BehaVR: User Identification Based on VR Sensor Data

Ismat Jarin*†
ijarin@uci.edu
University of California, Irvine

Yu Duan*
duany12@uci.edu
University of California, Irvine

Rahmadi Trimananda
rtrimana@uci.edu
University of California, Irvine

Hao Cui
cuih7@uci.edu
University of California, Irvine

Salma Elmalaki
salma.elmalaki@uci.edu
University of California, Irvine

Athina Markopoulou
athina@uci.edu
University of California, Irvine

Abstract

Virtual reality (VR) platforms enable a wide range of applications, however, pose unique privacy risks. In particular, VR devices are equipped with a rich set of sensors that collect personal and sensitive information (*e.g.,* body motion, eye gaze, hand joints, and facial expression). The data from these newly available sensors can be used to uniquely identify a user, even in the absence of explicit identifiers. In this paper, we seek to understand the extent to which a user can be identified based solely on VR sensor data, *within and across* real-world apps from diverse genres. We consider adversaries with capabilities that range from observing APIs available within a single app (app adversary) to observing all or selected sensor measurements across multiple apps on the VR device (device adversary). To that end, we introduce BehaVR, a framework for collecting and analyzing data from *all* sensor groups collected by *multiple* apps running on a VR device. We use BehaVR to collect data from real users that interact with 20 popular real-world apps. We use that data to build machine learning models for user identification within and across apps, with features extracted from available sensor data. We show that these models can identify users with an accuracy of up to 100%, and we reveal the most important features and sensor groups, depending on the functionality of the app and the adversary. To the best of our knowledge, BehaVR is the first to analyze user identification in VR comprehensively, *i.e.,* considering all sensor measurements available on consumer VR devices, collected by multiple real-world, as opposed to custom-made, apps.

1 Introduction

Virtual reality (VR) is a large and growing market [59] that enables a wide range of apps, from gaming to education [49], and work [44]. Meta Quest, one of the most popular consumer VR devices, has sold nearly 20 million units as of February 2023 [32]. SteamVR, the largest VR gaming platform, has over 7,800 VR apps as of May 2024 [67]. The VR ecosystem also comes with privacy concerns. Recent work showed that Oculus VR and its apps already collect personally identifying information [73], and can further infer sensitive attributes [56]. Some of this tracking and profiling are similar to practices in other app ecosystems, such as mobile [16, 64], smart TV [55, 80], web [10], etc. with some differences: the VR ecosystem is younger, more centralized, and not driven by ads, yet [73].

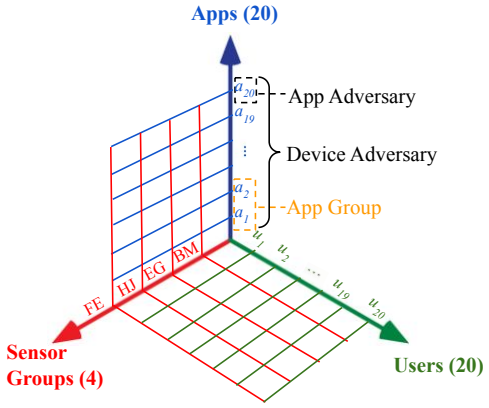


Figure 1: BehaVR problem space spans several dimensions: users, apps, and sensors. We consider four sensor groups: body motion (BM), eye gaze (EG), hand joints (HJ), facial expression (FE). We consider 20 real-world apps covering vast domains of VR apps. We have two types of adversaries: the app adversary has access only to one app; the device adversary has access across multiple apps. We further define App Groups as having similar activities and emotional states.

User Identification. VR has access to a rich set of sensors that capture sensitive, personal information. Consumer VR devices (*e.g.,* Meta Quest Pro), including their headsets and controllers, are equipped with sensors that collect measurements about head and body motion ("BM") [29, 50], eye gaze ("EG") [25, 45], hand joints ("HJ") [26, 46], and facial expression ("FE") [28, 51]. All these measurements are available on the device itself (*e.g.,* Quest Pro), can be sent to the platform (Meta), and a subset can be made available to app developers via APIs. Recent works [53, 57] have shown that some of these measurements can indeed be used for unique identification. The privacy implication is that a user's behavior in VR creates *implicit identifiers*[1] that can be used to identify users in the virtual world, even in the absence of *explicit identifiers* (*e.g.,* device IDs or user accounts) that are often well protected by permissions. Such implicit identifiers based on sensor measurements may remain effective in the case of shared devices (*e.g.,* shared among family members, coworkers, or public platforms), multiple accounts or

*The two authors made equal contributions and share first authorship.
†Corresponding author.

This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license visit https://creativecommons.org/licenses/by/4.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.
Proceedings on Privacy Enhancing Technologies 2025(1), 399–419
© 2025 Copyright held by the owner/author(s).
https://doi.org/10.56553/popets-2025-0022

[1]The combination of features extracted from the VR sensor data streams can produce unique fingerprints, such as behavioral biometrics (or "behaviometrics") [60, 69].

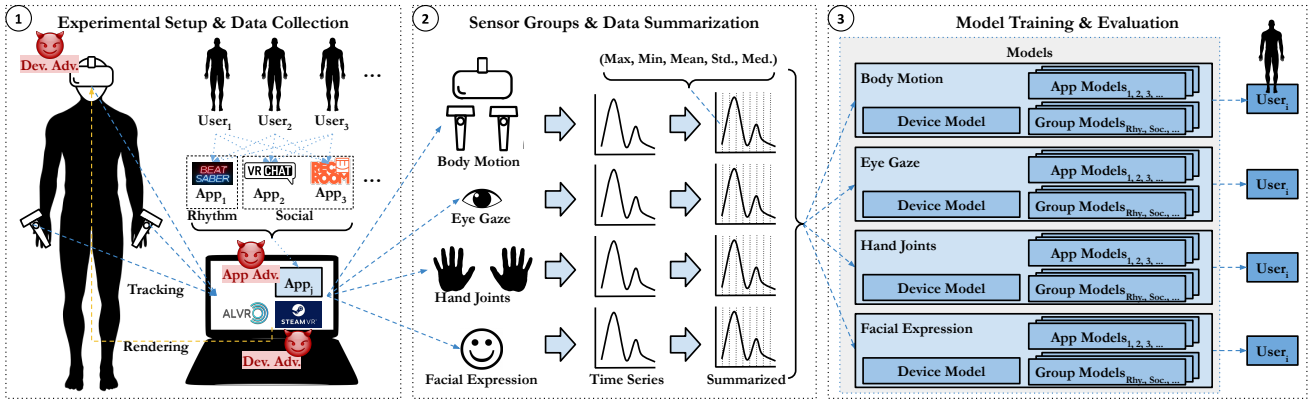


Figure 2: Overview of BehaVR. (1) Data Collection Setup: every user interacts with each app using Quest Pro; each app (*e.g.,* Beat Saber) runs on a PC and its VR environment is rendered on the Quest Pro headset; this enables the recording of sensor data sent from Quest Pro to the PC; apps are grouped based on similarity of activities and emotional states. (2) Data Processing: there are four groups of sensors, namely *body motion*, *eye gaze*, *hand joints*, and *facial expression*; we divide the time series generated by every sensor group into blocks, and we compute 5 *statistics* per block as features. (3) Model Training & Evaluation: using the previous features per block, we train different models (using data per app, across apps, even per group of apps) that an adversary can use to uniquely identify users.

devices per user, anonymized and released VR sensor dataset etc. (more details in Section 2.5). Identification using implicit identifiers has explored in mobile platforms [9] and recently in VR [53, 57].

In this paper, we broadly refer to the sensor measurements collected on VR devices, as well as to features extracted from them, as VR sensor data. We are interested in understanding *to what extent a user can be uniquely identified based on VR sensor data across 20 real-world apps* from diverse genres, from social (*e.g.,* VRChat) to education/training (*e.g.,* X-Plane 11), from entertainment (*e.g.,* BeatSaber) to virtual offices (*e.g.,* Job Simulator) among others; and which are the *top features, across real-world apps and sensor groups* , for an adversary that wants to uniquely identify a user with minimal effort. In particular, we consider two types of adversaries, depending on their vantage point for access to sensor data: (1) the *app adversary* mimics an app developer who has access to sensor data from APIs available within the app; and (2) the *device adversary* can have access to sensor data collected across multiple apps (see Section 2.5). The full problem space we consider is depicted in Fig. 1.

Comparison with Prior Work. Prior work has considered only parts of our problem space. In terms of sensor groups, user identification has been demonstrated based on body motion (*e.g.,* positional and rotational) sensor data from VR devices [53, 56, 57, 72], *i.e.,* the BM sensor group in our problem space. Newer VR devices, such as the Meta Quest Pro and Apple Vision Pro, are equipped with more sensors that track other body parts, including eyes, hands, and face [45, 51, 82]. Privacy aspects of these sensors have been studied before [39, 56], but their use for identification has not been, and neither has been their comparison for identification purposes. BehaVR, for the first time, explore all available sensor groups for identification. In terms of the experimental setup, prior work has focused on either one specific app and task (*e.g.,* Beat Saber in [57]), or custom apps specifically designed for their studies [53, 56, 72]. In BehaVR experiments, participants interact with 20 unmodified commercial VR apps, under limited guidance, better representing real-world scenarios of user identification.

Approach. We introduce BehaVR, a framework for collecting and analyzing data from all available sensor groups (*i.e.,* body motion, eye gaze, hand joints, and facial expression), and performing user identification within (*i.e.,* app adversary) and across (*i.e.,* device adversary) apps. To the best of our knowledge, BehaVR is the first to analyze user identification in VR comprehensively, *i.e.,* considering *all* sensor measurements available on a VR device, and across multiple commercial apps. Fig. 2 presents the overview of BehaVR. Next, we describe the BehaVR components and we highlight methodological contributions along the way.

(1) Collection of sensor data from real-world apps on the VR device. We develop an approach to observing, for the first time, all the sensor data in real-time during gameplay. We instrument ALVR, an open-source streaming app that is essential for Meta Quest devices (and other popular VR headsets) to play SteamVR apps [3], to record all sensor data by listening to the API calls. This gives visibility into data collected by real-world apps running unmodified on the VR device, which was not previously possible. Using the BehaVR setup, we perform a user study and collect a comprehensive dataset that covers all four sensor groups, consisting of around 400 sensor data records from users interacting with 20 popular apps on the SteamVR store (see Section 3.2).

(2) Sensor Data Analysis and Feature Engineering. BehaVR is the first to explore all available sensor groups for identification in real-world apps. The comprehensive sensor data and diverse app genres pose unique challenges for data processing. First, unlike prior work that focuses on specific tasks or deals with fixed time blocks [53] BehaVR dataset exhibits high variability in session time across users, apps, and sensors. To process variable-length time series of sensor data into time blocks we propose a new time block-division approach that is robust to the variability (see Section 4.1). Second,

in addition to the standard features extracted from the VR sensor data [53, 57], we introduce data augmentation and selection that are adapted for new sensor groups: for eye gaze, we add new features that correlate left and right eye's data; for facial expression, we explore facial elements and their combinations that represent users' emotions [18, 28] when interacting with an app. In addition, our feature analysis reveals how activities and valence/arousal states in different apps generate key identification features.

(3) Identification Models and Evaluation. We evaluate user identification in diverse real-world scenarios that cover different adversary capabilities. We train a Machine Learning model per sensor group for user identification. The identification model predicts on each time block and maximum voting [53] is used to produce the final label per user. Depending on the adversary's capability, the model is trained on sensor data from one app (app adversary), one app group, or all apps (device adversary) and evaluated on the same or different setting (*open-world-setting*) of the same app; or a completely different app (*zero-day scenario*) from same or different app groups (Section 5). We discuss the generality of BehaVR (Section 7.1) and provide recommendations for privacy practitioners (Section 7.2).

Identification Insights. Section 5 presents a comprehensive evaluation across sensor groups, apps or groups of apps, and adversary models, guided by the following research questions (RQs):

RQ1: How well a user can be identified using VR sensor data? We find that the adversaries can achieve up to 100% accuracy for many apps, especially using data from facial expression and body motion that perform better than eye gaze and hand joints.

RQ2: How long does it take to identify a user? The app adversary generally requires around 18 – 20 seconds of data across body motion and facial expression for up to 90% , and ∼50 seconds for eye gaze for up to 85% accuracy. The device adversary requires less data (∼9 seconds on average), since it combines data across apps.

RQ3: What are the top features for identification w.r.t. various apps and adversaries? We observe that for the unique identification, the top features describe the unique interaction between users and VR environment as well as user's physical characteristics (*e.g.,* height).

RQ4: Can we identify a user across different settings of same app or across different apps? We find that app adversary can provide 60-100% accuracy with new app settings (*i.e.,* open-world-settings). Device Adversary can identify users in a new app (*i.e.,* zero-day) from the same (70-100%) or different (5-40%) app group.

RQ5: What are the most important sensor groups for identification? Apps from different groups show sensor groups importance based on app activity and emotional states. Knowing the relative importance of different sensor groups allows the adversary to effectively train models or help users to decide which sensor groups to share.

Outline. The rest of the paper is structured as follows: Section 2 provides background and the problem setup. Section 3 presents the experimental setup and data collection. Section 4 presents the data analysis and model training. Section 5 presents the evaluation for app and device adversaries, for all sensor and app groups. Section 6 reviews related work. Section 7 and 8 provides discussion and conclusion respectively. The appendices provide additional results.

2 Background and Problem Setup

2.1 VR Hardware and Platform

There are many different VR platforms and setups that require varied software and hardware combinations. In this paper, we focus on SteamVR, the most popular VR gaming platform with over 7,800 VR apps [67] and millions of users [5]. In the SteamVR setup, the VR apps run on a personal computer (with either Windows or Linux system), and a compatible VR device is connected to stream the graphics and track user actions via its sensors. A streaming software needs to be installed on the PC and the VR device to transmit the graphics and sensor data [3, 4, 79].

In our experimental setup, we use ALVR [3] as the streaming app. ALVR is open-source and thus eases the instrumentation (see Section 3). As for the hardware, we choose Meta Quest Pro for testing [2], because the headset is equipped with the most comprehensive VR sensors, including body motion and eye gaze (which are supported by older VR devices like Quest 2), as well as hand joints and facial expression data (which are increasingly supported by newer generations of VR devices). BehaVR leverages SteamVR to run apps and ALVR [3] to record all sensor data by listening to the API calls (see Section 3). SteamVR (and ALVR) supports for many other consumer VR devices, notably HTC Vive Focus, ByteDance Pico and Apple Vision Pro [3]. We expect that our study is generalizable to any devices supported by SteamVR.

2.2 Sensor Groups

We explore all VR sensors available on today's consumer VR devices, *i.e.,* the following four groups: body motion (BM) [29, 50], eye gaze (EG) [25, 45], hand joints (HJ) [26, 46], and facial expression (FE) [28, 51]. These sensors are available to developers through device-independent OpenXR APIs [24], as well as captured by ALVR [3] in the BehaVR setup. Depending on the device and platform, additional permissions may be required to access specific sensors, *e.g.,* Quest Pro requires permissions for EG, and FE. However, in the SteamVR setup [78] that we use, apps run on PC and there is no permission check for collecting sensor data. On the Quest Pro, the ALVR app [3] requests initial permissions for FE and EG during installation. Thereafter, it operates without additional runtime permission requests for all 20 apps in our experiment.

We follow the data structure definitions from the OpenXR standards [24]. The main elements of the data structures are *position*, *rotation*, *linear* and *angular velocities*. Position, and linear and angular velocities are expressed in x, y, and z values of the Cartesian right-handed coordinate system, and rotation is expressed in x, y, z, and w values of the Quaternion coordinate system. Additional information regarding sensor groups is provided in the Appendix A.

2.3 VR Apps

App Selection. Starting from the top 100 apps from the "Most played VR games" list on Steam [68], we select 20 VR apps based on several criteria. First, we exclude apps that may cause inconvenience to most users, *e.g.,* horror or violence genre. This first criterion is mandated in 45 CFR § 46.111(a)(1) to minimize the experimental risk

[2]Note that SteamVR differs from Oculus VR, Meta's VR platform that runs VR apps natively on its Android-based system [35].

Table 1: Grouping apps (a_1, .., a_{20} listed in Table 6, Appendix B) based on their similarity of activities and emotional states (arousal/valence). *Sensor Groups:* BM, EG, HJ, FE. *Emotional States:* LA = low arousal, HA = high arousal, PV = positive valence, NV = negative valence.

App Groups	App No.	App-Specific Activities	Arousal/Valence	Important Sensors
Social	a_{12}, a_{15}, a_{18}	Walking, waving, grabbing and sightseeing/exploring virtual environment	LA/PV, HA/PV	BM, EG, FE, HJ
Flight Simulation	a_3, a_{19}, a_{20}	Holding onto the airplane control stick, interacting with control panel/buttons in an airplane cockpit	LA/NV, HA/NV, LA/PV	BM, HJ, FE
Golfing	a_6	Slow walk, holding a golf stick, and put the ball towards hole	LA/PV, HA/PV	BM
Interactive Navigation	$a_2, a_9, a_{10}, a_{11}, a_{16}, a_{17}$	Grabbing, moving objects, opening doors, *i.e.,* frequent interaction with virtual objects	Neutral, LA/PV, LA/NV	BM, EG, HJ
Knuckle-walking	a_7	Walking using an open fist like a gorilla, sightseeing/exploring virtual environment	LA/PV, HA/PV, LA/NV	BM, HJ, FE
Rhythm	a_1	Dancing-like moves and cutting objects in quick pace	All	BM, HJ, FE, EG
Shooting & Archery	a_{13}, a_{14}, a_5	Grabbing and holding a gun/arrow, aiming and shooting at objects	LA/NV, HA/NV	BM, EG, FE, HJ
Teleportation	a_4, a_8	sightseeing by teleportation (instead of walking) *i.e.,* without extensive body movement	All	FE

(*e.g.,* physical or psychological harm) on our study participants [23]. Second, we exclude apps without complete VR support, like those for VR devices other than Quest Pro or needing both VR controllers and a PC keyboard for input. Finally, we attempt to compile a rich set of apps from various genres, *e.g.,* social, entertainment, flight, gaming etc. The list of 20 SteamVR apps is shown in Table 6 in Appendix B, referred to as a_1, .., a_{20}, throughout the paper.

Apps Grouping. We group apps based on the similarity of their activities and emotional states, considering an adversarial point of view. Our motivation is to leverage app similarities for cross-app identification and zero-day scenarios, *i.e.,*, using data from multiple similar apps to better identify users within the same group. Although heuristic, our app grouping performs well (see Section 5.5.2 and Table 2) and serves as proof of concept, however, an adversary can further optimize it. We propose app grouping in Table 1. For BM and HJ, we group apps according to similar app-specific activities; *e.g.,* social apps require walking, waving, and exploring, contrarily shooting apps require targeting and shooting objects – leading to different motion patterns. Note that, even within same group, differences exist; *e.g.,* for shooting group, a_{13} requires teleporting, while a_{14} requires walking.

For facial expression only, app grouping further considers emotional *arousal* (*e.g.,* how calm or active an emotion is) and *valence* (*i.e.,* how positive or negative an emotion is) states induced by the VR environment of an app; we use the approach proposed in [7, 38, 70]. There are four types of arousal-valence states we have considered in our study, namely high arousal positive valence (HA/PV), *e.g.,* happiness; low arousal positive valence (LA/PV), *e.g.,* surprise; high arousal negative valence (HA/NV), *e.g.,* fear/stress; and finally low arousal negative valence (LA/NV), *e.g.,* sadness. Different app environment may induce any of these states. For example, we observe that social apps induce mostly joy or surprise (*i.e.,* HA/PV), and flying/shooting apps induce mostly fear/stress (*i.e.,* HA/NV). Furthermore, one app can induce multiple emotions (*e.g.,* when completing a level in a game, users feel happy if they succeed and sad if they fail). See Table 1 for detailed lists of app groups and their associated valence/arousal states.

The last column in Table 1 lists the important sensor groups for each app group. Sensor group importance arises from app-specific activities and emotional states, which increase with the active use of specific sensor groups or are influenced by strong valence/arousal states of the app. Consequently, these data are adequately available to the adversaries. For example, in flight apps, users use controller/hands and induce emotions like surprise, fear thus, body motion, hand joints and facial expression are listed as the important sensor groups for them. The importance of these sensor groups for user identification is confirmed in the evaluation (see Section 5.6.2).

2.4 Current Practices Regarding VR Sensor Data

Different VR platforms and apps have different practices regarding sensor data collection, use, and sharing. We looked into privacy policies and permissions to better understand those practices.

2.4.1 VR Sensor Data in App's Privacy Policies. Privacy laws, such as GDPR [17] and CCPA [66], require disclosure of data collection, use and sharing practices. Both CCPA (in §1798.140(c)) and GDPR (Article 4(14)) define *behavioral characteristics* as part of "biometric information" or "biometric data" that can uniquely identify a person. This motivates us to look into real-world VR apps and platforms and their disclosure of VR sensor data.

We look at the top 100 apps from the "Most played VR games" list on Steam [68] and download their privacy policies. As of May 2023, only 60 apps provided a privacy policy. Our authors first read and check all privacy policies to understand how they disclose the collection of VR sensor data. We looked for statements on "biometric data" or "sensory data", as well as more specific types (*e.g.,* "head movement") in any of the sections. Then, we used string matching and ChatGPT to scan the whole text again to ensure that we do not miss any content. We found that only a few (10 out of 60) privacy policies discuss the collection of VR sensor data, and some make conflicting statements. Additional details can be found in Appendix C. These observations are aligned with the findings in [73], that many VR apps did not provide a privacy policy or did not disclose VR sensory data collection adequately. Meta, the maker of Quest Pro, indicates in its privacy policy that they collect data and use it for personalization [34]. Unity, the top game engine that many VR apps build on, claims the collection of biometric information for the purpose of identifying an individual. Unity explicitly mentions "hand and face geometry" (HJ and FE sensor groups) as examples of biometrics that may be collected [75]. This further motivates us to study how well real-world VR apps and VR platforms can identify users based on their VR activities.

2.4.2 VR Sensor Data Permissions. What sensor groups an app has permission to access depends on the platform.

The BehaVR study in this paper, is based on 20 apps from the SteamVR Store [77]. Our review of each app's website on SteamVR revealed no information about which sensor data are collected or

what their collection purposes are. Furthermore, as detailed in Section 2.1, SteamVR apps do not have runtime permission constraints that prevent apps from reading any sensor data, whether the app needs them for functionality or not.

We also looked at how these same 20 apps are used beyond the SteamVR Store, on other popular platforms, particularly MetaQuest Store [48]. We found 10 of our 20 apps available there. Of these, 6 apps disclose the types of sensor data they collect. All 6 reports are collecting BM data by default. One app, VRChat (a_{18}) discloses collecting EG, HJ, and FE data, and another app, RecRoom (a_{15}), collects FE data only. We also identified that the Job Simulator app (a_9) discloses the collecting of HJ data. Meta apps have runtime permission checks to protect FE, HJ, and EG [34], while BM is more widely available without permission checks.

2.5 Threat Model

2.5.1 VR Threat Scenarios. The adversary's goal is to identify users based on VR sensor data. During training, the adversary observes users in VR apps, records and analyzes their sensor data streams, and creates models for user identification. During evaluation, the adversary observes new sensor data streams and uses the trained models to identify the user who generated them.

It is worth noting that if explicit identifiers (such as device IDs, user accounts, or software IDs) are available, they are straightforward to use for user identification. Instead, our focus is on using VR sensor data alone as implicit identifiers for user identification. Unique identification without explicit identifiers has been studied using different data in the past, ranging from mobile location data [9], to body motion data in VR recently [53, 57]. It has also privacy implications: an adversary can identify and track a user based on the VR sensor data *alone*, w/o necessarily having access to the explicit identifiers; *i.e.,* the adversary can obtain access to VR sensor data in various ways: directly (an honest-but-curious developer or 3rd party library without access to device IDs as they are often well protected by permissions), or by compromising any of the above, or through an anonymized and released dataset. The question has also implications for anonymity (or lack thereof) in the virtual world[3]: even if a user changes their VR device, account, app, or avatar, they can still be (re-)identified based on their sensor data. On a positive note, the uniqueness of VR sensor data can potentially be used for authentication [43].

2.5.2 BehaVR Adversaries. Next, we define two types of adversaries, depending on their vantage points and sensor data access, described next and depicted in "①" in Fig. 2. Both adversaries build models using individual sensor groups for identification. This emphasizes the importance of considering scenarios where an adversary may have access to only one sensor group rather than all sensor groups. Other scenarios include the adversary aiming to minimize its effort by utilizing fewer data or users may not use certain sensor groups; *e.g.,* for HJ models, we assume the adversary only utilizes HJ sensor groups for identification, not other groups such as BM, EG, or FE. [4] BehaVR adversaries are as follows:

App Adversary. The app adversary (adv_{app}) has access to sensor data collected from APIs available within a single app. This mimics an app developer and any third party that has the same permissions and receives the data from the app, *e.g.,* Unity[76]. When ALVR app [3] grants full sensor permissions, the SteamVR apps do not request any runtime permission. Therefore, we assume that SteamVR apps may access or collect all sensor groups (see more details in Section 2.1 and 2.2). The app adversary corresponds to the client adversary in the taxonomy in [22] and has been previously studied for unique identification [53, 56, 57, 72].

Device Adversary. The device adversary (adv_{dev}) has access to all four sensors collected from multiple VR apps. Realistic examples of the device adversary include the device manufacturer, a game company that releases multiple apps, third parties with access to multiple apps' data (*e.g.,* Unity [76], SteamVR[78]), or malware that records sensor data. Additionally, its capabilities are available to the VR device (*e.g.,* Quest Pro) and its operating system, as well as to the PC in our SteamVR setup or a compromised library with functionality similar to ALVR. The device adversary corresponds to the hardware or client adversaries in the taxonomy in [22]. To the best of our knowledge, it has *not* been previously studied for identification in VR, since collecting sensor data across multiple real-world apps was not previously possible.

Both adversaries collect users' sensor data trace D from the four sensor groups discussed in Section 2.2. Both train models to identify a user u_i among n users ($u_1, .., u_n$). Their main difference is, that the app adversary has access to data from one app, while the device adversary has access to *multiple* apps' data. Both adversaries train one model per sensor group, which predict the label for each sensor data block and can combine all labels of blocks through max voting per user [11]. Furthermore, the device adversary may train on data from all apps, or groups of similar apps as defined in Section 2.3. See Section 4.2 for details on the models [5] and 5 for their evaluation.

3 Experimental Setup

3.1 Using ALVR as a Vantage Point

In our study, we used Quest Pro, the latest state-of-the-art VR device from Meta, released in November, 2022. While other VR devices also work with SteamVR [78], Quest Pro collects eye gaze and facial expression data [45, 51], in addition to body motion and hand joints data [46]—collected by older VR devices; *e.g.,* Meta Quest 2.

The BehaVR data collection system is depicted in "①" in Fig. 2. Although Oculus OS on Quest Pro can natively run VR apps, we use the SteamVR setup [78] that allows us to intercept and record sensor data. In BehaVR, Quest Pro sends sensor data to a PC that runs a VR app and the Quest Pro and PC are connected via WiFi. BehaVR integrates the ALVR [3], an open-source software that can run VR apps on a PC. With the help of SteamVR, ALVR can run Steam apps that provide VR support on Quest Pro: the sensor data

[3]Several scenarios where implicit identifiers can be useful are:(1) unlike mobile devices that are personal, VR devices and user accounts can be shared by a group of people (*e.g.,* among family members, friends, public VR game-stores or education platforms [61], among coworkers [37]); (2) one user may use multiple accounts for one or multiple apps, multiple devices or avatars for privacy or other reasons.

[4]The adversary cannot utilize BM and HJ simultaneously since users use the controller and hand alternatively in VR. As a result, the adversary may receive zero or corrupted values from controllers (part of the BM sensor group) while using a hand. In addition, we project the scenario where FE/EG can be disabled by users as well.

[5]We use the terms "adversary" and "model" interchangeably.

sent from Quest Pro are received by ALVR and become input to the app to process and render the app's VR environment in real-time. Finally, the app sends the rendering results back to the Quest Pro, so the headset displays the VR environment to the user. While, the SteamVR was intended to enhance VR performance by performing heavy tasks on PC, we use it for passive data monitoring, for the first time. We instrument parts of ALVR's source code that receive sensor data from the Quest Pro (*i.e.,* by creating hooks on the four sensor groups data streams) and save the data as time series.

3.2 User Study and Data Collection

We conducted an IRB-approved user-study from our institution's IRB review committee. We recruited participants aged 20-40, with an equal gender split to better represent the diverse demographic of VR users [6]. Please see the participant distribution summary in Appendix D. The data collection was performed by three authors and required $\sim 5-6$ hours per participant (including briefing and training) to collect ~ 400 sensor data records (20 real-world apps per user, ~3 months in total for 20 users). Each participant was compensated $10/h, declared in IRB and participant consent form.

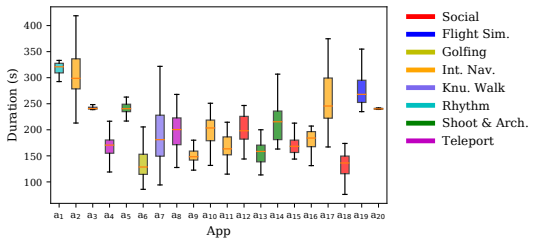
Each user was asked to wear a Quest Pro headset, and interact with all 20 apps in our corpus (see Table 6 in Appendix B). A research team member provided rough prompts to the VR user during user-app interaction.[6] These prompts guide users in interacting with each app, according to the purpose of the app, but users have freedom to interact with the app in their own way and pace. Meanwhile, BehaVR (*i.e.,* the instrumented version of ALVR) was running and recording sensor data from Quest Pro. For each app, a user completes the app-specific activity twice (*i.e.,* two sessions) and we collect two data traces, whose duration was typically $\sim$ 3–4 minutes: the first trace is for model training & validation. For evaluation, we utilize few/all (seconds) data from second session of the same app, or from new/different settings of the same app, or from different apps based on our adversarial set-up (see Section 4.2).

3.2.1 Dataset Summary and Size. The number of participants in our user study (20) is on-par with most prior user studies that collected data from participants, *e.g.,* [41, 54, 56, 60, 72] considered 16-50 participants. This number is smaller than 500 participants in [53], who however, performed simple tasks compared to our work that considers multiple real-world apps. It is also smaller than 50K users in [57], in which the authors considered one popular real-world app (Beat Saber) and body motion data, in a dataset provided by BeatLeader [62]; this number of users is obviously impossible to involve in in-person user studies. In terms of duration, we recorded around $3-4$ minutes per user for two sessions (see Fig. 3), which compares to [57] (median of ~3 minutes per session) and [53] (five 20-second videos for a total of 1 minute 40 seconds).

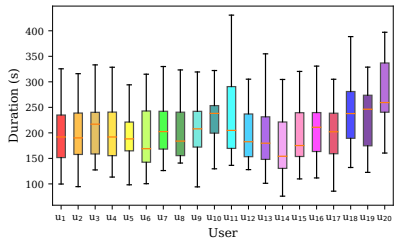
4 Data Analysis and Model Training

This section presents the BehaVR pipeline for processing the sensor data (Section 4.1, also see "②" in Fig. 2) and for model training (Section 4.2, also see "③" in Fig. 2) .

[6]For example, in Golf It! (a_6), we give users the prompt: "Please putt the golf ball into the hole, in the beginning with controller, then with bare hands". It is up to the user in what way and how many times they putt the ball.



(a) Total durations of sessions grouped by app



(b) Total durations of sessions grouped by user

Figure 3: Durations of sessions. There are 20 users, each interacts with 20 apps. Colors represent app groups.

4.1 Sensor Data Summarization

Here we convert the sensor data streams into feature vectors, which are suitable for a non-sequential model (*e.g.,* Random Forest).

4.1.1 Insight: Variability. The BehaVR dataset exhibits variability across users, apps, and sensors, even when they perform the same activity w.r.t. the same app. Designing for variability was a decision we made on purpose to capture users' natural behavior. From a *user* perspective, the user has the freedom on how, and at what pace, to perform the activity in each app. From the *app* perspective, variability is caused by different apps having different activities. From the *sensor* perspective, variability occurs as the four sensor groups operate with different sampling rates and time spans.

To illustrate the variability in *session duration* across users and apps, we plot the distribution of total durations of sessions per app (Fig. 3a) and per user (Fig. 3b). In Fig. 3a, We observe that users interact with the same app for varying durations: while average durations differ across apps, the variance for each app is relatively small. In Fig. 3b, we observe that the average durations are closer to each other, but have larger variance for each user. Thus, we summarize the sensor data on per app and per sensor group basis.

4.1.2 Data Processing. Next, we describe data processing. The details regarding data processing are available in Appendix E.1. The goal of this step is to segment sensor data into time blocks and summarizing each as a feature vector for model input. One challenge is to choose the length of the block. For block division, we first experimented with fixed-block length (FBL)—also been used in [53]. FBL divides time series data into blocks: each block has a fixed length (*e.g.,* 1 or 2 seconds), and it ignores the variability of users, apps, and sensors. Since there is much variability in BehaVR dataset, we develop an intuitive but robust method, refer as

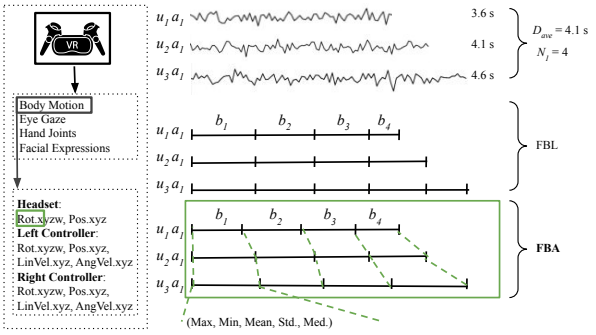


Figure 4: FBA illustration for the x value of headset rotation from the BM sensor group.

fixed-block amount (or FBA), guided by previous observations on variability across apps and users. FBA divides time series into a fixed amount (number) of blocks for each sensor group per app (*e.g.,* unlike FBL, FBA takes variability into account). FBA works comparatively better in our case (See Fig. 7 in Appendix E.2).

Fig. 4 shows FBA applied to BM, processing a headset rotation value (x) for app a_1. With session durations of 3.6, 4.1, and 4.6 seconds for 3 users, the average duration is 4.1 seconds, rounded down to 4 blocks. Each user's time series is divided into 4 blocks of ~1 second. A parameter $r \in (0, 2]$ to adjust block division: $N_{FBA_j} = r \cdot N_j$, increasing r increases block count and decreases block length. Finally, we summarize the time series of each reading in each block with a vector of five statistics, *i.e.,* maximum, minimum, mean, standard deviation, and median. This summarization previously was used in [53] and [57] for BM.

4.1.3 Feature Selection and Engineering. Next, we select and augment the features, as follows:

Body Motion. We use all 33 BM sensor readings, including 3 position and 4 rotation readings from each controller and the headset, and 3 linear velocity and 3 angular velocity[7] readings from each controller. After computing five statistics (max, min, mean, stdev, median) for each reading, we obtain 165 BM features per block.

Eye Gaze. The EG features are derived from one position and 3 rotation readings per eye[8]. From position readings, we derive the *interpupillary distance* (IPD, *i.e.,* the x position difference between two eyes). Prior work has shown that IPD is a top feature for gender identification [56]. Inspired by this, we also augment rotation readings by calculating the differences in the same reading between two eyes. By calculating the five statistics for 3 rotation readings per eye (6 total) and the 3 differential values, and including IPD as a separate feature[9], we obtain 46 EG features per block.

Hand Joints. There are 182 readings per hand that describe 3 position and 4 rotation readings from each of 26 joints. After calculating five statistics, we have 1,820 HJ features per block for two hands. In order to limit model complexity and reduce run time, we reduce the number of features, using information gain [42] – a popular technique for feature selection [31, 65] employed in prior work [53, 57]. We compute the information gain based on the model's performance on the training data. We exclude features with negligible importance and choose the top 400 HJ features per block.

Facial Expression. BehaVR captures 64 sensor readings (see Appendix A). After calculating five statistics per reading, we obtain 320 FE features per block for model training and evaluation (see Section 5.4). Next, we select FE readings that describe emotion-related actions (*i.e.,* Action Units, AU) for each emotion (see Table 5) [18]. For example, there are 4 sensor readings corresponding to happiness, described by AU6 (facial elements 5 and 6) and AU12 (facial elements 33 and 34). After computing five statistics per reading for each emotion, we obtain the final features for that emotion. For example, for happiness, the final feature set consists of 20 FE features after computing the statistics. Final features for each emotion are utilized to train and evaluate individual models for each specific emotion in Section 5.4.1. Additionally, we selected 25 FE readings for all emotions, resulting in 125 FE features, to evaluate a model on all facial emotions combined.

Final Features. After data processing and feature engineering, we obtain the final set of features per group, summarized in Table 8 Appendix E.2. These are used to train and apply the adversary models, described next.

4.2 User Identification Models

4.2.1 Classification Task. We perform a multi-class classification task to uniquely identify user u_i among the set of n (*i.e.,* 20) users.

Train-Test Split. We gathered sensor data from users in two sessions per app, with each session involving the completion of app-specific activities (see details in Section 3.2). Data from the first session were split, with 90% for training (D_{train}) and 10% for validation (D_{val}). For evaluation, data from the same app (under similar or different settings) or a completely different app can be used.

Model Architecture and Hyperparameters Tuning. We explored models including Random Forest (RF) [40], Gradient Boosting (XGB) [19], Support Vector Machine (SVM) [8], and Long Short-Term Memory Networks (LSTM) [33]. We found that RF and XGB performed best on the BehaVR dataset; this aligns with the closest prior works [53, 57]. More details on algorithm selection and hyperparameter tuning can be found in Appendix E.2 and E.3.

Model Training. FBA is applied to divide each session into N_{FBA_j} blocks per user in an app a_j, *i.e.,* $[b_1, b_2, ..., b_{N_{FBA_j}}]$. The duration per session per user is T, and each block's duration is t. All blocks from the first session is used for training (D_{train}). For evaluation, we pick s number of blocks per user from second session; s represents a *sub-session* that has $[b_1, b_2, ..., b_s]$ blocks, where $s \leq N_{FBA_j}$. S_t for the sub-session is $s \times t$. S_t allows us to investigate the minimum time we need per sensor group per user for identification. S_t equals to the whole evaluation session per user when $s = N_{FBA_j}$ and $S_t = T$. Finally, we perform a classification task for each block (*i.e.,* predict the label for each $b_1, b_2, ..., b_s$) and use maximum voting [11] across all blocks to determine the final label for each user.

4.2.2 Different Adversaries and Their Models. In this section, we describe the experimental setup (*e.g.,* model) for different adversary.

[7] Unlike prior works [53, 54, 57] that focused *only* on position and rotation, we additionally consider angular and linear velocity.
[8] The EG position y, z and rotation z are always zero, presumably because eyes cannot move in these directions relative to the headset.
[9] We do not compute statistics for position, since they do not change much over time.

App Adversary Models. The app adversary trains an *app model* on each app's data. Initially, we assume that the app models are trained and tested in similar app settings (*e.g.,* difficulty level, virtual rooms, songs) from the two different logins of the same user.

Next, we relax some constraints of app adversary and show that user identification works even in more open-world scenarios, where the app models are trained and tested in separate settings of the same app. We refer to this as *open world settings* for app adversary. We chose five representative apps from five distinct app groups. In the case of Beatsaber(a_1), a popular rhythm app, training, and testing data were collected from different songs and difficulty levels based on users' preferences. For RecRoom (a_{15}), a social app, we gather training and testing data from separate virtual locations such as MacDonald's virtual restaurant, virtual campus or party venue. Similarly, in the case of Gorilla Tag (a_7), a knuckle-walking app, training, and testing data were collected in separate virtual spaces. For Elven Assasin (a_5), belonging to shoot.& archery app group, training, and testing occur across different difficulty levels of the gameplay. Finally, for Chess (a_{17}), an interactive navigation app, training, and testing data were gathered across different gaming rounds, while the users freely moved chess pieces.

Device Adversary Models. The device adversary has access to multiple apps' data. As device adversary aims to identify users across different apps, this setting undergoes *open world settings* for across app evaluation. We start from the *universal model* that uses data from all apps (a_1, ..., a_{20}); adversary can choose to train on a subset–a *group model* for an app group (see details in Section 2.3). Suppose an app group (*e.g.,* social) has n_g number of apps. The device adversary trains an app group model on all n_g apps' training data. First, we consider the scenario where the adversary identifies a user of an app in a similar app group: the adversary can apply the model on each app's test data to identify users. We identify users across different app groups (n_g) by evaluating the app group model with an average data representation (a_{avg}).[10]

Next, device adversary can initiate attacks under *zero-day scenario*, where the adversary attempts to identify a user from an app that it has not previously trained on. To that end, we train an app group model with $n_g - 1$ apps' training data and test on n_g^{th} app (n_g^{th} app's data is not in D_{train}). We refer this type of attack as *zero-day attack*. We perform leave one out method and report the average accuracy to report effectiveness of *zero-day attack*.

Top Features. For each model, we analyze the feature importance for RF and XGB using information gain [42]. This helps both an adversary or privacy designer who wants to minimize its work.

5 Evaluation Results

In this section, we evaluate the performance of BehaVR's adversaries. Table 2 summarizes the results. For each sensor group (in Sections 5.1, 5.2, 5.3, and 5.4), we evaluate 20 models (*i.e.,* one model per app) guided by the following research questions (RQs):

- *RQ1 (Accuracy):* How well can a user be identified using different VR sensor group? How do these groups compare to each other?
- *RQ2 (Sub-session Time S_t):* How long does identification take?
- *RQ3 (Top Features):* What are the top features for identification w.r.t. various apps and adversaries?

In Section 5.5, we evaluate our open-world experiment, discussed in Sections 2.3 and 4.2, answering the following:

- *RQ4 (Open-World Setting):* Can we identify a user across different settings within same app or a user across similar or different apps (app groups)? What if the app is *not included* in the training of the in adversary's model (*zero-day* scenario)?

Next, Section 5.6 discusses relative sensor group importance (among sensor groups and w.r.t. app groups) by answering the following:

- *RQ5 (Sensor Group Importance):* What are the most important sensor groups in general, and as they relate to particular app groups? Moreover, can combining weak sensor models help to generate a comparatively stronger attack?

5.1 Body Motion Models Evaluation

RQ1 (Accuracy). The identification accuracy for body motion app models is 100% in 3 apps and ≥95% (*i.e.,* at most 1 out of 20 users is falsely identified) in 14 apps (see adv_{app} results from BM column of Table 2). These results are consistent with previous studies (see Section 6). Most apps, such as Beat Saber (a_1; extensively studied in [57]), archery (a_5), and shooting (a_{13}), demand significant body motion (headset and controllers movement). Other apps that require less body motion (*e.g.,* in a_4, users move through teleportation) provide ~70-80% accuracy. As the device adversary considers a larger amount of data and tasks for training, the accuracy is up to 100% (see adv_{dev}, Table 2) compared to a single app.

RQ2 (Sub-session Time S_t). For the app models, accuracy is ~80% with an average user sub-session time (S_t) of 4s. The app models require at least 16s of S_t to reach 90% accuracy (see Fig. 8a in Appendix F). The device model achieves higher accuracy with similar S_t by training on all 20 apps (see Fig. 8e in Appendix F), accumulating comprehensive user behavior knowledge.

RQ3 (Top Features). In Appendix F, Table 10 presents the top-3 features for identification for each app. The top features are influenced by app-specific activities and user measurements; *e.g.,* flight apps require users to sit and make left-right head movements to control flight making the x-position of the headset (left-right movements) as top feature. Shoot.& archery apps (*e.g.,*a_5) require both headset and controller movement/velocity that influence as top features. For the *device model*, the y, x, and z positions of the headset are top features (see Fig. 9a in Appendix F), indicating height, left-right and forward-backward extent of the head movements influence identification. Fig. 10a in Appendix F shows the importance of headset features: 5% to 35% higher accuracy compared to controller features alone (~21% on average) across all apps.

Key Takeaways. BM identification relies on both app-activity specific and users unique measurement (*e.g.,* height) features.

5.2 Eye Gaze Models Evaluation

RQ1 (Accuracy). App models provide ≥90% accuracy for 8 apps, ≥85% for 12 apps, ≥75% for the remaining. We observe that identification accuracy is influenced by frequent object-eye interactions, *e.g.,* Beat Saber (a_1) model gives 95% as users frequently look at and follow the movement of virtual objects in this app. Similarly,

[10] For example, with 3 apps in a group, we use 33.33% S_t from each app for evaluation. This is to make a fair comparison using the same amount of data for evaluation.

Table 2: Identification accuracy (%) for app adversary (adv_{app}) and device adversary (adv_{dev}) w.r.t. sensor groups. The app adversary (adv_{app}) trains and evaluates an app model on sensors data from a single app (listed in App No column). The device adversary (adv_{dev}) has two rows. The first row (*e.g.,* a_{12}, a_{15}, a_{18}) reports results from adv_{dev} training a group model on all apps in that group and evaluating on each individual app. The second row for adv_{dev} reports results from training a group model and evaluating on average data of that group (*e.g.,* a_{avg} indicates each app contributes 50% of the data if $n_g = 2$). Each group, Golfing, Rhythm, and Knuckle-walking has exactly one app; thus, adv_{app} and adv_{dev} are the same (filled with "both" in Adver. column).

App Group	Adver.	App No.	Sensor Group			
			BM	EG	HJ	FE
Social	adv_{app}	a_{12}, a_{15}, a_{18}	85, 95, 95	80, 90, 90	60, 65, 75	95, 100, 100
	adv_{dev}	a_{12}, a_{15}, a_{18}	95, 95, 95	75, 90, 85	70, 85, 75	100, 100, 100
		a_{avg}	100	95	90	100
Flight Sim.	adv_{app}	a_3, a_{19}, a_{20}	95, 100, 95	85, 90, 75	80, 75, 75	100, 95, 95
	adv_{dev}	a_3, a_{19}, a_{20}	95, 100, 100	90, 90, 90	80, 80, 75	100, 100, 95
		a_{avg}	100	95	95	100
Interactive Navigation	adv_{app}	$a_2, a_9, a_{10}, a_{11}, a_{16}, a_{17}$	95, 80, 95, 95, 95, 100	80, 80, 85, 95, 75, 80	60, 40, 60, 60, 70, 90	100, 100, 90, 95, 100, 100
	adv_{dev}	$a_2, a_9, a_{10}, a_{11}, a_{16}, a_{17}$	95, 85, 90, 95, 95, 100	75, 60, 80, 80, 60, 75	65, 40, 60, 75, 75, 90	100, 100, 95,100,100,100
		a_{avg}	100	95	85	100
Shooting & Archery	adv_{app}	a_5, a_{13}, a_{14}	95, 90, 100	85, 75, 90	70, 60, 80	90, 100, 100
	adv_{dev}	a_5, a_{13}, a_{14}	95, 100, 100	85, 80, 90	70, 65, 80	90, 100, 100
		a_{avg}	100	90	85	100
Teleport.	adv_{app}	a_4, a_8	70, 80	90, 70	35, 45	95, 95
	adv_{dev}	a_4, a_8	75, 85	90, 75	35, 50	100, 95
		a_{avg}	85	90	50	100
Golfing	both	a_6	80	70	50	90
Rhythm	both	a_1	95	90	75	100
Knu.-walk.	both	a_7	95	80	65	100
All	adv_{dev}	$a_1, a_2, ..., a_{20}$	90 − 100	50 − 80	45 − 95	100
		a_{avg}	100	90	100	100

archery, shooting, and flight simulation app models show high accuracy due to frequent eye-object interaction. The device model's accuracy can be up to 100% (see EG column in Table 2).

RQ2 (Sub-session Time S_t). The app models accuracy is ∼50% with 5s of S_t per user on average. It increases to ∼70% with 19s of S_t (accuracy may vary depending on apps, see Fig. 8b in Appendix F). Device model (Fig. 8e in Appendix F) shows 80% with 17s of S_t.

RQ3 (Top Features). For both app and device models (see Table 10, Appendix F and Fig. 9b , Appendix F) show that augmented features contribute the most to user identification for EG. The top features are the y-rotation that correlates left and right eyes (*i.e.,* "Quat.y Left Right"), that matches our intuition: for EG, augmented features (*i.e.,* f^a_{LR}) are important for unique identification. Fig. 10b in Appendix F shows that augmented features (f^a_{LR}) improve model accuracy significantly (5 − 35% or ∼20% on average) across all apps.

Key Takeaways. Augmenting the standard features with the distance between the eyes improves identification accuracy.

5.3 Hand Joints Models Evaluation

RQ1 (Accuracy). The app models provide ≥70% accuracy for 9 apps, which involve diverse hand movements and gestures (see Table 6); *e.g.,* in a_1 (Beat Saber), users swing light sabers using hands, involve claw position and frequent hand movements; for a_{17} (chess), users grab and move chess pieces. Conversely, several app groups, such as teleportation, lack hand-specific activities that cause low identification accuracy; *e.g.,* teleportation provides the lowest accuracy (∼35%). For the device models, the accuracy is ≥85% in most cases.

RQ2 (Sub-session Time S_t). For the app models, the accuracy is ≥ 60% with S_t of at least 20s (see Fig. 8c in Appendix F). For the device model, accuracy is 90% with 120s of S_t (see Fig. 8f in Appendix F).

RQ3 (Top Features). For app models, (see Table 10, Appendix F) shows that the top features are influenced by app-activities. See Table 3 for description.*e.g.,* for a_1 (Beat Saber), top features are the positions of joints 1 and 3 (thumb metacarpal and palm) of the right and joint 24 (little intermediate) of the left hand. These joints are exercised when making a fist for holding sabers. For a_{17} (chess), joints 22 and 25 (little metacarpal and distal) of right hand (use for moving chess pieces) are top features. For device model (see Fig. 9c, Appendix F), positions of left-hand joints 1 (palm), 2 (wrist), 7 (index metacarpal), and right-hand joints 3 (palm), 2 (wrist) and 5 (thumb distal) are top features. They represent users natural hand positions, emphasizing joint positions (*e.g.,* making an open fist) and activities–emphasizing joint rotations (grabbing, waving, etc.).

Key Takeaways. Apps with more hand-related activities show higher attack accuracy using HJ sensor group.

5.4 Facial Expression Models Evaluation

RQ1 (Accuracy). Facial Expression is highly effective for user identification; the app models provide ≥ 95% accuracy for 17 and 90% for the remaining 3 apps. The device models achieve up to 100%, consistent with other sensor groups (see FE column of Table 2).

RQ2 (Sub-session Time S_t). For app adversary, most apps provide ≥85% accuracy with S_t of only 5s, and ≥90% accuracy with 18s (see Fig. 8d in Appendix F). For the device model, accuracy is 95% with just 3s and 100% within 17s (see Fig. 8e in Appendix F), demonstrating the high effectiveness of FE.

RQ3 (Top Features). Facial features are correlated to the app-specific activities (*e.g.,* in a_9:job simulation, users eat a doughnut, relates to element 27—jaw movement as a top feature) and valence/arousal states (*e.g.,* social, rhythm: elements 5 and 6, which correspond to AU6—action unit for happiness) (see Table 10 in Appendix F). See Table 4 for description. For the device model (see Fig. 9d in Appendix F), both emotions and natural expressions are key features, *e.g.,* elements 5 (cheek raiser), 6 (jaw drop), and 25 (jaw thrust) are part of user expression of joy and surprise respectively; 28 (jaw thrust) and 51 (lips toward) contribute to natural expressions representing the outward (lower lip) and opening (both lips) movements.

5.4.1 Facial Emotion Models Evaluation. In this section, we focus on facial elements/AU combinations that represent an emotion (see descriptions in [18, 28] and results in Table 11 in Appendix F), rather than all/other facial expression. We argue that arousal/valence states in VR may induce certain emotions, similar to what happens in the real world. For example, socializing, whether in-person or virtually, can make a person happy (HA/PV), or seeing a positive/new environment can induce joy/surprise (PV). From Table 1, we pick one or two apps from seven groups, representing the rest of the apps and groups to evaluate our hypothesis.

Our results confirm that facial elements/AUs indicating emotions can be used for identification, correlating strongly with the app's arousal/valence states. Social apps' models use AU combinations that represent happiness and surprise, provide ≥95%. Flight simulation or shooting apps induce mostly negative valence, thus identification based on happiness facial elements induce low accuracy, *i.e.,* apps a_{14} (shooting) and a_{20} (flight simulation), give 80% and 75% respectively, however, both apps provide ≥90% based on facial elements/AUs representing fear. In some apps, app-specific activities and arousal/valence states may induce mixed emotions. For instance, in Beat Saber (a_1), users may experience happiness due to the music/beat, fear from the tension of cutting blocks or avoiding obstacles, and even sadness or anger when missing some blocks. The models for these apps achieve high accuracy by considering almost all emotions. Apps with mostly neutral arousal/valence states (*e.g.,* interactive navigation apps) may achieve low accuracy (~80%) for high arousal emotions, such as happiness/sadness as the VR environment may not strongly induce these emotions.

Finally, combining all AUs representing all emotions provides high accuracy across all apps (*i.e.,* ≥90% for most apps). Fig. 10d in Appendix F shows that AU combinations representing emotions provide better accuracy compared to other facial expression AUs (that do not represent emotions) by $5-25\%$ in most apps (with some exceptions) and 5% on average; *e.g.,* considering arousal/valence, accuracy improves by 25% for a_{20}–flight and 10% for a_{14}–shooting.

Key Takeaways. Our findings suggest that an adversary can select AU combinations that represent emotions w.r.t. the app's arousal/valence state, to identify a user with low effort.[11]

5.5 Open-World Setting Evaluation

Next, we relax some constraints of Section 5.1- 5.4 and train and test on data from different settings and activities within and across apps, referred as *Open-World setting* (see Section 4.2.2). We show that users can be identified, not only across similar sessions of the same app (Sections 5.1 - 5.4), also across different settings of the same app (Section 5.5.1) and across different apps (Section 5.5.2).

5.5.1 User Identification across Different Settings in the Same App. In this section, we have considered different settings (*e.g.,* difficulty levels, virtual spaces, songs) in the same app and as a proof of concept, we performed additional experiments for five representative apps from five app groups (see experiment details on 4.2.2).

Body Motion For BM, accuracy of open-world ranges from 80-100% (close to 5.1 results), showing that users can be reliably identified across various settings within same app. Since motion patterns are unique to users *and* their activities within apps can be similar across different settings (not identical which adds variability that results in negligible accuracy drops) making identification feasible.

Eye Gaze For EG, the identification accuracy is between 60-80%.

Hand Joints For HJ, accuracy ranges from 60-80%. Accuracy is slightly lower in the open-world-setting compared to 5.3 due to added variability from different settings of the same app as BM.

Facial Expression For FE, accuracy is higher (90-100%) compared to other sensor groups as it relies heavily on facial emotion (see Fig. 10d), influenced by app's arousal/valence (see Section 2.3). If different app settings maintain a consistent arousal-valence state, users remain in similar emotional conditions, minimizing variability for FE and thus accuracy varies less across settings.

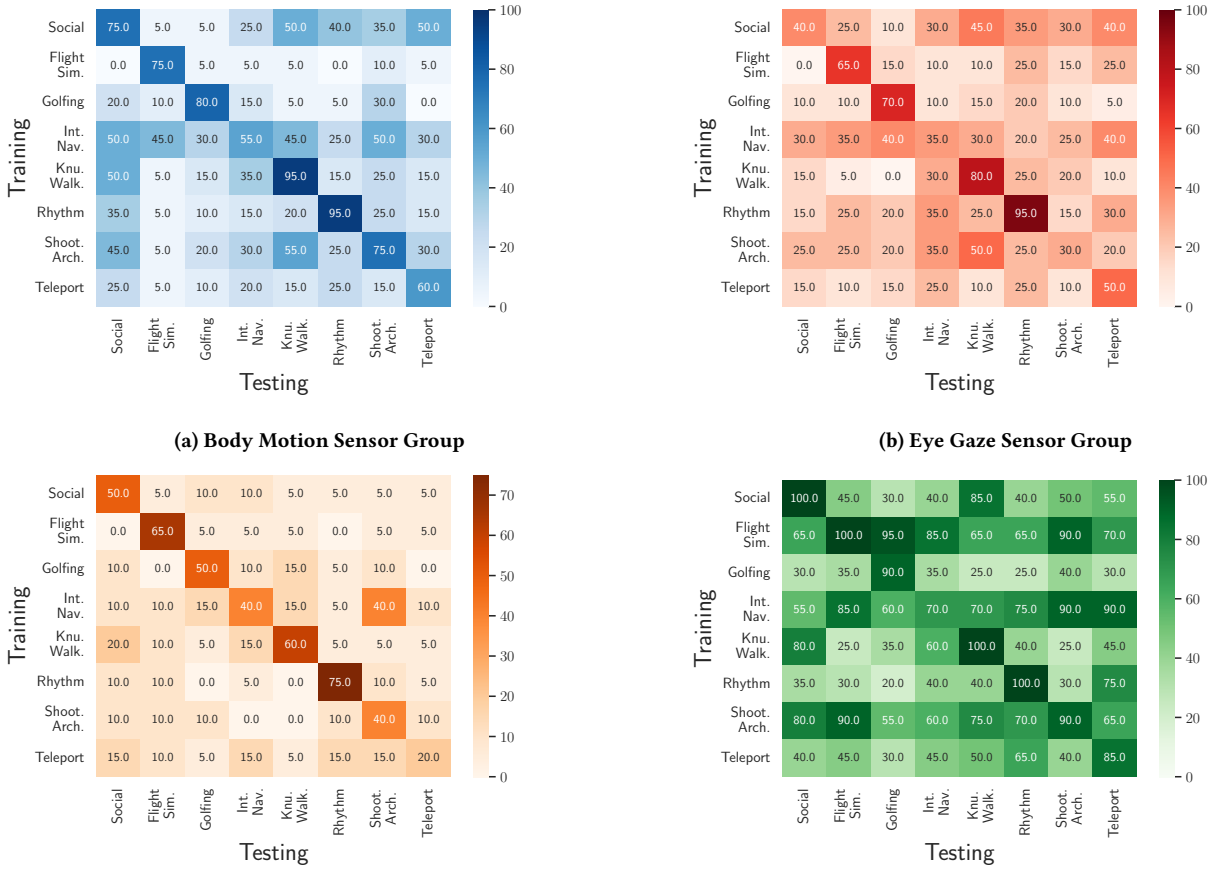
5.5.2 User Identification across App Groups. To address *RQ4*, we conduct experimental evaluation (see adv_{dev} row in Table 2) for each app group described in Table 1 and for the *zero-day* scenario (see Fig. 5) defined in Section 4.2.

Body Motion. The performance of the *app group models* (trained on all apps data within a group) and the *universal device model* (trained on all apps) using BM is comparable, *e.g.,* both achieve up to 100% accuracy using a_{avg}. The group models outperform the app models in general, which encourage the device adversary to choose a group model over app models; *e.g.,* for a_{13}, shoot.&arch. group model provides 10% higher accuracy than app model. In the *zero-day* scenario, Fig. 5a shows that models evaluated on new apps in the same group perform well ($75-95\%$ accuracy), but performs poorly on different groups (*i.e.,* $0-50\%$ accuracy). This evaluation confirms that group models are effective in *zero-day* scenarios.

Eye Gaze. No app grouping based on eye gaze is observed due to the lack of app-specific activity. Table 2 indicates that group models perform similarly to app models. However, in the *zero-day* scenario, grouping can be valuable as within similar groups, there is a higher occurrence of eye-object interactions that potentially help to identify users within a new app. Fig. 5b shows a diagonal in the heatmap with adequate accuracy, supporting our claim.

Hand Joints. For hand joints, app group models and the universal device model have comparable performance, *e.g.,* using a_{avg}, the accuracy difference is within the range of $10-15\%$ mostly. Akin to BM, HJ group models outperform the app models, *e.g.,* the social group model provides 20% higher accuracy for a_{15} than the app model. For *zero-day* scenario, a group model evaluated on its native apps provides higher accuracy ($40-75\%$) compared to an app from different groups (~20%) (see Fig. 5c).

[11] For example, adversary can use a few facial features (*e.g.,* 20 for happiness in social apps) instead of all (*i.e.,* 325) facial features for adequate unique identification.



(a) Body Motion Sensor Group

(b) Eye Gaze Sensor Group

(c) Hand Joints Sensor Group

(d) Facial Expression Sensor Group

Figure 5: Identification accuracy (in percent) in the *zero-day* scenario. The adversary trains on the data from other apps in a group, and tests in a new app (for which it did not have training data) in the same group. The *diagonal* shows the accuracy for apps within the same group, whereas the *other values* show the accuracy for apps from other app groups.

Facial Expression. Both the app group and the universal models achieve up to 100% accuracy using FE data, with group models performing similarly or better than app models. For the *zero-day* scenario, a model tested on apps from the same group provides higher accuracy (70 – 100%) than apps from different groups (20 – 65%) for most of the cases (see Fig. 5d). However, several apps provide high accuracy within different groups; *e.g.*, social group model accurately predicts a_7 with 85%, the shooting&archery group model provides 90% when tested on flight apps (*e.g.*, a_{20}) as they share similar arousal/valence states (LA/NV, HA/NV) (see Table 1).

5.6 Sensor Group Importance Evaluation

We compare the importance of different sensor groups, in general (Section 5.6.1) and for specific app groups (Section 5.6.2). Finally, we evaluate whether combining multiple weaker sensor group models (ensemble) can enhance attack performance (see Section 5.6.3).

5.6.1 Model Accuracy across Sensor Groups. This section partly addresses *RQ5*. FE and BM sensor groups outperform the EG and HJ. For BM, 14 out of 20, and for FE, 17 out of 20 apps achieve an accuracy of ≥ 95%. Conversely, only 5 out of 20 apps offer ≥90% for EG, which is intuitive since BM and FE cover more diverse activities compared to EG. HJ shows low attack performance in specific context: 9 among 20 apps provide 70 – 90% due to a lack of HJ-based activities in several apps. Both app and device models for EG and HJ require longer S_t (see *RQ2* of Sections 5.1-5.4) than BM, FE. Therefore, from an attacker's perspective, if the goal is to minimize effort and given that the attacker has access to any of the sensor groups, FE would be the optimal choice.

5.6.2 Important Sensors per App Group. This section partly addresses *RQ5* by discussing the importance of individual sensor groups relative to app groups. This comparison is useful: (1) for attackers to optimize which sensor groups' data to train and test models on, to efficiently utilize resources and maintain accuracy; and (2) for defense strategies regarding users' decisions to share sensor data, *e.g.*, users can revoke permissions for some sensor groups (including BM [47]) that are not essential for that app group.

Body Motion. Most app groups require body motion for app-specific activities. For example, social app groups involve walking and exploring, rhythm apps require dance-like continuous movements. Consequently, BM and its associated motion features are available to adversaries in those apps and thus provide ≥85% accuracy in most app groups. Contrarily, BM is less crucial for certain groups *e.g.,* teleportation since minimal body movement occurs for teleporting to different virtual locations, providing ≤80% accuracy.

Eye Gaze. For EG, there are no defined app-specific activities for our selected apps. Meta indicates EG is employed for realistic avatars and to estimate directions of where users are looking[52]. Intuitively, certain app groups require frequent eye-object interactions while performing app activities; *e.g.,* in shoot.&archery or rhythm, users aim to shoot/cut, resulting in frequent eye-object interactions. Consequently, they provide high identification accuracy ($85-95\%$) and important from adversarial perspective. Conversely, certain groups require minimal eye-object interactions as users mostly sightseeing in those apps (*e.g.,* knuckle-walking, provides $70-80\%$ accuracy). Thus, we can argue that, EG is optional from a user's perspective for our app groups except social. Thus given the available permission system, users might disable it.

Hand Joints. HJ for certain app groups that require active hand movements/gestures (*e.g.,* archery, flight, interactive navigation, & rhythm) provide higher accuracy (see Table 2), *e.g.,* flight apps models provide $75-80\%$ as require lots of hand activities for controlling flight. Conversely, HJ models for the teleportation group lack hand-related activities and achieve $35-45\%$. However, users may prefer to turn off HJ in certain groups where hand movements are optional (teleportation) and presumably prefer HJ in groups, where hand movements are crucial (*e.g.,* interactive navigation).

Facial Expression. Facial Expression or FE is crucial for identification in all app groups, as apps' arousal/valence states can trigger specific facial expression. Thus, FE achieves $\geq 90\%$ accuracy across all groups. Sharing FE data is particularly relevant in app-groups where user interaction is significant and realistic avatars enhance the experience (*e.g.,* social, job simulator). However, in groups where the realistic avatars are unnecessary, specifically in single user mode, where multi user interaction is not necessary, (*e.g.,* Rhythm, Flight), users may disable FE.

5.6.3 Sensor Group Model Ensemble. So far, our study assumes the adversaries use individual sensor groups to identify users (from Section 5.1 to 5.6.2). Next, we consider settings where the adversary combines multiple sensor group models to improve attack accuracy from individual sensor group models (*i.e.,* $< 90\%$). In this experiment, we excluded FE as an attacker may exploit FE alone for a successful attack (with $\geq 90\%$ accuracy) or users may disable FE (see Section 5.6.2). Moreover, we did not consider any combination of BM and HJ since the adversary can not collect data from both simultaneously. For combining multiple models, we used the model ensemble technique [21], where the final identification result is obtained through multiple sensor models using maximum voting mechanisms [11] of blocks per user.

Body Motion and Eye Gaze Model Ensemble. The adversary may consider combining BM and EG, given that, it has access to both BM and EG but not FE and HJ, and individual model accuracy of BM and EG are relatively low (*e.g.,* 80% and 70% for a_6, see Table 2). Our results show that ensembling EG and BM together can improve attack accuracy up to 10% (see Appendix F, Table 13).

Eye Gaze and Hand Joints Model Ensemble. Adversary considers combining HJ and EG, assuming they have access to both sensor groups; and BM/FE is unavailable (*i.e.,* not used by users) or corrupted. Under this assumption, if individual sensor models for EG and HJ yield low identification accuracy (70% and 45% for *e.g.,* a_8, see Table 2), ensemble techniques using both sensor models can enhance attack accuracy by 5-10% (see Appendix F and Table 13).

6 Related Work

Privacy in VR. Adams et al. studied the awareness of users and developers on data collection practices on VR devices [2]. Trimananda et al. [73] analyzed the network traffic generated popular Oculus VR apps, and reported personal information (device and user identifiers and some VR sensory data) collection and their inconsistencies with app's privacy policies [73]. Recently, Nair et al. developed an adversarial app to demonstrate tasks that can harvest users' personal information; *e.g.,* physical characteristics, location, gender [56]. The privacy of sensor data and APIs receives growing attention. VREED demonstrates emotion recognition in VR through eye tracking and physiological signals [71]. Kalɛido introduces Differential Privacy (DP) for safeguarding eye tracking, emphasizing user interests revealed in eye gaze heatmaps [39]. MetaGuard [58] safeguards user privacy using DP through feature obfuscation.

User Identification in VR. Most closely related to this paper is a body of prior works that focuses on identification based on sensor data collected on VR [12, 41, 53, 54, 56, 57, 60, 72, 81]. Most prior works focused on identifying users using BM within pre-defined tasks/custom apps. In [69], Stephenson et al. compared various authentication mechanisms for AR/VR, which include head/hand/eye biometrics. Tricomi et al. identified users in VR/AR based on body motion and eye movements [72]. Pfeuffer et al. focused on identification in various controlled tasks (*i.e.,* pointing, grabbing, walking) as correlated body and eye tracking data together [60]. Miller et al. used body motion for identification as users randomly select and watch 360-degree videos on VR [53]. Next, Miller et al. used spatial features from head and controller identification [54]. Nair et al. [57] analyzed a large dataset (50K users) of one commercial app (BeatSaber [20]), provided by the BeatLeader scoreboard [62]. They utilized body motion and contextual features for identification.

7 Discussion

7.1 BehaVR in Perspective

To the best of our knowledge, BehaVR is the first to analyze user identification in VR comprehensively, *i.e.,* considering (1) all VR sensors available (including HJ, FE, in addition to BM, EG); (2) data we collected from several real-world, unmodified apps and (3) considering identifiability within and across different apps, allow us covering a wide range of adversarial settings.

Generalization. Although our study is limited to 20 real-world apps, we believe that our methodology and evaluation results are generalizable. First, as explained in Sections 2.1 and 3.1, BehaVR is capable of collecting *all* sensor data from *any* of the thousands

of SteamVR apps that are compatible with ALVR setup. Second, BehaVR analyzes sensor data through device-independent standard OpenXR APIs. Thus, our user identification models and evaluations work with any apps and VR platforms that support the APIs. Finally, *zero-day-attack* and sensor and feature importance analysis can be applied to other apps if they fall under our app groups (based on activities and emotional states) as described in Section 2.3.

7.2 Recommendations for Mitigation

Based on our experience with BehaVR, we provide some recommendations for best practices and mitigation, including setting up permissions across VR platforms, auditing sensor data collection to offer users recommendations on data sharing practices, and implementing privacy-preserving mechanisms.

Use Permissions on all Platforms. While modern VR platforms such as Oculus VR have provided additional permission checks to protect FE, HJ and EG [34, 63], SteamVR apps lack any permission system and disclosure about sensor data collection in their websites (see Section 2.4.2), leaving users with no control over these sensor data (see Section 2.2). We recommend that all VR platforms and app developers (specifically SteamVR platform and apps) should implement permission systems for collecting sensor data, similar to Oculus VR. Additionally, we recommend that developers should disclose clearly which sensor data they collect, and limit that collection to what is needed for the functionality of the apps.

Provide Privacy Recommendation Systems for Users. Not all sensor groups are necessary for users to share for every app group (see Section 5.6). For example, FE is crucial for generating realistic avatars, is important for social apps but not for flight or interactive navigation apps. Moreover, certain sensor groups pose high identification accuracy; *e.g.,* FE, thus users may avoid sharing FE in general (for privacy reasons) or in later app groups (not necessary for app activities). Users can also decide to share less privacy-sensitive sensor groups; *e.g.,* for flight simulation, controller (parts of BM) can be replaced by HJ as later shows low attack accuracy (see Fig. 10a and 10c). Default recommendations can be offered to users via privacy nudges [1] or implementation of privacy recommendation systems based on static or policy analysis of apps (to analyze what they actually collect for which purpose) [30] guided by BehaVR.

Need for Privacy Preserving Mechanisms. We hope that our observations, particularly our feature analysis across different apps and sensor groups (see Section 5), can guide the design of defense mechanisms. One potential defense strategy could be to obfuscate sensor data. This can be implemented, for example, locally through local differential privacy (LDP) [13] either at the (1) device firmware level, before the sensor data leaves the device, or (2) software level, before the sensor data is transmitted to the server, as outlined in the framework described in [22]. The design depends on the adversary type: if the device is trusted, (2) is sufficient, if the adversary can intercept the device, (1) needs to be implemented, to be effective against the threat model described in Section 2.5.2. Guided by our feature analysis (Section 5), LDP can be applied to top features, which significantly contribute to user identification. For example, obfuscating the y-axis positional readings from the headset in the BM sensor group, which is the top feature, can significantly reduce identification accuracy. Future work can optimize the privacy-utility trade-off of this defense approach.

7.3 Limitations and Future Work

Study Size. A limitation of the user study, described in Section 3.2, is the number of participants (20). This number is on par with similar studies [54, 56, 72], but smaller than in [53] (500 users, one task) or [57] (50K users, crowd-sourced)[12]. The limitation in the number of participants comes from the time-consuming nature of our experiments, interacting with several real-world apps for a significant amount of time, in-person, under IRB guidelines; see Section 3.2 for details. A related limitation is that our dataset may not be representative of all VR users, such as younger users (age ≤ 18), older adults (age > 40), or w.r.t. other demographics. These factors may introduce bias and limit the generality of the results.

While acknowledging the study size as a limitation, we hope this work provides new insights into VR privacy by expanding the problem space in other dimensions, *i.e.,* several sensors (4 groups, 475 readings) and diverse real-world apps (20 from 8 groups). We will make the BehaVR system available to enable future research to expand the study to a larger scale, if so desired.

Auditing Data Collection. On SteamVR, apps are neither restricted by permissions nor required to disclose the collection of sensor data on their store pages (see Sections 2.1 and 2.4.2). Therefore, an app adversary may technically collect any sensor data without restrictions from the platform. However, we do not claim that individual apps do so. Auditing the data collection would involve network or program analysis [73] beyond the scope of BehaVR. We leave this as future work.

Advanced User Identification and Profiling. Although RF and XGB models can already achieve good performance (see Section 5), an adversary may minimize its work by feature minimization or leveraging more powerful models. Furthermore, given the rich behavioral information embedded in the sensor data, an adversary may go beyond identification and draw more inferences about (*i.e., profile*) users, such as demographics, physical conditions, and preferences. A natural next step is to exploit our dataset for profiling.

8 Conclusion

We present BehaVR, a framework for collecting and analyzing VR sensor data from four sensor groups. We applied it to Quest Pro and conducted a user study where real users interacted with real-world VR apps. We build models that an adversary can use to identify users within similar or different settings of an app, across different apps, or within a group of similar apps. We show that these models perform well, and we compare their performance across different sensor groups and apps. We also investigate the minimum time and top features for identification, and the importance of sensor groups on the apps or app groups. Additionally, we provide insights on how BehaVR can be generalized and effective for diverse studies in VR, and recommend strategies for privacy practitioners, including setting permissions and implementing privacy measures.

[12]We show that accuracy does not drop significantly with varying participant numbers, align with prior studies with 500 [53] or 5000 [57] participants (see Appendix D).

Acknowledgments

We would like to thank Diana Romero for her help with part of our VR app selection process. We would also like to thank Professor Habiba Farrukh for her valuable discussions, including on data access and permissions for VR apps. This work was supported in part by the National Science Foundation under award numbers 1956393, 1900654 and 2339266, and a gift from the Noyce Initiative.

References

[1] Alessandro Acquisti, Idris Adjerid, Rebecca Balebako, Laura Brandimarte, Lorrie Faith Cranor, Saranga Komanduri, Pedro Giovanni Leon, Norman M. Sadeh, Florian Schaub, Manya Sleeper, Yang Wang, and Shomir Wilson. 2017. Nudges for Privacy and Security: Understanding and Assisting Users' Choices Online. *ACM Comput. Surv.* 50, 3 (2017), 44:1–44:41. https://doi.org/10.1145/3054926

[2] Devon Adams, Alseny Bah, Catherine Barwulor, Nureli Musaby, Kadeem Pitkin, and Elissa M. Redmiles. 2018. Ethics Emerging: the Story of Privacy and Security Perceptions in Virtual Reality. In *SOUPS*.

[3] ALVR. 2023. ALVR - Air Light VR. https://github.com/alvr-org/alvr.

[4] AMD. 2024. Relive for VR FAQ. https://www.amd.com/system/files/documents/amd-radeon-relive-for-vr-faq.pdf.

[5] Tomislav Bezmalinovic. 2023. SteamVR in February 2023: PC VR usage is seemingly declining. https://mixed-news.com/en/steamvr-february-2023/.

[6] Ivan Blagojević. 2023. Virtual Reality Statistics. https://99firms.com/blog/virtual-reality-statistics/.

[7] Stéphane Bouchard and G Labonté-Chartrand. 2011. *Emotions and the emotional valence afforded by the virtual environment.* IntechOpen.

[8] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[9] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. 2013. Unique in the Crowd: The Privacy Bounds of Human Mobility. *Scientific reports* 3, 1 (2013), 1–5.

[10] Martin Degeling, Christine Utz, Christopher Lentzsch, Henry Hosseini, Florian Schaub, and Thorsten Holz. 2019. We value your privacy... now take some cookies: Measuring the GDPR's impact on web privacy. In *NDSS*.

[11] Thomas G. Dietterich. 2000. Ensemble Methods in Machine Learning. In *Multiple Classifier Systems, First International Workshop, MCS 2000, Cagliari, Italy, June 21-23, 2000, Proceedings (Lecture Notes in Computer Science, Vol. 1857)*, Josef Kittler and Fabio Roli (Eds.). Springer, 1–15. https://doi.org/10.1007/3-540-45014-9_1

[12] Yuming Du, Robin Kips, Albert Pumarola, Sebastian Starke, Ali Thabet, and Artsiom Sanakoyeu. 2023. Avatars Grow Legs: Generating Smooth Human Motion from Sparse Tracking Inputs with Diffusion Model. *arXiv preprint arXiv:2304.08577* (2023).

[13] Cynthia Dwork. 2006. *Differential Privacy.* Springer, Berlin. Tutorial presentation at the International Conference on Automata, Languages and Programming (ICALP).

[14] Paul Ekman and Wallace V. Friesen. 1978. Facial Action Coding System: Manual.

[15] Paul Ekman, Wallace V Friesen, and Joseph C Hager. 2002. *Facial Action Coding System: Facial action coding system: the manual: on CD-ROM.* Research Nexus.

[16] William Enck, Peter Gilbert, Byung-Gon Chun, Landon P. Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol N. Sheth. 2010. TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones. In *OSDI*.

[17] European Union (EU). 2018. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data (General Data Protection Regulation). https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679.

[18] Bryn Farnsworth. 2022. Facial Action Coding System (FACS) – A Visual Guidebook. https://imotions.com/blog/learning/research-fundamentals/facial-action-coding-system/.

[19] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[20] Beat Games. 2023. Beat Saber. https://www.beatsaber.com/.

[21] M. A. Ganaie, Minghui Hu, Ashwani Kumar Malik, Muhammad Tanveer, and Ponnuthurai N. Suganthan. 2022. Ensemble deep learning: A review. *Eng. Appl. Artif. Intell.* 115 (2022), 105151. https://doi.org/10.1016/J.ENGAPPAI.2022.105151

[22] Gonzalo Munilla Garrido, Vivek Nair, and Dawn Song. 2023. SoK: Data Privacy in Virtual Reality. *arXiv preprint arXiv:2301.05940* (2023).

[23] US Federal Government. 2023. § 46.111 Criteria for IRB approval of research (eCFR). https://www.ecfr.gov/current/title-45/subtitle-A/subchapter-A/part-46/subpart-A/section-46.111.

[24] The Khronos OpenXR Working Group. 2023. The OpenXR Specification. https://registry.khronos.org/OpenXR/specs/1.0/html/xrspec.html.

[25] The Khronos OpenXR Working Group. 2023. The OpenXR Specification: § 12.28. XR_EXT_eye_gaze_interaction. https://registry.khronos.org/OpenXR/specs/1.0/html/xrspec.html#XR_EXT_eye_gaze_interaction.

[26] The Khronos OpenXR Working Group. 2023. The OpenXR Specification: § 12.30. XR_EXT_hand_tracking. https://registry.khronos.org/OpenXR/specs/1.0/html/xrspec.html#XR_EXT_hand_tracking.

[27] The Khronos OpenXR Working Group. 2023. The OpenXR Specification: § 12.31.6. Conventions of hand joints. https://registry.khronos.org/OpenXR/specs/1.0/html/xrspec.html#convention-of-hand-joints.

[28] The Khronos OpenXR Working Group. 2023. The OpenXR Specification: § 12.53.7. Conventions of blend shapes. https://registry.khronos.org/OpenXR/specs/1.0/html/xrspec.html#_conventions_of_blend_shapes.

[29] The Khronos OpenXR Working Group. 2023. The OpenXR Specification: § 2.16. Coordinate System. https://registry.khronos.org/OpenXR/specs/1.0/html/xrspec.html#coordinate-system.

[30] Hanyang Guo, Hong-Ning Dai, Xiapu Luo, Zibin Zheng, Gengyang Xu, and Fengliang He. 2024. An Empirical Study on Oculus Virtual Reality Applications: Security and Privacy Perspectives. In *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering, ICSE 2024, Lisbon, Portugal, April 14-20, 2024*. ACM, 159:1–159:13. https://doi.org/10.1145/3597503.3639082

[31] Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research* 3, Mar (2003), 1157–1182.

[32] Alex Heath. 2024. This is Meta's AR / VR hardware roadmap through 2027. https://www.theverge.com/2023/2/28/23619730/meta-vr-oculus-ar-glasses-smartwatch-plans.

[33] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[34] Meta Inc. 2023. Supplemental Meta Platforms Technologies Privacy Policy. https://www.meta.com/legal/privacy-policy/.

[35] Meta Inc. 2024. Meta Store. https://www.meta.com/experiences/.

[36] VRChat Inc. 2023. VRChat Privacy Policy. https://hello.vrchat.com/privacy.

[37] Alexandra D. Kaplan, Jessica Cruit, Mica R. Endsley, Suzanne M. Beers, Ben D. Sawyer, and Peter A. Hancock. 2021. The Effects of Virtual Reality, Augmented Reality, and Mixed Reality as Training Enhancement Methods: A Meta-Analysis. *Hum. Factors* 63, 4 (2021). https://doi.org/10.1177/0018720820904229

[38] Peter Kuppens, Francis Tuerlinckx, James A Russell, and Lisa Feldman Barrett. 2013. The relation between valence and arousal in subjective experience. *Psychological bulletin* 139, 4 (2013), 917.

[39] Jingjie Li, Amrita Roy Chowdhury, Kassem Fawaz, and Younghyun Kim. 2021. Kal€ido: Real-Time Privacy Control for Eye-Tracking Systems. In *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, Michael Bailey and Rachel Greenstadt (Eds.). USENIX Association, 1793–1810. https://www.usenix.org/conference/usenixsecurity21/presentation/li-jingjie

[40] Andy Liaw and Matthew Wiener. 2002. Classification and Regression by randomForest. *R News* 2, 3 (2002), 18–22. https://CRAN.R-project.org/doc/Rnews/

[41] Jonathan Liebers, Mark Abdelaziz, Lukas Mecke, Alia Saad, Jonas Auda, Uwe Gruenefeld, Florian Alt, and Stefan Schneegass. 2021. Understanding User Identification in Virtual Reality through Behavioral Biometrics and the Effect of Body Normalization. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–11.

[42] Gilles Louppe. 2014. *Understanding Random Forests: From Theory to Practice.* Ph. D. Dissertation. University of Liège, Belgium. https://orbi.uliege.be/handle/2268/170309

[43] Shiqing Luo, Anh Nguyen, Chen Song, Feng Lin, Wenyao Xu, and Zhisheng Yan. 2020. OcuLock: Exploring Human Visual System for Authentication in Virtual Reality Head-mounted Display. In *NDSS*.

[44] Meta. 2023. Bring your teams together in Meta Horizon Workrooms. https://www.meta.com/work/workrooms/?utm_source=www.usenix.org&utm_medium=oculusredirect.

[45] Meta. 2023. Eye tracking on Meta Quest Pro. https://www.meta.com/help/quest/articles/getting-started/getting-started-with-quest-pro/eye-tracking/.

[46] Meta. 2023. Getting started with Hand Tracking on Meta Quest headsets. https://www.meta.com/help/quest/articles/headsets-and-accessories/controllers-and-hand-tracking/hand-tracking-quest-2/.

[47] Meta. 2023. Meta Quest Headset Tracking. https://www.meta.com/help/quest/articles/headsets-and-accessories/using-your-headset/turn-off-tracking/.

[48] Meta. 2023. Meta Quest Store. https://www.oculus.com/experiences/quest/.

[49] Meta. 2023. Our story: enter the future of learning. https://about.meta.com/immersive-learning/our-story.

[50] Meta. 2023. This is Meta Quest Pro. https://www.meta.com/quest/quest-pro/tech-specs/.

[51] Meta. 2023. Use Natural Facial Expressions on Meta Quest Pro. https://www.meta.com/help/quest/articles/getting-started/getting-started-with-quest-pro/facial-expressions/.

[52] Meta. Accessed: 2024. *App Privacy Data Types on Meta Quest.* https://www.meta.com/help/quest/articles/accounts/privacy-information-and-settings/app-privacy-data-types-meta-quest/

[53] Mark Roman Miller, Fernanda Herrera, Hanseul Jun, James A Landay, and Jeremy N Bailenson. 2020. Personal Identifiability of User Tracking Data During Observation of 360-degree VR Video. *Scientific Reports* 10, 1 (2020), 1–10.

[54] Robert Miller, Natasha Kholgade Banerjee, and Sean Banerjee. 2022. Combining Real-world Constraints on User Behavior with Deep Neural Networks for Virtual Reality (VR) Biometrics. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 409–418.
[55] Hooman Mohajeri Moghaddam, Gunes Acar, Ben Burgess, Arunesh Mathur, Danny Yuxing Huang, Nick Feamster, Edward W Felten, Prateek Mittal, and Arvind Narayanan. 2019. Watching You Watch: The Tracking Ecosystem of Over-the-Top TV Streaming Devices. In *ACM CCS*.
[56] Vivek Nair, Gonzalo Munilla Garrido, Dawn Song, and James F. O'Brien. 2023. Exploring the Privacy Risks of Adversarial VR Game Design. In *Proceedings on Privacy Enhancing Technologies (PoPETs)*.
[57] Vivek Nair, Wenbo Guo, Justus Mattern, Rui Wang, James F O'Brien, Louis Rosenberg, and Dawn Song. 2023. Unique Identification of 50,000+ Virtual Reality Users from Head & Hand Motion Data. *arXiv preprint arXiv:2302.08927* (2023).
[58] Vivek C. Nair, Gonzalo Munilla Garrido, and Dawn Song. 2023. Going Incognito in the Metaverse: Achieving Theoretically Optimal Privacy-Usability Tradeoffs in VR. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, UIST 2023, San Francisco, CA, USA, 29 October 2023- 1 November 2023*, Sean Follmer, Jeff Han, Jürgen Steimle, and Nathalie Henry Riche (Eds.). ACM, 61:1–61:16. https://doi.org/10.1145/3586183.3606754
[59] PR Newswire. 2023. Virtual Reality (VR) In Healthcare Global Market Report 2023: Virtual Reality Gains Acceptance in Remote Home Assessments. https://finance.yahoo.com/news/virtual-reality-vr-healthcare-global-030000025.html.
[60] Ken Pfeuffer, Matthias J Geiger, Sarah Prange, Lukas Mecke, Daniel Buschek, and Florian Alt. 2019. Behavioural Biometrics in VR: Identifying People from Body Motion and Relations in Virtual Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.
[61] David A. Plecher, Maximilian Wandinger, and Gudrun Klinker. 2019. Mixed Reality for Cultural Heritage. In *IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2019, Osaka, Japan, March 23-27, 2019*. IEEE, 1618–1622. https://doi.org/10.1109/VR.2019.8797846
[62] Viktor Radulov. 2023. BeatLeader. https://www.beatleader.xyz/..
[63] Sarah Radway and Daniel Votipka. 2023. Identifying New Challenges In The Oculus Permissions Framework. In *Proceedings of the Symposium On Usable Privacy and Security (SOUPS) 2023, Poster Session*. USENIX Association.
[64] Anastasia Shuba, Athina Markopoulou, and Zubair Shafiq. 2018. NoMoAds: Effective and Efficient Cross-App Mobile Ad-Blocking. In *PETS*.
[65] Hardeep Singh and Gurvinder Singh. 2021. Feature selection methods in data mining: A comparative study. *Materials Today: Proceedings* 47, 2 (2021), 204–208.
[66] State of California Department of Justice - Office of the Attorney General. 2018. California Consumer Privacy Act (CCPA). https://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?lawCode=CIV&division=3.&title=1.81.5.&part=4.
[67] Steam. 2023. Virtual Reality Titles. https://store.steampowered.com/vr.
[68] SteamDB. 2023. Most played VR games. https://steamdb.info/charts/?tagid=21978.
[69] Sophie Stephenson, Bijeeta Pal, Stephen Fan, Earlence Fernandes, Yuhang Zhao, and Rahul Chatterjee. 2022. Sok: Authentication in augmented and virtual reality. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 267–284.
[70] Nazmi Sofian Suhaimi, Chrystalle Tan Bih Yuan, Jason Teo, and James Mountstephens. 2018. Modeling the affective space of 360 virtual reality videos based on arousal and valence for wearable EEG-based VR emotion classification. In *2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA)*. IEEE, 167–172.
[71] Luma Tabbaa, Ryan Searle, Saber Mirzaee Bafti, Md. Moinul Hossain, Jittrapol Intarasirisawat, Maxine Glancy, and Chee Siang Ang. 2021. VREED: Virtual Reality Emotion Recognition Dataset Using Eye Tracking & Physiological Measures. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 4 (2021), 178:1–178:20. https://doi.org/10.1145/3495002
[72] Pier Paolo Tricomi, Federica Nenna, Luca Pajola, Mauro Conti, and Luciano Gamberini. 2022. You Can't Hide Behind Your Headset: User Profiling in Augmented and Virtual Reality. *arXiv preprint arXiv:2209.10849* (2022).
[73] Rahmadi Trimananda, Hieu Le, Hao Cui, Janice Tran Ho, Anastasia Shuba, and Athina Markopoulou. 2022. {OVRseen}: Auditing Network Traffic and Privacy Policies in Oculus {VR}. In *31st USENIX security symposium (USENIX security 22)*. 3789–3806.
[74] Unity. 2023. Eye Gaze Interaction. https://docs.unity3d.com/Packages/com.unity.xr.openxr@1.0/manual/features/eyegazeinteraction.html.
[75] Unity. 2023. Game Player and App User Privacy Policy. https://web.archive.org/web/20230830004711/https://unity.com/legal/game-player-and-app-user-privacy-policy.
[76] Unity. 2023. Unity Analytics. https://docs.unity.com/ugs/manual/analytics/manual/overview.
[77] Valve. 2023. Steam Store. https://store.steampowered.com/.
[78] Valve. 2023. SteamVR. https://store.steampowered.com/app/250820/SteamVR/.
[79] Valve. 2024. Relive for VR FAQ. https://www.meta.com/experiences/5841245619310585/.
[80] Janus Varmarken, Hieu Le, Anastasia Shuba, Athina Markopoulou, and Zubair Shafiq. 2020. The TV is Smart and Full of Trackers: Measuring Smart TV Advertising and Tracking. In *PETS*.
[81] Alexander Winkler, Jungdam Won, and Yuting Ye. 2022. QuestSim: Human Motion Tracking from Sparse Sensors with Simulated Avatars. In *SIGGRAPH Asia 2022 Conference Papers*. 1–8.
[82] Yash and Jibin. 2023. Apple Vision Pro features, specs, price, and release date. https://www.igeeksblog.com/apple-vision-pro-features/.

A Details on Sensor Groups

In this appendix, we expand on Sections 2.2 and 4.1.3 and provide additional details and discussion of the sensor groups.

Body Motion (BM). BehaVR captures the position (x, y, z), rotation (x, y, z, w), angular (x, y, z) and linear velocity (x, y, z) from the two controllers and only position and rotation from the headset [29]. This sensor group has received much attention in prior work [41, 53, 54, 57, 81]. However, the focus was *only* on position and rotation values.

Eye Gaze (EG). BehaVR captures the position and rotation of eye gaze for both left and right eyes (7 values per eye) [25, 74]. Some of the prior work has also looked into eye data, but from different angles [60, 72]. In [60], the authors analyzed eye gaze data together with body motion data. Meanwhile, [72] looked into eye parameters (*i.e.,* pupil size and eye openness). In BehaVR, we analyze eye gaze as an independent sensor group (see Section 4.1.3).

Hand Joints (HJ). The OpenXR standard tracks the motion of each hand as a composition of *26 individually articulated joints*. See Table 3 for the full list and descriptions that follows data structure of the `XrHandJointEXT` in [27]. BehaVR captures the position and rotation of each joint [26] for each hand.

Facial Expression (FE). The OpenXR standard tracks 64 facial elements. See Table 4 to find full list and descriptions derived from the data structure of the *XrFaceExpressionFB* [28]. The 64 facial elements can be mapped to 31 Action Units (AUs) as per the Facial Action Coding System (FACS) [14]. Each AU in the FACS standard represents one facial muscle movement. The combinations of the AUs may correspond to a particular emotion. For example, the combination of AU6 (Cheek Raiser) and AU12 (Lip Corner Puller) may indicate a person smiling, which can be correlated with the emotion happiness [18]. Details regarding OpenXR facial expression elements [28] mapping to emotion AUs are in Table 5.

B List of SteamVR Apps

In Section 3, we discuss our experimental setup that includes how we choose 20 SteamVR apps from the list of top 100 SteamVR apps. Table 6 lists the 20 SteamVR apps and the activity that users perform during data collection.

C Privacy Policy Reading

In Section 2.4.1, we discuss our findings in privacy policies of 100 most played VR games. Here, we present additional details.

Availability of privacy policies. For the top 100 apps from the "Most played VR games" list on Steam, we manually visit their websites and locate the link to their privacy policies. We find that 60 of them provide privacy policies.

Reading privacy policies. We read each privacy policy and look for statements on "biometric data" or "sensory data", as well as

Table 3: List of 26 joints in the hand joints data structure per OpenXR convention [26].

No.	OpenXR Data Structure	Joint Name
1.	XR HAND JOINT PALM EXT	Palm
2.	XR HAND JOINT WRIST EXT	Wrist
3.	XR HAND JOINT THUMB METACARPAL EXT	Thumb Metacarpal
4.	XR HAND JOINT THUMB PROXIMAL EXT	Thumb Proximal
5.	XR HAND JOINT THUMB DISTAL EXT	Thumb Distal
6.	XR HAND JOINT THUMB TIP EXT	Thumb Tip
7.	XR HAND JOINT INDEX METACARPAL EXT	Index Metacarpal
8.	XR HAND JOINT INDEX PROXIMAL EXT	Index Proximal
9.	XR HAND JOINT INDEX INTERMEDIATE EXT	Index Intermediate
10.	XR HAND JOINT INDEX DISTAL EXT	Index Distal
11.	XR HAND JOINT INDEX TIP EXT	Index Tip
12.	XR HAND JOINT MIDDLE METACARPAL EXT	Middle Metacarpal
13.	XR HAND JOINT MIDDLE PROXIMAL EXT	Middle Proximal
14.	XR HAND JOINT MIDDLE INTERMEDIATE EXT	Middle Intermediate
15.	XR HAND JOINT MIDDLE DISTAL EXT	Middle Distal
16.	XR HAND JOINT MIDDLE TIP EXT	Middle Tip
17.	XR HAND JOINT RING METACARPAL EXT	Ring Metacarpal
18.	XR HAND JOINT RING PROXIMAL EXT	Ring Proximal
19.	XR HAND JOINT RING INTERMEDIATE EXT	Ring Intermediate
20.	XR HAND JOINT RING DISTAL EXT	Ring Distal
21.	XR HAND JOINT RING TIP EXT	Ring Tip
22.	XR HAND JOINT LITTLE METACARPAL EXT	Little Metacarpal
23.	XR HAND JOINT LITTLE PROXIMAL EXT	Little Proximal
24.	XR HAND JOINT LITTLE INTERMEDIATE EXT	Little Intermediate
25.	XR HAND JOINT LITTLE DISTAL EXT	Little Distal
26.	XR HAND JOINT LITTLE TIP EXT	Little Tip

Table 4: List of elements in the facial expression data structure as per OpenXR convention [28] mapped into Action Units (AU). There are a total of 64 elements of facial expression that are mapped into 31 AUs.

No.	Facial Elements in OpenXR Data Structure	Action Unit (AU)	AU#
1.	XR FACE EXPRESSION BROW LOWERER L FB	Brow Lowerer	AU4
2.	XR FACE EXPRESSION BROW LOWERER R FB		
3.	XR FACE EXPRESSION CHEEK PUFF L FB	Cheek Puff	AU34
4.	XR FACE EXPRESSION CHEEK PUFF R FB		
5.	XR FACE EXPRESSION CHEEK RAISER L FB	Cheek Raiser	AU6
6.	XR FACE EXPRESSION CHEEK RAISER R FB		
7.	XR FACE EXPRESSION CHEEK SUCK L FB	Cheek Suck	AU35
8.	XR FACE EXPRESSION CHEEK SUCK R FB		
9.	XR FACE EXPRESSION CHIN RAISER B FB	Chin Raiser	AU17
10.	XR FACE EXPRESSION CHIN RAISER T FB		
11.	XR FACE EXPRESSION DIMPLER L FB	Dimpler	AU14
12.	XR FACE EXPRESSION DIMPLER R FB		
13.	XR FACE EXPRESSION EYES CLOSED L FB	Eyes Closed	AU43
14.	XR FACE EXPRESSION EYES CLOSED R FB		
15.	XR FACE EXPRESSION EYES LOOK DOWN L FB	Eyes Look Down	AU64
16.	XR FACE EXPRESSION EYES LOOK DOWN R FB		
17.	XR FACE EXPRESSION EYES LOOK LEFT L FB	Eyes Look Left	AU61
18.	XR FACE EXPRESSION EYES LOOK LEFT R FB		
19.	XR FACE EXPRESSION EYES LOOK RIGHT L FB	Eyes Look Right	AU62
20.	XR FACE EXPRESSION EYES LOOK RIGHT R FB		
21.	XR FACE EXPRESSION EYES LOOK UP L FB	Eyes Look Up	AU63
22.	XR FACE EXPRESSION EYES LOOK UP R FB		
23.	XR FACE EXPRESSION INNER BROW RAISER L FB	Inner Brow Raiser	AU1
24.	XR FACE EXPRESSION INNER BROW RAISER R FB		
25.	XR FACE EXPRESSION JAW DROP FB	Jaw Drop	AU26
26.	XR FACE EXPRESSION JAW SIDEWAYS LEFT FB	Jaw Sideways	AU30
27.	XR FACE EXPRESSION JAW SIDEWAYS RIGHT FB		
28.	XR FACE EXPRESSION JAW THRUST FB	Jaw Thrust	AU29
29.	XR FACE EXPRESSION LID TIGHTENER L FB	Lid Tightener	AU7
30.	XR FACE EXPRESSION LID TIGHTENER R FB		
31.	XR FACE EXPRESSION LIP CORNER DEPRESSOR L FB	Lip Corner Depressor	AU15
32.	XR FACE EXPRESSION LIP CORNER DEPRESSOR R FB		
33.	XR FACE EXPRESSION LIP CORNER PULLER L FB	Lip Corner Puller	AU12
34.	XR FACE EXPRESSION LIP CORNER PULLER R FB		
35.	XR FACE EXPRESSION LIP FUNNELER LB FB	Lip Funneler	AU22
36.	XR FACE EXPRESSION LIP FUNNELER LT FB		
37.	XR FACE EXPRESSION LIP FUNNELER RB FB		
38.	XR FACE EXPRESSION LIP FUNNELER RT FB		
39.	XR FACE EXPRESSION LIP PRESSOR L FB	Lip Pressor	AU24
40.	XR FACE EXPRESSION LIP PRESSOR R FB		
41.	XR FACE EXPRESSION LIP PUCKER L FB	Lip Pucker	AU18
42.	XR FACE EXPRESSION LIP PUCKER R FB		
43.	XR FACE EXPRESSION LIP STRETCHER L FB	Lip Stretcher	AU20
44.	XR FACE EXPRESSION LIP STRETCHER R FB		
45.	XR FACE EXPRESSION LIP SUCK LB FB	Lip Suck	AU28
46.	XR FACE EXPRESSION LIP SUCK LT FB		
47.	XR FACE EXPRESSION LIP SUCK RB FB		
48.	XR FACE EXPRESSION LIP SUCK RT FB		
49.	XR FACE EXPRESSION LIP TIGHTENER L FB	Lip Tightener	AU23
50.	XR FACE EXPRESSION LIP TIGHTENER R FB		
51.	XR FACE EXPRESSION LIPS TOWARD FB	Lips Toward	AU8
52.	XR FACE EXPRESSION LOWER LIP DEPRESSOR L FB	Lip Depressor	AU16
53.	XR FACE EXPRESSION LOWER LIP DEPRESSOR R FB		
54.	XR FACE EXPRESSION MOUTH LEFT FB	Mouth Stretch	AU27
55.	XR FACE EXPRESSION MOUTH RIGHT FB		
56.	XR FACE EXPRESSION NOSE WRINKLER L FB	Nose Wrinkler	AU9
57.	XR FACE EXPRESSION NOSE WRINKLER R FB		
58.	XR FACE EXPRESSION OUTER BROW RAISER L FB	Outer Brow Raiser	AU2
59.	XR FACE EXPRESSION OUTER BROW RAISER R FB		
60.	XR FACE EXPRESSION UPPER LID RAISER L FB	Upper Lid Raiser	AU5
61.	XR FACE EXPRESSION UPPER LID RAISER R FB		
62.	XR FACE EXPRESSION UPPER LIP RAISER L FB	Upper Lip Raiser	AU10
63.	XR FACE EXPRESSION UPPER LIP RAISER R FB		
64.	XR FACE EXPRESSION COUNT FB	Count	

Table 5: Mapping between emotions, arousal/valence states (LA = low arousal, HA = high arousal, PV = positive valence, NV = negative valence), from Table 4 derived from OpenXR facial expression elements [28], and Action Units (AU) [15].

Emotion	Arousal/Valence	Facial Element No.	AUs No.
Happiness	HA/PV	(5, 6) + (33, 34)	AU6 + AU12
Surprise	LA/PV	(23, 24) + (58, 59) + (60, 61) + (25)	AU1 + AU2 + AU5 + AU26
Anger	HA/NV	(1, 2) + (60, 61) + (29, 30) + (49, 50)	AU4 + AU5 + AU7 + AU23
Contempt	HA/NV	33 + (11, 12)	AU12 + AU14
Disgust	HA/NV	(56, 57) + (31, 32) + (52, 53)	AU9 + AU15 + AU16
Fear	LA/NV	(23, 24) + (58, 59) + (1, 2) + (60, 61) + (29, 30) + (43, 44) + (25)	AU1 + AU2 + AU4 + AU5 + AU7 + AU20 + AU26
Sadness	LA/NV	(23, 24) + (1, 2) + (31, 32)	AU1 + AU4 + AU15
All	All	All emotion elements	All AUs

specific types (*e.g.,* "head movement") in relevant sections about data collection, use and sharing. However, human reading is prone to omission due to the lengthy text and extra work required to reveal some contents (*e.g.,* collapsible text). In addition, 4 privacy policies are not in English. To complement the reading, we write a script to call the ChatGPT (GPT-4) model to read the whole text and ask it to report any statements about data types of interest. We also use simple string matching to search for relevant content.

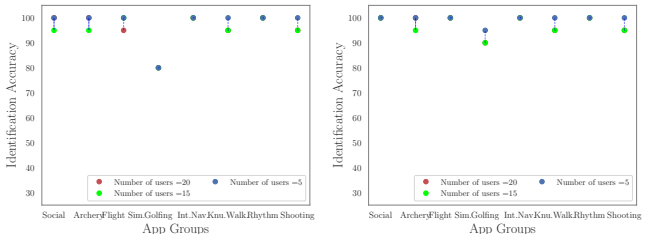
VR sensor data in apps' privacy policies. Among 60 games that provide privacy policies, we find that 10 of them discuss the collection of sensor data. Table 7 shows the list of 10 apps and what sensor data they disclose to collect. We find that only seven

Table 6: List of 20 VR apps in the BehaVR app corpus.

App No.	App Title	Tasks
a_1	Beat Saber	Cut objects with light-sabers: with the controllers and then with bare hands.
a_2	BONEWORKS	Explore the welcome scene; in front of a shelf, the user is prompted to grab dumbbells with bare hands and exercise.
a_3	DCS World Steam Edition	Fly a military aircraft: the user first control the aircraft with controllers and then with bare hands.
a_4	Derail Valley	Explore the scene in a train station; in front of a table, the user is interacting with a book and a walkie-talkie using bare hands.
a_5	Elven Assasin	Shoot arrows to monsters: with the controllers and then with bare hands.
a_6	Golf It!	Putt a golf ball with the controllers; once it gets close to the hole, the user is prompted to continue with bare hands.
a_7	Gorilla Tag	Perform gorilla movement (walk like gorilla to explore the environment): first with the controllers, then with bare hands.
a_8	Hot Dogs, Horseshoes & Hand Grenades	Explore a virtual park; in front of a vending machine, the user will interact with it with bare hands.
a_9	Job Simulator	Explore office-worker simulation; The user is to interact with a virtual office objects with controllers and then with bare hands.
a_{10}	Keep Talking and Nobody Explodes	Defusing a bomb with the controllers; then, the user is prompted to defuse the bomb with bare hands.
a_{11}	McOsu	Explore the welcome scene; The user is also asked to interact with the virtual objects with bare hands.
a_{12}	Neos VR	Explore a futuristic building; the user interacts with books in a bookshelf first using the controllers and then with bare hands.
a_{13}	No Man's Sky	Explore an unknown planet by teleporting; the user interacts with a laser gun (shoot targets) with controllers and then with bare hands.
a_{14}	Pavlov VR	Play & practice the basic and shootings; The user is interacting with a panel with bare hands.
a_{15}	Rec Room	Explore a school or a McDonald or virtual recreation center; The user will wave their hands at an avatar with bare hands.
a_{16}	Space Engine	Explore a virtual planetarium by teleporting to space objects (*e.g.,* planets, stars, etc.); the user is asked to interact with the planetarium first with the controllers and then with bare hands.
a_{17}	Tabletop Simulator	Move chess pieces: first with the controllers and then with bare hands.
a_{18}	VRChat	Explore the virtual scene by walking around; The user will wave or greet with bare hands.
a_{19}	VTOL VR	Fly a helicopter; The user interact with the control panel and stick with controller and then with bare hands.
a_{20}	X-Plane 11	Fly a civilian aircraft and interact with the virtual objects with the controllers and with bare hands.

Table 7: Sensory and biometric data types discussed in the privacy policies of the top 100 VR apps on Steam.

App	Collected Data Types
iRacing	sensory data, biometric data
Arizona Sunshine	motion sensor information, motion tracker information
Rec Room	sensory data, head movement, facial expressions
DeoVR Video Player	motion sensor events
Gorilla Tag	movement data (hands and head)
Microsoft Flight Simulator (2 app versions)	skeletal tracking data, sensor data
VRChat	sensory information, biometric information (ambiguous)
One-armed Cook	biometric information (ambiguous)
WGT Golf	biometric information (ambiguous)



(a) Body Motion **(b) Facial Expression**

Figure 6: Visualization of identification accuracy changes by varying number of users.

privacy policies *clearly* mention the collection of "biometric data" and/or "sensory data". Some of them mention more specific data types, such as "head and hand movement", "facial expressions", and "skeletal tracking". In addition, three privacy policies give *ambiguous* statements about the collection of these data. For example, in VRChat's privacy policy [36], "California Resident Privacy Notice" table marks no collection and disclosure in the row "sensory information", but the "Disclosure of Personal Information" section states "sensory information" is shared with vendors.

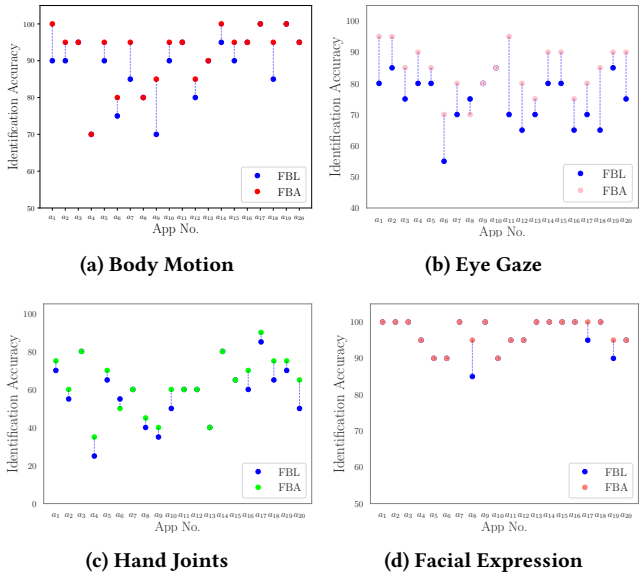
VR sensor data in Meta's privacy policy. We additionally read the privacy policy of Meta, the vendor of Quest Pro. In contrast to the scarce discussion of sensor data in VR apps' privacy policies, Meta provides a long list of sensor and biometric data that are collected in its privacy policy and supplemental articles [34, 46, 51]. In the paragraph of "Physical characteristics and movements", it discloses the collection of the position and orientation of the headset and controllers, the speed of controller movement, hand tracking, eye tracking, facial expression and other data types. The list clearly covers the types of sensor data explored in this work. Indeed, as the platform, Meta has the best vantage point to collect these data, which can potentially be used for user identification, *i.e.,* personalization [34].

D More about the User Study Participants

In this appendix, we expand Section 3.2 and provide additional details regarding BehaVR user study participants.

The demographic distributions of the participants are as follows: female is 9 (45%), male is 11 (55%). The age ranges for the participants is between 20-40 with a median age of 26 and mean age of ∼ 28. The nationality of the participants are 4 (20%) Indian , 3 (15%) Chinese , 6 (30%) other Asian , 3 (15%) American, 2 (10%) European and 2 (10%) Undisclosed. Height distributions of the users are, 4 users (20%) <160cm, 9 (45%) between 160 to 175cm, 5 (25%) >175cm and 2 (10%) undisclosed. Dominant hand (using mostly left or right hand to interact with virtual objects) of the users are 19 (95%) right-handed and 1 ambidextrous (5%).

Among them, 10 (50%) of users have prior VR experiences , 9 (45%) of them was trained during our study by the authors and 1 (5%) did not disclose his/her experience. Prior works show that even with 500 [53] or 5000[57] users, the identification accuracy does not drop with a significant amount using a simple RF or XGB models. As a proof of concept, we conducted experiments on both the BM and FE groups (for one app from each app group), varying the number of users. Our results show that accuracy does not fluctuate largely through varying numbers of users (See Fig. 6).



(a) Body Motion (b) Eye Gaze

(c) Hand Joints (d) Facial Expression

Figure 7: Identification accuracy comparison between FBA and FBL methods for the four sensor groups.

E More Details on Data Processing and User Identification Models

This appendix expands Section 4.1.2, where we outlined the process of converting time series data into feature blocks, and Section 4.2, where we discussed building the BehaVR models. Additionally, we provide insights into optimizing FBA and how we select specific model architecture for user identification in BehaVR.

E.1 More about Data Processing

Pre-processing. This step aims to obtain valid time series data with unique timestamps. First, we de-duplicate timestamps and delete invalid columns (*e.g.,* columns with only zeros). Next, we check any data corruption (*e.g.,* rows that contain error messages) and replace the invalid values using neighboring rows.

Block Division. In order to be able to divide the time series in more or less number of blocks, with much shorter or longer duration than 1 second accordingly, we introduce parameter $r \in (0, 2]$, which controls the final amount of blocks ("final block amount") for each app a_j: $N_{FBA_j} = r \cdot N_j$. When we increase the ratio r, we increase the final block amount while decreasing the block length (amount of time per block). Thus, to align r values across all sensor groups, we choose $r = 1$. The key insight here is, unlike FBL, FBA takes into account the variability across users to scale the number of blocks for each app, N_{FBA_j}, as to align similar user-app interactions in the time series.

Summarization. We summarize the information in the time series of each block with a vector of 5 statistics, *i.e.,* maximum, minimum, mean, standard deviation, and median within each block, which will serve as features next. This summarization was originally proposed in [53] for Body Motion and was also used in [57].

Table 8: Feature dimensions and block counts for summarized sensor data using the FBA method for different r, the parameter that adjusts the block numbers as described in Section 4.1.2. We report the (number of blocks, number of features) for each sensor group and corresponding r.

Sensor Group	$r = 2$	$r = 1$	$r = 0.5$	$r = 0.2$	$r = 0.1$
Body Motion	(150658, 165)	(75342, 165)	(37834, 165)	(15133, 165)	(7468, 165)
Eye Gaze	(168400, 46)	(84200, 46)	(41920, 46)	(16520, 46)	(8080, 46)
Hand Joints	(58480, 400)	(29240, 400)	(14360, 400)	(5480, 400)	(2520, 400)
Facial Expression	(168400, 320)	(84200, 320)	(41920, 320)	(16520, 320)	(8080, 320)

Table 9: Performance analysis for algorithm selection.

Algorithm	App No.	Accuracy (%)			
		BM	EG	HJ	FE
RF	a_1	100	100	100	100
RF	a_{15}	100	100	100	100
XGB	a_1	100	100	100	100
XGB	a_{15}	85.71	85.71	71.42	100
SVM	a_1	57.14	57.14	85.71	100
SVM	a_{15}	38.23	38.23	71.42	38.23

Block Post-Processing. In this step, we verify the block's validity by checking each block (rows) and then each feature (columns). Initially, we eliminate invalid blocks and estimate missing values (e.g., filling missing values in HJ data with related ones). Finally, we refine the feature vectors for the four sensor groups by removing undesirable features (e.g., those with all zero/one values or irrelevant to the classification task).

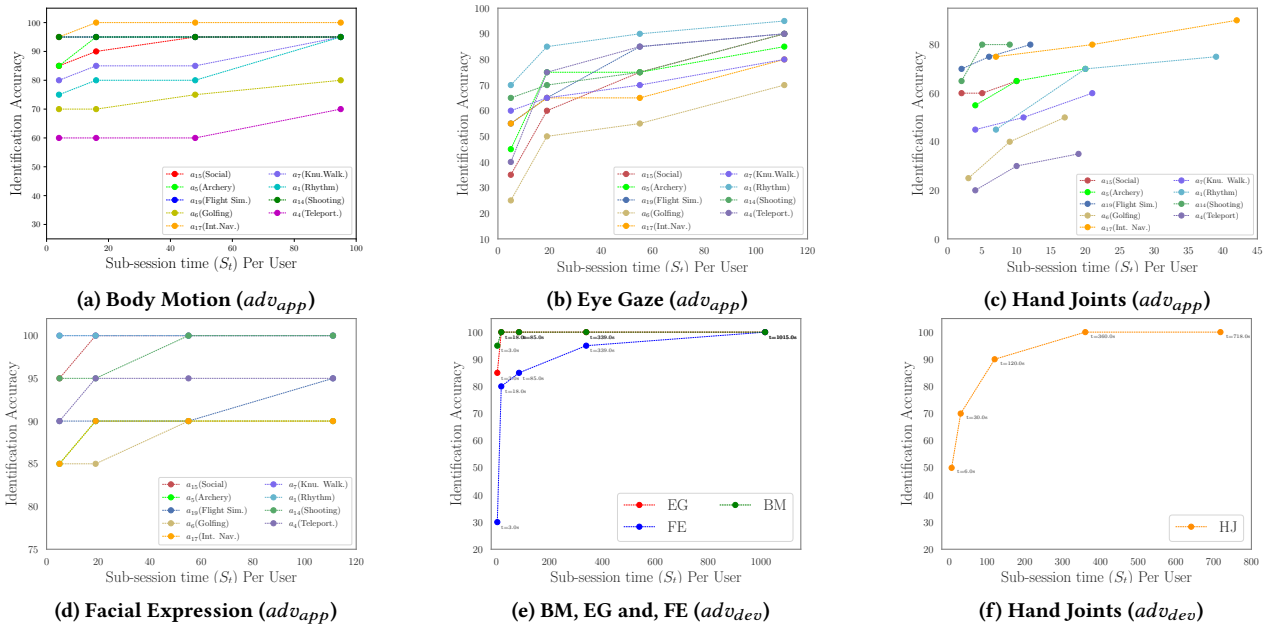
E.2 FBA Evaluation and Optimization

We evaluate and compare FBL and FBA in Figures 7a, 7b, 7c and 7d. We can observe that FBA improves app model identification accuracy ($5-15\%$ for body motion, $5-25\%$ for eye gaze, and $5-10\%$ for hand joints) for most apps compared to FBL, supporting the decision to use FBA over FBL across our experiments.

Hyperparameter Tuning. In BehaVR, for RF, first we tune hyperparameters by varying *n-estimators* and *max-depth* from $(50, 200)$ and $(1, 20)$ respectively in five iterations. Then, we select the best model based on five-fold cross-validation. Finally, based on the accuracy obtained from the primary analysis, we choose the optimal point of FBA ratio r: our evaluations rely on this final model.

Choosing Optimal Ratio r. The main challenge when using FBA is finding the optimal FBA block division ratio r. If r is too high, summarized data become noisy. On the contrary, if r is too low, important information would be missing from the summary. Finding the right balance is crucial to preserve relevant information. We perform a preliminary experiment based on 7 participants to find the optimal value of r. For body motion and eye gaze, the results suggest $r = 1$. For hand joints, the results suggest $r = 0.5$; this is intuitive as a meaningful hand gesture can be captured in a longer block length. For facial expression, the results suggest that any r values will be optimal.

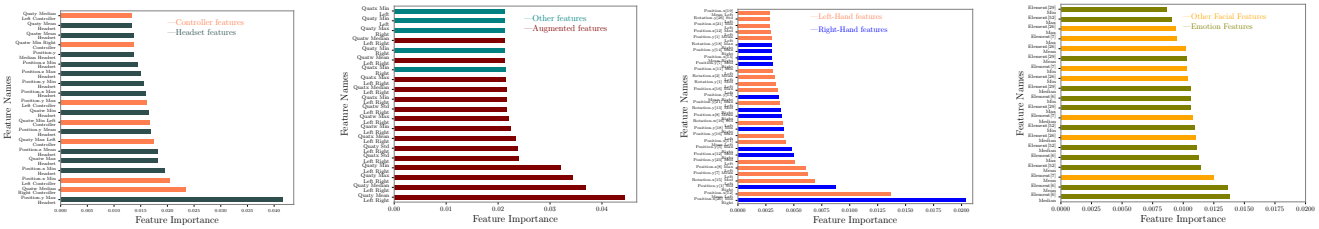
Data Summarization Output. Table 8 presents the dimensions of the output of data summarization using FBA.



(a) Body Motion (adv_{app}) (b) Eye Gaze (adv_{app}) (c) Hand Joints (adv_{app})

(d) Facial Expression (adv_{app}) (e) BM, EG and, FE (adv_{dev}) (f) Hand Joints (adv_{dev})

Figure 8: User identification accuracy for app and device models across four sensor groups, with respect to the average sub-session time (S_t in seconds) per user.



(a) Top-20 features for BM (b) Top-20 features for EG (c) Top-30 features for HJ (d) Top-20 features for FE

Figure 9: Top features for user identification for device adversary w.r.t. each of the four sensor groups.

E.3 Algorithm Selection

We initially explore various ML models such as Random Forest (RF) [40], Gradient Boosting (XGB) [19], Support Vector Machine (SVM) [8], and Long Short-Term Memory Networks (LSTM) [33] across two apps : consist of one social app, namely Rec Room (a_{15}), and rhythm app, namely Beat Saber (a_1). We analyze the two apps (out of 20), which are among the most popular VR apps, as they contain common activities (*e.g.,* walking, waving, grabbing, etc.). Table 9 shows that RF achieves the highest identification accuracy; We argue that LSTM is intended to perform sequence prediction, whereas BehaVR focuses on identification (*i.e.,* a classification task); LSTM performs poorly (~81% accuracy for body motion in app a_1), thus, we do not consider LSTM further in our evaluation.

F More Evaluation Results

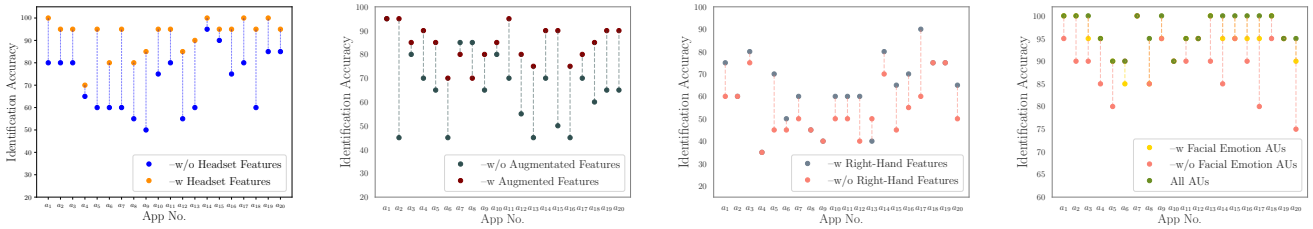
In Section 5, we presented evaluation results of BehaVR's app and device models for user identification. In this appendix, we present additional tables and figures related to evaluation.

Sub-session Time Characterization. Figures 8a, 8b, 8c, and 8d (for app adversary); and Figures 8e and 8f (for device adversary) show identification accuracy w.r.t. sub-session time.

Top Features. We list the top features for user identification using app models trained with data from the four sensor groups in Table 10. Further, Figures 9a, 9b, 9c, and 9d show the top features for user identification for device adversary. In Figure 10a, 10b, 10c and 10d shows importance of headset features for BM, augmented features for EG, right-hand features of HJ and Finally, AUs/elements of emotion for FE respectively.

Identification Accuracy Based on Emotion Action Units. In Table 11 shows the identification accuracy based on combinations of AUs that represent emotions based on different app groups.

Identification Accuracy for Open-World Settings. In Table 12, the identification accuracy for 5 representative apps from five different app-groups is shown, given that the training and testing data are collected from different settings, difficulty levels, or songs (open world settings).



(a) Headset features of BM **(b) Augmented features of EG** **(c) Right-hand features of HJ** **(d) Emotion AUs of FE**

Figure 10: Visualization of identification accuracy improvement for each of the four sensor groups w.r.t. top-features.

Table 10: Top-3 features in user identification for app models for each of the four sensor groups.

App No.	Body Motion	Eye Gaze	Hand Joints	Facial Expression
a_1	Position.z Mean Left Controller, Position.z Min Headset', Position.y Median Right Controller	Quat.y Mean Left Right, Quat.y Median Left Right, Quat.y Max Left Right	Position.z[3] Max Right, Rotation.z[24] Mean Left, Position.z[1] Max Right	Element[23] Min, Element[5] Median, Element[6] Mean
a_2	Position.x Max Headset, Position.y Max Headset, Position.x Mean Headset	Quat.y Mean Left Right, Quat.y Median Left Right, Quat.x Mean Right	Position.z[26] Med Left, Rotation.z[2] Med Right, Rotation.z[18] Min Left	Element[5] Min, Element[57] Median, Element[5] Median
a_3	Position.x Mean Headset, Position.z Max Headset, Quat.y Median Headset	Quat.y Mean Left Right, Quat.y Mean Left, Quat.x Max Right	Quat.y Mean Left Right, Quat.y Max Left Right	Element[28] Min, Element[51] Min, Element[51] Median
a_4	Position.z Min Headset, Position.y Max Headset, Position.z Max Headset	Quat.y Max Left Right, Quat.y Mean Left Right, Rotation.w Med Left	Position.y Mean Right, Rotation.z[25] Max Right, Position.x[26] Mean Right	Element[30] Min, Element[29] Mean, Element[29] Min
a_5	Position.y Min Left Controller, Lin.0 Std Right Controller, Quat.z Mean Right Controller	Quat.y Min Left Right, Quat.y Mean Left Right, Quat.y Max Left Right	Position.x Mean Right, Rotation.z[3] Max Left, Rotation.z[11] Med Left	Element[6] Min, Element[57] Mean, Element[5] Mean
a_6	Position.x Min Headset, Position.x Max Headset, Quatw Max Headset	Quat.y Mean Left Right, Quat.y Median Left Right, Quatw Mean Left	Rotation.z Min Left, Rotation.x Max Left, Position.y Max Right	Element[26] Min, Element[57] Mean, Element[5] Mean
a_7	Quat.w Mean Headset, Position.x Mean Headset, Quat.x Min Right Controller	Quat.y Mean Left Right, Quat.y Median Left Right, Quat.y Max Left Right	Position.x Mean Left, Position.z Max Left, Rotation.y Min Left	Element[5] Median, Element[2] Min, Element[6] Median
a_8	Position.y Max Headset, Position.y Mean Headset, Position.y Median Headset	Quat.y Mean Left Right, Quat.y Median Left Right, Quat.y Max Left Right	Rotation.z Max Right, Rotation.x Mean Right, Position.x Min Right	Element[30] Min, Element[29] Min, Element[6] Median
a_9	Quat.y Min Right Controller, Position.x Mean Right Controller, Quat.x Max Right Controller	Quat.y Mean Left Right, Quat.y Median Left Right, Quat.y Max Left Right	Position.z Max Left, Position.y Mean Right, Position.z[2] Max Left	Element[30] Min, Element[6] Mean, Element[27] Max
a_{10}	Position.x Median Headset, Position.x Max Headset, Position.x Mean Headset	Quat.y Mean Left Right, Quat.y Median Left Right, Quatw Mean Left	Position.y Mean Right, Rotation.y[12] Min Left, Rotation.z[6] Max Left	Element[29] Min, Element[25] Min, Element[2] Min
a_{11}	Position.y Max Headset, Position.y Mean Headset, Position.x Min Headset	Quat.y Mean Left Right, Quat.y Median Left Right, Quat.y Max Left Right	Position.z Min Left, Position.x[11] Max Right, Position.x Mean Left	Element[51] Min, Element[51] Median, Position.x Mean Left
a_{12}	Position.y Max Headset, Position.y Mean Headset, Position.y Min Headset	Quat.y Mean Left Right, Quat.y Median Left Right, Quat.y Max Left Right	Position.x Min Right, Position.x[24] Mean Right, Position.x[17] Min Right	Element[51] Median, Element[51] Min, Element[51] Mean
a_{13}	Position.y Max Headset, Position.x Mean Headset, Position.y Mean Headset	Quat.y Mean Left Right, Quat.y Median Left Right, Quat.y Min Left Right	Position.x Mean Right, Position.z[15] Min Right, Position.x[5] Mean Right	Element[51] Min, Element[6] Min, Element[25] Max
a_{14}	Quat.x Mean Right Controller, Position.z Min Left Controller, Quat.w Mean Left Controller	Quat.y Mean Left Right, Quat.x Max Left, Quatw Max Right	Position.x Mean Left, Position.y[24] Mean Left, Position.x[7] Med Right	Element[51] Median, Element[25] Median, Element[51] Min
a_{15}	Position.y Max Headset, Position.x Max Headset, Position.x Mean Headset	Quat.y Mean Left Right, Quat.y Median Left Right, Quat.y Max Left Right	Position.x Mean Right, Position.x[14] Mean Right, Position.x[6] Med Left	Element[51] Min, Element[23] Min, Element[25] Median
a_{16}	Position.y Max Headset, Position.x Min Headset, Position.x Mean Headset	Quat.y Mean Left Right, Quat.y Median Left Right, Quat.y Min Left Right	Position.z Min Left, Rotation.z[3] Med Left, Position.x[12] Mean Right	Element[25] Min, Element[5] Mean, Rotation.z[3] Med Left
a_{17}	Position.x Min Headset, Position.x Mean Headset, Position.y Max Headset	Quat.y Mean Left Right, Quat.y Median Left Right, Quat.y Max Left Right	Position.y Med Left, Rotation.z[22] Min Right, Rotation.x[25] Mean Right	Element[50] Min, Element[41] Mean, Element[54] Mean
a_{18}	Position.y Max Headset, Position.z Mean Headset, Position.y Median Headset	Quat.y Mean Left Right, Quat.y Median Left Right, Quat.x Mean Right	Position.z[13] Mean Right, Position.z[8] Max Left, Position.y[18] Med Left	Element[25] Median, Element[2] Min, Position.z[8] Max Left
a_{19}	Quat.x Min Right Controller, Quatw Max Left, Position.y Median Right Controller	Quat.y Mean Left Right, Quat.x Min Right, Quatw Max Left	Position.z Max Left, Position.z[18] Med Left, Position.y[1] Med Left	Element[51] Min, Element[51] Mean, Element[25] Median
a_{20}	Position.x Min Headset, Position.x Median Headset, Position.x Mean Headset	Quat.y Mean Left Right, Quat.y Median Left Right, Quat.y Max Left Right	position.x[11] Max Left, Position.x[12] Med Left, Position.x[4] Max Left	Element[30] Min, Element[30] Mean, Element[5] Mean

Table 11: Identification accuracy (in %) based on combinations of AUs that represent emotions w.r.t. app groups; *Emotional States:* LA = low arousal, HA = high arousal, PV = positive valence, NV = negative valence.

Emotion	Arousal/Valance	Identification accuracy (%) in App Groups									
		Social		Flight Sim.		Int. Nav.		K.-walk.	Rhy.	Shooting	Archery
		a_{18}	a_{15}	a_{19}	a_{20}	a_{16}	a_{10}	a_7	a_1	a_{14}	a_5
Happiness	HA/PV	100.0	100.0	85.0	70.0	80.0	75.0	95.0	95.0	80.0	70.0
Surprise	LA/PV	100.0	95.0	85.0	80.0	80.0	85.0	100.0	100.0	90.0	85.0
Anger	HA/NV	95.0	95.0	95.0	85.0	85.0	85.0	90.0	95.0	90.0	85.0
Disgust	HA/NV	75.0	75.0	70.0	55.0	60.0	75.0	70.0	70.0	75.0	75.0
Fear	LA/NV	90.0	95.0	90.0	90.0	90.0	95.0	100.0	100.0	95.0	90.0
Sadness	LA/NV	85.0	90.0	100.0	90.0	80.0	80.0	90.0	90.0	95.0	85.0
All Emotion AUs	All	95.0	100.0	95.0	90.0	95.0	90.0	100.0	100.0	100.0	85.0
All AUs	All	95.0	100.0	100.0	95.0	100.0	90.0	100.0	100.0	100.0	90.0

Table 12: Evaluation Results for the Open-World Setting.

App No.	App Group	Accuracy (%)			
		BM	EG	HJ	FE
Social	a_{15}	80	60	60	100
Int.Nav.	a_{17}	100	80	70	90
Knu.walk.	a_7	90	70	60	90
Rhythm	a_1	80	60	60	80
Shoot.& Arch.	a_5	100	70	80	90

Table 13: Evaluation Results for Model Ensemble (BM = Body Motion, EG = Eye Gaze, HJ = Hand Joints, BM&EG = Ensemble of Body Motion and Eye Gaze models, EG&HJ = Ensemble of Eye Gaze and Hand Joints models).

App No.	App Group	Accuracy (%)				
		BM	EG	HJ	BM&EG	EG&HJ
Social	a_{12}	85	80	60	95	80
Teleportation	a_4, a_8	75,80	90,70	35,45	100, 90	90, 80
Flight Simulation	a_3, a_{20}	95,95	85,75	80,75	–,–	90,85
Knu.Walking	a_7	95	80	65	–	85
Int. Nav.	a_2, a_9	95,80	80,80	60,60	–,80	90,80
Golfing	a_6	80	70	50	90	80

Sensor Group Model Ensemble Results. Table 13 represents the identification accuracy for the attacker that ensemble multiple sensor group models and then calculates the final attack accuracy. The first three sub-columns of the Accuracy column represent individual sensor group accuracy (*e.g.,* either for BM, EG, or HJ). If individual sensor group identification accuracy is low, the attacker further ensemble those weak models of multiple sensor groups, as presented in the last two columns (BM&EG and EG&HJ). Any empty value on the table indicates that the individual sensor model provides high identification accuracy, the attacker further did not optimize it.