research Fellow in Nanyang Technological University, Singapore, and a faculty member with College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He is currently an Assistant Professor at Intelligent Transportation Thrust, System Hub, The Hong Kong University of Science and Technology (Guangzhou). His research interests include Artificial Internet of Things, mobile computing, urban computing, and big data analytic. He is a senior member of IEEE and CCF.

**Yongpan Zou** (Member, IEEE) received the PhD degree from the Department of Computer Science and Engineering (CSE), Hong Kong University of Science and Technology, in 2017. He is currently an associate professor with the College of Computer Science and Software Engineering, Shenzhen University. His research interests include ubiquitous sensing, mobile computing, and human-computer interaction.



**Zhenjiang Li** (Member, IEEE) received the BE degree from Xi'an Jiaotong University, Xi'an, China, in 2007, and the MPhil and Ph.D. degrees from the Hong Kong University of Science and Technology, Hong Kong, China, in 2009 and 2012, respectively. He is currently a professor with the Department of Computer Science, City University of Hong Kong. His research interests include the Internet of Things, edge AI systems, and smart sensing.

**Kaishun Wu** received his Ph.D. degree in Computer Science and Engineering at The Hong Kong University of Science and Technology (HKUST). Before joining HKUST(GZ) as a Full Professor at DSA Thrust and IoT Thrust in 2022, he was a distinguished Professor and Director of Guangdong Provincial Wireless Big Data and Future Network Engineering Center at Shenzhen University. Prof. Wu is an active researcher with more than 200 papers published on major international academic journals and conferences, as well as more than 100 invention patents, including 12 from the USA. He received the 2012 Hong Kong Young Scientist Award, the 2014 Hong Kong ICT Awards: Best Innovation, and 2014 IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award. He is a Fellow of IEEE, IET, and AAIA.