SwipeSense: Exploring the Feasibility of Back-of-Device Swipe Interaction Using Built-In IMU Sensors

NEEL SHAH, Ontario Tech University, Canada
BENEDICT LEUNG, Ontario Tech University, Canada
MARIANA SHIMABUKURO, Ontario Tech University, Canada
ALI NESHATI, Ontario Tech University, Canada

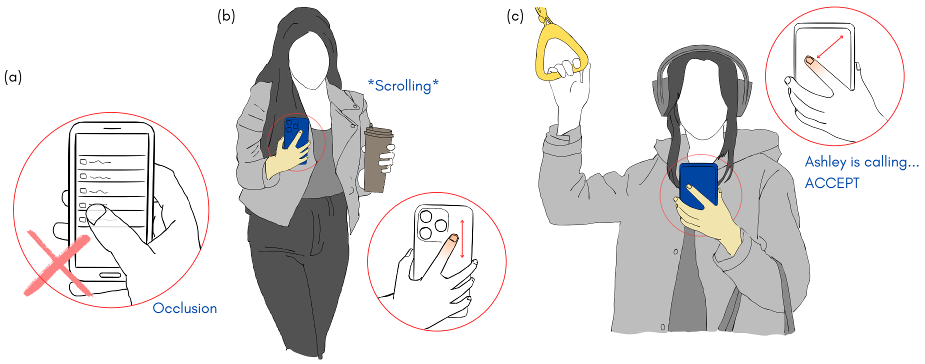


Fig. 1. a) Common "fat finger" problem during direct touchscreen interaction where the user's finger blocks the display content. b) One-handed scrolling using a back-of-device swipe, allows seamless interaction while holding another object. c) Answering an incoming call with a diagonal back-of-device swipe, enabling quick interaction while holding onto a support handle in a busy environment.

The growing dimensions of smartphones have intensified the challenges associated with screen reachability. Back-of-device (BoD) interaction expands the range of reachability and offers a promising solution to mitigate screen occlusion while enhancing one-handed interactions. However, much of the existing research relies on incorporating additional hardware components. In this paper, we present SwipeSense a technique for exploring the feasibility of directional swipe interactions on the back of devices, utilizing built-in inertial measurement unit (IMU) sensors and machine learning models. We conducted a user study with 12 participants who performed 9600 BoD swipes in 8 distinct directions while holding the device naturally. The results of our machine learning models indicate that various directional swipes on the back of the device can be accurately distinguished using only the built-in IMU sensors of the phone, achieving a range of model accuracy between 72% and 95%. Furthermore, we showcase potential applications for these gestures.

CCS Concepts: • **Human-centered computing → Gestural input**; *Smartphones*; • **Computing methodologies → Multi-task learning**.

Additional Key Words and Phrases: back-of-device interaction, input techniques, swipe gesture, smartphones, screen occlusion

Authors' Contact Information: Neel Shah, Ontario Tech University, Oshawa, Ontario, Canada, neelsamirkumar.shah@ontariotechu.net; Benedict Leung, Ontario Tech University, Oshawa, Ontario, Canada, benedict.leung1@ontariotechu.net; Mariana Shimabukuro, Ontario Tech University, Oshawa, Ontario, Canada, mariana.shimabukuro@ontariotechu.ca; Ali Neshati, Ontario Tech University, Oshawa, Ontario, Canada, ali.neshati@ontariotechu.ca.

ACM 2573-0142/2025/9-ARTMHCI030
https://doi.org/10.1145/3743734

ACM Reference Format:
Neel Shah, Benedict Leung, Mariana Shimabukuro, and Ali Neshati. 2025. SwipeSense: Exploring the Feasibility of Back-of-Device Swipe Interaction Using Built-In IMU Sensors. *Proc. ACM Hum.-Comput. Interact.* 9, 5, Article MHCI030 (September 2025), 24 pages. https://doi.org/10.1145/3743734

1 Introduction

Despite mobile devices increasing in processing power and size, interaction with smartphone displays still presents significant challenges, particularly with one-handed interaction which is crucial for accessibility and convenience [16, 45, 51]. These issues are exacerbated when users are multitasking or carrying objects like bags or cups (see Figure 1b) — generating a need for one-handed interaction [21, 26]. Additionally, the "fat finger" problem (Figure 1a), a well-known issue with small displays, complicates usability as it typically occurs when users attempt to interact with the screen using their other hand, often blocking the content underneath the finger during direct interaction [22, 23].

To enhance mobile user experience by reducing screen occlusion and providing more ergonomic options for interaction, recent studies have identified the back or sides of smartphones as potential areas to solve these issues through off-screen gestures [21, 33, 46, 58]. In contrast, numerous one-handed interaction techniques have thus emerged to address these multitasking limitations. These solutions range from software-based approaches [16, 28], to context-aware sensing [14, 51], and alternate input surfaces [21, 45, 54]. Yet, many existing solutions concentrate on discrete taps and/or device modifications including specialized hardware or form-factor modifications. Thus, continuous gestures, particularly back-of-device (BoD) swipes are under-explored.

User-defined studies confirmed that swipes feel intuitive and familiar, often mirroring front-of-screen interactions like map navigation or call handling [46, 55]. This suggests that BoD swipes can reduce screen occlusion, maintain fluid interaction, and minimize grip changes without requiring users to learn new gestures. However, further research is needed to translate this potential into commercially viable interactions — an area where major smartphone manufacturers are already exploring off-screen input techniques. For instance, Apple's "Back Tap"[1] enables shortcut activation through rear taps, while Samsung's fingerprint sensor swipe[2] provides quick access to notifications. Similarly, Google's Active Edge [40] allows squeezing gestures for specific functions. While these techniques reduce visual clutter, they rely on specialized hardware (e.g., fingerprint sensors, pressure-sensitive edges) and focus on discrete interactions. In contrast, directional swipes are continuous interactions and they remain largely unexplored despite their potential for richer, more natural interactions that build on familiar user behaviours without requiring grip adjustments or screen obstruction.

In the context of training models for gestures recognition, multi-task learning (MTL) — a widely used machine learning approach that optimizes multiple related tasks in parallel, has been successfully applied in areas for BoD tap recognition [24]. However, this approach has not been explored for detecting and classifying swipe gestures. Unlike taps, which generate static data, swipe gestures produce continuous, dynamic patterns, requiring advanced processing techniques. This work investigated a unified convolutional neural network (CNN) capable of handling two tasks: 1) swipe detection to identify the presence of a swipe and 2) swipe classification to differentiate swipe directions.

To investigate the feasibility of identifying BoD gestures, we conducted a user study in which our participants performed PERPENDICULAR and DIAGONAL swipes (Figure 2) for a range of gesture

[1]https://support.apple.com/en-ca/111772, accessed February 5, 2025.
[2]https://eu.community.samsung.com/t5/galaxy-z-fold-z-flip/notification-swipe-down/td-p/3897717, accessed February 5, 2025.

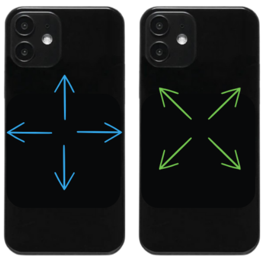


Fig. 2. Illustration of PERPENDICULAR (left, blue arrows) and DIAGONAL (right, green arrows) swipes performed on the back-of-device.

directions. Quantitatively, our system achieved swipe detection accuracies, frequently surpassing 90%, with some decline when generalizing across different users. Qualitatively, participants described BoD swipes as intuitive, notably for common tasks such as scrolling and navigation, answering/rejecting calls, and media control. Although DIAGONAL swipe gestures introduced occasional challenges related to grip adjustments and fatigue, we learned that DIAGONAL swipes should be reserved for less frequent actions (i.e., zoom in/out in a map application), underscoring the feasibility of this off-screen interaction paradigm in broader mobile user experience contexts.

The main contributions of this paper are twofold. 1) SwipeSense explored the feasibility of BoD swipe interactions using built-in inertial measurement unit (IMU) sensors, eliminating the need for external hardware. 2) The data was collected from 12 participants, each performing 800 swipe gestures at a 10Hz sampling rate, resulting in a comprehensive dataset of over 250K sensor readings. This dataset will be made publicly available upon acceptance on a GitHub repository, paving the way for further research on swipe-based interactions and can be repurposed to evaluate alternative recognition techniques.

2 Related Work

In this section, we discuss related work on peripheral and BoD techniques, one-handed interaction techniques, and background on machine learning approaches for detecting and classifying interactions in mobile and wearable devices.

2.1 Peripheral and Back-of-Device Input Techniques

Early BoD input explorations often relied on specialized or external hardware, limiting commercial viability and increasing costs. For instance, BackXPress [6] featured a "sandwiched" phone design that measured rear-surface finger pressure, while BackC&P [4] demonstrated how BoD touch can augment mobile copy-and-paste workflows via its prototype, which simply mounted two off-the-shelf smartphones back-to-back to enable efficient back taps. Back-Mirror [54] employed a mirror attachment to direct the camera's view to the back, and BackTrack [56] implemented a thin, battery-free capacitive trackpad on the back that maps 2D finger motion to front-screen touches. InfiniTouch [34] extended touch sensititvity beyond the display area, and other methods, such as Finexus [3] leveraged acoustic sensing for touch on the device [49]. Electrick [62] and AuraSense [63] worked on touch detection via electric fields in mobile and wearable, respectively. Although these techniques showcased the potential of off-screen interaction, they require additional transducers, sensor arrays, or custom instrumentation, making mass adoption challenging.

Later research sought to reduce reliance on custom hardware by using onboard sensors or minor modifications. For instance, TapSense [17] identified touch variations via acoustic signatures, while HandSee [60] employed a stereo cameras for full-hand gestures tracking. In wearables, solutions like Sidetap & Slingshot [59] and SynchroWatch [43] used smartwatch sensors to support additional gestures without hardware customization. Similarly, WatchOut [61] used built-in IMUs on smartwatches to detect around-device gestures. However, many of these approaches relied on computation offloaded to external devices (e.g., cameras, external processing units) or focus on wearables.

However, a small number of systems avoid both extra hardware and off-device processing. For example, VSkin [48] leveraged built-in speakers and microphones to capture structure-borne and air-borne acoustic signals for fine-grained back-surface gesture sensing, but is tied to a specific speaker–microphone layout and raises privacy concerns by exposing raw audio streams. Similarly, Granell and Leiva [15] systematically analyzed built-in smartphone sensors (accelerometer, gyroscope, gravity, microphone) to identify a minimal subset of features for reliable and efficient BoD tap detection, demonstrating high accuracy while ensuring low energy consumption. Woodpecker [32] offered secure mobile authentication by fusing built-in accelerometer and microphone data, applying a lightweight fast Fourier transform, dynamic time warping, and wavelet-based peak detection to verify secret back-tap rhythms. Likewise, TapNet [24] implements an entirely on-device multi-task convolutional network that processes IMU signals and device form factor in parallel, jointly estimating multiple tap properties for robust off-screen input without additional hardware or external computation.

Inspired by TapNet [24], which demonstrated BoD tap detection using only IMU sensors, and other IMU-based techniques [14, 35, 45, 61], we present SwipeSense, it extends TapNet's tap concept to continuous swipe gestures. By leveraging swipe gestures familiarity, it preserves standard smartphone configuration while enabling practical, lightweight BoD interactions. Like TapNet, it provides a robust, easily deployable solution for off-screen interaction without requiring custom hardware and offloaded computation.

2.2 One-Handed Interaction Techniques

One-handed interaction techniques address the challenges of operating increasingly large smartphones using only one hand, where the thumb is the primary input digit [16]. This is particularly useful when the other hand is occupied (e.g., carrying items), allowing users to perform tasks like typing or swiping one-handed. Software-based enhance the thumb's reachability, such as EyeMU [31] — which pairs gaze estimation with IMU gestures, pressure-based cursor for distant display targets [16], and ThumbSpace [28], which remaps distant targets into smaller proxy regions. Similarly, other methods dynamically adapt interactions using grip sensing (GripSense [14]) or alternative input modalities (EarTouch [51] supporting ear-based gestures for visually impaired users).

Beyond touchscreen interfaces, researchers have explored alternative form factors and hardware to enhance one-handed interactions. Examples include bendable displays [13], mid-air thumb gestures [18], and BackPat [45] — which used device microphones and gyroscopes for patting gestures. MagTap [29] localized BoD single and double taps by detecting a button-cell tapping tool's perturbations in the phone's magnetometer, accelerometer, and gyroscope. Wearables, including PinchWatch [36] and side-press interactions [21] also contributed to this space. Collectively, these approaches aimed to minimize grip changes, reduce occlusion, and maintain fluid interaction. However, most rely on discrete taps, specialized hardware, or offer limited support for continuous BoD gestures.

Recent user-defined studies [46, 55] highlighted the potential of BoD swipe gestures for off-screen interactions. While tap-based BoD solutions have been widely explored [21, 24, 25, 45], swipes remain underutilized despite their prevalence in front-screen interactions like scrolling and navigation [55]. Shimon et al. [46] found that users frequently replicated front-screen swipes on the back for tasks such as map panning, call handling, and home-screen navigation — often preferring them over taps. Their elicitation study showed strong consensus on the suitability of BoD swipes, with 14 out of the 15 participants using swipe for the “Next” command. Additionally, swipe gestures were seen as intuitive for answering or rejecting calls, aligning with the widely recognized Tinder [3] acceptance/rejection metaphor.

Despite these user-driven insights, existing BoD research has primarily focused on discrete tap detection [21, 25, 45], simplistic off-screen gestures, or used specialized hardware to detect swipes [9, 11, 50, 54, 56, 57]. In contrast, our technique, SwipeSense, expanded the BoD interaction design space by enabling robust, directional swipe recognition and classification using only built-in IMU signals — eliminating additional hardware. SwipeSense employs a MTL (machine learning) framework to achieve reliable BoD swipe detection and direction classification (F1 scores: 72% – 95% across various conditions) while maintaining computational efficiency (see latency scores in subsection 6.1) and preserving front-screen visibility.

2.3 Machine Learning Approaches for Interaction Recognition on Mobile and Wearable Devices

Machine learning can enable robust, real-time gesture recognition on minimal or unmodified mobile hardware — with onboard sensors. Some approaches used a phone’s RGB camera to detect in-air gestures [47], while wearable-centric methods relied on IMU signals with transfer learning to adapt user-defined gestures [8]. However, most employed single-task models focused solely on classification or detection, limiting efficiency and missing cross-task insights.

Multi-task learning (MTL) optimizes related subtasks simultaneously, improving accuracy, efficiency, and generalization [2, 19, 42]. For example, Liu et al. [35] showed that jointly predicting orientation and location on a smartwatch improves precision, while TapNet [24] employed MTL for BoD tap recognition on smartphones. Despite these efforts, continuous gestures like BoD swipes — requiring both temporal and directional analysis — remain underexplored in MTL contexts. SwipeSense, extends MTL to BoD swipe interactions, integrating swipe detection and directional classification into a single neural network. Inspired by TapNet [24] and Liu et al. [35], SwipeSense reduces computational overhead while achieving robust, real-time performance. In contrast to prior single-task or hardware-dependent methods, our approach relies solely on the device’s IMU, enabling accurate off-screen interaction without additional hardware or external computation offload.

3 SwipeSense Implementation

Building on previous works in IMU-based input methods [14, 24, 35, 45, 61], our research expands the interaction space on mobile devices by enabling robust recognition of BoD swipes through a multi-task neural network. We propose a pipeline for delivering two key outputs from the same model: 1) swipe detection (i.e., whether a swipe gesture has occurred) and 2) swipe classification (i.e., identifying the direction of the swipe).

3.1 Formal Problem Definition

Let $\mathrm{a}^t \in \mathbb{R}^3$ and $\boldsymbol{\omega}^t \in \mathbb{R}^3$ denote the accelerometer and gyroscope readings at timestamp t. Our goal is to learn a function

$$f\left(\mathrm{a}^t, \boldsymbol{\omega}^t\right)$$

that produces two types of outputs for each window of IMU data:

[3]https://tinder.com/, accessed February 5, 2025.

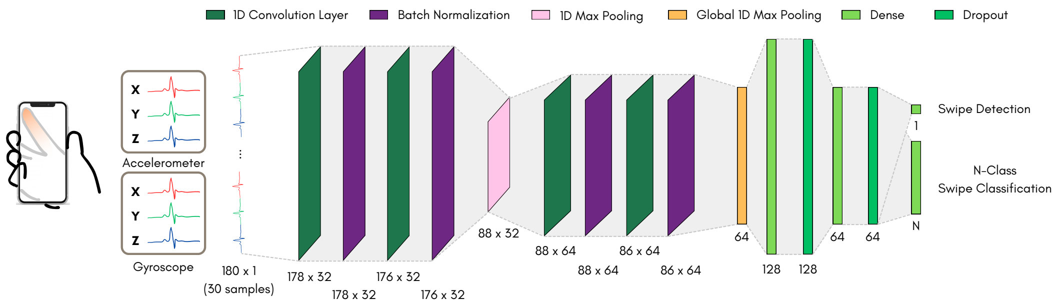


Fig. 3. Overview of the multi-task learning model for back-of-device swipe classification. The model processes accelerometer and gyroscope data (X, Y, Z axes) using a series of 1D convolution layers, batch normalization, max pooling, dense and dropout layers. It detects swipe gestures and classifies them into N swipe categories.

- $\hat{y}_t^{\text{swipe}} \in \{0, 1\}$: Binary indicator for swipe detection, determining if the input contains a swipe gesture.
- $\hat{\mathbf{y}}_t^{\text{type}} \in [0, 1]^C$: Probability distribution for swipe classification, identifying one among C possible swipe directions.

Formally, we write:

$$\hat{y}_t^{\text{swipe}}, \quad \hat{\mathbf{y}}_t^{\text{type}} = f(\mathbf{a}^t, \boldsymbol{\omega}^t),$$

where f encapsulates the shared and task-specific parameters learned by the multi-task neural network.

3.2 Pipeline Overview

The pipeline starts with data preparation, capturing six-channel IMU signals (accelerometer and gyroscope, three axes each) at a 100ms sampling rate. A 30-sample window (~3000 ms) represents the duration of a typical swipe movement. First-order derivatives emphasize motion dynamics, improving orientation invariance and reducing sensitivity to device alignments. SwipeSense, a convolutional neural network (CNN) with shared layers, processes this input to extract task-agnostic features with two specialized branches for swipe detection and classification (Figure 3).

3.3 Design of SwipeSense Architecture

SwipeSense's multi-task CNN architecture is illustrated in Figure 3. It concurrently performs two tasks: swipe detection (binary classification) and swipe classification (multiclass). The model input (six-channel IMU data) is flattened into a one-dimensional, 180-element feature vector, enabling convolutional filters to operate across all IMU features. This one-channel CNN approach simplifies the architecture, reducing parameters while effectively capturing both local motion dynamics and global context [10, 37, 39].

Unlike prior work [24] that discards low-amplitude gestures via heuristic gating (e.g., peak detection), SwipeSense omits such gating, ensuring subtle swipes are recognized. While this slightly increases computational overhead, it removes reliance on domain-specific thresholds, improving robustness to dynamic user behaviours and phone orientations.

The shared feature extraction module comprises convolutional layers with batch normalization, followed by dense layers with dropout [30]. These layers progressively extract motion patterns at varying abstraction levels. Two output branches handle the separate tasks: a sigmoid neuron for binary swipe detection and a softmax layer for multiclass direction classification. This MTL approach enables efficient and accurate BoD swipe recognition for short input windows.

Swipe Detection Loss. We treat swipe detection as a binary classification task, with labels $y_t^{\text{swipe}} \in \{0, 1\}$. The network outputs $\hat{y}_t^{\text{swipe}} \in [0, 1]$ via a sigmoid activation:

$$\hat{y}_t^{\text{swipe}} = \text{sigmoid}\big(f(\mathrm{a}^t, \boldsymbol{\omega}^t)\big).$$

We adopt binary cross-entropy for this branch:

$$L_{\text{swipe}} = -\frac{1}{T}\sum_{t=1}^{T}\Big[\, y_t^{\text{swipe}} \, \log\big(\hat{y}_t^{\text{swipe}}\big) + \big(1 - y_t^{\text{swipe}}\big) \, \log\!\big(1 - \hat{y}_t^{\text{swipe}}\big)\Big],$$

where T is the number of samples in the batch.

Swipe Classification Loss. Swipe classification is formulated as a multiclass classification problem over C classes, with one-hot labels $\mathrm{y}_t^{\text{type}} \in \{0, 1\}^C$. The output is $\hat{\mathrm{y}}_t^{\text{type}}$, a probability vector from a softmax layer:

$$\hat{\mathrm{y}}_t^{\text{type}} = \text{softmax}\big(f(\mathrm{a}^t, \boldsymbol{\omega}^t)\big).$$

We use categorical cross-entropy:

$$L_{\text{type}} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{c=1}^{C} y_{t,c}^{\text{type}} \, \log\!\big(\hat{y}_{t,c}^{\text{type}}\big),$$

where $y_{t,c}^{\text{type}}$ is the label for class c, and $\hat{y}_{t,c}^{\text{type}}$ is its predicted probability.

Combined Multi-task Loss. Finally, we integrate both tasks into a single multi-task objective:

$$L_{\text{total}} = \alpha \, L_{\text{swipe}} + \beta \, L_{\text{type}},$$

where α and β control the relative weighting of swipe detection vs. classification. In practice, we set $\alpha = \beta = 1$ to give equal importance to both tasks, but these weights can be tuned based on dataset characteristics or application needs. Training the network under this combined loss ensures that features beneficial for swipe detection and classification are simultaneously learned, leading to a more robust, shared representation of BoD gestures.

4 Data Collection

To explore the feasibility of SwipeSense and ensure its practicality, we gathered data from a group of 12 participants. We collected one-handed BoD interactions, where participants were instructed to swipe the back of the phone with their index finger in multiple directions. This data collection was carried out in accordance with the guidelines that our institution's Research Ethics Board had reviewed. The study session took approximately 90 minutes, and participants received a $20 gift card as compensation for their time.

4.1 Apparatus and Software

For our data collection process, we selected the iPhone 12 due to its moderate size, avoiding the smaller dimensions of the "Mini" or the larger scale of the "Pro Max" models. This choice ensured that the phone would comfortably fit in the hands of most participants, providing a standard baseline for our experiments across various hand sizes and usability scenarios. The smartphone was connected to a Mac Mini (M2 processor with 16G RAM) throughout the experiment to collect and process data efficiently. The study system was implemented using Swift[4].

[4]https://developer.apple.com/swift/, accessed in February 5, 2025.

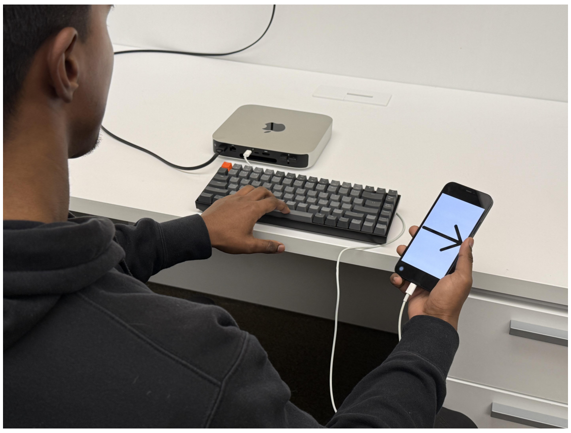


Fig. 4. Experimental Setup: Participant holding a smartphone to perform back-of-device directional gestures while using a keyboard to manually start and stop each gesture sequence during the lab study.

4.2 Study Design Overview

We designed our experiment using a within-subjects study design with one primary independent variable: GESTURE SETS with two levels (DIAGONAL and PERPENDICULAR gestures). There were four DIRECTION variations for each GESTURE SETS (PERPENDICULAR: ↓, ↑, →, ←; DIAGONAL: ↘, ↙, ↖, ↗), abbreviated for brevity (e.g., ↖ – bottom right to top left). Each direction variation (trial) within a technique was randomly displayed (uniform distribution). A trial was repeated 200 times per GESTURE SETS, representing a block (1×200). The study had two blocks per GESTURE SETS to increase the number of trials (4×200). The order for GESTURE SETS blocks was counter-balanced using a Latin square. The primary measurements taken in our study were six-channel IMU signals. These signals capture essential motion data across three axes of acceleration and three axes of gyroscopic movement. This allows for a comprehensive analysis of the user's interactions with the device during the experimental tasks. In addition, the questionnaire provides subjective measures. All measures are continuous. In summary: 4 blocks $\times$ 200 repetitions = 800 swipes per participant, totalling 9600 for the 12 participants.

4.3 Procedure

Pre-study: Upon arriving at the lab, participants were first briefed on the project's objective. They read and signed a consent form, confirming their understanding and agreement to participate in the study. After completing the demographics questionnaire, their hand proportions were measured for further analysis and deeper insights from the collected data.

Pre-trial instructions and training: Participants were instructed to hold the device with their dominant hand and adjust it until they were comfortable. Their non-dominant hand started the trials on the Mac Mini (Figure 4). They were also instructed not to rest their dominant hand on the desk to mimic one-handed interaction scenarios.

Training: Participants were asked to train and perform all the gestures. The training was organized into two blocks: in one block, participants practiced PERPENDICULAR swipes, and in the other, they practiced DIAGONAL swipes. Each block consisted of at least 20 swipes. Participants were allowed to continue practicing until they felt comfortable performing all gestures with confidence, ensuring they were ready to proceed with the experiment. However, participants were not given feedback or specific guidance on executing trials beyond their direction to avoid suggesting that they perform the gestures in any specific manner (i.e., cadence of movement or specific grip).

Study blocks: After the training phase, the participants were allowed to start working on the four blocks of trials. Based on insights from the pilot study, a mandatory break was implemented every 50 swipes to minimize fatigue and allow hand and finger positioning changes. Additionally, participants took at least a five-minute break after completing each block before proceeding to the next block.

Post-study: After completing all task blocks in our study, participants filled out a questionnaire that assessed their preferences, comfort levels, and potential applications of the interaction technique they interacted with during the experiment. The comfort questions were structured on a scale from 1 to 5, where 1 represented the least comfortable and 5 indicated the most comfortable. This allowed us to quantify participants' comfort levels with the device and interaction technique used during the experiment.

4.4 Participants

We recruited 12 right-handed participants, ages 19 to 23, of which six identified as female and six as male. This selection was made to ensure control and consistency across the experiment by using participants who all used their dominant hand, the right hand, to interact with the phone. Focusing on right-handed users helps eliminate any confounding effects that might arise from variations in hand dominance, thus maintaining uniformity in the data collected. Participants were recruited using a mass email sent to the students.

5 Machine Learning Experiments

We conducted our experiments under five training conditions to assess how well SwipeSense generalizes across users and gestures:

- **Multi-Person:** All participants' data were pooled, then split into 70% training, 20% test, and 10% validation. This configuration simulates a typical scenario where diverse user data is available.
- **Single-Person:** Each participant's data was split (e.g., 70–20–10) for training, testing, and validation on that same individual, reflecting a personalized model. We repeated this for all 12 participants and averaged the results.
- **Cross-Person:** One participant's data was used for training, and the other 11 participants' data served as the test set. This process was repeated for each participant to measure how well the model trained on one user generalizes to others.
- **Leave-One-User-Out (LOO):** For each of the 12 participants, we held out that user's entire dataset for testing, and trained/validated on the other 11 users' data (70% train, 20% test, 10% validation).
- **LOO + Fine-Tuning (LOO + FT):** Starting from each LOO-trained model (70% train/20% test/10% validation on 11 users), we fine-tuned on the held-out user's data (20% train, 10% validation; the remaining 70% reserved for final testing) using a reduced learning rate and early stopping.

Baselines and Proposed Multi-task Approach: Similarly to TapNet [24], we implemented (using TensorFlow[5]) and evaluated three machine learning techniques:

- **SVM:** A conventional machine learning baseline using a Support VectorMachine [7, 20]. Unlike deep nets, SVMs rely on static kernels and cannot be fine-tuned in place, to incorporate a held-out user's data (as in LOO/LOO + FT), we retrained the SVM from scratch by augmenting its training set with that user's samples.

[5]https://www.tensorflow.org/,accessed accessed February 5, 2025.

Table 1. Weighted F1 Scores for SVM, SISO SwipeSense, and MIMO SwipeSense across Various Gesture Sets and Training Scenarios with Cross-Validation (CV) Results.

		SVM Detection		**SVM Classification**		**SISO Detection**		**SISO Classification**		**MIMO Detection**		**MIMO Classification**	
		F1	**CV**	**F1**	**CV**	**F1**	**CV**	**F1**	**CV**	**F1**	**CV**	**F1**	**CV**
Perpendicular	Multi-Person	0.90	0.88 ± 3.1	0.83	0.81 ± 2.8	0.91	0.88 ± 3.0	0.84	0.82 ± 3.4	**0.93**	0.90 ± 2.5	**0.91**	0.88 ± 3.2
	Single-Person	0.85	0.80 ± 3.3	0.81	0.77 ± 3.0	0.86	0.82 ± 3.2	0.83	0.79 ± 3.1	**0.88**	0.86 ± 3.0	**0.86**	0.84 ± 2.8
	Cross-Person	0.78	0.75 ± 3.5	0.69	0.67 ± 3.6	0.80	0.77 ± 3.4	0.75	0.72 ± 3.3	**0.82**	0.78 ± 3.2	**0.79**	0.75 ± 3.5
	LOO	0.86	0.84 ± 3.2	0.78	0.76 ± 3.2	0.87	0.85 ± 3.1	0.81	0.79 ± 3.0	**0.89**	0.87 ± 3.0	**0.89**	0.87 ± 3.0
	LOO+FT	0.92	0.90 ± 2.8	0.88	0.86 ± 2.8	0.93	0.91 ± 2.6	0.88	0.86 ± 2.8	**0.95**	0.93 ± 2.5	**0.94**	0.92 ± 2.5
Diagonal	Multi-Person	0.86	0.85 ± 3.0	0.80	0.78 ± 2.8	0.88	0.86 ± 3.2	0.87	0.87 ± 3.0	**0.89**	0.89 ± 2.7	**0.87**	0.87 ± 3.3
	Single-Person	0.82	0.82 ± 3.5	0.74	0.74 ± 3.9	0.81	0.81 ± 3.2	0.78	0.78 ± 3.3	**0.83**	0.83 ± 3.5	**0.80**	0.80 ± 3.4
	Cross-Person	0.75	0.75 ± 3.6	0.70	0.70 ± 3.4	0.76	0.76 ± 3.4	0.72	0.72 ± 3.6	**0.76**	0.76 ± 3.5	**0.72**	0.72 ± 3.7
	LOO	0.84	0.82 ± 3.1	0.77	0.75 ± 3.2	0.85	0.83 ± 2.8	0.82	0.80 ± 3.1	**0.86**	0.84 ± 2.9	**0.85**	0.83 ± 3.2
	LOO+FT	0.89	0.87 ± 2.9	0.82	0.80 ± 2.7	0.90	0.88 ± 2.7	0.87	0.85 ± 2.9	**0.92**	0.90 ± 2.6	**0.91**	0.89 ± 2.8
Combined	Multi-Person	0.89	0.89 ± 2.8	0.87	0.87 ± 2.9	0.90	0.90 ± 3.1	0.88	0.88 ± 2.7	**0.91**	0.91 ± 2.6	**0.89**	0.89 ± 2.8
	Single-Person	0.84	0.84 ± 3.3	0.80	0.80 ± 3.2	**0.86**	0.86 ± 3.4	0.82	0.82 ± 3.5	0.85	0.85 ± 3.3	**0.84**	0.84 ± 3.6
	Cross-Person	0.76	0.76 ± 3.4	0.72	0.72 ± 3.5	0.78	0.78 ± 3.5	0.73	0.73 ± 3.7	**0.79**	0.79 ± 3.4	**0.77**	0.77 ± 3.5
	LOO	0.86	0.84 ± 2.6	0.82	0.80 ± 3.2	**0.88**	0.86 ± 2.5	0.82	0.80 ± 3.3	0.87	0.85 ± 2.4	**0.84**	0.82 ± 3.3
	LOO+FT	0.91	0.89 ± 2.4	0.88	0.86 ± 2.7	0.91	0.89 ± 2.3	0.88	0.86 ± 2.8	**0.94**	0.92 ± 2.8	**0.93**	0.91 ± 2.6

- **Single-Input Single-Output (SISO) SwipeSense:** A single-task neural network variant that separates swipe detection from classification.
- **Multi-Input Multi-Output (MIMO) SwipeSense:** Our proposed multi-task architecture, which jointly performs swipe detection (binary) and swipe classification (multiclass).

When comparing SVM, SISO, and MIMO, we can assess how advanced neural networks (single-task or multi-task) can perform against a well-known machine learning baseline (SVM) in different user scenarios. subsection 6.2 presents the model's evaluation and comparisons.

Model tuning: To further improve model performance, we performed hyperparameter tuning using Optuna[6]. This automated framework systematically searches the parameter space for optimal configurations. In addition, we conducted this search on the multi-person training scenario, as it offered the most user diversity, serving as a robust proxy for tuning. The best trial yielded: learning rate: 2.91×10^{-4}, dropout rate (Conv layers): 0.218, and dropout rate (Dense layers): 0.188.

These hyperparameters were applied across the single-person, cross-person, and multi-person conditions for consistency and easy comparisons. The clarity of a unified configuration outweighed any minor performance trade-offs from using a single hyperparameter set. After tuning, we retrained SwipeSense (MIMO) from scratch for up to 100 epochs (or until convergence). We used an early stopping with a patience of five epochs to halt training once performance no longer improved.

These procedures were carried out for each gesture subset (PERPENDICULAR, DIAGONAL, and COMBINED) under all five training conditions. We evaluated the final models on both the binary (swipe vs. non-swipe) and multiclass (swipe direction) tasks. Across conditions, Optuna consistently provided stable training configurations, while deeper CNN layers and selected dropout rates yielded robust classifications of nuanced swipe motions.

Deployable model: Furthermore, for the mobile deployment test of our approach, MIMO SwipeSense, we exported it to a TensorFlow `SavedModel` and converted it to a `.mlmodel` file using Apple's `coremltools`. This `.mlmodel` was integrated into an Xcode project to build a Swift application for the iPhone 12.

6 Results

This section presents the outcomes of our machine learning experiments (section 5) along with model latency and qualitative feedback from the post-study questionnaire.

[6]https://optuna.org/, accessed February 5, 2025.

Table 2. Comparison of five training configurations (Multi-Person, Single-Person, Cross-Person, Leave-One-User-Out(LOO), and LOO + Fine-Tuning) for MIMO SwipeSense. Multi-Person leverages data diversity for the highest F1; Single-Person demonstrates per-user customization; Cross-Person exhibits the largest drop, reflecting user-specific swipe patterns; LOO further stresses cross-user generalization with a held-out user; LOO + FT recovers and often exceeds the Multi-Person baseline by adapting on a small amount of held-out user data.

Condition	**Description**	**Observation**
Multi-Person	Data pooled from all participants (70% train, 20% test, 10% val).	Highest F1 scores, diversity aids generalization.
Single-Person	Train/test on the same individual's data (70–20–10). Repeated for all 12 participants.	Slightly lower F1 than Multi-Person; feasible for per-user customization.
Cross-Person	Train on one participant, test on the remaining 11. Repeated for each participant.	Largest F1 drop, reflecting user-specific swipe patterns.
LOO	Data pooled from 11 users (70 % train / 20 % test / 10 % validation), the held-out user's entire dataset used only for testing.	Modest F1 drop (≈2–5%) relative to Multi-Person, reflecting the challenge of extrapolating to a completely unseen user.
LOO + FT	Starting from each LOO model, fine-tuned top dense layers on 20 % of the held-out user's swipes (10 % - validation), with convolution layers frozen and a reduced learning rate.	Recovers and even exceeds Multi-Person F1 (≈3-4% boost), demonstrating that a small amount of user-specific data markedly improves personalization.

6.1 Real-Time Inference and Mobile Deployment

To demonstrate the computational feasibility of our MIMO SwipeSense model, we evaluated its inference performance on both a Mac mini and an iPhone 12. We used 30 sensor samples per batch, corresponding to six IMU features across 30 timesteps, and flattened these readings into a $(180, 1)$ input shape. The 30-sample window (3s) was selected as a conservative, fixed-length segment. This choice ensures that the model captures the complete swipe and any additional time added by the study protocol (i.e., space bar activations). Real-time latency doesn't require waiting for the entire window, but as soon as the model confidence crosses a threshold, often within the first 10-15 samples (100-150ms). Below, we summarize the setup and observations for each platform.

Mac mini. The average inference time was 80.89ms per batch. This result indicates that the model can be further optimized (e.g., GPU acceleration) and may be needed for tighter latency constraints in desktop environments. However, the current inference time is still operational for prototyping and testing deployments in desktops.

iPhone 12. To deploy on a commercial smartphone, we converted the MIMO SwipeSense model into `CoreML` [7] format. During inference of a single batch of 30 IMU samples, it achieved a latency of 1.57ms. This suggests that even a relatively deep, multi-task CNN can run efficiently on modern smartphones. This result highlights the viability of our model for real-time, on-device swipe recognition without additional hardware.

6.2 Model Evaluation Results

As previously discussed, we evaluated the three approaches (SVM, SISO SwipeSense, and MIMO SwipeSense) on three gesture sets (PERPENDICULAR, DIAGONAL, COMBINED) and five training scenarios (Multi-Person, Single-Person, Cross-Person, LOO, LOO + FT). Table 2 summarizes each scenario, and Table 1 reports both single-test weighted F1 scores and 10-fold Cross-Validation (CV) (mean ± standard deviation).

[7]https://developer.apple.com/documentation/coreml/, accessed 5 February, 2025

Table 3. MIMO SwipeSense: Precision (P), Cross-Validation (CV), Recall (R), and AUC with Detection AUC (Det AUC), Classfication AUC (Clf AUC) across Training Scenarios.

		Detection				**Classification**				**AUC**			
Gesture Set	**Scenario**	**P**	**CV**	**R**	**CV**	**P**	**CV**	**R**	**CV**	**Det AUC**	**CV**	**Clf AUC**	**CV**
Perpendicular	Multi-Person	0.95	0.91 ± 2.5	0.93	0.88 ± 2.5	0.91	0.90 ± 3.2	0.91	0.89 ± 3.2	0.97	0.94 ± 2.3	0.95	0.91 ± 3.0
	Single-Person	0.91	0.88 ± 3.0	0.88	0.86 ± 3.0	0.86	0.81 ± 3.4	0.86	0.85 ± 3.4	0.92	0.88 ± 3.0	0.86	0.84 ± 2.8
	Cross-Person	0.87	0.83 ± 3.2	0.82	0.77 ± 3.2	0.79	0.75 ± 3.5	0.79	0.77 ± 3.5	0.93	0.88 ± 3.3	0.90	0.85 ± 2.6
	LOO	0.93	0.90 ± 2.5	0.90	0.87 ± 2.5	0.89	0.87 ± 3.0	0.89	0.86 ± 3.0	0.96	0.93 ± 2.9	0.91	0.88 ± 3.0
	LOO + FT	0.96	0.93 ± 2.2	0.94	0.91 ± 2.2	0.94	0.89 ± 2.8	0.94	0.91 ± 2.8	0.98	0.95 ± 2.4	0.97	0.93 ± 2.8
Diagonal	Multi-Person	0.92	0.87 ± 2.7	0.89	0.87 ± 2.7	0.87	0.84 ± 3.3	0.87	0.84 ± 3.3	0.93	0.90 ± 2.6	0.91	0.90 ± 2.8
	Single-Person	0.89	0.85 ± 3.5	0.85	0.83 ± 3.5	0.79	0.74 ± 3.5	0.79	0.75 ± 3.5	0.91	0.88 ± 3.1	0.90	0.88 ± 2.8
	Cross-Person	0.84	0.82 ± 3.5	0.79	0.76 ± 3.5	0.72	0.71 ± 3.7	0.72	0.70 ± 3.7	0.90	0.88 ± 3.5	0.83	0.82 ± 2.7
	LOO	0.90	0.88 ± 2.9	0.87	0.82 ± 2.9	0.84	0.83 ± 3.1	0.84	0.80 ± 3.1	0.95	0.91 ± 2.7	0.89	0.84 ± 3.1
	LOO + FT	0.94	0.90 ± 2.6	0.92	0.90 ± 2.6	0.90	0.85 ± 2.7	0.90	0.88 ± 2.7	0.98	0.95 ± 2.1	0.96	0.93 ± 2.5
Combined	Multi-Person	0.92	0.89 ± 2.6	0.90	0.85 ± 2.6	0.89	0.86 ± 2.8	0.88	0.87 ± 2.8	0.95	0.91 ± 2.2	0.93	0.91 ± 2.6
	Single-Person	0.89	0.84 ± 3.3	0.87	0.86 ± 3.3	0.85	0.82 ± 3.4	0.84	0.82 ± 3.4	0.92	0.89 ± 3.0	0.91	0.88 ± 2.8
	Cross-Person	0.85	0.81 ± 3.4	0.81	0.80 ± 3.4	0.78	0.77 ± 3.5	0.75	0.73 ± 3.5	0.93	0.90 ± 3.3	0.90	0.87 ± 2.6
	LOO	0.91	0.87 ± 2.4	0.88	0.86 ± 2.4	0.85	0.83 ± 3.2	0.84	0.82 ± 3.2	0.96	0.94 ± 2.2	0.94	0.91 ± 2.5
	LOO + FT	0.95	0.93 ± 2.2	0.93	0.92 ± 2.2	0.93	0.91 ± 2.5	0.93	0.89 ± 2.5	0.99	0.98 ± 2.0	0.97	0.92 ± 2.5

Precision. Precision measures the fraction of predicted swipes (or swipe directions) that were actually correct. For MIMO SwipeSense detection, as shown in 3, we observe very high precision in the Multi-Person condition (0.95, CV ± 2.5%), which gradually declines under more stringent scenarios—dropping to 0.87 (± 3.2%) in Cross-Person and holding at 0.93 (± 2.5%) for LOO. Fine-tuning the held-out user (LOO + FT) pushes detection precision up further to 0.96 (± 2.2%). A similar pattern holds for the classification task: precision starts at 0.91 (± 3.2%) in Multi-Person, dips to 0.79 (± 3.5%) in Cross-Person, and recovers to 0.94 (± 2.8%) after LOO + FT. The relatively low CV values (2–-3%) underscore that these estimates are stable across different data folds.

Recall. Recall (true positive rate) quantifies how many of the actual swipes (or swipe directions) were correctly identified. In detection, as seen in 3, recall is 0.93 (± 2.5%) for Multi-Person, falls to 0.82 (± 3.2%) under Cross-Person, and improves to 0.90 (± 2.5%) with simple leave-one-user-out evaluation. After fine-tuning (LOO + FT), detection recall jumps to 0.94 (± 2.2%). Classification recall follows the same trend: 0.91 (± 3.2%) ⟶ 0.77 (± 3.5%) ⟶ 0.86 (± 3.0%) ⟶ 0.91 (± 2.8%). Together, these recall values show that even under the hardest generalization tests, fine-tuning can recover nearly all true positives.

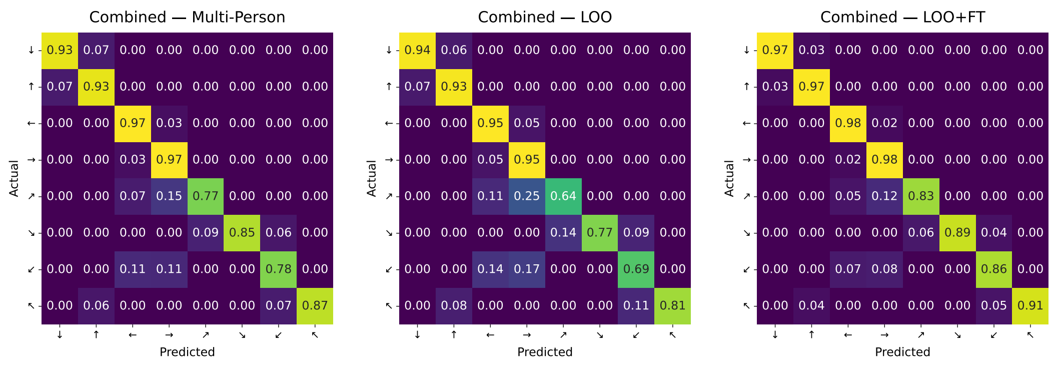
Single-Test Weighted F1 Scores. We evaluated each condition using a standard approach (a 70–20–10 split or equivalent) that separates the data into training, validation, and test sets. A high F1 score indicates the system achieves a good balance between rarely missing gestures (high recall) and rarely misclassifying gestures (high precision). The weighted F1 score combines both precision and recall into one metric, providing a balanced measure of model performance, especially when some gestures appear more frequently than others. As shown in Table 1, MIMO SwipeSense consistently outperforms both SISO and the simpler SVM model across most gestures. Importantly, when testing on a completely unseen user (LOO), performance only slightly decreases (by 2–5%), demonstrating that the model generalizes well to new users. Further, fine-tuning on a small amount of data from the previously unseen user (LOO + FT) restores performance, even surpassing the original baseline by 3–4%. Overall, gestures performed in the PERPENDICULAR directions were recognized more accurately than those in the DIAGONAL directions, likely because perpendicular swipes produce clearer motion patterns.

AUC. The Area Under the ROC Curve (AUC) summarizes the trade-off between sensitivity and specificity across all detection thresholds. For detection, as seen in 3, AUC is 0.97 (± 2.3%) in Multi-Person, dips slightly to 0.93 (± 3.3%) in Cross-Person, and reaches 0.99 (± 2.0%) after LOO

+ FT. Classification AUC is similarly strong: 0.95 (± 3.0%) ⟶ 0.90 (± 2.6%) ⟶ 0.97 (± 2.5%) for Multi-Person, Cross-Person, and LOO + FT, respectively. These high AUC values (all well above 0.9) indicate that MIMO SwipeSense can robustly discriminate between true and false swipe events—and between the eight directional classes—even when exposure to a new user is limited. Because AUC integrates performance over all possible decision thresholds (rather than at a single cutoff), it often appears higher: the model's output probabilities remain well-separated for correct vs. incorrect classes, even if a fixed-threshold F1 score is lower.

10-Fold Cross-Validation. We further validated our models using a 10-fold CV to mitigate potential biases from a single train–test split. Each CV result is presented as "Mean ± SD" (Table 1), offering insights into the model's stability across multiple data folds. Smaller standard deviation values indicate consistent performance, whereas larger values may suggest sensitivity to training splits or class imbalance.

Key Comparison Takeaways. **PERPENDICULAR × DIAGONAL:** Across all methods, PERPENDICULAR swipes consistently achieve higher F1, precision, and recall than DIAGONAL, reflecting the more distinct IMU signatures of cardinal motions. **Cross-User Generalization (LOO):** Holding out an entire user (LOO) incurs only a moderate F1 drop of 2–5% relative to Multi-Person, and similarly causes small declines in precision (≈ 0.95% ⟶ ≈ 0.93%) and recall (≈ 0.93% ⟶ ≈ 0.90%). This underscores that our model's features transfer reasonably well to new users. **User-Specific Adaptation (LOO + FT):** Fine-tuning just the top dense layers on 20% of the held-out user's data not only recovers but often exceeds the Multi-Person baseline—boosting F1 by 3–4%, raising precision back up to 0.96, and recall to 0.94. This demonstrates that a small amount of per-user data yields rapid personalization. **Precision & Recall Trajectories:** Both precision and recall follow the same pattern—high in Multi-Person, dip under Cross-Person, rebound with LOO, and peak with LOO + FT—while their CV ranges (±2–3%) indicate stable estimates across folds. **AUC Robustness:** MIMO SwipeSense maintains AUCs above 0.90 in all scenarios (e.g. detection AUC: 0.97 ⟶ 0.93 ⟶ 0.99), since its soft outputs remain well-separated even when a fixed-threshold F1 dips. **SVM × Deep Learning:** Both SISO and MIMO SwipeSense outperform the SVM baseline in nearly every condition, with MIMO generally 2–3% ahead of SISO. Even with LOO, deep models adapt better than static kernels, and fine-tuning in MIMO yields the largest gains.



(a) Multi-Person (b) LOO (c) LOO + FT

Fig. 5. Confusion matrices for the Combined gesture set under three training scenarios: (a) Multi-Person, (b) Leave-One-User-Out, and (c) LOO + Fine-Tuning.

6.3 Confusion Matrices

To illustrate where errors occur, Figure 5a, Figure 5b, and Figure 5c show the 8×8 confusion matrices for the Combined gesture set under (1) Multi-Person, (2) Leave-One-User-Out (LOO), and (3) LOO + FT. As expected, LOO (middle) exhibits a clear rise in misclassifications—particularly among gestures whose IMU signatures produce similar motion dynamics. After user-specific fine-tuning (right), these confusions are largely depleted: for instance, the misclassification rate of $\nearrow$ as $\longrightarrow$ drops from $\approx$ 24.9% to $\approx$ 11.6%, and $\swarrow$ as $\longleftarrow$ errors fall from $\approx$ 14.0% to $\approx$ 6.5%. By freezing the convolutional feature extractor we preserve general swipe representations, and by adapting only the final dense layers on a small held-out user sample, we realign the classifier to that user's unique motion patterns, restoring and even improving upon the Multi-Person baseline in key cell accuracies.

6.4 Qualitative Results

Participants were asked how comfortable the gestures were on a 5-point Likert scale from (1) "least comfortable" to (5) "most comfortable" (Figure 6). Using the Wilcoxon signed rank test with continuity correction, PERPENDICULAR swipes ($MD = 4.5, IQR = 1.0$) were significantly more comfortable than DIAGONAL swipes ($MD = 2.0, IQR = 0.25$) ($W = 0.0, p < 0.0005, r = 0.66$).

When asked which gestures were the most comfortable, all participants agreed PERPENDICULAR (Figure 2) gestures were the most comfortable compared to DIAGONAL (Figure 2) as our participants found that the movements were more natural for their fingers and wrists (e.g., *"Easier on the wrist. Movement is more natural. You are used to doing perpendicular swipes anyways."* — P1 *"They're easier to control, and the motions feel very familiar."* — P8). In particular, $\uparrow$ and $\downarrow$ were preferred over $\longleftarrow$ or $\longrightarrow$ (N = 8) (e.g., *"Felt like bottom to top was most comfortable as it felt natural. It did not restrict my movement a lot. "* — P2 and *"Swiping top to bottom was particularly satisfying because it felt ergonomic and natural."* — P12). Some of our participants observed that the BoD interaction closely mimics regular smartphone actions. They noted that the gestures and movements required for BoD

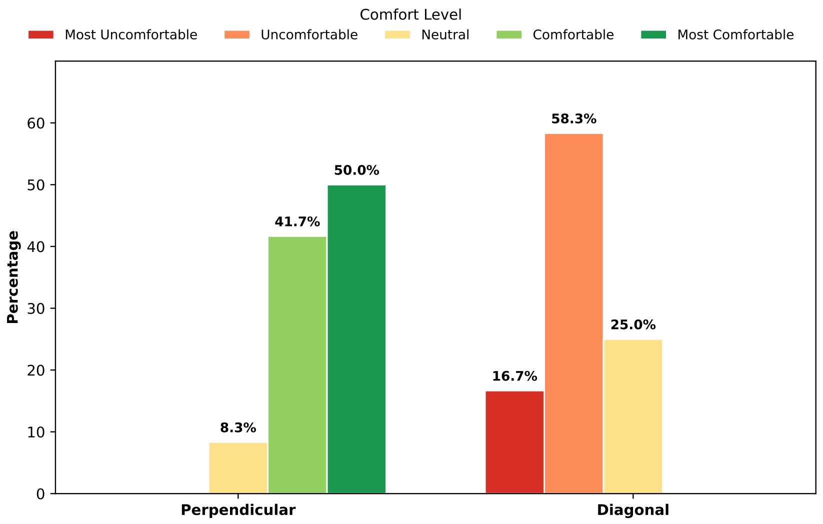


Fig. 6. Participant comfort ratings on a 5-point Likert scale reveal that perpendicular swipes (left) elicited overwhelmingly positive responses—91.7% of ratings fell in the "Comfortable" or "Most Comfortable" categories—whereas diagonal swipes (right) were predominantly uncomfortable, with 75% of ratings in the "Uncomfortable" or "Most Uncomfortable" categories.

interaction were similar to those typically used on the front screen of smartphones, making the transition to this method feel intuitive and familiar. This similarity could potentially facilitate easier adoption and more natural interactions for users accustomed to conventional touchscreen usage. (e.g., *"They felt very natural, like using a touchscreen. Swiping up and down especially mimics scrolling behaviour we do daily."* — P6 and *"These gestures are straightforward and feel like an extension of the natural way I already interact with the phone."* — P7).

On the other hand, DIAGONAL were the least comfortable. It was very restrictive to the point where the phone is slipping (e.g., *"There is a lot of pressure on the wrist and you are struggling to keep the phone popping out of the hand."* — P5 and *"They felt clumsy and were hard to execute properly. My fingers struggled to move diagonally without tilting the phone."* — P10). Participants also note ↗ and ↙ were the most unnatural among DIAGONAL gestures (e.g., *"My fingers kept slipping when trying to get the angle right. It was especially difficult to swipe bottom left to top right smoothly."* — P4 and *"Going from top right to bottom left was particularly awkward."* — P9). Due to the restrictiveness, participants were not confident if the angle was correct (e.g., *"They were challenging to execute with precision. I often second-guessed whether my gesture was diagonal enough"* — P12).

7 Discussion

7.1 Swipe Detection and User Preferences

Preference for PERPENDICULAR Swipes. Our findings indicate a clear preference for PERPENDICULAR swipes, aligning with patterns from prior user-defined gesture studies. Notably, Shimon et al. [46] report that *"the vast majority of elicited gestures were phone-oriented, meaning swipes were made along the vertical and horizontal axes of the phone"*. In our study, 8 out of 12 participants explicitly identified ↓ and ↑ swipes as the most comfortable gestures. This aligns with Shimon's [46] observation that users often replicate familiar front-screen interactions. Supporting this, recent research highlights that scrolling comprises a significant portion of overall smartphone interaction events [38], potentially explaining why participants naturally adapted (↓ and ↑)swipes for BoD interactions.

Indeed, F1 scores corroborate this user preference: PERPENDICULAR swipes demonstrated higher classification accuracy than DIAGONAL gestures. For instance, under single-person training, ↓ and ↑ achieved true positive rates of 87.5% and 86.8%, respectively, but dropped to 76.6% and 75.7% in cross-person testing. These performance reductions are likely influenced by varied grip styles and swipe velocities observed in our video recordings, even when participants believed they were executing "similar" PERPENDICULAR gestures. This discrepancy underlines the influence of individual biomechanics and real-time phone handling techniques, factors that may be less pronounced in simpler tap-based interactions [24].

Challenges with DIAGONAL Swipes. DIAGONAL swipes emerged as a potential point of discomfort, with 4 of the 12 participants dropping their phones during these gestures. This observation underscores the practical ergonomics of grip maintenance. Viet Le's [33] study of BoD input corroborates this by showing that users rely on a "comfortable area" for index-finger swipes. Once DIAGONAL gestures force the finger beyond that zone, slippage or errors increase, as manifested in our phone-drop incidents. In addition to grip discomfort, DIAGONAL gestures also introduce ambiguities in angle and trajectory, making them more prone to classification errors, especially in cross-person settings where individual biomechanics (speed, strength, angle) diverge widely.

Single-Person vs. Cross-Person F1. Comparisons between single-person and cross-person F1 scores further highlight the user-specific nature of swipes [1]. Whereas a single-person model yields fewer false positives, cross-person generalization remains more elusive. These findings differ from earlier tap-based research [24], where training on one individual sometimes generalized surprisingly well to others. In our study, DIAGONAL swipes such as ↗ and ↙ were often misclassified as → or ←. Specifically, ↗ was misclassified as → 20.5% of the time and as ← 9.8%. Similarly, ↙ was misclassified as → 14.2% of the time and as ← 14.5%. This misclassification pattern and the performance decline in cross-person evaluation imply that swipe gestures incorporate a stronger biomechanical component, unlike taps. Each individual's speed, strength, and angle appear to affect classification outcomes more than discrete taps do [1, 24].

7.2 Failure Cases and Mitigations

While BoD swipe gestures offer clear benefits, possible errors for real-world use need to be kept in consideration. The confusion matrices reveal that certain diagonals routinely get confused with specific cardinal directions. For example, ↗ and ↙ can be recognized as → and ← ($\sim 6-11\%$ after fine-tuning). For real-world use case, these gestures should not be combined into one context.

During everyday activities such as walking, typing, or cooking, small IMU perturbations can easily be mistaken for back-of-device swipes. Fortunately, prior IMU-based gesture work has demonstrated that false-positive rates can be driven below 1% by combining three complementary techniques—context-aware gating [41, 52], temporal buffering [27], and adaptive classification thresholds [41]. Building on these proven methods, we can gate swipe detection to dedicated UI contexts (e.g., only when a "back-swipe" mode is active), employ temporal buffering (require swipes to exceed both a minimum duration and magnitude), or adaptively raise classification thresholds while the user is "in motion". We expect that, by integrating these strategies together with selectively avoiding the worst-offender gestures, we can substantially reduce false positives and make Bod swipes both powerful and safe in everyday scenarios.

7.3 Potential Applications

In this section, we next document some potential use cases where SwipeSense can enhance user interaction. Prior work confirms that swiping offers compelling benefits over traditional taps or scrolling: Shimon et al. [46] report that participants naturally replicate front-screen swipe gestures on the phone's back. In contrast, Dou and Sundar [12] showed that introducing swipes increases user engagement and reduces navigational effort in mobile websites. Similarly, Choi et al. [5] found distinct advantages for swipe-based interactions in mobile shopping contexts.

In our post-study questionnaire, we also asked participants to suggest where BoD swipes could fit into their daily routines. Many drew parallels to existing front-screen gestures: for instance, P2 proposed using DIAGONAL swipes to control zoom functions, and P9 recommended horizontal swipes for next/previous song selection. These suggestions mirror our findings that PERPENDICULAR swipes are best suited for frequent, repetitive tasks, whereas DIAGONAL swipes — perceived as more restrictive — can be reserved for less frequent operations. Figure 7 illustrates three potential applications inspired by these insights.

(a) **Email Navigation:** A bounding box shifts via PERPENDICULAR swipes (↑, ↓) to highlight different emails. A ← swipe opens the highlighted email. This design eliminates the need for thumb stretches, making browsing emails one-handed while holding a cup or grocery bag possible.
(b) **Map Interaction:** Swiping DIAGONAL (↖ or ↘) performs zoom in/out, minimizing on-screen occlusion, a significant issue in map-based mobile tasks [44]. Users retain a clear view of the map by offloading zoom gestures to the phone's backside.
(c) **Music Controls:** Horizontal PERPENDICULAR swipes (→ or ←) skip forward or backward between songs, while a ↖ swipe toggles pause/play. Because pausing or resuming tracks is less frequent, assigning a DIAGONAL gesture aligns well with participants' feedback that diagonal swipes feel

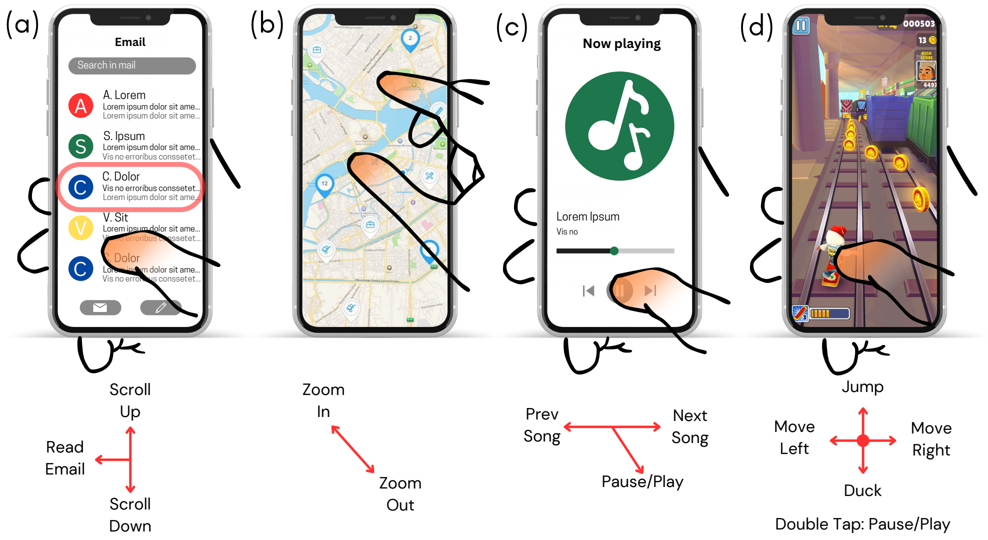


Fig. 7. Example applications that off-load common front-screen actions to back-of-device (BoD) gestures.The semi-transparent orange overlays, shown consistently in all four mock-ups, depict the screen area that a conventional front-screen gesture would occlude. **(a)** Email navigation: vertical PERPENDICULAR swipes (↑, ↓) select messages, and a ← swipe opens the highlighted email. **(b)** Maps: DIAGONAL swipes (↖, ↘) provide zoom in/out with minimal occlusion. **(c)** Music controls: horizontal PERPENDICULAR swipes (→, ←) skip tracks, and a ↖ swipe pauses/plays. Frequent actions are mapped to PERPENDICULAR gestures, while less frequent commands use DIAGONAL swipes. **(d)** Subway-Surfer-Style Runner: PERPENDICULAR swipes (↑ jump, ↓ duck, →/← lane-change) and a BoD Double Tap (pause/play) keep the player's thumb off the display.

less comfortable for repetitive actions. Moreover, this example illustrates how PERPENDICULAR and DIAGONAL swipes can coexist in a single interface, further expanding the overall gesture space.

(d) Subway-Surfer-Style Runner: Fast-paced endless-runner games are driven entirely by front-screen swipes (↑ jump, ↓ duck, →/← lane-change). When the game speeds up, When the game speeds up, a one-handed player's thumb must pivot quickly across the display, hiding up to half the screen at the critical moment before impact [44] (see orange overlay in d. Fig. 7). Off-loading those swipes to the back keeps the track and oncoming obstacles visible, reducing costly crashes. The small pause icon in the top-left corner suffers from the classic occlusion & "fat-finger" problems—finger covers the target and its contact area dwarfs a single pixel [53].

In each scenario, PERPENDICULAR swipes facilitate high-frequency commands (scrolling, skipping tracks), whereas DIAGONAL swipes handle lower-frequency operations (zooming, pausing, playing) without cluttering the front screen. This balance capitalizes on users' existing swipe habits while mitigating ergonomics issues, suggesting wide-ranging possibilities for incorporating BoD gestures into future mobile interfaces. Finally, the Subway Surfers [8] gaming scenario highlights the need to envision the design of jointly sensing BoD tapping and swiping. A practical way forward can be a two-stage IMU pipeline: (i) a lightweight temporal-energy detector that separates impulsive taps from longer-duration swipes using windowed variance and peak-to-RMS ratio, followed by (ii) a shared convolutional direction classifier for the swipe branch. Such a unified model would enlarge

[8]https://subwaysurfers.com/, accessed 7 May, 2025

the BoD gesture vocabulary while keeping computation wholly on-device, and is an exciting avenue for future work.

7.4 Limitations and Future Work

Although SwipeSense demonstrated robust performance in our controlled lab study, our participant pool was small and homogeneous, only 12 right-handed university students (ages 19–23), which limits statistical power and may not reflect broader grip styles or demographics. Evaluating SwipeSense in everyday scenarios such as those proposed in Section 7.3 would help verify its resilience not only to environmental noise and user distractions, but also to varied motion contexts like walking, device handling in transit, and background vibration.

Additionally, our collected data and analysis did not incorporate extensive personalization for individual variations, such as finger size or hand ergonomics. While single-person results indicated strong performance with 800 swipes per user, future work could gather larger, more diverse datasets per participant to refine model stability and further investigate personal calibration mechanisms. Such an approach would allow SwipeSense to tailor swipe sensitivity and threshold parameters to each user's biomechanics, thereby improving both comfort and accuracy.

Furthermore, while our study relied on post-trial observations, future designs could explore real-time contextual awareness to differentiate intentional swipes from incidental movements. For instance, leveraging motion state (e.g., walking vs. sitting) or ongoing app usage (e.g., media playback, navigation) could help SwipeSense interpret swipe data more accurately and thus mitigate false positives. This approach would be especially valuable for continuous gestures like DIAGONAL swipes, where small grip adjustments or background motion may influence classification. Incorporating such contextual cues can better prepare SwipeSense for adaptability and user satisfaction in real-world scenarios.

Overall, addressing these limitations — real-world validation, user-specific calibration and contextual adaptability — stands to strengthen SwipeSense's applicability and user satisfaction in practical, everyday mobile interactions.

8 Conclusion

We introduced SwipeSense, a novel back-of-device swipe interaction technique leveraging built-in smartphone IMU sensors. Designed to enhance usability when the front screen is inaccessible — such as during physical activities or when carrying objects — SwipeSense reduces screen occlusion, enabling more seamless and intuitive interactions.

Our study evaluates the feasibility of detecting swipe directions using only built-in sensors, eliminating the need for additional hardware. SwipeSense accurately recognizes distinct swipes with 72%–95% accuracy, supporting both basic navigation and complex commands through simple BoD gestures. Thus, SwipeSense complements the existing research and expands the BoD gesture set. SwipeSense supports two primary gesture sets: PERPENDICULAR swipes, enabling cardinal-direction interactions, and DIAGONAL swipes, allowing for more advanced and less frequent commands. By expanding gesture-based controls, SwipeSense enhances mobile usability across various scenarios, setting a new standard for intuitive smartphone interactions.

Acknowledgments

We would like to thank all our participants. We acknowledge funding from the NSERC Discovery Grant (210587) to the last author

References

[1] Chris Bevan and Danaë Stanton Fraser. 2016. Different strokes for different folks? Revealing the physical characteristics of smartphone users from their swipe gestures. *International Journal of Human-Computer Studies* 88 (2016), 51–61.

[2] Rich Caruana. 1997. Multitask learning. *Machine learning* 28 (1997), 41–75.

[3] Ke-Yu Chen, Shwetak N. Patel, and Sean Keller. 2016. Finexus: Tracking Precise Motions of Multiple Fingertips Using Magnetic Sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, San Jose California USA, 1504–1514. https://doi.org/10.1145/2858036.2858125

[4] Liang Chen. 2024. BackC&P: Augmenting Copy and Paste Operations on Mobile Touch Devices Through Back-of-device Interaction. *International Journal of Advanced Computer Science and Applications* 15, 11 (2024), xx–yy. https://doi.org/10.14569/IJACSA.2024.0151193

[5] Ben C. F. Choi, Samuel N. Kirshner, and Yi Wu. 2016. Swiping vs. Scrolling in Mobile Shopping Applications. In *HCI in Business, Government, and Organizations: eCommerce and Innovation*, Fiona Fui-Hoon Nah and Chuan-Hoo Tan (Eds.). Springer International Publishing, Cham, 177–188.

[6] Christian Corsten, Bjoern Daehlmann, Simon Voelker, and Jan Borchers. 2017. BackXPress: Using Back-of-Device Finger Pressure to Augment Touchscreen Input on Smartphones. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, Denver Colorado USA, 4654–4666. https://doi.org/10.1145/3025453.3025565

[7] Corinna Cortes and Vladimir Vapnik. 1995. Support-Vector Networks. *Mach. Learn.* 20, 3 (Sept. 1995), 273––297. https://doi.org/10.1023/A:1022627411411

[8] Gabriele Costante, Lorenzo Porzi, Oswald Lanz, Paolo Valigi, and Elisa Ricci. 2014. Personalizing a smartwatch-based gesture interface with transfer learning. In *2014 22nd European Signal Processing Conference (EUSIPCO)*. IEEE, IEEE Press, Piscataway, NJ, USA, 2530–2534.

[9] Wenzhe Cui, Suwen Zhu, Zhi Li, Zheer Xu, Xing-Dong Yang, IV Ramakrishnan, and Xiaojun Bi. 2021. BackSwipe: Back-of-device Word-Gesture Interaction on Smartphones. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 196, 12 pages. https://doi.org/10.1145/3411764.3445081

[10] Agastasya Dahiya, Dhruv Wadhwa, Rohan Katti, and Luigi G. Occhipinti. 2024. Efficient Hand Gesture Recognition Using Artificial Intelligence and IMU-Based Wearable Device. *IEEE Sensors Letters* 8, 12 (2024), 1–4. https://doi.org/10.1109/LSENS.2024.3501586

[11] Alexander De Luca, Emanuel Von Zezschwitz, Ngo Dieu Huong Nguyen, Max-Emanuel Maurer, Elisa Rubegni, Marcello Paolo Scipioni, and Marc Langheinrich. 2013. Back-of-device authentication on smartphones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Paris France, 2389–2398. https://doi.org/10.1145/2470654.2481330

[12] Xue Dou and S Shyam Sundar. 2016. Power of the swipe: Why mobile websites should add horizontal swiping to tapping, clicking, and scrolling interaction techniques. *International journal of human-computer interaction* 32, 4 (2016), 352–362.

[13] Audrey Girouard, Jessica Lo, Md Riyadh, Farshad Daliri, Alexander Keith Eady, and Jerome Pasquero. 2015. One-Handed Bend Interactions with Deformable Smartphones. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1509–1518.

[14] Mayank Goel, Jacob Wobbrock, and Shwetak Patel. 2012. GripSense: using built-in sensors to detect hand posture and pressure on commodity mobile phones. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, Cambridge Massachusetts USA, 545–554. https://doi.org/10.1145/2380116.2380184

[15] Emilio Granell and Luis A. Leiva. 2016. Less Is More: Efficient Back-of-Device Tap Input Detection Using Built-in Smartphone Sensors. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces* (Niagara Falls, Ontario, Canada) *(ISS '16)*. Association for Computing Machinery, New York, NY, USA, 5––11. https://doi.org/10.1145/2992154.2992166

[16] Kyohei Hakka, Toshiya Isomoto, and Buntarou Shizuki. 2019. One-Handed Interaction Technique for Single-Touch Gesture Input on Large Smartphones. In *Symposium on Spatial User Interaction*. ACM, New Orleans LA USA, 1–2. https://doi.org/10.1145/3357251.3358750

[17] Chris Harrison, Julia Schwarz, and Scott E. Hudson. 2011. TapSense: enhancing finger interaction on touch surfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, Santa Barbara California USA, 627–636. https://doi.org/10.1145/2047196.2047279

[18] Khalad Hasan, Junhyeok Kim, David Ahlström, and Pourang Irani. 2016. Thumbs-Up: 3D Spatial Thumb-Reachable Space for One-Handed Thumb Interaction on Smartphones. In *Proceedings of the 2016 Symposium on Spatial User Interaction*. ACM, Tokyo Japan, 103–106. https://doi.org/10.1145/2983310.2985755

[19] Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sathiamoorthy, Yihua Chen, Rahul Mazumder, Lichan Hong, and Ed Chi. 2021. Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning. *Advances in Neural Information Processing Systems* 34 (2021), 29335–29347.

[20] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their applications* 13, 4 (1998), 18–28.

[21] David Holman, Andreas Hollatz, Amartya Banerjee, and Roel Vertegaal. 2013. Unifone: designing for auxiliary finger input in one-handed mobile interactions. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*. ACM, Barcelona Spain, 177–184. https://doi.org/10.1145/2460625.2460653

[22] Christian Holz and Patrick Baudisch. 2010. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 581–590.

[23] Christian Holz and Patrick Baudisch. 2011. Understanding touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2501–2510.

[24] Michael Xuelin Huang, Yang Li, Nazneen Nazneen, Alexander Chao, and Shumin Zhai. 2021. TapNet: The Design, Training, Implementation, and Applications of a Multi-Task Learning CNN for Off-Screen Mobile Input. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 282, 11 pages. https://doi.org/10.1145/3411764.3445626

[25] Michael Xuelin Huang, Yang Li, Nazneen Nazneen, Alexander Chao, and Shumin Zhai. 2021. TapNet: The Design, Training, Implementation, and Applications of a Multi-Task Learning CNN for Off-Screen Mobile Input. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–11.

[26] Zheng Huang, Jed Limke, and Jun Kong. 2016. A Comparative Study on Inter-Device Interaction: One-Handed Interaction VS Two-Handed Interaction. In *Proceedings of the 9th International Symposium on Visual Information Communication and Interaction*. ACM, New York, NY, USA, 67–74.

[27] Peiqi Kang, Jinxuan Li, Bingfei Fan, Shuo Jiang, and Peter B Shull. 2021. Wrist-worn hand gesture recognition while walking via transfer learning. *IEEE Journal of Biomedical and Health Informatics* 26, 3 (2021), 952–961.

[28] Amy K. Karlson and Benjamin B. Bederson. 2007. ThumbSpace: Generalized One-Handed Input for Touchscreen-Based Mobile Devices. In *Human-Computer Interaction – INTERACT 2007*, Cécilia Baranauskas, Philippe Palanque, Julio Abascal, and Simone Diniz Junqueira Barbosa (Eds.). Vol. 4662. Springer Berlin Heidelberg, Berlin, Heidelberg, 324–338. https://doi.org/10.1007/978-3-540-74796-3_30 Series Title: Lecture Notes in Computer Science.

[29] Insu Kim, Jaemin Choi, Suhyeon Shin, JunSeob Kim, Mucheol Kim, Sungrae Cho, and Hyosu Kim. 2023. Back-of-device tap recognition by leveraging magnetometer in commodity smartphones. In *2023 14th International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, IEEE Press, Piscataway, NJ, USA, 656–658.

[30] Joonhyun Kim, Jungsoo Lee, and Wansoo Kim. 2024. Transferable Convolutional Neural Networks for IMU-based Motion Gesture Recognition in Human-Machine Interaction. In *2024 24th International Conference on Control, Automation and Systems (ICCAS)*. IEEE Press, Piscataway, NJ, USA, 61–66. https://doi.org/10.23919/ICCAS63016.2024.10773204

[31] Andy Kong, Karan Ahuja, Mayank Goel, and Chris Harrison. 2021. EyeMU Interactions: Gaze + IMU Gestures on Mobile Devices. In *Proceedings of the 2021 International Conference on Multimodal Interaction*. ACM, Montréal QC Canada, 577–585. https://doi.org/10.1145/3462244.3479938

[32] Satvik Kulshreshtha and Ahmed Sabbir Arif. 2020. Woodpecker: Secret Back-of-Device Tap Rhythms to Authenticate Mobile Users. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE Press, Piscataway, NJ, USA, 2727–2733. https://doi.org/10.1109/SMC42975.2020.9283239

[33] Huy Viet Le, Sven Mayer, Patrick Bader, and Niels Henze. 2018. Fingers' Range and Comfortable Area for One-Handed Smartphone Interaction Beyond the Touchscreen. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–12.

[34] Huy Viet Le, Sven Mayer, and Niels Henze. 2018. InfiniTouch: Finger-Aware Interaction on Fully Touch Sensitive Smartphones. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. ACM, Berlin Germany, 779–792. https://doi.org/10.1145/3242587.3242605

[35] Miaomiao Liu, Sikai Yang, Wyssanie Chomsin, and Wan Du. 2023. Real-Time Tracking of Smartwatch Orientation and Location by Multitask Learning. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems* (Boston, Massachusetts) *(SenSys '22)*. Association for Computing Machinery, New York, NY, USA, 120–133. https://doi.org/10.1145/3560905.3568548

[36] Christian Loclair, Sean Gustafson, and Patrick Baudisch. 2010. PinchWatch: A Wearable Device for One-Handed Microinteractions. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services Workshop on Ensembles of On-Body Devices (MobileHCI '10)*. ACM, New York, NY, USA, xx–yy.

[37] H. Nematallah and S. Rajan. 2020. Comparative Study of Time Series-based Human Activity Recognition using Convolutional Neural Networks. In *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)* (Dubrovnik, Croatia). IEEE Press, Piscataway, NJ, USA, 1––6. https://doi.org/10.1109/I2MTC43012.2020.9128582

[38] Beryl Noë, Liam D Turner, David EJ Linden, Stuart M Allen, Bjorn Winkens, and Roger M Whitaker. 2019. Identifying indicators of smartphone addiction through user-app interaction. *Computers in human behavior* 99 (2019), 56–65.

[39] Farzan Majeed Noori, Enrique Garcia-Ceja, Md. Zia Uddin, Michael Riegler, and Jim Tørresen. 2019. Fusion of Multiple Representations Extracted from a Single Sensor's Data for Activity Recognition Using CNNs. In *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE Press, Piscataway, NJ, USA, 1–6. https://doi.org/10.1109/IJCNN.2019.8851898

[40] Philip Quinn, Seungyon Claire Lee, Melissa Barnhart, and Shumin Zhai. 2019. Active Edge: Designing Squeeze Gestures for the Google Pixel 2. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1––13. https://doi.org/10.1145/3290605.3300504

[41] Samir A Rawashdeh, Derek A Rafeldt, and Timothy L Uhl. 2016. Wearable IMU for shoulder injury prevention in overhead sports. *Sensors* 16, 11 (2016), 1847.

[42] Huimin Ren, Sijie Ruan, Yanhua Li, Jie Bao, Chuishi Meng, Ruiyuan Li, and Yu Zheng. 2021. MTrajRec: Map-Constrained Trajectory Recovery via Seq2Seq Multi-task Learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 1410––1419. https://doi.org/10.1145/3447548.3467238

[43] Gabriel Reyes, Jason Wu, Nikita Juneja, Maxim Goldshtein, W. Keith Edwards, Gregory D. Abowd, and Thad Starner. 2018. SynchroWatch: One-Handed Synchronous Smartwatch Gestures Using Correlation and Magnetic Sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (Jan. 2018), 1–26. https://doi.org/10.1145/3161162

[44] Anne Roudaut, Stéphane Huot, and Eric Lecolinet. 2008. TapTap and MagStick: Improving one-handed target acquisition on small touch-screens. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (Napoli, Italy) *(AVI '08)*. Association for Computing Machinery, New York, NY, USA, 146––153. https://doi.org/10.1145/1385569.1385594

[45] Karsten Seipp and Kate Devlin. 2014. BackPat: one-handed off-screen patting gestures. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*. ACM, Toronto ON Canada, 77–80. https://doi.org/10.1145/2628363.2628396

[46] Shaikh Shawon Arefin Shimon, Sarah Morrison-Smith, Noah John, Ghazal Fahimi, and Jaime Ruiz. 2015. Exploring User-Defined Back-Of-Device Gestures for Mobile Devices. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, New York, NY, USA, 227–232.

[47] Jie Song, Gábor Sörös, Fabrizio Pece, Sean Ryan Fanello, Shahram Izadi, Cem Keskin, and Otmar Hilliges. 2014. In-air gestures around unmodified mobile devices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 319––329. https://doi.org/10.1145/2642918.2647373

[48] Ke Sun, Ting Zhao, Wei Wang, and Lei Xie. 2018. VSkin: Sensing Touch Gestures on Surfaces of Mobile Devices Using Acoustic Signals. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, New Delhi India, 591–605. https://doi.org/10.1145/3241539.3241568

[49] Yu-Chih Tung and Kang G. Shin. 2016. Expansion of Human-Phone Interface By Sensing Structure-Borne Sound Propagation. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, Singapore Singapore, 277–289. https://doi.org/10.1145/2906388.2906394

[50] Lei Wang, Xiang Zhang, Yuanshuang Jiang, Yong Zhang, Chenren Xu, Ruiyang Gao, and Daqing Zhang. 2021. Watching your phone's back: Gesture recognition by sensing acoustical structure-borne propagation. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 2 (2021), 1–26.

[51] Ruolin Wang, Chun Yu, Xing-Dong Yang, Weijie He, and Yuanchun Shi. 2019. EarTouch: Facilitating Smartphone Use for Visually Impaired People in Mobile and Public Scenarios. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–13. https://doi.org/10.1145/3290605.3300254

[52] Shaofan Wang, Tao Zhang, Yuangan Li, Pengjiao Li, Haopeng Wu, and Ke Li. 2024. Continuous Hand Gestures Detection and Recognition in Emergency Human-Robot Interaction Based on the Inertial Measurement Unit. *IEEE Transactions on Instrumentation and Measurement* 73 (2024), 1–15. https://doi.org/10.1109/TIM.2024.3440381

[53] Daniel Wigdor, Clifton Forlines, Patrick Baudisch, John Barnwell, and Chia Shen. 2007. Lucid touch: A see-through mobile device. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology* (Newport, Rhode Island, USA) *(UIST '07)*. Association for Computing Machinery, New York, NY, USA, 269–278. https://doi.org/10.1145/1294211.1294259

[54] Pui Chung Wong, Hongbo Fu, and Kening Zhu. 2016. Back-Mirror: back-of-device one-handed interaction on smartphones. In *SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications*. ACM, Macau, 1–5. https://doi.org/10.1145/2999508.2999522

[55] Haijun Xia, Michael Glueck, Michelle Annett, Michael Wang, and Daniel Wigdor. 2022. Iteratively designing gesture vocabularies: A survey and analysis of best practices in the HCI literature. *ACM Transactions on Computer-Human Interaction (TOCHI)* 29, 4 (2022), 1–54.

[56] Chang Xiao, Karl Bayer, Changxi Zheng, and Shree K. Nayar. 2021. BackTrack: 2D Back-of-device Interaction Through Front Touchscreen. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 7, 8 pages. https://doi.org/10.1145/3411764.3445374

[57] Xiang Xiao, Teng Han, and Jingtao Wang. 2013. LensGesture: Augmenting mobile interactions with back-of-device finger gestures. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction* (Sydney, Australia) *(ICMI '13)*. Association for Computing Machinery, New York, NY, USA, 287–294. https://doi.org/10.1145/2522848.2522850

[58] Yen-Ting Yeh, Quentin Roy, Antony Albert Raj Irudayaraj, and Daniel Vogel. 2020. Expanding Side Touch Input on Mobile Phones: Finger Reachability and Two-Dimensional Taps and Flicks using the Index and Thumb. *Proceedings of the ACM on human-computer interaction* 4, ISS (2020), 1–20.

[59] Hui-Shyong Yeo, Juyoung Lee, Andrea Bianchi, and Aaron Quigley. 2016. Sidetap & Slingshot Gestures on Unmodified Smartwatches. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, Tokyo Japan, 189–190. https://doi.org/10.1145/2984751.2984763

[60] Chun Yu, Xiaoying Wei, Shubh Vachher, Yue Qin, Chen Liang, Yueting Weng, Yizheng Gu, and Yuanchun Shi. 2019. HandSee: Enabling Full Hand Interaction on Smartphone with Front Camera-based Stereo Vision. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–13. https://doi.org/10.1145/3290605.3300935

[61] Cheng Zhang, Junrui Yang, Caleb Southern, Thad E. Starner, and Gregory D. Abowd. 2016. WatchOut: extending interactions on a smartwatch with inertial sensing. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*. ACM, Heidelberg Germany, 136–143. https://doi.org/10.1145/2971763.2971775

[62] Yang Zhang, Gierad Laput, and Chris Harrison. 2017. Electrick: Low-Cost Touch Sensing Using Electric Field Tomography. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, Denver Colorado USA, 1–14. https://doi.org/10.1145/3025453.3025842

[63] Junhan Zhou, Yang Zhang, Gierad Laput, and Chris Harrison. 2016. AuraSense: Enabling Expressive Around-Smartwatch Interactions with Electric Field Sensing. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, Tokyo Japan, 81–86. https://doi.org/10.1145/2984511.2984568

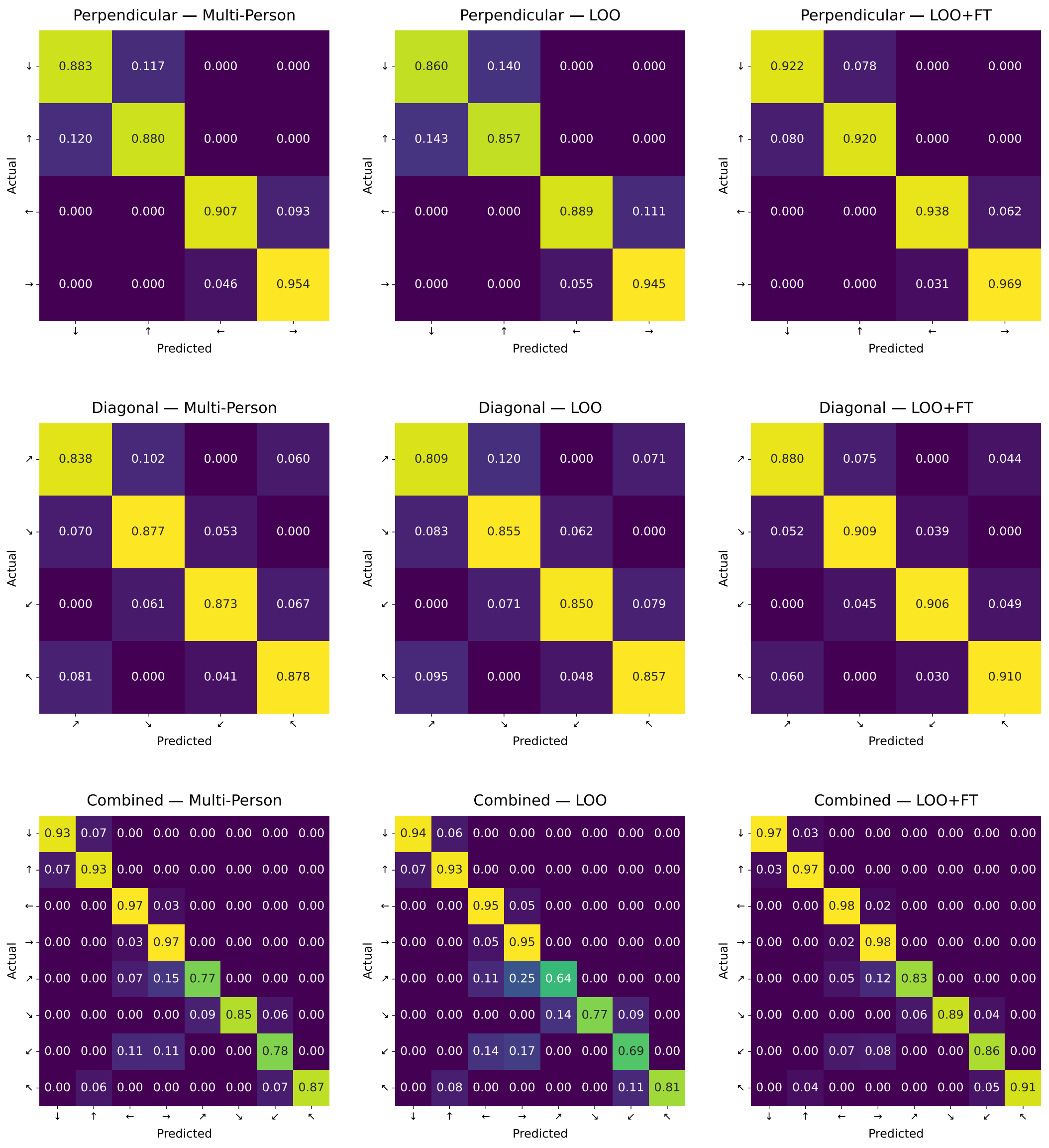
A Appendix

Table 4. SVM baseline: Precision (P), Recall (R), and AUC (Detection / Classification) with 10-fold cross-validation (CV) for each gesture set and training scenario.

		Detection				**Classification**				**AUC**			
Gesture	**Scenario**	P	CV	R	CV	P	CV	R	CV	Det	CV	Clf	CV
Perpendicular	Multi-Person	0.91	0.87±3.0	0.89	0.86±3.0	0.86	0.84±2.8	0.84	0.82±2.8	0.94	0.89±2.5	0.83	0.79±3.0
	Single-Person	0.88	0.86±3.0	0.85	0.80±3.0	0.85	0.82±3.0	0.82	0.78±3.0	0.90	0.87±3.0	0.88	0.87±3.2
	Cross-Person	0.82	0.78±3.4	0.78	0.73±3.4	0.78	0.75±3.6	0.75	0.70±3.6	0.88	0.87±3.1	0.85	0.83±3.4
	LOO	0.90	0.85±2.7	0.88	0.84±2.7	0.88	0.86±2.9	0.85	0.82±2.9	0.93	0.88±2.8	0.87	0.83±2.9
	LOO + FT	0.93	0.88±2.3	0.91	0.88±2.3	0.90	0.89±2.7	0.88	0.87±2.7	0.97	0.92±2.4	0.93	0.89±2.7
Diagonal	Multi-Person	0.89	0.85±2.9	0.86	0.82±2.9	0.84	0.81±2.8	0.81	0.79±2.8	0.91	0.89±2.8	0.89	0.87±3.0
	Single-Person	0.85	0.83±3.5	0.82	0.79±3.5	0.80	0.78±3.9	0.78	0.77±3.9	0.88	0.84±3.2	0.83	0.78±3.1
	Cross-Person	0.79	0.76±3.5	0.75	0.74±3.5	0.76	0.73±3.4	0.73	0.72±3.4	0.85	0.83±3.4	0.78	0.74±3.5
	LOO	0.87	0.83±2.8	0.84	0.82±2.8	0.82	0.81±2.8	0.79	0.77±2.8	0.92	0.89±2.9	0.86	0.85±2.8
	LOO + FT	0.91	0.90±2.4	0.88	0.85±2.4	0.86	0.81±2.7	0.83	0.79±2.7	0.96	0.93±2.6	0.92	0.90±2.5
Combined	Multi-Person	0.90	0.87±2.7	0.87	0.84±2.7	0.88	0.86±2.9	0.86	0.82±2.9	0.92	0.89±2.6	0.90	0.86±2.8
	Single-Person	0.87	0.85±3.2	0.83	0.78±3.2	0.86	0.83±3.2	0.82	0.78±3.2	0.89	0.85±3.1	0.87	0.85±3.2
	Cross-Person	0.81	0.79±3.3	0.76	0.71±3.3	0.79	0.74±3.5	0.73	0.70±3.5	0.88	0.84±3.1	0.85	0.82±3.3
	LOO	0.89	0.87±2.5	0.86	0.81±2.5	0.87	0.84±2.9	0.82	0.77±2.9	0.94	0.91±2.6	0.88	0.86±2.9
	LOO + FT	0.93	0.92±2.2	0.91	0.88±2.2	0.90	0.88±2.7	0.88	0.86±2.7	0.98	0.95±2.3	0.93	0.92±2.6

Table 5. SISO SwipeSense: Precision (P), Recall (R), and AUC (Detection / Classification) with 10-fold cross-validation (CV) for every gesture set and training scenario.

		Detection				**Classification**				**AUC**			
Gesture	**Scenario**	P	CV	R	CV	P	CV	R	CV	Det	CV	Clf	CV
	Multi-Person	0.92	0.90±3.0	0.90	0.87±3.0	0.88	0.83±3.4	0.86	0.85±3.4	0.84	0.81±2.8	0.93	0.90±3.1
	Single-Person	0.89	0.86±3.2	0.87	0.84±3.2	0.86	0.82±3.1	0.84	0.79±3.1	0.90	0.86±3.1	0.88	0.83±3.4
Perpendicular	Cross-Person	0.83	0.78±3.4	0.80	0.78±3.4	0.80	0.76±3.3	0.76	0.72±3.3	0.88	0.83±3.3	0.87	0.82±3.7
	LOO	0.91	0.90±2.9	0.89	0.88±2.9	0.90	0.86±3.2	0.87	0.83±3.2	0.92	0.90±2.9	0.90	0.86±3.2
	LOO + FT	0.94	0.93±2.5	0.92	0.89±2.5	0.92	0.90±2.8	0.90	0.86±2.8	0.96	0.93±2.5	0.94	0.90±2.7
	Multi-Person	0.90	0.86±2.9	0.88	0.85±2.9	0.89	0.85±3.0	0.86	0.84±3.0	0.91	0.88±2.9	0.92	0.88±2.8
	Single-Person	0.86	0.85±3.2	0.84	0.82±3.2	0.85	0.84±3.3	0.82	0.77±3.3	0.90	0.85±2.9	0.88	0.85±2.9
Diagonal	Cross-Person	0.80	0.79±3.4	0.77	0.73±3.4	0.78	0.73±3.6	0.74	0.70±3.6	0.88	0.86±3.6	0.83	0.79±3.7
	LOO	0.88	0.86±2.8	0.85	0.83±2.8	0.88	0.84±3.1	0.84	0.80±3.1	0.89	0.86±2.8	0.90	0.89±2.8
	LOO + FT	0.92	0.91±2.7	0.89	0.87±2.7	0.91	0.86±2.9	0.88	0.86±2.9	0.93	0.89±2.4	0.94	0.92±2.6
	Multi-Person	0.91	0.89±2.4	0.89	0.87±2.4	0.90	0.89±2.7	0.88	0.87±2.7	0.93	0.90±2.4	0.92	0.89±2.7
	Single-Person	0.88	0.86±3.4	0.85	0.82±3.4	0.87	0.85±3.5	0.84	0.82±3.5	0.90	0.89±3.3	0.88	0.83±3.5
Combined	Cross-Person	0.82	0.81±3.5	0.78	0.76±3.5	0.81	0.76±3.7	0.77	0.72±3.7	0.89	0.87±3.4	0.87	0.86±3.5
	LOO	0.90	0.89±2.3	0.88	0.83±2.3	0.89	0.86±3.3	0.86	0.84±3.3	0.94	0.90±2.4	0.92	0.90±2.8
	LOO + FT	0.93	0.90±2.1	0.91	0.90±2.1	0.92	0.90±2.8	0.90	0.85±2.8	0.97	0.94±2.1	0.95	0.94±2.7



Perpendicular — Multi-Person

	↓	↑	←	→
↓	0.883	0.117	0.000	0.000
↑	0.120	0.880	0.000	0.000
←	0.000	0.000	0.907	0.093
→	0.000	0.000	0.046	0.954

Perpendicular — LOO

	↓	↑	←	→
↓	0.860	0.140	0.000	0.000
↑	0.143	0.857	0.000	0.000
←	0.000	0.000	0.889	0.111
→	0.000	0.000	0.055	0.945

Perpendicular — LOO+FT

	↓	↑	←	→
↓	0.922	0.078	0.000	0.000
↑	0.080	0.920	0.000	0.000
←	0.000	0.000	0.938	0.062
→	0.000	0.000	0.031	0.969

Diagonal — Multi-Person

	↗	↘	↙	↖
↗	0.838	0.102	0.000	0.060
↘	0.070	0.877	0.053	0.000
↙	0.000	0.061	0.873	0.067
↖	0.081	0.000	0.041	0.878

Diagonal — LOO

	↗	↘	↙	↖
↗	0.809	0.120	0.000	0.071
↘	0.083	0.855	0.062	0.000
↙	0.000	0.071	0.850	0.079
↖	0.095	0.000	0.048	0.857

Diagonal — LOO+FT

	↗	↘	↙	↖
↗	0.880	0.075	0.000	0.044
↘	0.052	0.909	0.039	0.000
↙	0.000	0.045	0.906	0.049
↖	0.060	0.000	0.030	0.910

Combined — Multi-Person

	↓	↑	←	→	↗	↘	↙	↖
↓	0.93	0.07	0.00	0.00	0.00	0.00	0.00	0.00
↑	0.07	0.93	0.00	0.00	0.00	0.00	0.00	0.00
←	0.00	0.00	0.97	0.03	0.00	0.00	0.00	0.00
→	0.00	0.00	0.03	0.97	0.00	0.00	0.00	0.00
↗	0.00	0.00	0.07	0.15	0.77	0.00	0.00	0.00
↘	0.00	0.00	0.00	0.00	0.09	0.85	0.06	0.00
↙	0.00	0.00	0.11	0.11	0.00	0.00	0.78	0.00
↖	0.00	0.06	0.00	0.00	0.00	0.00	0.07	0.87

Combined — LOO

	↓	↑	←	→	↗	↘	↙	↖
↓	0.94	0.06	0.00	0.00	0.00	0.00	0.00	0.00
↑	0.07	0.93	0.00	0.00	0.00	0.00	0.00	0.00
←	0.00	0.00	0.95	0.05	0.00	0.00	0.00	0.00
→	0.00	0.00	0.05	0.95	0.00	0.00	0.00	0.00
↗	0.00	0.00	0.11	0.25	0.64	0.00	0.00	0.00
↘	0.00	0.00	0.00	0.00	0.14	0.77	0.09	0.00
↙	0.00	0.00	0.14	0.17	0.00	0.00	0.69	0.00
↖	0.00	0.08	0.00	0.00	0.00	0.00	0.11	0.81

Combined — LOO+FT

	↓	↑	←	→	↗	↘	↙	↖
↓	0.97	0.03	0.00	0.00	0.00	0.00	0.00	0.00
↑	0.03	0.97	0.00	0.00	0.00	0.00	0.00	0.00
←	0.00	0.00	0.98	0.02	0.00	0.00	0.00	0.00
→	0.00	0.00	0.02	0.98	0.00	0.00	0.00	0.00
↗	0.00	0.00	0.05	0.12	0.83	0.00	0.00	0.00
↘	0.00	0.00	0.00	0.00	0.06	0.89	0.04	0.00
↙	0.00	0.00	0.07	0.08	0.00	0.00	0.86	0.00
↖	0.00	0.04	0.00	0.00	0.00	0.00	0.05	0.91

Fig. 8. Confusion matrices for *Perpendicular*, *Diagonal*, and *Combined* gesture sets (rows) under three training scenarios—Multi-Person, Leave-One-User-Out (LOO), and LOO + Fine-Tuning (columns). The orange overlay in each heat-map illustrates how front-screen interaction can occlude critical content, reinforcing the value of back-of-device input.