Toward Safer Augmented Reality: Securing Input, Output, and Interaction

Kaiming Cheng

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2025

Reading Committee:

Franziska Roesner, Chair

Tadayoshi Kohno, Chair

Jon E. Froehlich

Program Authorized to Offer Degree:
Computer Science & Engineering

University of Washington

Abstract

Toward Safer Augmented Reality: Securing Input, Output, and Interaction

Kaiming Cheng

Co-Chairs of the Supervisory Committee:
Franziska Roesner
Paul G. Allen School of Computer Science & Engineering

Tadayoshi Kohno
Paul G. Allen School of Computer Science & Engineering

Augmented Reality (AR) technologies have evolved significantly over the years. Once considered niche and expensive research prototypes, AR devices are becoming increasingly accessible and powerful. In addition to the hardware advancements, the application development ecosystems and AI capabilities integrated into these devices have also rapidly expanded. These advancements in AR will soon empower individuals to use AR on an everyday basis. As millions of users begin to explore AR technologies and incorporate them into their daily lives, safeguarding users' security and privacy from unwanted threats becomes ever more imperative. Due to AR devices' ability to alter users' perceptions of the physical world, the nature of their three-dimensional user interface, and multi-modal sensing capabilities, many of these threats are fundamentally different from known risks of non-immersive technologies like web and mobile interfaces.

In this dissertation, I identify critical security and privacy risks, evaluate these risks in cutting-edge AR systems, and propose mitigation solutions to enhance user safety. My approach centers on analyzing the three core phases of the AR system data flow — input, output, and interaction — each of which introduces distinct classes of vulnerabilities.

For threats related to AR input, I investigated the emerging sensory permission models, such as eye-tracking and hand-tracking, for three major AR platforms (HoloLens 2, Oculus Quest Pro, and Vision Pro). My collaborators and I surveyed 280 participants on Prolific

to investigate their comfort, perceived and actual comprehension, and decision factors. We explicitly recruited participants who had no prior experience with AR, in order to capture people's comfort and comprehension on their first exposure to these permission-granting flows, rather than relying on their past experiences. Based on the results, we identify design principles for how future AR platforms can better communicate existing privacy protections, enhance privacy-preserving designs, and more effectively communicate potential risks.

For threats related to AR output, I present my work that formalizes the security-related properties of the 3D UI output in AR. My collaborators and I demonstrate the security implications of different instantiations of these properties through five proof-of-concept attacks between distrusting AR application components (i.e., a main app and an included library) — including a clickjacking attack and an object erasure attack. We then empirically investigate these UI security properties on five current AR platforms: ARCore (Google), ARKit (Apple), HoloLens (Microsoft), Oculus (Meta), and WebXR (browser), finding that all platforms enable at least three of our proof-of-concept attacks to succeed. We provide concrete recommendations for platform developers, including adaptations of existing 2D UI security measures and novel AR-specific defense techniques to prevent these attacks.

For threats related to AR interaction, I describe my work that characterizes perceptual manipulation attacks (PMA) in AR, which involves manipulating users' multi-sensory (e.g., visual, auditory, haptic) perceptions of the world when users are interacting with AR content. Through immersive adversarial overlaid content, PMA influence users' judgments and following actions to induce incorrect perception, cognition, or resulting reaction. To provide a foundation for understanding and addressing PMA, my collaborators and I conducted an in-person study with 21 participants with three PMA that attacked different perceptions: visual, auditory, and situational awareness. Our findings reveal the effectiveness of these attacks and inform design guidelines for defending against PMA in AR environments.

Together, this thesis represents significant theoretical and empirical progress toward secure, privacy-preserving, and trustworthy AR systems for mainstream adoption.

Table of Contents

Page

List of Figures

Figure Number Page

List of Tables

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisors, Franziska Roesner and Tadayoshi Kohno. This work would not have been possible without their unwavering support, guidance, and encouragement, which have formed the very foundation of my PhD journey. Both of you have not only been exceptional mentors in advancing my research but also inspiring role models in shaping my approach to life. I am incredibly fortunate to have had the opportunity to work with you both, and I hope my future career will make you proud.

I sincerely thank my committee member, Jon E. Froehlich, for his time and dedication in supporting my PhD thesis. Working with Jon on research at the intersection of augmented reality and accessibility has been invaluable to my development as a researcher. His genuine care for people, commitment to building community, and innovative work toward creating accessible technology have been truly inspiring. I am deeply grateful for his mentorship and guidance.

The Security and Privacy Lab at UW CSE has been my home for the past five years, and I feel extremely fortunate to be a part of this wonderful group. Special thanks to the past and present members of the Security and Privacy Lab: Arka Bhattacharya, Maddie Burbage, Inyoung Cheong, Aarushi Dubey, Yael Eiger, Pardis Emami-Naeini, Ivan Evtimov, Michael Flanders, Chris Geeng, Gregor Haas, Rachel Hong, Umar Iqbal, Karl Koscher, Aroosh Kumar, Evan Lam, Kiron Lebeck, Michelle Lin, Rachel McAmis, Alexandra Michael, Peter Ney, Kentrell Owens, Basia Radka, Lucy Simko, Mattea Sim, Anna Kornfeld Simpson, Jeffery Tian, Miranda Wei, Henry Wong, Tina Yeung, and Eric Zeng. I will deeply miss the happy hours, fun activities, summer kayaking, and winter snowboarding trips.

I'm grateful to all of the friends who have enriched my life throughout grad school, both within and beyond UW CSE: Orevaoghene Ahia, Qiwen Cui, Weixin Deng, Brandon Yushan Feng, Chenpeng Gao, Ken Gu, Xinlu Guo, Xiaochuang Han, Dong He, Daniel Jiang, Liwei Jiang, Preston Jiang, Weizhao Jin, Jaehun Jung, Tiernan Kennedy, Jaewook Lee, Jialin Li, Jeffery Li, Inna Wanyin Lin, Chuanyuan Liu, Zheyuan Liu, Alisa Liu, Sirui Lu, Anton Lykov, Fanyi Ma, Yunbi Nam, Innocent Obi Jr, Lisa Orii, Rock Pang, Zijie Pan, Cameron Perry, Ananditha Raghunath, Shwetha Rajaram, Esteban Safranchik, Reshabh Sharma, Weijia Shi, Weihao Song, Xia Su, Yujie Tao, Austin Underwood, Zihao Ye, and Kevin Zheng.

My graduate school work would not have been possible without the foundation I received at the University of Virginia. I thank Professor Mark Sherriff for introducing me to the world of computer science. I thank Professor Luke Dahl, Professor Leah Reid, and Professor Matthew Burtner for your encouragement in interdisciplinary research bridging music, computer science, and Augmented/Virtual Reality. I thank Professor Yuan Tian and Professor David Evans for their mentorship in conducting research in security and privacy.

Thanks also to my cat, Dota, for her energetic meowing and companionship during the long nights of coding and paper writing.

I also want to thank Tottenham Hotspur F.C., a team I've been supporting since the beginning of my PhD. It was such a wholesome moment to see the team finally win the European championship as I'm also graduating. To dare is to do. Come on you Spurs!

My deepest gratitude goes to my family. Thank you to my parents, Jiangang Cheng and Peiying Gong, and my grandparents, Xuezhong Cheng, Shuqin Ma, Weiguo Pei, and Jinfang Gong, for their unconditional love and support. They are my heroes who worked so hard to provide a better living for our family. It's my honor to carry forward your legacy.

Dedication

To my parents, Jiangang Cheng and Peiying Gong, for their continuous support, encouragement, motivation, and love.

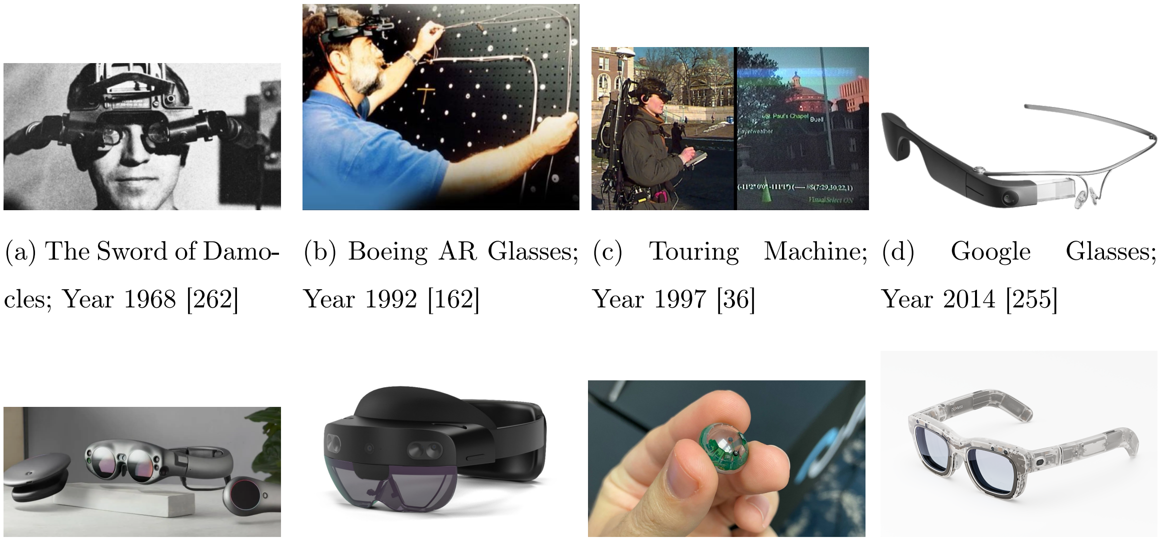
Chapter 1

Introduction

Humans evolved to navigate and understand a complex physical world, developing sophisticated cognitive processes for learning, spatial understanding, and reasoning through direct interaction and exploration. Augmented Reality (AR) — technologies that seamlessly blend three-dimensional virtual content with the physical surroundings — extend our innate, spatial way of thinking by transforming our world itself into a computational medium.

This vision for AR was introduced back in the late 1960s, when Ivan Sutherland developed the first head-mounted display prototype [262]. In the early 1990s, Caudell and Mizell demonstrated the first practical application of AR in assisting human-involved operations in aircraft manufacturing [266]. In 1997, the "touring machine" was designed as the first mobile outdoor AR systems, taking AR outside of controlled environments. With breakthroughs in hardware technologies as well as the advancements in platforms and application development ecosystems in the past decade, a series of AR devices have emerged targeting consumer markets, increasing public awareness and adoption. Google Glass marked the first consumer head-mounted AR device with voice-activated functionality in 2014. Magic Leap One in 2018 separated the display unit from the processing unit to distribute weight away from the head for enhanced comfort. Microsoft's HoloLens 2 in 2019 provided advanced spatial mapping and gesture control capabilities. Mojo's Smart Contact Lens demonstrated the first on-eye micro-LED display technology. More recently, Meta introduced Orion, aimed at achieving lightweight, all-day wearable AR glasses. Figure 1.1 represents the recent evolution of AR devices, evolving from bulky, rudimentary devices to today's lightweight, ergonomic, yet remarkably powerful form factors.

Moreover, the recent explosion of generative and contextual AI has significantly improved AR functionality and capability. Embedding AI models and biometric sensors into AR systems has enabled intelligent, context-aware interfaces capable of dynamically responding to users' surroundings and actions. Ergonomic interaction techniques such as microgestures [165], neural interfaces [49], and gaze-driven selection [257] were developed to reduce user fatigue and enhance usability. The proliferation of generative AI models offers new avenues for personalizing AR content and multimodal understanding of the physical world [186, 253, 283]. With the market for AR devices surpassing two million units in 2024 and experiencing an over 200% growth rate, AR is rapidly evolving from niche research technology to mainstream consumer adoption.



(a) The Sword of Damocles; Year 1968 [262]

(b) Boeing AR Glasses; Year 1992 [162]

(c) Touring Machine; Year 1997 [36]

(d) Google Glasses; Year 2014 [255]

(e) Magic Leap; Year 2018 [153]

(f) HoloLens 2 by Microsoft; Year 2019 [16]

(g) Mojo Smart Contact Lens; Year 2022 [21]

(h) Meta Orion AR Glasses, Year 2025 [251]

Figure 1.1: The evolution of Augmented Reality Devices

At the same time, the immersive and sensor-rich nature of AR introduces unprecedented risks to security, privacy, and safety for both users and bystanders. Due to AR devices' always-on sensing needs and ability to alter users' perceptions of the physical world, many of these threats are fundamentally different from known risks of non-immersive technologies (e.g., web and mobile interfaces). As AR technologies rapidly evolve, the computer security community and AR industry have begun to identify key security and privacy challenges in this space from more than a decade ago [233]. Roesner et al. [234] first proposed security threat modeling taxonomies for AR, which included input, data access, and output; Guzman et al. [127] extended these three aspects to include user interaction and device protection. While existing research lays a strong foundation for anticipating, identifying, and mitigating these risks, significant questions remain about how these threats manifest in real-world AR deployments and what new vulnerabilities emerge from the latest AR capabilities.

1.1 Thesis Statement

The primary goal of this thesis is to uncover novel security and privacy risks in the core phases of the AR system data flow, which include input, output, and interaction, as illustrated in Figure 1.2. To achieve this goal, I employ a combination of methods: (1) identifying and analyzing novel security and privacy risks specific to today's AR capabilities; (2) systematically evaluating these threats in current AR platforms; and (3) developing practical mitigation strategies and design guidelines that empower both researchers and developers to build safer AR applications and systems.

With AR technology rapidly evolving and new devices like Meta Ray-Ban and Apple Vision Pro entering mainstream markets, I conducted thorough security evaluations on the state-of-the-art AR hardware around these three key phases across current AR platforms. For each phase, I identified concrete examples of threats that are no longer theoretical but actively exploitable in today's AR ecosystem. Below, I organize the identified risks according to the data flow pipeline, highlighting how each phase introduces distinct security and privacy risks.

Risk with AR Sensor Input While the powerful suite of sensors on modern AR devices is necessary for enabling immersive experience, they need to collect a substantial amount of data about user data, often encompassing personal or sensitive information. This data, especially when combined with other types, has the potential to reveal intimate details. For example, the outward-facing camera create potential privacy threats to the bystanders (i.e., those surrounding the device during its use), capturing them without their awareness or

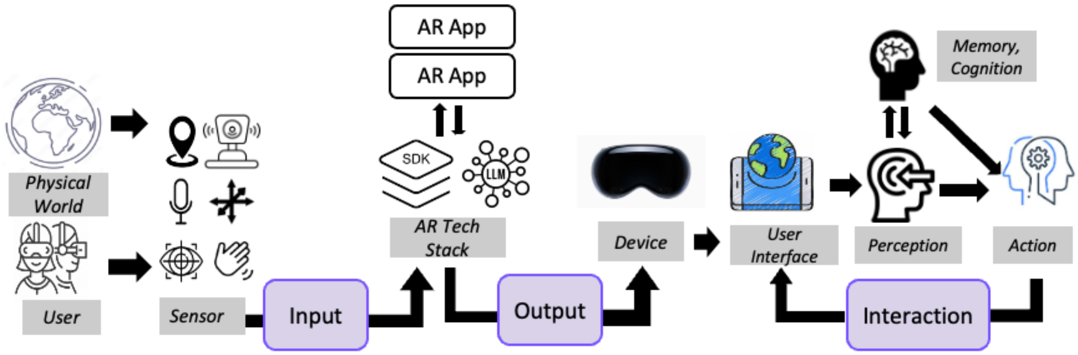


Figure 1.2: Conceptualized Data Flow Diagram for AR Systems

consent. The spatial sensing data can be used to refer to activity happening in the physical world [135], eye-tracking data can be used to reveal sensitive user attributes [111, 174], and body motion data can be used to infer the sensitive information that the user typed [260]. Existing consumer-facing AR headsets, such as Microsoft's HoloLens 2, Meta's Oculus Quest Pro, and Apple's Vision Pro, are already equipped with advanced sensors to perform eye-tracking and hand-tracking. In Chapter 3, we investigate the current eye-tracking and hand-tracking permission models of these platforms, and the extent to which users feel comfortable and informed about these sensors. Based on (mis)alignments we identify between comfort, perceived and actual comprehension, and decision factors, we discuss how future AR platforms can better communicate existing privacy protections, improve privacy-preserving designs, or better communicate risks.

Risk with AR Output After processing sensory input data, AR systems generate output that seamlessly integrates into the user's environment. This output is rendered through a sophisticated pipeline involving spatial computing SDKs, computer vision algorithms, rendering engines, and AI capabilities such as large language models (LLMs) that contextually generate content. However, unlike traditional computing interfaces, the three-dimensional nature of AR output immerses the user *inside* a three-dimensional user interface — including, in some contexts, the real world itself — in contrast to the user merely interacting it from the outside. This introduces new UI security threats, including 3D UI vulnerabilities in AR that lead to attacks such as clickjacking or denial-of-service [117]. Here, we consider a threat model where multiple entities might be interacting within the AR UI, such as third-party embedded code (e.g., a library) running inside an AR application in which the embedded code (e.g., the library) seeks to compromise a property of the AR application or vice versa. In Chapter 4, we formalize these 3D UI security properties and demonstrate their implications through proof-of-concept attacks across major AR platforms.

Risk with Human-AR Interaction Beyond security vulnerabilities in 3D AR UI, the immersive nature of AR output creates opportunities for attacks that target human perception itself when interacting with AR content. In particular, one class of potential attacks, termed *Perceptual Manipulation Attacks* (PMA) by Tseng et al. [269], aims to manipulate the human multi-sensory perceptions of the physical world to influence users' decision-making and even lead to physical harm through the presented AR output stimuli. As the level of immersion in AR technology continues to increase, the potential for such manipulations becomes even more profound and impactful. In Chapter 5, we created a variety of tasks and PMA, evaluated their effectiveness in controlled studies, and observed how users responded, adapted to, and reasoned about them. We derive recommendations for future investigation and defensive directions based on our findings.

Research Methodology For each class of risks, I address the following primary research questions:

(a) What are the unique security and privacy risks introduced by recent advancements in AR hardware?

(b) How does the immersive, three-dimensional nature of AR content create new security vulnerabilities?

(c) How do end-users perceive, understand, and respond to security and privacy threats in AR environments?

(d) What different design decisions have been implemented by today's AR platforms, and what are their security implications?

To investigate these questions, I employ a diverse set of research methodologies, including program analysis, software testing, user study, and online survey. Together, this thesis presents both empirical findings and theoretical frameworks that advance our understanding of security and privacy challenges in AR to inform more secure design practices for future AR systems.

1.2 Thesis Overview

This thesis presents the required background and closely related work in Chapter 2. Chapter 3 focus on permission models for eye-tracking and hand-tracking input for three major platforms (HoloLens 2, Quest Pro, and Vision Pro). We understand difference in platforms' privacy permission flows and investigate how well do people comprehend the permissions, their capabilities, and the associated privacy risk. Chapter 4 presented the empirical analysis of five current AR platforms, systematically investigating how they handle security related properties with 3D AR output. We found that these current AR platforms, including Apple's ARKit, Google's ARCore, Meta's Oculus, Microsoft's HoloLens, and WebXR, are all designed and implemented in ways that enable our attacks to succeed. Chapter 5 presents our experimental findings of the spectrum of end-user reactions, perceptions, and defensive strategies when interacting with AR-based perceptual manipulation attacks (PMA). Finally, Chapter 6 concludes and provides some future directions.

Chapter 2

Background and Related Work

This chapter begins with background context of Augmented Reality (AR), including its definition, technological components, and key application areas. I then review prior work on AR security and privacy, followed by research contributions in adjacent domains that relate to this dissertation.

2.1 Definition of Augmented Reality (AR)

In the early 90s, Milgram et al. observed that although the term “Augmented Reality” had begun to appear more frequently in academic and technical literature, its usage lacked a clear, consistent definition [205]. They developed the Reality-Virtuality Continuum [204], as shown in Figure 2.1, to facilitate a better understanding of AR, Mixed Reality (MR), and Virtual Reality (VR) and how these concepts interconnect. This continuum has two extremes: on one end, a real environment, and on the other, a fully virtual environment. The key distinguishing characteristic of AR is that it preserves the user’s connection to their physical surroundings while enhancing it with contextually relevant digital information. For this dissertation, I define “AR” as technologies that place virtual content in a user’s view of a real-world environment, whether embedded in it or overlaid on it. This virtual content can be multi-sensory, including but not limited to visual elements (3D models, text, images), auditory cues (spatial audio, sound effects), haptic feedback (vibrations, force feedback), or combinations of them. Other researchers and industry practitioners may use alternative terms to refer to the same or related concepts, including MR and extended reality (XR).

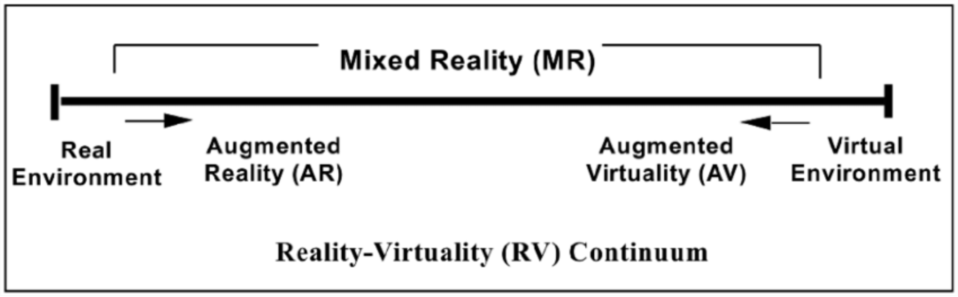


Figure 2.1: Milgram and Kishino on the Reality-Virtuality Continuum

2.1.1 AR Platforms and SDKs

Modern AR application development relies on a complex ecosystem of hardware and software platforms. AR SDKs (Software Development Kits) provide essential tools for sensing, spatial mapping, rendering, and interaction. They abstract the complexities of computer vision, sensor fusion, and 3D graphics into developer-friendly APIs. Examples include Apple's ARKit [8], Google's ARCore [4], Meta's Oculus Integration [24], Microsoft's MRTK [46] for HoloLens, and WebXR [44] for the web. These platforms cover all three AR hardware form factors currently available: handheld mobile devices, video passthrough AR headsets, and optical see-through AR headsets. Not surprisingly, they also differ in their implementation choices, from rendering pipelines to input modalities. Some SDKs include native rendering engines (e.g., ARCore and ARKit), while others integrate with third-party engines like Unity [41] or Unreal Engine [42]. WebXR, as a browser-based standard for AR experience, leverages libraries like Three.js [37] and Babylon.js [9].

2.1.2 AR User Interface

Unlike traditional 2D contexts where users interact with user interface (UI) content on flat screens, the AR UI is a conjunction of visual elements from the physical world, the AR virtual world, and the user's interactions within the immersive 3D world. The AR context requires that the system process and understand the physical world surrounding the user [6, 7, 15, 22, 130], and virtual content is often "anchored" to physical-world surfaces, requiring rapid updates in response to change's in the user's physical surroundings or position.

Figure 2.2 illustrates the AR UI processing pipeline, showing how platform-specific SDKs integrate these three elements. Environmental sensors (cameras, depth sensors, IMUs) capture physical world data, which is processed through SDKs to create environmental understanding. Simultaneously, user input is captured through various modalities and interpreted within the spatial context. These inputs are then processed by the platform's rendering engine to generate the final composite UI that is presented to the user through AR displays.

2.1.3 Current Uses of AR

Augmented Reality has evolved from experimental prototypes to practical applications across a wide range of domains. Each field leverages AR's unique capabilities to create new possibilities for interaction, learning, and productivity. The following paragraphs highlight several key application areas where AR has demonstrated significant impact or shown promising potential. Although these examples are not exhaustive, they demonstrate the breadth and variety of AR deployment.

Education AR has transformed educational practices by providing immersive, interactive learning experiences across disciplines. In medical education, platforms like Microsoft HoloLens enable students to visualize and manipulate 3D anatomical models to improve their overall spatial understanding and learning experience [208]. For STEM subjects, AR-Math [167] demonstrates how everyday objects can be transformed into an interactive mathematical learning experience, helping children explore arithmetic and geometry concepts through lively and contextualized experiences.

Accessibility AR technologies can provide real-time object recognition, scene understanding, and semantic segmentation to enhance situational awareness for users with accessibility needs. CookAR augments kitchen tools with visual feedback to support people with low vision in cooking environments [185]. RASSAR leverages LiDAR, camera data, machine learning, and AR to semi-automatically identify, categorize, and localize indoor accessibility and safety issues [259]. SoundShift explores how spatialized audio can enhance environmental awareness for people with visual impairments [112].

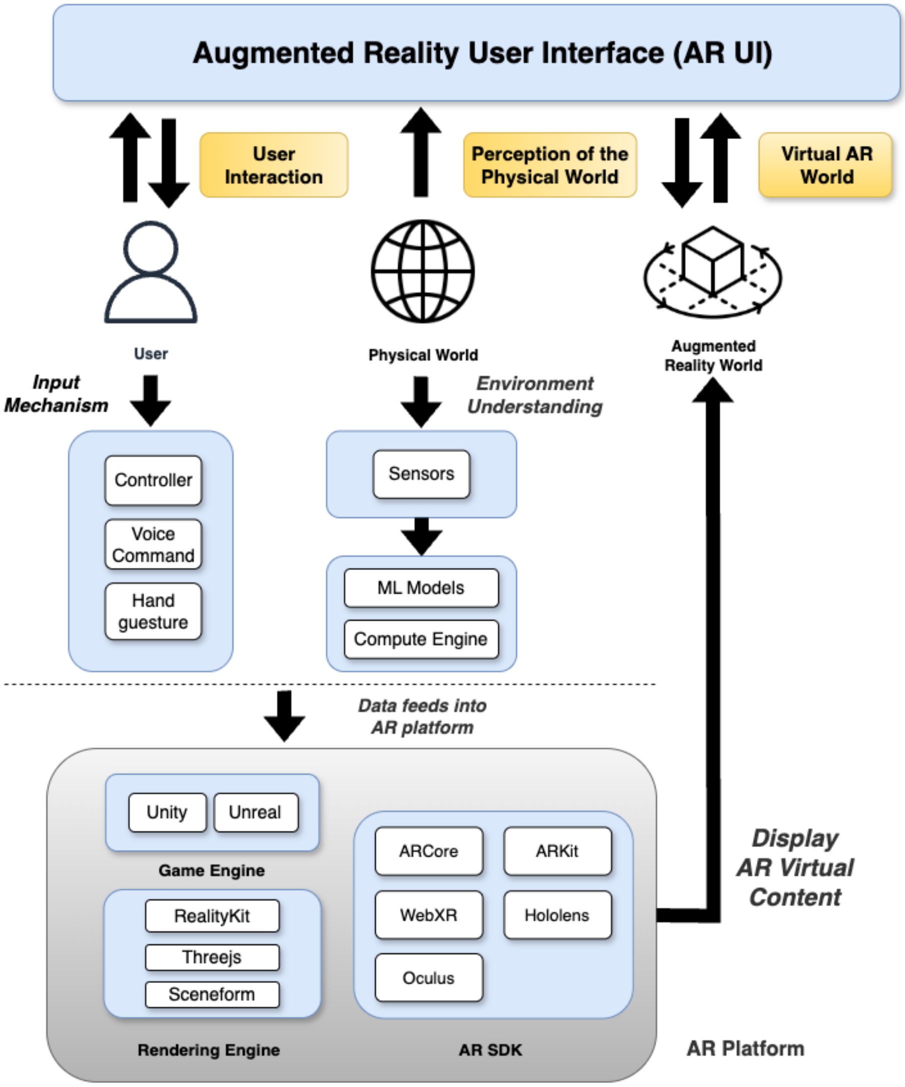


Figure 2.2: Overview of Augmented Reality User Interface. The yellow box represents the three main components: perception of the physical world, virtual AR world, and user interaction.

Cultural Preservation and Tourism AR has been increasingly used to preserve cultural heritage and enhance tourism experiences. At archaeological ruins and museums, AR applications allow visitors to visualize reconstructions of ancient structures, interact with virtual guides, and explore contextual information [294]. Museums like the Kaua'i Museum have developed immersive virtual AR artifacts, allowing visitors to explore Kaua'i's history in a unique and engaging way [48]. Studies show that such AR-enhanced experiences significantly improve visitor engagement and knowledge retention [123].

Entertainment Gaming has long been a driving force in the advancement of AR technologies, with games like *Pokémon GO* demonstrating the potential of location-based, real-world AR experience to engage millions of users globally. Beyond gaming, AR is now shaping new forms of interactive media and live content creation. For instance, Ray-Ban Meta smart glasses enable users to live-stream their first-person perspective in real time, allowing friends and family to see through their eyes and share personal experiences remotely.

2.2 AR Security and Privacy

As AR technologies rapidly evolve, the computer security community and AR industry have begun to identify key security and privacy challenges that are posed to end-users. Below, I discuss prior works that have significantly contributed to a deeper understanding of these challenges.

2.2.1 Risk from Perceptual Manipulations

AR's immersive overlays pose unique threats by hijacking or distorting users' sensory and cognitive processes. In this section, we examine AR-specific manipulation attacks and recent work on detecting and mitigating these threats.

Perceptual Manipulations in AR Given its immersive nature, AR can be an even more powerful medium for perceptual manipulation. Previous work has explored different techniques in AR to manipulate various kinds of human perception. Schmidt et al. [241] leverage visual illusions to manipulate the perceived spatial relationships between the user and objects in AR. Nakano et al. [211] developed a generative adversarial network-based AR application that changes the appearance of food in order to manipulate users' gustatory sensations. Punpongsanon et al. [223, 224] investigated how AR visual output can affect human perception of haptic softness and bending stiffness. Researchers have also developed techniques that manipulate users' visual perception to imperceptibly redirect their movement in the physical space [176, 261]. Recently, security researchers started to explore the potential of attacks based on perceptual manipulation. Baldassi et al. [98] considered direct impacts on the human brain, identifying sensory and perceptual risks (e.g., from accidentally or maliciously induced visual adaptations, motion-induced blindness, and photosensitive epilepsy). Casey et al. [110] present several proof-of-concept attacks that manipulate user visual perception to direct their physical movement, collide with real-world objects, and induce motion sickness. Lebeck et al. [178] discuss adversarial visual output that obscures important real-world content (such as traffic signs).

Secure AR Output Management To defend against adversarial visual output, Lebeck et al. [180] enforce output policies by developing a framework to prevent virtual content from obscuring safety-critical physical objects. Their system demonstrated the need for context-aware permissions that adapt based on environmental conditions rather than static grant models. Lee et al. [184] presented AdCube, addressing WebVR ad fraud through practical confinement techniques for third-party immersive content. Their work introduced isolation mechanisms specifically designed for 3D interactive environments, preventing clickjacking, sensory manipulation, and other AR-specific attack vectors. More recently, Xiu et al. [282] leveraged vision-language models (VLMs) to identify adversarial overlay. In evaluations on pre-collected datasets and live AR streams, ViDDAR achieves up to 92.15% % accuracy in obstruction detection.

2.2.2 Privacy Risk from Sensitive Input

AR devices, equipped with advanced sensors, collect a substantial amount of user data, often encompassing personal or sensitive information. In this section, we briefly discuss prior work that examines security and privacy risks from these data and explore the various mitigation strategies proposed to safeguard against these risks.

Eye-tracking Eye-tracking streams reveal not only where a user is looking, but also sensitive user attributes, including gender, age, race, geographic origin, and a wide array of personal characteristics and preferences [174, 194, 277]. Moreover, fine-grained fixations and gaze patterns can be leveraged for targeted marketing based on a user's estimated interest level [111, 284]. A line of work focuses on building privacy-preserving eye-tracking data. For example, Li et al. [191] applied a formal privacy guarantee on raw eye gazes through differential privacy. David-John et al. [125, 126] explored various mechanisms, such as temporal downsampling and spatial downsampling to mitigate potential privacy risks from eye-tracking data.

Motion Data Here we defined motion data as the user's hand-tracking data and body movement data. Recent studies showed users can be profiled and deanonymized based on their hand-tracking data. For example, Pfeuffer et al. [221] demonstrated the feasibility of identifying users from body motion. Nair et al. [210] performed a large-scale identification on over 50,000 users by using head and hand movement data. They later proposed a new technique named MetaGuard++ [20] that leveraged deep motion masking to anonymize MR motion data.

Spatial Data Guzman et al. [149] first identified the security implications of 3D spatial maps accessed by mobile MR applications. They performed spatial inference attacks over various 3D spatial data captured to infer object semantic classes using HoloLens and ARCore. Nama et al. [212] later designed a privacy framework for regenerating 3D point cloud data to defend against spatial inference attacks. Recently, Farrukh et al. [135] exploited the spatial data collected from an iPad to infer a user's indoor location type. They propose techniques like noise injection or restricting access to raw 3D spatial maps as countermeasures.

Bystander Privacy Unlike handheld cameras, AR headsets operate continuously, making it unclear to bystanders when they are being recorded. Denning et al. [128] found that bystanders of AR headset users are concerned about being identified, highlighting the need to grant permission before being included in the AR recording. O'Hagan et al. [216] conducted an online survey to examine bystanders' privacy preferences and comfort with various AR functionalities on hypothesized AR applications. This "stealth recording" risk is still understudied, but platform vendors have begun exploring hardware indicators (e.g., LEDs or audio prompts) to signal active recording. Corbett et al. [122] proposed BystandAR to protect bystander visual (camera and depth) data in real-time with only on-device processing.

Permission Control Mechanisms To give users finer-grained control over sensitive AR inputs, several permission frameworks have been proposed. In [159], Jana et al. presented a new OS abstraction named recognizer to process raw sensor data and expose high-level objects to AR applications, providing a least-privilege approach for sensor input. Roesner et al. [235] build upon recognizer to specify permission policy on real-world objects. However, these frameworks can be too restrictive in scenarios where rich visual data is desired for functionality reasons. LensCap [156] by Hu el al. separates visual process, network process, and storage process to give users more control over application usage. Erebus [170] by Kim et al. is a domain-specific language (DSL) based access control framework that allow users to define granular access control policies for sensitive functions like object detection, location detection, and image detection.

2.2.3 Hardware as the Attack Surface

AR systems rely on complex hardware components including sensors, displays, processors, and network, which each presenting potential attack vectors. Recent research has highlighted how these hardware components can be exploited to compromise user security, privacy, and even physical well-being.

GPU and Graphics Exploits The graphical processing pipeline in AR systems presents multiple attack surfaces. Odeleye et al. [215] identified GPU-based attacks that deliberately overload rendering resources, leading to dropped and missed frames that induce cyber sickness for users. Taking a different approach, Wang et al. [275] exploited 3D avatar rendering in the Apple Vision Pro to track gaze patterns and reconstruct keyboard inputs, demonstrating how realistic avatar intended for social interaction and remote collaboration can be used for sensitive input inference.

Charging Cable Vulnerabilities The charging cable for AR devices create unexpected side channels. By observing minute fluctuations in the cable's power and audio lines, Li et al. [190] demonstrated the feasibility of inferring visual activities and audio output from connected AR devices. This allows attackers to gather information about user activity without direct access to the device or its primary communication channels.

Sensor-Based Attacks The rich sensor in AR systems create multiple surfaces for privacy attacks. Zhang et al. [291] revealed how high-level performance counters (e.g., CPU/GPU frame rate, frame time, thread times, draw calls, vertex counts) can leak information about user activity, as CPU frame rates plummet during complex spatial or hand-tracking tasks and GPU frame rates shift under function calls for key application. In related work, Zhang et al. [290] presented FaceReader, which reconstructs high-quality vital sign signals (breathing and heartbeat patterns) based on motion sensor data. Motion sensors have also been exploited to recover keyboard inputs: Meteriz-Yıldıran et al. [203] leveraged hand-tacking data to recover typed content, achieving accuracy between 40% and 87% within the top-500 guesses. Slocum et al. [249] improved this attack by applying machine-learning models to high-resolution head-pose time series, achieving over 90% accuracy in recovering passwords under controlled conditions.

Network Attacks While previous work demonstrates that users' motion and behavior can leak sensitive input, Su et al. [260] consider a threat model where they were able to reverse engineer network packets from popular multi-user VR applications to infer the typed content. Through user studies in Rec Room and other popular VR apps, their attack achieves up to 98 percent top-1 accuracy with inferred typed content.

2.3 Miscellaneous and Orthogonal Work

Cognitive Vulnerabilities to External Stimuli Prior work in non-AR contexts already showed how human cognition and perception can be distracted or manipulated by exogenous cues (i.e., external stimuli). For example, a line of research [136, 155, 177, 265] studied how visual reaction time is sensitive to visual stimuli. Yantis and Jonides [287] showed that an object with sudden onset was always processed first. Neyens and Boyle [213] suggested that cell phone usage while driving is associated with cognitive, auditory, and visual distractions, causing a high likelihood of vehicle accidents. Simons and Chabris conducted the famous *attentional blindness experiment* in 1999 [247]. When asked to perform a task that required full attentional resources, subjects often failed to see a gorilla in the midst of the experiment. Our study design in Chapter 5 were built on top of these studies from classic psychology and cognitive science literature.

Empirical Analysis of AR Platforms As AR platforms continue to emerge and develop, recent work has compared and evaluated AR platforms in various performance and functionality criteria. Scargill et al. [240] investigate AR object placement stability in mobile AR platforms. Slocum et al. [248] measure the spatial inconsistency when placing virtual objects in the real world on ARCore, while Lee et al. [183, 230] analyze the AR object placement deviation on WebXR. Other works have proposed functionality metrics, such as general performance (CPU/memory use) [214] body movement and marker-based tracking [93, 270], accessibility and ease-of-use [267], lighting estimation [218], and plane and feature point detection [218] that allow direct comparison across multiple AR platforms. Our work in Chapter 3 and Chapter 4 builds upon these evaluation approaches but focuses specifically on empirically evaluating the security and privacy across these AR platforms.

2D UI Security UI security in 2D has been well-studied. For example, early work in this space included secure windowing systems like Trusted X [133] and EROS [242]. More recently, a line of work considered UI security requirements and threats on Android and iOS [94, 103, 116, 187, 232]. Luo et al. [197] provide a thorough analysis of UI vulnerabilities in mobile browsers. As mentioned, there is also significant prior work mitigating clickjacking attacks on the web and in other contexts (e.g., [157]). Our work in Chapter 4 takes the next step in the broader space of UI-level security, studying emerging AR platforms

Permission Perception Users rely on dialogs in the permission-granting process to learn about the potential utility and privacy risks associated with certain permissions, all of which allow users to make informed decisions. Many previous works aim to understand what concerns users have when granting permissions [102, 137–140, 209], and how to better design the permission/warning dialog to increase transparency for the users [141, 278]. Prior work assessing the efficacy of permission systems has used comprehension to determine the extent to which users are informed about the permissions being requested. Felt et al. [140] first studied the effectiveness of Android install-time permission, and Shen et al. [243] investigated users' comprehension of the runtime permission model on iOS and Android. Harborth et al. [151] evaluated user comprehension of permissions requested in mobile AR applications.

Their results suggested that users are concerned with current permissions in AR, such as speech and face recognition, yet the mobile system did not request permission to collect such data. As discussed in Chapter 3, my collaborators and I examine how users perceive emerging AR-specific permissions for eye-tracking and hand-tracking. We identify opportunities for AR platforms to effectively communicate about utility and privacy through permission UI flows.

Chapter 3

Input: Eye-Tracking and Hand-Tracking Permission Design

This chapter considers input, the first phase of the AR system data flow highlighted in Chapter 1. It explores challenges within the permission design of novel input modalities that are equipped by current AR devices, such as eye-tracking and hand-tracking sensors. More specifically, this chapter investigates the current technical landscape of these new sensing permissions and how end-users perceive and understand associated privacy and utility implications from the permission-granting flow. While the literature on permission design for mobile or web platforms with sensory input data is rich, there have been no empirical studies on permission design in the context of Augmented Reality headsets. In pursuit of this objective, we conducted an online survey with 280 participants to investigate user comfort, comprehension, and willingness regarding eye-tracking and hand-tracking permissions designs for three major AR platforms (HoloLens 2, Quest Pro, and Vision Pro). Based on the results, we discuss how current AR platforms can better communicate existing privacy protections, enhance privacy-preserving designs, or more effectively communicate risks. The work in this chapter, User Comprehension and Comfort with Eye-Tracking and Hand-Tracking Permissions in Augmented Reality, was first published and presented at the 15th Usable Security and Privacy Symposium (USEC) in 2025 [118].

3.1 Introduction

Augmented reality (AR) technologies have reached the cusp of commercial viability, transforming how we interact with the real world, the digital world, and ourselves. Unlike traditional 2D contexts where users interact with content on flat screens, extensive research from industry and academia aims to reinvent how users naturally and smoothly interact with the virtual 3D world. Eye-tracking [50, 70, 106, 172] and hand-tracking [67, 68, 124, 199] are integral to this evolution, enhancing user immersiveness [97, 106] and bringing different yet more intuitive and natural input modalities.

Existing consumer-facing AR headsets, such as Microsoft's HoloLens 2, Meta's Oculus Quest Pro, and Apple's Vision Pro, are already equipped with advanced sensors to perform eye-tracking and hand-tracking. As core input mechanisms for AR systems, these sensors enable exciting functionalities, such as navigating and interacting with the virtual space using eye gaze [50, 56] and hand gestures [51, 59, 66], or system performance optimizations [61].

Despite the potential benefit these new features bring, existing research has highlighted privacy concerns associated with both eye-tracking and hand-tracking sensors. Because these sensors continuously monitor and record streams of fine-grained biometric data, they introduce unprecedented risks related to the collection, use, and potential misuse of user input. For instance, the data captured by these devices could be used for inferring sensitive user attributes [174, 194, 277], predicting interest level [111, 284], and revealing user identity [193, 210, 221].

Depending on the system design, AR systems or applications may access the data from these sensors by asking users for permission, or access may be passively enabled by default. End users may grant or deny permission requests based on their expectations of the utility-privacy tradeoff. If users consent to these sensors without fully understanding the associated risks, they may unintentionally expose themselves to privacy violations and security threats [198, 246]. On the other hand, clear communication of the data collection and privacy techniques can effectively increase users' willingness to adopt new technologies [281]. While the literature on mobile or web platforms is rich, to our knowledge, there have been no empirical studies on permission-granting in the space of Augmented Reality headsets.

Given that eye-tracking and hand-tracking serve as primary input channels in today's AR systems, understanding how these input permissions are managed becomes critical for the broader AR ecosystem. Thus, our first foundational research question is focused on comprehensively assessing how permission management works on exemplar examples of modern

AR technologies:

- **RQ1: Current Landscape.** What is the current technical landscape for eye-tracking and hand-tracking permissions in AR platforms?

For this work, we focus on three leading examples of AR technologies: the Microsoft HoloLens 2, the Meta Oculus Quest Pro, and the Apple Vision Pro. We base our analysis on experimentation with real devices and publicly-available information. Informed by our findings to RQ1, we next explore the answers to the following two research questions. At a high level, these research questions ask: how do users feel after being presented with the permission dialogs from the HoloLens 2, the Oculus, and the Vision Pro (e.g., how do they feel about their privacy) (RQ2), and do they understand what it means to grant a permission on these devices (e.g., what are the privacy implications of granting permission) (RQ3)?

More precisely, our next two research questions are:

- **RQ2: User Perceptions.** How do people perceive different platforms' privacy permission flows for eye-tracking and hand-tracking in AR? We explore the extent to which people feel comfortable and informed about these permissions.

- **RQ3: User Comprehension.** After viewing the information provided by the permission flow, how well do people comprehend the permissions, their capabilities, and the associated privacy risk?

To answer RQ2 and RQ3, we conducted a survey of 280 participants. In this survey, we showed participants screenshots of the permission-granting interfaces for the HoloLens 2, Oculus, and Vision Pro. We asked participants to what extent they felt comfortable and informed about the permission, confident about the protection of sensitive data, and how clear they found the permission flow to be. We explicitly recruited participants who had no prior experience with AR, in order to capture people's comfort and comprehension on their *first* exposure to these permission-granting flows, rather than relying also on their past experiences.

Among our findings, we observe that: (1) the extent to which participants felt comfortable and informed depended on the device, sensor, and whether they were considering system-level or app-level access (Section 3.4.1). (2) Participants experienced greater difficulty understanding privacy implications compared to utility, and are generally less informed at the app-level compared to the system-level (Section 3.4.2). (3) Participants were largely uninformed about data handling processes, for example, whether the system or application shares their data with external servers, has access to the raw data, or accesses their data in the background (Section 3.4.3).

Additionally, we investigate what factors participants report would contribute to their willingness to try eye- and hand-tacking enabled AR technologies (RQ4, Section 3.4.4). For example, how do participants weigh the importance of understanding who has access to their data or why these data are being collected?

- **RQ4: Factors that Impact User Decisions.** What permission-related factors do people report as important in their decision-making process around whether or not to try eye- and hand-tacking enabled AR technologies in the future?

Stepping back, we then compare the results between perception (RQ2), comprehension (RQ3), and self-reported decision factors (RQ4) to identify (mis)alignments (Section 3.5). For example, we identify cases where comfort may be found in part in a misunderstanding of the actual implications or implementation of a permission, meaning that people may believe a permission is more or less privacy-invasive than it actually is. We discuss how future AR systems could improve the permission-granting flow for eye-tracking and hand-tracking input while better communicating privacy implications, and/or implementing privacy protections currently lacking.

Disclosure We have reported all of our findings to Apple, Meta, and Microsoft.

3.2 Current AR Permission Granting Landscape

3.2.1 Methodology

To understand the current landscape of eye-tracking and hand-tracking permissions in today's AR platforms, we investigated three high-profile publicly available platforms: HoloLens

2 (from Microsoft), Quest Pro (Oculus, from Meta), and Vision Pro (from Apple). Our team conducted multiple rounds of structured brainstorming to generate and refine properties relevant to eye- and hand-tracking permission granting (e.g., whether applications have access to eye-tracking data when running in the background).

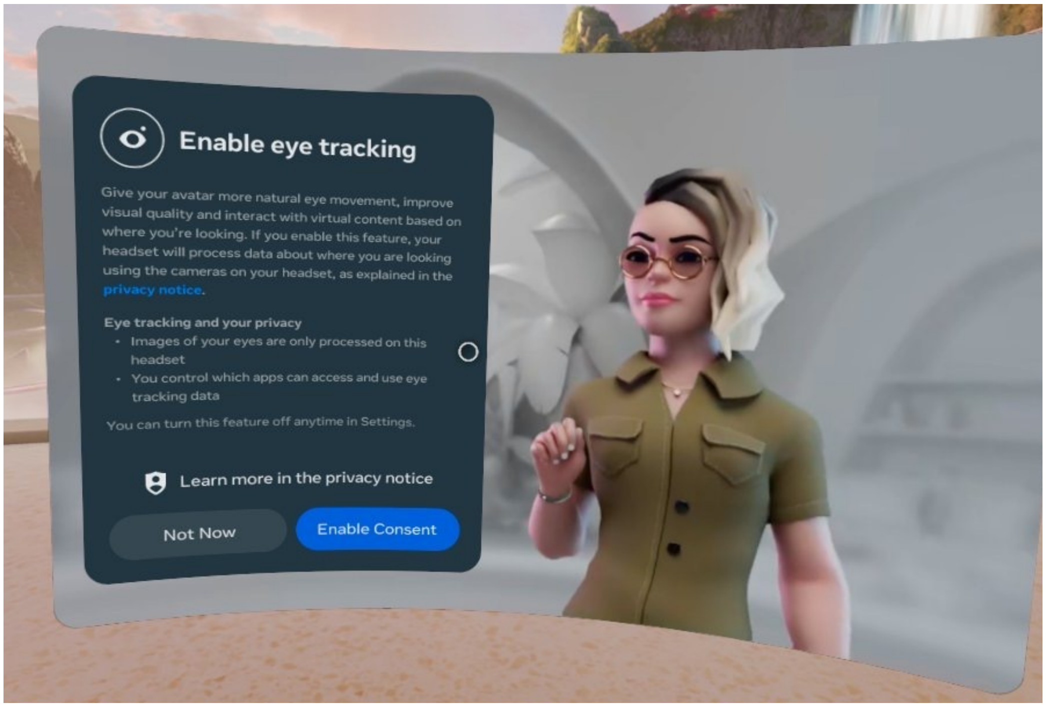
After finalizing the properties, the lead author examined the documentation and the privacy policies, and built applications on each device to evaluate each property. We performed our initial evaluation in October 2023 and verified them on the up-to-date AR operating system (Holographics version 24H1, Oculus Quest version 65, and visionOS version 1.1) in May 2024. All authors iteratively validated the findings and resolved disagreements.

We highlighted that our findings are based on snapshots of the ever-changing AR permission ecosystem, and the results might be subject to change in future upgrades. For example, we noticed several changes in the permission UI for Oculus hand-tracking and eye-tracking privacy notice, though these changes didn't affect system capability. Nevertheless, our findings can serve as a benchmark to evaluate how the permission landscape evolves. We summarized the selected properties in the "Permission Comprehension" column in Appendix A.5, highlighting our findings for each AR platform using an underline. The complete list of reasoning and supporting references is available in Appendix A.2.

3.2.2 Eye-Tracking Permission

Permission Request We find that only Oculus requests the user's permission to perform eye tracking on a system level, as shown in Figure 3.1. The permission dialog from the system illustrates the potential utility of eye-tracking and the privacy-preserving techniques Oculus deploys. In contrast, eye-tracking capability is enabled by default for HoloLens or Vision Pro on the system level, given it's one of the primary input modalities (as opposed to controllers for Oculus). Developers could request eye-tracking permission on Oculus and HoloLens as shown in Figure 3.2, but not on Vision Pro.

Data Granularity All three platforms prevent applications from accessing raw eye-tracking images due to significant privacy concerns. For Oculus and HoloLens, the provided eye-tracking APIs [50, 60] include abstracted eye-tracking data, comprising a stream of gaze vectors to represent the user's eye orientation and movement patterns [63, 70]. However,



Enable eye tracking

Give your avatar more natural eye movement, improve visual quality and interact with virtual content based on where you're looking. If you enable this feature, your headset will process data about where you are looking using the cameras on your headset, as explained in the privacy notice.

Eye tracking and your privacy

- Images of your eyes are only processed on this headset
- You control which apps can access and use eye tracking data

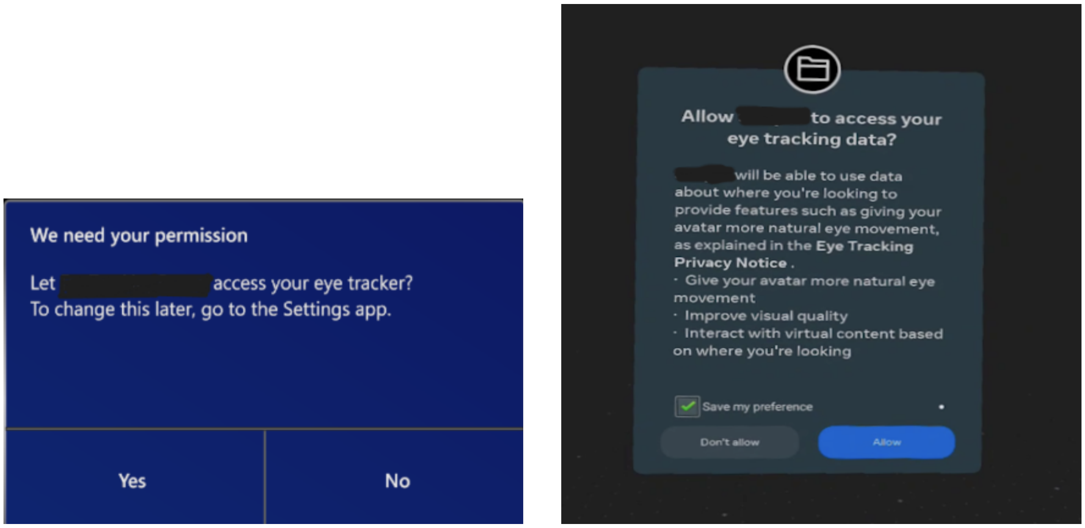
You can turn this feature off anytime in Settings.

Figure 3.1: Oculus: System-level eye-tracking permission.

neither platform controls how third-party entities use, store, or share users' abstracted gaze data [63].

Compared with Oculus and HoloLens, Vision Pro employs a different, arguably more privacy-preserving, data collection model. According to their Privacy Overview report [83], Apple acknowledges that (abstracted) eye-tracking data, including the content the user looked at or the duration they looked at it, could potentially reveal a user's thought processes. As a result, while Vision Pro enables eye-tracking permission by default, the processed eye-tracking data is not available to Apple, third-party entities, or websites. Instead, developers utilize Apple's native event-handling mechanisms, such as UIKit [89] or SwiftUI [88], to manage user interactions automatically. As users navigate applications, visionOS processes and renders visual effects that respond to where they look on the device.

Data Transmission While Oculus is the only platform that requests permission to enable eye-tracking on a system level, we also find that it is the only platform to collect and retain



We need your permission

Let [redacted] access your eye tracker?
To change this later, go to the Settings app.

Allow [redacted] to access your eye tracking data?

[redacted] will be able to use data about where you're looking to provide features such as giving your avatar more natural eye movement, as explained in the **Eye Tracking Privacy Notice** .

- Give your avatar more natural eye movement
- Improve visual quality
- Interact with virtual content based on where you're looking

☑ Save my preference

(a) HoloLens 2: App-level permission (app name blurred for anonymity)

(b) Oculus: App-level permission (app name blurred for anonymity)

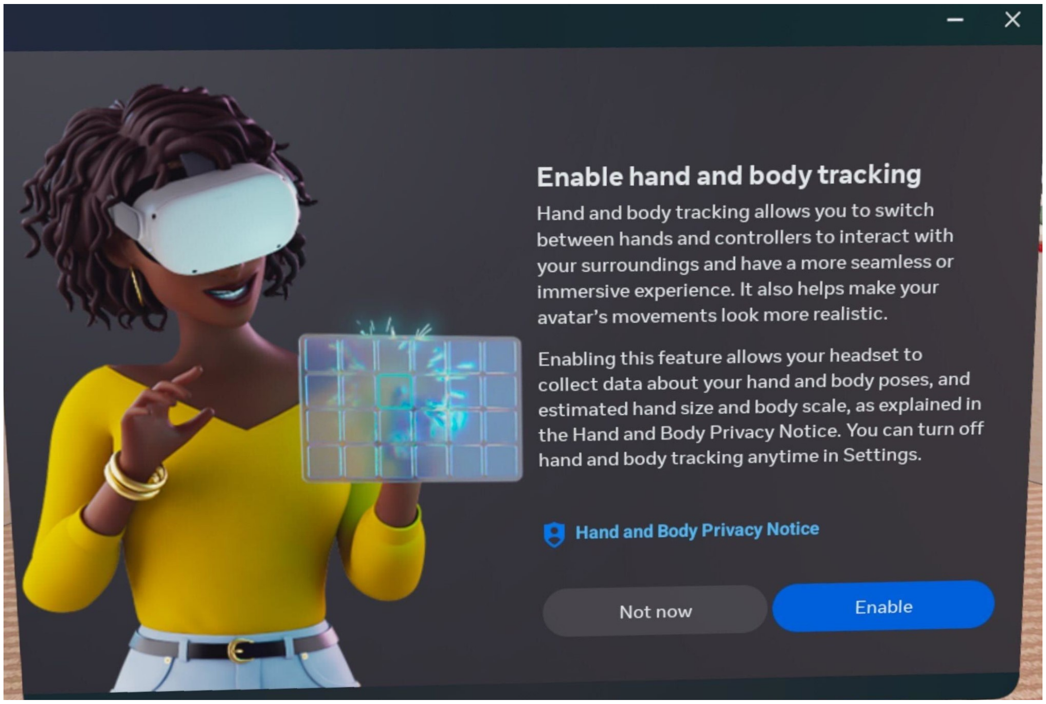
Figure 3.2: App-level Eye-tracking dialogs

user's eye-tracking data. Specifically, Oculus stored the abstracted gaze data and users' interactions with eye tracking in their company server. As stated in their privacy policy [63], the eye-tracking data will be associated with users' accounts until Meta "no longer need it to provide the service or improve the eye-tracking feature".

3.2.3 Hand-Tracking Permission

Permission Request Similar to eye-tracking, only Oculus requests the user's permission to perform hand-tracking on a system level, as shown in Figure 3.3. The permission dialog illustrates the potential utility and provides a reference link to the privacy policy. Vision Pro is the only platform that requests app-level permission for hand-tracking, as shown in Figure 3.4b, whereas the other two platforms automatically grant applications access to the hand-tracking API. The only platform that supports background access for hand-tracking is HoloLens, as shown in Figure 3.4a.

Data Granularity All platforms provide an abstract representation of the user's hand-tracking through hand skeleton data. With the underlying recognition model, the system can



Enable hand and body tracking

Hand and body tracking allows you to switch between hands and controllers to interact with your surroundings and have a more seamless or immersive experience. It also helps make your avatar's movements look more realistic.

Enabling this feature allows your headset to collect data about your hand and body poses, and estimated hand size and body scale, as explained in the Hand and Body Privacy Notice. You can turn off hand and body tracking anytime in Settings.

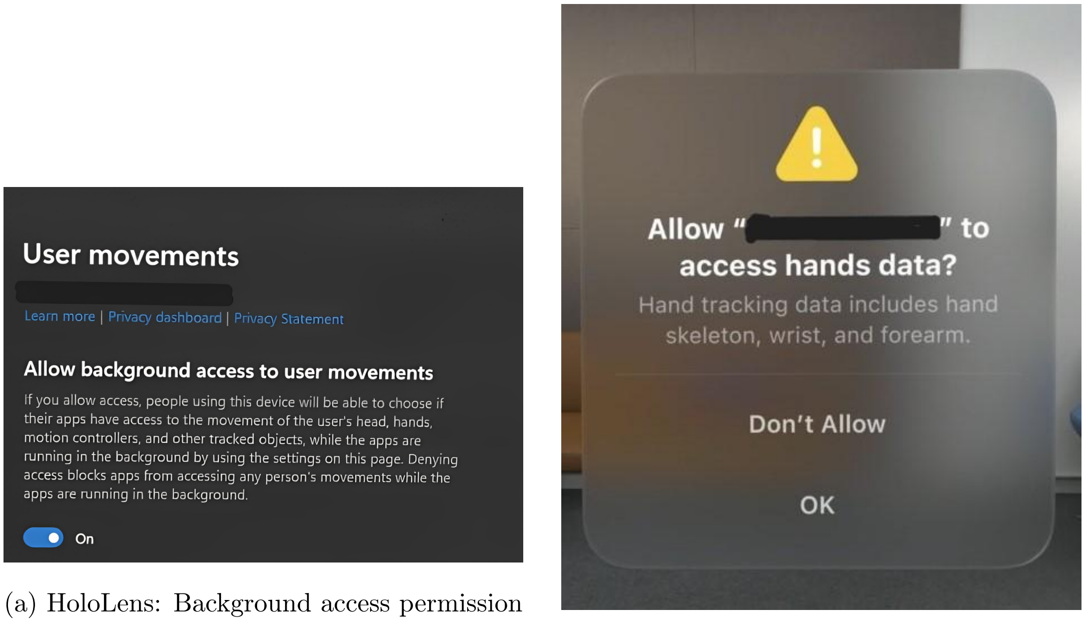
Figure 3.3: Oculus: System-level hand-tracking permission.

understand users' gestures, hand position, relative hand size, and hand movement. The only difference is that the developers can get access to the user's hand-tracking data without an additional prompt on HoloLens and Oculus (if the user already granted it to the system). For Vision Pro, the hand-tracking data is only available to the developer when the application is in an immersive space [68].

Data Transmission While Oculus is the only platform that requests permission to enable hand-tracking at the system level, it also processes and shares the hand-tracking data with the Oculus server, where it is retained for 90 days [64]. For HoloLens, the hand-tracking data is processed on the device and is not stored [86] and for Vision Pro, the hand-tracking data is only stored on-device [83].

3.3 User Study Methodology

To answer RQ2-RQ4, we designed and ran a user study.



User movements

[redacted]

Learn more | Privacy dashboard | Privacy Statement

Allow background access to user movements

If you allow access, people using this device will be able to choose if their apps have access to the movement of the user's head, hands, motion controllers, and other tracked objects, while the apps are running in the background by using the settings on this page. Denying access blocks apps from accessing any person's movements while the apps are running in the background.

On

Allow "[redacted]" to access hands data?

Hand tracking data includes hand skeleton, wrist, and forearm.

Don't Allow

OK

(a) HoloLens: Background access permission for applications. (system name blurred for anonymity)

(b) Vision Pro: App-level permission (app name blurred for anonymity)

Figure 3.4: Hand-tracking permission dialogs

3.3.1 Survey Design and Procedure

Inspired by the different permission-granting processes across different sensors and platforms we documented in Section 3.2 (RQ1), we designed a survey to study users' comfort, the extent to which users perceive themselves as informed by the permission granting processes, their comprehension of the permissions, and what factors impact their likelihood of using these devices in the future. This survey, launched online on Prolific in May 2024, assessed perceptions of three AR platforms, with questions designed to answer our research questions of interest. The complete list of survey questions and instructions are available in Appendix A.1.

After consenting to participate, participants read that we were investigating perceptions of augmented reality technologies. Participants saw several image examples of AR headsets, and were asked about their familiarity and experience with AR headsets, both broadly and with the headsets investigated in this study specifically. Participants were excluded from analyses (but still received payment) if they indicated they had used any of the three headsets investigated here. Next, participants read that AR headsets have different sensors recording data while the headsets are in use, and that users typically view permission dialogs prompting them to allow or deny the headset access to these data. Participants were told they would view permission dialogs and rate their impressions for two different sensors. Participants were randomly assigned on a between-subjects basis to evaluate one of three mainstream AR headsets: Meta's Quest Pro, Microsoft's HoloLens 2, or Apple's Vision Pro. The company and device names were anonymized in the survey to avoid biasing evaluations. Participants evaluated the device's eye-tracking and hand-tracking permissions in random order on a within-subjects basis.

For each sensor, participants were asked to imagine they were using an AR/MR headset with the sensor feature. First, participants read that they were navigating the system-level permission settings for a given sensor. In general, this was followed by a real screenshot of the platform's permission dialog, or several dialogs depending on the platform's interface, with all screenshots accompanied by alt text. We also presented other screenshots to simulate the experience of enabling eye- or hand-tracking, such as the hand visualizations that users see when they put on the headset. For platforms that did not explicitly ask for the user's permission for a given sensor, we told participants that the permission was enabled by default. This part of the survey was designed to follow a user's actual permission-granting process within a given platform as closely as possible. See Appendix A.1 for screenshots.

To assess the extent to which people feel comfortable and informed while experiencing the permission flow (RQ2), participants answered several questions about their perceptions of the dialogs and the device more broadly. Participants responded to a series of 5-point Likert scale questions assessing how informed they felt about both the utility of the permission and its associated privacy risks, their confidence that their data will be securely stored, the extent to which they know what data will be collected and how it will be used based on the permission screenshots presented, and how comfortable they felt using the device (see full questions and scales in Appendix A.1).

We then sought to explore whether the interfaces impacted users' actual understanding or misperceptions of the system's capabilities and privacy protections (RQ3). Participants responded to a series of True or False questions about the system's capabilities and privacy (e.g., "The system can identify which real-world objects you are looking at;" "The system can retain the image of your hand on the AR/MR headset"). For each statement, participants indicated whether they believed it was True or False, or indicated "I don't know." Our team conducted multiple rounds of interactive brainstorming and preliminary experiments to generate questions and finalize answers. These questions are inspired by prior studies on mobile permissions (e.g., [140]).

After answering the above questions, participants were then told to imagine they were opening an app on the headset to navigate the app-level permission settings for the device. Here again, participants saw screenshots of permission dialog(s), or received alternative information about the permissions as applicable. Participants responded to the same questions as for the system-level, assessing comfort with the app, how informed they feel, and a similar series of true/false/I don't know questions about the app's capability and privacy protections.

Finally, participants read that we wanted to understand what information about the system and app would help them feel more comfortable using this technology in the future. Participants were shown five factors relevant to permission dialogs (i.e., knowing who will have access to the data, how the data will be stored, how the data will be transmitted, what type of data will be collected, and the purpose of collecting the data). Participants selected their top three most important factors (in no particular order).

Participants answered all questions for a given sensor before evaluating the next sensor. After evaluating both sensors, participants responded to an attention check, reported demographic information, and received payment through Prolific.

3.3.2 Ethics

The study was deemed Exempt by the university's Human Subjects Review Board (IRB). Participants were anonymous and identifying data were removed or not obtained. Participants could leave the survey at any time. Participants were compensated based on Prolific's guidelines (see below).

Gender		Age		Race/Ethnicity	
Man	48.2%	18-24	7.4%	White	87.5%
Women	47.5%	25-34	26.2%	Black or African American	4.6%
Undiscl.	4.3%	35-44	21.4%	Asian	2.5%
		45-54	21.0%	American Indian / Alaskan Native	1.4%
		55-64	14.4%	Native Hawaiian / Pacific Islander	0.4%
		65+	9.6%	Mixed	0.4%
				Undisclosed	3.2%

Table 3.1: Breakdown of participant demographics by gender, age, and race/ethnicity.

3.3.3 Participants

We conducted an a priori power analysis using G*Power to determine how many participants were needed to detect a moderate effect size. This analysis determined that 260 participants would be sufficient to detect an effect size of d = 0.35 at 80% power in an independent-samples t-test. This sample size also provides sufficient power to detect effect sizes of $n_p^2 < .010$ in mixed-model ANOVAs.[1] In actuality, 292 adult U.S. crowdworkers on Prolific completed the 13-minute survey online in exchange for payment, with compensation set based on Prolific's guidelines ($12 hourly rate). We excluded participants from analyses who failed to pass an attention check and who indicated they had used either Oculus, HoloLens, or Vision Pro. Participants excluded from analyses still received payment. After exclusions, our analyses includes 280 participants. Participants' demographics are included in Table 3.1.

3.3.4 Limitations

We consider several limitations of our study's design. First, a survey with screenshots may not fully capture the complete experiences of a user wearing an AR headset. Beyond the different modality, there may also be additional information in the device's initial setup flow, such as a 3D video, that helps communicate permission-related impressions to users that are not captured by our survey design. Similarly, app developers can customize the permission dialog text on Vision Pro or provide justifications before the dialog on Oculus and HoloLens,

[1] These data do not meet all normality assumptions for ANOVAs. However, prior work shows that ANOVAs are robust against non-normality when the sphericity assumption is met, as it is in our data [105].

meaning that the information shown in the app-level dialog may depend heavily on that customization in practice. Second, our attempt to anonymize company and device names in the survey may not have always been successful. Since certain UI characteristics are manufacturer-specific, they may have been recognizable to some participants, influencing their perceptions. Third, our analysis of participant comprehension depends on our own understanding of the correct answers to the true/false questions (see Appendix A.2 for our understanding). Nevertheless, we believe it is valuable to understand what participants believe the answers are based on the permission dialogs they see as this understanding will influence user perception and decisions. Lastly, permission designs are subject to change as platforms evolve and update their SDKs. The observations and analyses presented here are based on our understanding of the systems in May 2024. Despite these limitations, our study sheds light on people's perception and comprehension of novel AR platform permissions and evaluates key aspects of the current designs of these platforms' permission models and dialogs. Future work must continue to revisit these questions as the technology and app ecosystems evolve, just as a decade or more of research studied the smartphone permission and app ecosystem.

3.4 Survey Results

We investigated perceptions of eye-tracking and hand-tracking based on the permission flow (RQ2), comprehension of utility and privacy implications (RQ3), and information deemed particularly important to include in the permission dialog (RQ4).

3.4.1 RQ2: Perceptions of Permission Flows Differ Across Devices, Sensors, and Use Level

We investigated the extent to which participants felt comfortable and informed using the AR headset. In the sections below, we explore how participants' perceptions depended on the device and sensor type. Thus, we conduct a series of mixed-methods ANOVAs and t-tests on each dependent variable. We focus on system-level perceptions to avoid inflating Type I errors with additional comparisons at the app-level.

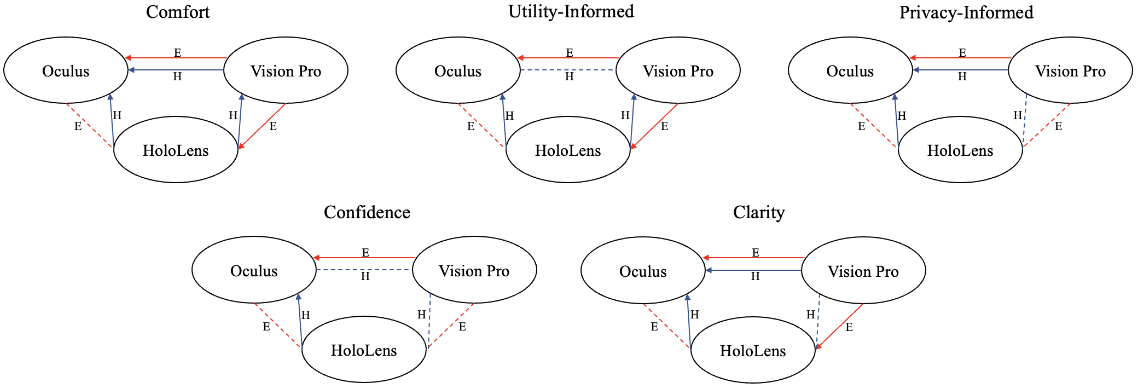


Figure 3.5: System-level perceptions compared across devices. Red lines (labeled "E") represent eye-tracking and blue lines (labeled "H") represent hand-tracking. Arrows point to the device that was rated significantly higher on the item. Dashed lines = non-significant.

Comfort

We conducted a mixed-method ANOVA on participants' comfort level with sensor type (eye-tracking, hand-tracking) as a within-subjects variable and device (Oculus, HoloLens, Vision Pro) as a between-subjects variable. Participants' comfort was impacted by both the device and the sensor type, indicated by a significant interaction between sensor type and device, $F(2, 277) = 16.108$, $p < .001$, $n_p^2 = .104$ (see all system-level comparisons in Figure 3.5).

We conducted t-tests across devices and sensors to decompose this interaction. We first observed differences in comfort across devices. In the context of eye-tracking, participants felt similarly comfortable using Oculus and HoloLens, ($p = .387$, $d = 0.13$), but felt significantly more comfortable using both Oculus and HoloLens as compared to Vision Pro ($ps < .009$, $ds > 0.38$). In the context of hand-tracking, participants felt significantly more comfortable using Oculus compared to both HoloLens ($p < .001$, $d = 0.63$) and Vision Pro ($p = .040$, $d = 0.32$). Participants also felt significantly less comfortable using HoloLens than Vision Pro for hand-tracking ($p = .034$, $d = -0.31$).

Differences in comfort between the sensors, on the other hand, emerged only within HoloLens. Participants who saw dialogs from Oculus or Vision Pro were similarly comfortable with eye-tracking and hand-tracking (*ps* > .150, *ds* < .16). But participants who saw HoloLens dialogs felt significantly more comfortable with the eye-tracking sensor than the hand-tracking sensor ($p < .001$, $d = 0.58$).

Feeling Informed about Permission Utility

We next investigated the extent to which participants felt informed about the utility of the permissions. At the system level, there was a significant interaction between device and sensor type, $F(2, 277) = 11.394$, $p < .001$, $n_p^2 = .076$. For eye-tracking, participants felt similarly informed about the utility of Oculus and HoloLens ($p = .694$, $d = 0.06$). However, participants felt significantly more informed about the utility of both Oculus and HoloLens as compared to Vision Pro (*ps* < .040, *ds* > 0.30). For hand-tracking, participants felt significantly more informed about the utility of Oculus than HoloLens ($p < .001$, $d = 0.63$). There was no significant difference between Oculus and Vision Pro ($p = .122$, $d = 0.24$). Participants felt significantly less informed about the utility of HoloLens compared to Vision Pro ($p = .013$, $d = -0.36$). We next compared differences on the system-level in the extent to which people felt informed about the utility across eye-tracking and hand-tracking. For both Oculus and HoloLens, participants felt significantly more informed about the utility of eye-tracking as compared to hand-tracking (*ps* < .030, *ds* > 0.24). This difference was non-significant amongst participants who saw Vision Pro ($p = .320$, $d = 0.11$).

Feeling Informed about Privacy

We next investigated the extent to which participants felt informed about the associated privacy risk of the permissions. Once again, at the system-level, there was a significant interaction between device and sensor type, $F(2, 277) = 4.027$, $p = .019$, $n_p^2 = .028$. In the context of eye-tracking, participants who saw Oculus felt more informed about the privacy risks than participants who saw Vision Pro ($p = .031$, $d = 0.33$), but no other device comparisons were significant (*ps* > .200, *ds* < 0.18). In the context of hand tracking, participants who saw Oculus felt more informed about the privacy risks than participants who saw either HoloLens or Vision Pro (*ps* < .035, *ds* > 0.32), whereas participants in the latter two conditions did not significantly differ (*ps* = .184, $d = -0.19$).

Comparing across sensors at the system-level, we found that participants who saw both

Oculus and HoloLens felt more informed about the privacy risks of eye-tracking than hand-tracking ($ps < .022$, $ds > 0.24$). This difference was non-significant amongst participants who saw Vision Pro ($p = .103$, $d = 0.18$).

Confidence in Security

We investigated how confident participants felt about the system's ability to securely store their data. There was a significant interaction between device and sensor type, $F(2, 277) = 4.888$, $p = .008$, $n_p^2 = .034$. For eye-tracking, participants felt more confident about Oculus than Vision Pro ($p = .048$, $d = 0.30$), and all other comparisons were non-significant ($ps > .110$, $ds < 0.24$). For hand-tracking, participants felt more confident about Oculus than HoloLens ($p = .011$, $d = 0.37$), and all other comparisons were non-significant ($ps > .190$, $ds < 0.20$). Comparing across sensors at the system-level, participants who saw either Oculus or HoloLens felt more confident in the system securely storing their eye-tracking data than their hand-tracking data, ($ps < .015$, $ds > 0.27$). There was no difference in confidence across sensors for participants who saw Vision Pro ($p = .334$, $d = 0.11$).

Data Use Clarity

Finally, we investigated the extent to which participants felt they knew what data would be collected and how it would be used (i.e., data clarity) based on the permission flow. At the system-level, there was a significant interaction between device and sensor type, $F(2, 277) = 10.376$, $p < .001$, $n_p^2 = .070$. In the context of eye-tracking, there was no significant difference in data clarity across Oculus and HoloLens ($p = .817$, $d = 0.03$). However, participants felt more data clarity from both Oculus and HoloLens as compared to Vision Pro ($ps < .022$, $ds > 0.33$). In the context of hand-tracking, participants felt more data clarity from Oculus as compared to both HoloLens and Vision Pro ($ps < .008$, $ds > 0.40$), and participants in the latter two conditions did not significantly differ ($p = .141$, $d = -0.22$).

Comparing across sensors, across all three devices, participants felt more data clarity about the eye-tracking permission than the hand-tracking permission ($ps < .026$, $ds > 0.24$).

Table 3.2: Participant comprehension correctness summary. Hol is HoloLens, Oc is Oculus, Vis is Vision Pro, Avg is the performance on each category, Avg-S is the performance on each sensor, and Avg-T represents the overall performance across all questions.

		Hol	Oc	Vis	Avg	Avg-S	Avg-T
Eye	Util	54.7%	59.8%	43.2%	52.8%	42.6%	43.3%
	Priv	30.2%	45.5%	21.7%	32.4%		
Hand	Util	56.0%	62.1%	56.3%	58.0%	44.0%	
	Priv	30.3%	31.0%	28.5%	30.0%		

Relationship Between Comfort and Feeling Informed

The findings above clearly demonstrate that the extent to which participants feel comfortable, informed, and confident are impacted by the permission dialogs and the sensor tracking data in nuanced ways. At a higher level, we were also interested in whether participants who feel more informed also feel more comfortable using the device. Collapsed across all devices and sensors, feeling informed about the utility of the permission (r = .628. p < .001) and feeling informed about the associated privacy risks of the permission (r = .595, p < .001) were each correlated with comfort using the device or app. This correlation underscores the importance of *felt* comprehension. Similarly, the extent to which people felt confident that the device was securely storing their data (r = .792. p < .001) and felt clear about the data use policies (r = .668. p < .001) were also each correlated with comfort using the device or app. Regardless of actual understanding, feeling more informed and confident after reading permission dialogs may create a more comfortable experience for users — though not necessarily a more privacy-preserving one.

3.4.2 RQ3: Permission Comprehension Overview

Users can only make informed security and privacy decisions if they understand the implications of those decisions. The trade-offs between utility and privacy represent the benefits and risks inherent in these choices. Hence, it is crucial that systems are designed to clearly com-

Table 3.3: Comparing the level of comprehension regarding the system-level permission and application-level permission. See Table A.5 for details.

	Category	System	App	Diff
Hololens-Eye	Utility	73.0%	36.4%	-36.6%
	Privacy	30.9%	29.5%	-1.4%
Hololens-Hand	Utility	57.1%	54.9%	-2.2%
	Privacy	40.2%	32.8%	-7.4%
Oculus-Eye	Utility	67.9%	51.7%	-16.2%
	Privacy	49.5%	41.4%	-8.1%
Oculus-Hand	Utility	62.9%	61.4%	-1.5%
	Privacy	40.2%	21.7%	-18.5%
Vision-Eye	Utility	70.7%	15.6%	-55.1%
	Privacy	17.6%	25.9%	+8.3%
Vision-Hand	Utility	56.6%	56.1%	-0.5%
	Privacy	26.3%	30.7%	+4.4%

municate these factors, enabling users to navigate this balance with clarity and knowledge. In addressing RQ3, we investigate this dynamic by analyzing comprehension differences (1) across various sensors, (2) between system-level and app-level permissions, and (3) among different devices. Appendix A.2 provides the "answer key", to the best of our knowledge.

Comprehension Across Sensors

We first scored participants' answers to the true/false questions. We found that participants had a slightly better understanding of hand-tracking (average 44.0% across three platforms) than eye-tracking (average 42.6% across three platforms). Although participants generally understood the utility of eye-tracking and hand-tracking, on average scoring 52.8% and 58.0% on utility-related questions respectively, their understanding of privacy implications was noticeably lower. Specifically, participants only correctly answered an average of 32.4% of the privacy questions for eye-tracking and 30.0% for hand-tracking.

Comprehension Across System-Level and App-Level Permissions

We explored whether respondents' comprehension differs between system-level permissions and app-level permissions, where we see different technical and UX designs. As shown in Table 3.3, in all conditions examined, participants tended to be less informed regarding the utility of permissions within the application compared to their understanding of the same permission within the system. We observe a sharp decline in the understanding of eye-tracking utility at the app level for HoloLens (a decrease of 36.6%) and Vision Pro (a decrease of 55.1%). For participants' comprehension of privacy, we observe a similar declining pattern in the understanding of eye-tracking and hand-tracking privacy at the app level for both HoloLens and Oculus. The only exception is Vision Pro, where app-level privacy comprehension is better than system-level.

Comprehension Across Devices

Lastly, we assess whether users' comprehension differs across the three devices' permission-granting flows. Table 3.2 summarizes the comprehension score across devices. We conducted two-sample Z-tests to compare across devices. Participants who saw Oculus had a significantly higher comprehension of the eye-tracking utility ($z = 2.1842$, $p = .029$) and privacy ($z = 3.3082$, $p < .001$) compared to participants who saw Vision Pro. Participants who saw Oculus also showed significantly higher comprehension of privacy than participants who saw HoloLens ($z = 2.2008$, $p = .028$), but utility comprehension did not significantly differ ($z = 0.7158$, $p = .472$).

For hand-tracking, although participants who saw Oculus showed descriptively higher comprehension of the sensor utility, comprehension did not significantly differ from participants who saw Vision Pro ($z = 0.7762$, $p = .435$) or HoloLens ($z = 0.8611$, $p = .390$). Similarly, hand-tracking privacy comprehension among participants who saw Oculus was descriptively, but not significantly, higher than comprehension amongst participants who saw Vision Pro ($z = 0.3595$, $p = .719$) or HoloLens ($z = 0.1055$, $p = .912$).

3.4.3 RQ3: Specific Permission Comprehension

We next deeply investigate specific permission comprehension questions. We investigate the questions participants frequently answered incorrectly, indicated in a red color in Table A.5. We identified six topics where misperceptions commonly occur. In some cases, participants underestimate privacy risks, and in other cases, they overestimate them (i.e., underestimate privacy protections). We also identified topics where participants showed good comprehension.

Overestimating Access to Raw Data Retention

Our results revealed a significant gap in participants' awareness regarding the system's or application's capability to retain unprocessed images of the eye or hand. When asked if the device could retain raw data, on average over 42% of respondents across all three platforms believed that the system could store unprocessed eye-tracking data, and over 48% believed the system could store raw hand-tracking data. In terms of the raw eye-tracking data, both HoloLens and Vision Pro retain identifiable iris data from users. While this information is encrypted on the device, it is important to ensure transparency in how these data are generated and how the raw data will be processed after iris patterns are generated. Oculus explicitly states that it does not store any raw eye-tracking data [57], yet only 14.8% of participants answered this question correctly. For hand-tracking, both HoloLens and Vision Pro stored processed hand-tracking, such as hand gestures for system interactions [86] and size and the shape of your hand [83]. Oculus again states that it does not store any raw hand-tracking data [64], but only 9.1% of participants answered this question correctly.

Uninformed of Data Uploaded to System Servers

Participants were also largely uninformed about these platforms' eye-tracking and hand-tracking data-sharing practices. At the system-level, we found that 30% of participants believed that Oculus does not share eye-tracking data with external servers, and another 40% were unsure. Oculus's privacy policy states that abstracted gaze data is sent to and *stored* on their servers and will be dissociated from individual accounts when they no longer need it [63]. Many participants who saw HoloLens (49.5%) and Vision Pro (56.5%) were also unsure whether eye-tracking data would be shared with an external server, though Vision Pro explicitly mentions that eye data will not be shared with Apple, and HoloLens states that it avoids passing any identifiable information in their privacy statement. Similarly, we found that participants were largely uninformed about the hand-tracking data-sharing practice for HoloLens (57.9%), Oculus (56.8%), and Vision Pro (64.7%). Based on the privacy policy, we find that both HoloLens [86] and Vision Pro [83] stored abstracted hand-tracking information on the device, while Oculus shared the hand-tracking data with the Oculus server [64].

Overestimating Permission Model for HoloLens and Vision Pro

On a system level, we found that participants overestimated their ability to control the platform's access to their data through permissions. For example, 92.5% of participants believed they could control the system's access to eye-tracking data on HoloLens, and 87.1% believed the same for Vision Pro. For hand-tracking, 57.9% and 82.4% of participants held this assumption for HoloLens and Vision Pro, respectively. In reality, eye-tracking and hand-tracking for these systems are enabled by default. While both HoloLens and Vision Pro have adopted privacy-enhancing techniques to protect eye- and hand-tracking data, which are the primary source of interaction, our findings highlight a gap in user understanding of data control and practices on these platforms.

Overestimation of Background Data Access for Oculus and Vision Pro

Another common misunderstanding for Oculus and Vision Pro was the belief that applications could access eye-tracking or hand-tracking data in the background. From our experimentation, there was no direct API that allowed such background access. However, only a small percentage of participants answered correctly: 21.5% for eye-tracking and 11.6% for hand-tracking. For hand-tracking on HoloLens, the majority of participants (87.9%) answered correctly, likely due to the permission dialog clearly illustrating this capability.

Overestimation Application Access To Calibration and Biometric Eye-tracking data

At the application level, we found that participants overestimated the ability of applications to access their eye calibration data and biometric data (e.g., iris representation). For example, less than 5% of participants correctly identified that eye-tracking calibration data is not available to applications on HoloLens and Oculus. For platforms that support iris authentication, we found that on average, 67.3% of participants incorrectly believed that such information is accessible by applications.

Uninformed about the Privacy Practice for Vision Pro

Vision Pro acknowledges that even abstracted eye-tracking data could lead to serious privacy threats [83]. As a result, neither Apple nor third-party entities have access to these data. Only the final selection, rather than the eye movements, is available to the system and application. While Vision Pro arguably deploys the most privacy-preserving practices, we found that participants were largely uninformed. For example, when asked whether the system only collects the final selection, only 18.8% of participants answered correctly. Similarly, only 27.1% correctly understood that applications could not access eye-tracking data even with permission, and only 23.5% understood that only final selection is available to applications.

Comprehension for Eye- and Hand-Tracking Utility

Finally, we highlight questions where participants, with no prior AR experience, showed good comprehension. When asking participants about the main utility for eye-tracking (simulating your eye movement), and for hand-tracking (simulating your hand movement), we found that the majority of the participants understand these utilities, especially when such utility is clearly illustrated. For example, in the case of the hand-tracking utility for the Oculus system, participants showed a 100% comprehension rate. The only exception is the application utility for eye-tracking on Vision Pro, which may be a result of our finding in Section 3.4.3.

3.4.4 RQ4: Factors Impacting User Decisions

Given our scope focusing on participants with no XR experience, we asked them to rate what information about the system and the app can help them feel more comfortable using the technology in the future. Post-experience, they selected three out of five factors we provided. Figure 3.6 illustrates the distribution of these factors, and we observe consistency in the factors across devices.

We observed that participants preferred information about who would have access to this data. Given the sensitivity of the collected biometric data, it is important to provide clear and comprehensive information on whether the system, external device (company server), or application developers will have access to their data. Additionally, clarity on how the data will be transmitted—whether it is encrypted, stored locally, or shared with remote servers—is crucial in building trust and comfort with the technology.

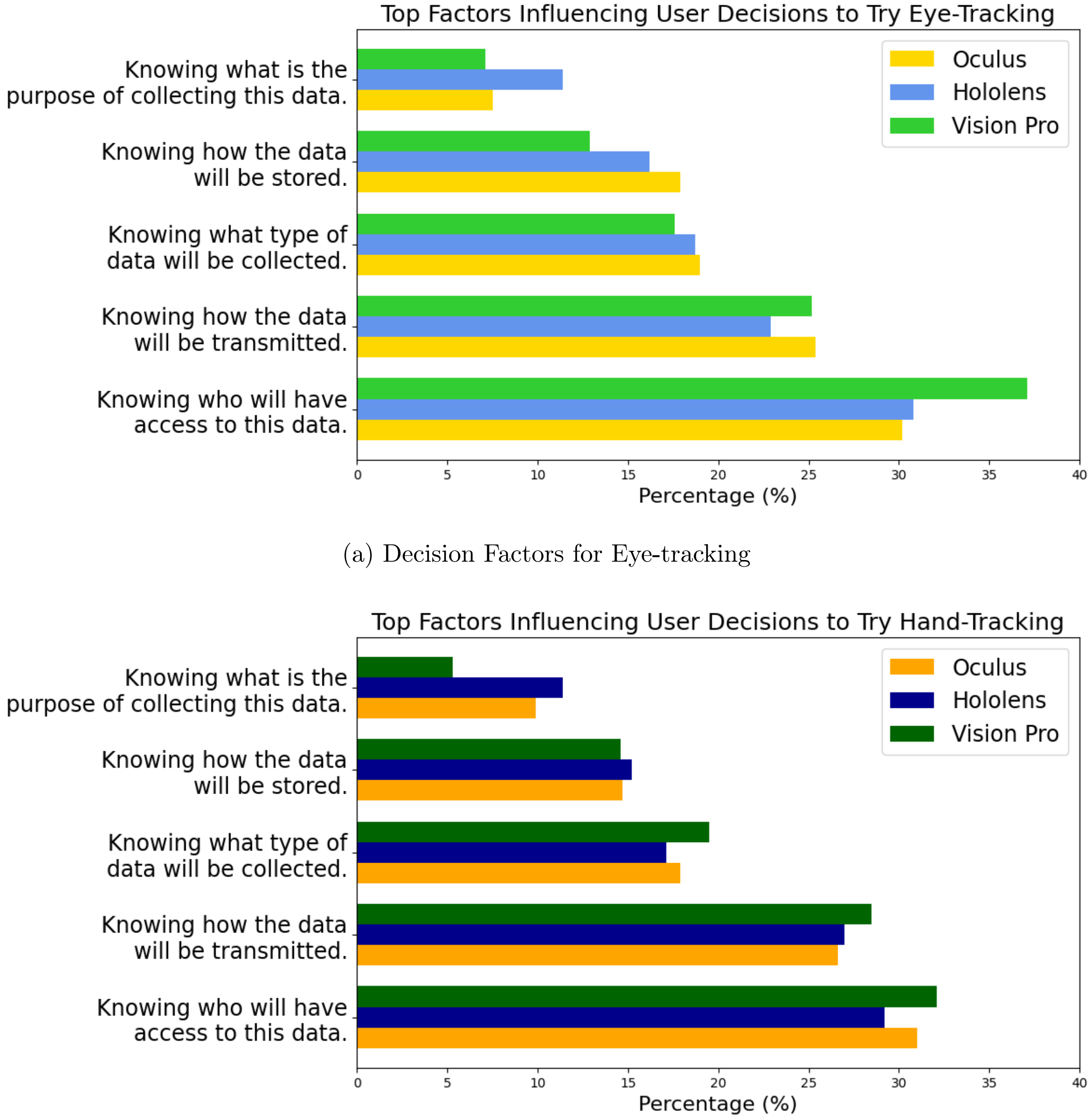
We were surprised to find less than 20% of participants regarded the type of collected data as a significant factor. This finding underscores a possible underestimation of the privacy risks associated with raw data access, and the need for more user education on the significance of raw data protection.

In addition, the distribution of factors considered important across hand-tracking and eye-tracking was highly consistent. This uniformity indicates that the factors that matter to users when considering the adoption of new technologies may remain consistent across various types of sensor data being collected. As AR technology advances, integrating more sophisticated sensors and collecting more data, we are hopeful that our findings will provide insights applicable to these emerging contexts as well.

3.5 Discussion

User's preference can be influenced by many factors, including the previous knowledge of these sensors, different data access models, dialog content, visualization, and UI flows. Informed by the results in Section 3.4, we identified several key lessons from our work that could enhance user's comfort and comprehension with implications for MR designers.

First, we found that effective communication about utility and privacy through permission UI flows enhances people's comfort and willingness to use the technology, which aligns



(a) Decision Factors for Eye-tracking

(b) Decision Factors for Hand-tracking

Figure 3.6: Decision factors breakdown for influencing user decisions to try eye-tracking and hand-tracking technologies on Hololens, Oculus, and Vision Pro.

with previous studies [137, 139, 281]. For example, the permission flow on Oculus provides more detailed descriptions, compared to HoloLens and Vision Pro, regarding the utility and privacy implications of eye- and hand-tracking sensors. Consequently, people who interacted with the Oculus interface were better informed, which not only aligned with how informed they *felt*, but also increased their comfort compared to the other platforms. Our findings in Section 3.4.3 suggest the necessity of including relevant descriptions to enhance topics where users tend to underestimate the system's privacy protections, such as preventing the retention of raw data.

Suggestion 1: AR platforms and developers should provide clear communication on potential utility and privacy to enhance user comfort and comprehension.

Second, our findings in Section 3.2 illustrate the different approaches AR platforms have taken in handling users' eye-tracking data. While HoloLens and Oculus both took active steps to protect users' privacy by only providing abstracted eye-tracking data, recent studies have shown that even abstracted eye-tracking data can contain significant privacy risks, such as revealing user intention [111, 134], psychological state [254], age [90], and cultural background [228]. Hence, we encourage these platforms to explore potential privacy-preserving mechanisms, including limiting system and application's access [83] or adding stronger privacy guarantees over the abstracted eye-tracking data stream [191, 195, 256]. However, we also noticed that while Vision Pro adopted stronger privacy-preserving techniques (providing to apps only final UI selections the system derives from eye-tracking data), people often failed to fully comprehend their implications (see Section 3.4.3). Apple may be missing an opportunity to enhance user comfort and understanding by clearly explaining the deployed protections.

Suggestion 2: Given the potential privacy risks with even abstracted eye-tracking data, we encourage platforms such as Oculus and HoloLens to provide stronger privacy protection. We also advocate for better communication with users if such practice is adopted.

Third, we found that while many participants considered data transmission information important for both sensors, many were inadequately informed about this factor. For example, around 70% of the participants were unaware that Oculus shares eye-tracking data to its own external server (and over 90% for hand-tracking). Despite Oculus outlining this in their privacy policy, given the low likelihood of users reading privacy policies [158, 168, 236], such information is not effectively communicated. We argue that it is essential to implement opt-in/opt-out features, allowing users to control their data-sharing preferences. In addition, the eye-tracking data retention period needs a clearer definition than “deletion when no longer needed”. Stepping back, platforms should also consider whether this data needs to be shared with external servers at all, and at what granularity and for which purposes.

Suggestion 3: For platforms that do upload data, such as Oculus, we suggest: (a) implementing an opt-in/opt-out feature for users to choose whether they wish to share eye-tracking data with external servers, and (b) providing a transparent explanation for this data collection, including the retention period, in the permission flow.

Fourth, our findings in Section 3.2 suggest that HoloLens and Oculus grant applications automatic access to users’ hand-tracking data, but a minority of participants understood this. Recognizing that hand-tracking is the main interaction modality and cannot be realistically opted out of entirely, we recommend that HoloLens and Oculus still provide finer-grained permission to limit applications’ access to certain hand-tracking data. For example, precise estimation of hand skeleton data could be limited, given its potential privacy implications for inferring sensitive attributes [193, 210].

Suggestion 4: AR platforms should implement fine-grained permissions for hand-tracking to provide users more control over their data, e.g., by restricting applications’ access to specific types of hand-tracking data.

Finally, it is important to consider the trade-off between privacy and usability in our recommendations. For example, deploying fine-grained hand-tracking permissions might put an extra burden on users. However, our results in Section 3.4.3 suggest that the majority of participants expect this control from HoloLens (71.0%) and Oculus (82.4%). In addition, results in Section 3.4.1 suggested that participants who saw HoloLens felt less comfortable with sharing hand-tracking data compared to Oculus and Vision Pro, possibly due to the lack of hand-tracking permissions. Similarly, if AR systems clearly illustrate their privacy mechanisms for eye- or hand-tracking data, this transparency might deter new users who are concerned about potential privacy issues. Although AR technologies have grown significantly over the past few years, with initial Vision Pro sales estimate of 200,000 units in 2024 [82], Quest Pro sales estimate of 100,000 units [72–75, 85], and HoloLens 2 sales estimate of 300,000 units [54] since release, they are still in the early stages of mass adoption. To position these technologies for broader acceptance, it is crucial to enhance users' trust through effective privacy mechanisms [100], preparing the mainstream market to bridge the adoption chasm [206]. We encourage future research to further explore this direction.

Suggestion 5: Proper privacy-enhancing techniques can better prepare AR technologies for future widespread adoption.

3.6 Conclusion

As AR technologies advance, the input mechanisms that enable natural user interaction also introduce novel privacy concerns. This chapter explored how three existing AR headsets — Meta's Oculus Pro, Microsoft's HoloLens 2, and Apple's Vision Pro — navigate permissions for eye- and hand-tracking input data, and the extent to which users feel comfortable and informed about granting access to these input channels. We find that people's experiences with and comprehension of permissions are affected by both the different design choices across devices and the nature of the input sensors themselves. Based on our findings, we suggest how current and future AR platforms can design permissions that effectively communicate information that is particularly important and that often goes misunderstood by end users.

Chapter 4

Output: Three-dimensional User Interface Security Properties

This chapter considers output, the second phase of the AR system data flow highlighted in Chapter 1. It explores security vulnerabilities in the context of the three-dimensional user interface (3D UI), which is a combination of AR virtual output, the visual elements from the physical world, and the user's input interacting with the immersive 3D world. More specifically, this chapter investigates security properties unique to three-dimensional output through five proof-of-concept attacks — including a clickjacking attack and an object erasure attack. We then present the first systematic evaluation of 3D UI security properties across five leading AR platforms: ARCore (Google), ARKit (Apple), Hololens (Microsoft), Oculus (Meta), and WebXR (browser). Our findings reveal that all platforms allow at least three of our proof-of-concept attacks to succeed. We conclude by discussing potential defense strategies, including adapting lessons from 2D UI security and exploring new directions for securing AR output. The work in this chapter, "When the User Is Inside the User Interface: An Empirical Study of UI Security Properties in Augmented Reality," was first published and presented at the 33rd USENIX Security Symposium in 2024 [117].

4.1 Introduction

Extensive past research and practice have considered user interface (UI) security for two-dimensional screens (desktop, browser, and mobile), e.g., clickjacking attacks that trick the user into interacting with UI elements [92, 99, 120, 157, 198, 238], information leakage via the user interface [43, 116], and isolating UI components from other entities [14, 222, 271]. In this chapter, we explore UI security in emerging augmented reality (AR) platforms.[1] AR immerses the user *inside* a three-dimensional user interface — including, in some contexts, the real world itself — in contrast to the user merely observing it from the outside.

Scope and Threat Model UI-level security for AR includes potential attacks on: (1) the user's *perception of the physical world*, (2) the virtual world or other *virtual content*, and (3) the user's *interactions with virtual content*. Regarding the first threat model, recent work has begun to study security and privacy for emerging AR platforms more generally, including some UI-related issues related to attacks on physical world perception. For example, it has discussed or demonstrated attacks in which malicious AR content is used to obscure important real-world (or virtual) content [119, 180] and side-channel attacks that allow malicious applications to infer information about the user's physical surroundings [210, 291, 293]. In this work, we consider a threat model where multiple entities might be interacting within the AR UI, such as third-party embedded code (e.g., a library) running inside an AR application in which the embedded code (e.g., the library) seeks to compromise a property of the AR application or vice versa. As the ecosystem continues to develop, we envision our threat model extending to future multi-user/multiple AR applications simultaneously augmenting the user's view of the world. Given our focus on multiple principals, we thus particularly consider threat models (2) and (3), i.e., from one principal on another principal's virtual content and on the user's interaction with another principal. This chapter examines how AR platforms handle the output and rendering of virtual content in multi-principal environments, where different entities may seek to manipulate what users see and how they interact with virtual objects.

AR UI Security Attacks and Properties Given this problem scope, we prototype several multi-principal UI security attacks on currently available AR platforms. These include:

1. A *clickjacking attack* on ARKit (Apple), where a malicious AR application component tricks the user into interacting with another component's virtual object. This attack is achieved by placing both objects at the same 3D coordinates and taking advantage

[1] We focus on AR and while we believe some findings might be possible in VR too, we are not in a position to fully clarify all differences between AR and VR.

of ARKit's inconsistency about which object is visible and which receives user input.

2. A *user input denial-of-service attack* on Hololens (Microsoft), where a malicious AR application component blocks the user from interacting with another or *any* virtual object. This attack is achieved by surrounding the target AR object in an invisible 3D box and/or entirely surrounding the user with an invisible 3D virtual box that captures all user input. This attack is possible on platforms that support invisible objects that are allowed to receive inputs.

3. An *input forgery* attack in ARCore (Google), where a malicious AR application component impersonates the user's interaction with virtual objects, even when they are not within the user's field of view. This attack is achieved by programmatically generating synthetic user interactions, taking advantage of the absence of input provenance, i.e., the ability to verify the origin of input.

4. A *object-in-the-middle attack* on Oculus (Meta), where a malicious AR application component snoops on the user's interactions with another component's virtual object. This attack is achieved by a combination of (a) an invisible object that intercepts the user's intended input, and (b) synthetic user input that is then dispatched to the original target object.

5. An *object erasure* attack in WebXR (browser), where a malicious AR application component uses an invisible virtual object to cause an underlying victim virtual object to disappear completely. This attack is possible due to WebXR's method for rendering invisible objects.

These proof-of-concept attacks are possible because of the way each platform implements three *specific UI-related properties* that we identified: What happens when multiple virtual objects are placed at the same 3D physical coordinates (*"Same Space"* property)? (2) How do "invisible" virtual objects work (*"Invisibility"* property)? (3) Do platforms allow synthetic user input (*"Synthetic User Input"* property)?

Empirical Evaluation of Current AR Platforms and SDKs After identifying these properties and demonstrating their security and privacy implications through the proof-of-concept attacks, we conduct a *systematic empirical investigation* of how current AR platforms and SDKs handle them. Specifically, we conduct experiments with ARCore (Google), ARKit (Apple), Hololens (Microsoft), Oculus (Meta), and WebXR (browser). We find inconsistencies in how current platforms handle the AR UI security properties that we identify and observe that many current implementations of these platforms enable attacks such as those we describe above.

Towards Future Defenses Our results highlight the necessity for emerging AR platforms to implement UI-level security precautions in their designs and implementations. However, defenses are not necessarily straightforward. In some cases, we recommend that platforms adopt known approaches from 2D UI security (e.g., adding user input provenance information [10]), though we found that no current platform has implemented this feature. In other cases, AR-specific approaches may need to be devised [184] or AR-specific tradeoffs considered (e.g., aligning the physics and rendering engines around UI security properties and handling UI isolation in a 3D, interleaved context [179]) when 2D mitigation fails to work.

Contributions We contribute the following:

1. We identify **three AR UI properties that have security and/or privacy implications** and provide criteria for evaluating them (Section 4.2).

2. Based on these properties, we demonstrate **five proof-of-concept AR UI security attacks** that are possible on today's AR platforms (Section 4.3).

3. We present **results of an empirical analysis** of these AR UI security properties in five commercially available AR platforms and associated SDKs: ARCore, ARKit, Hololens, Oculus, and WebXR (Section 4.4).

Finally, we reflect on **foundations for future defenses**, including known defenses that have not to date been applied to AR platforms and SDKs, as well as potential novel defensive directions (Section 4.5).

Disclosure We have reported all of our findings to Apple, Google, Meta, Microsoft, WebXR, and Unity.

4.2 Selected Properties and Evaluation Metrics

In Section 4.1, we described five proof-of-concept attacks on AR UI security. Before we return to these attacks in Section 4.3, we first identify and introduce three key AR UI security related properties that underpin these types of attacks.

Properties Our team, which included eight researchers, conducted multiple rounds of interactive threat modeling, structured brainstorming, and preliminary experiments to generate and refine ideas for design choices, properties, and/or test cases for AR platforms. Six (of eight) authors participated in the brainstorming process. This process involved: (1) each author independently generating security or privacy related questions and associated testable properties about AR platform designs (using sticky notes and a spreadsheet), (2) multiple authors reviewing and refining each row of the spreadsheet, and (3) clustering the properties according to themes (e.g., user input, multiple applications or components, hardware, sensors). From there, we chose to focus on AR UI security issues because we found them particularly interesting given the relationship between the UI, the environment, and the user and underexplored in current platforms.

1. **Same Space.** How do AR systems manage objects that share the same physical world mapping? For instance, when two AR objects with identical shapes and sizes are anchored at the same 3D coordinates, which object(s) become visible to the user? Which receive (s) the user's input? We leverage this property in the clickjacking attack (Section 4.3.2).

2. **Invisibility.** How do AR systems handle virtual objects in the AR world that are transparent? To what extent, if any, does an object's visibility influence its functionality? For example, are transparent objects capable of receiving user input? What happens when a transparent object renders over another virtual object? We leverage

this property in the user input denial-of-service (Section 4.3.3), object-in-the-middle (Section 4.3.5), and object erasure (Section 4.3.6) attacks.

3. **Synthetic User Input.** How do AR systems handle synthetic user input? For example, can adversarial code generate synthetic input to mimic human interaction, such as via a programmatically generated raycast? We leverage this property in the input forgery (Section 4.3.4) and the object-in-the-middle (Section 4.3.5) attacks.

Evaluation Metrics Moving from these properties and questions to *specific evaluation metrics* that we can use in our empirical investigation of platforms in Section 4.4:

For **Same-Space,** if two virtual objects from different application components are created with the same size and placed at the same 3D coordinates, we evaluate:

- *Rendering Order:* Is the object placed first or the object placed second visible?

- *Interaction Order:* Does the object placed first or the object placed second receive user input?

- *Rendering Flicker-Free:* Is rendering order consistent within a single trial, or does it flicker?

- *Rendering Consistency:* Is rendering order consistent across trials?

- *Interaction Consistency:* Is interaction order consistent across trials?

- *Rendering-Interaction Consistency:* Are the object that is visible and the object that receive user input the same object?

For **Invisibility**, if objects are transparent, we evaluate:

- *Create Invisible Object:* Using each of the following possible invisibility mechanisms, can an invisible object be created? Mechanisms include: (1) setting a zero alpha value, (2) disabling the object's renderer, (3) using a null material for the object, and (4) using a custom transparent material for the object?

- *Invisible User Interaction:* Can invisible objects based on the preceding mechanism take user input?

- *Composes as Expected with Virtual Objects:* If an invisible object is rendered overlapping another virtual object, what is the resulting visible rendering? Is the other virtual object visible? [2]

For **Synthetic Input**, if input is not created by real AR users, we evaluate:

- *Create Synthetic Input:* Does the platform support synthetic user input, such as through a simulated raycast?

- *Invisible Synthetic Input:* When synthetic input is dispatched, has any visible indication to the user occurred (e.g., via a visible raycast)?

- *Input Provenance:* When a virtual object receives user input, is there a way for it to distinguish real user input from synthetic input?

We stress that completeness, in properties or metrics, is not our goal. Instead, we focus on metrics derived from our brainstorming activity that we considered important, challenging, and interesting in the AR context.

Valid Use Cases for Properties While we focus on the security implication of these properties, we emphasize that they can also enable necessary features in many AR applications. Thus, seemingly simple solutions such as disallowing objects from occupying the same space, disabling invisible objects, or disallowing synthetic input will not be tenable.

The **Same Space** property allows designers and developers greater flexibility when creating AR experiences, enabling them to design complex scenes and arrangements where objects can interact, stack, or blend with each other in creative ways. In architectural or interior design AR applications, the ability for virtual objects to overlap becomes particularly useful as they can represent multiple layers or illustrate the relationships between

[2] We were motivated to add this evaluation metric after discovering the object erasure attack for WebXR, described in Section 4.3.6.

different elements, providing a rich visual representation of the design. Moreover, the Same Space property facilitates intuitive user interaction and manipulation. Users can freely move around in the physical space, effortlessly viewing, moving, rotating, or scaling individual objects. The dynamic updating of visuals in response to changes in the user's physical surroundings or interaction occurs smoothly without rigid collision constraints, ensuring a seamless and natural experience.

The **Invisibility** property also contributes to a broad range of practical functionalities. For example, Pokemon Go, one of the most popular AR games, uses invisible objects as a placeholder for Pokemon that are still under development [47]. Moreover, existing research corroborates the value of this functionality in medical training scenarios [104]. Here, the AR objects are designed to be context-sensitive and can transition to an invisible state, allowing surgeons to look through overlaid content without obstruction. In addition, invisibility is also widely used for handling occlusion between virtual objects and real-world objects to ensure that virtual objects are realistically occluded by real-world objects, improving overall immersiveness [5].

The **Synthetic User Input** property, commonly implemented via raycasting, is integral to the functioning of AR features. Its primary function is to indicate user interaction and selection in the real world by returning information about the selection, such as the distance, position, or a reference to a real-world object (plane, surface) or virtual AR object. The many legitimate use cases of synthetic user input include, for example, an AR shooting game that generates synthetic input as a shooting ray to intersect with the designated object.

Hence, it is crucial not to simply disable the features associated with these properties solely due to their potential security implications. Rather, we advocate that AR platform and application designers carefully consider the potential security implications in addition to the desirable use cases.

4.3 Threat Model and Proof-of-Concept Attacks

We now return to the proof-of-concept AR UI security attacks that we previewed in Section 4.1. First, we detail our threat model. Then, we describe in detail our proof-of-concept attacks on five AR platforms (ARCore, ARKit, Hololens, Oculus, and WebXR). Though we choose one platform on which to implement each attack, our empirical investigation in Section 4.4 will reveal that multiple platforms provide the preconditions for implementing most attacks.

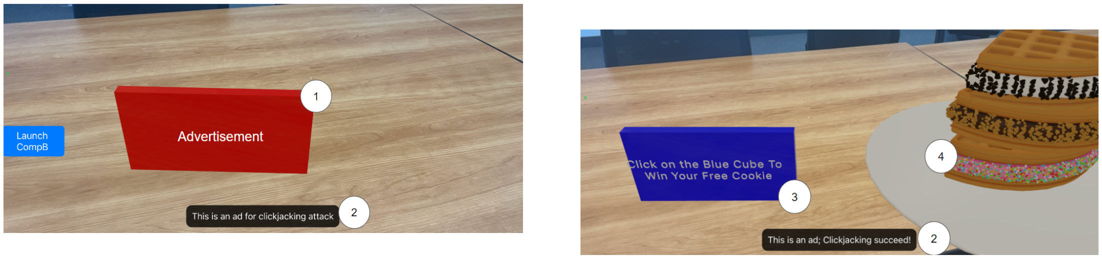
4.3.1 Threat Model And Attack Preconditions

We assume adversarial behavior might extend from one entity (e.g., an included ad library) to another entity (e.g., the main app), or vice versa. Attackers could directly call the API from the SDK to either gather sensitive information from the AR application (e.g., location of a target AR object) or place content within the AR virtual world. Our threat model resembles those used in other platforms, e.g., malicious third-party iframes [96, 289] or included third-party libraries in the mobile ecosystem [288, 292]. However, the current AR environment is arguably more vulnerable given there are no iframe-like primitives that isolate the execution of third-party code; instead, it shares a portion of the displayed AR scene.

All proof-of-concept attacks rely on at most these three preconditions (in addition to the individual attack-specific metrics we test): (1) the location of the targeted object, (2) the ability to generate virtual content interleaved with the victim's content, and (3) the execution of synthetic input. These preconditions are straightforwardly met in today's systems, where third-party libraries are included in applications as either the attacker or victim components.

4.3.2 Clickjacking: Leveraging Inconsistency between Rendering and Interaction Orders

Attack Motivation: Bait User Interaction Clickjacking is an attack that fools users into thinking they are clicking on one thing when they are actually clicking on another. As they can on web and mobile platforms today, app developers will be able to use third-party ad libraries on AR to display ads and generate revenue. AR advertising has attracted huge interest, and revenue in this market is projected to reach US $1.05 billion in 2023 [3]. Attackers are well-incentivized to mount clickjacking attacks on ads it includes, tricking users into clicking on them and thereby increasing ad revenues. Prior work has hypothesized and demonstrated clickjacking attacks through hijacking the cursor in the context of VR [184]. Here, we demonstrate one type of clickjacking attack in AR without modifying the user's


Launch CompB
Advertisement
1
This is an ad for clickjacking attack
2
Click on the Blue Cube To Win Your Free Cookie
4
3
This is an ad; Clickjacking succeed!
2

(a) User's view of an AR advertisement object.

(b) Demonstration of the clickjacking attack

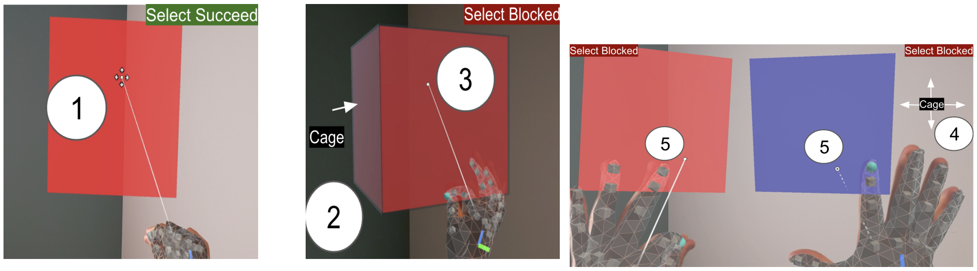
Figure 4.1: **Clickjacking attack on ARKit.** ① The advertisement object displayed in the AR world. ② A prompt after the advertisement is clicked. ③ A bait AR object rendered on top of the advertisement object exploiting the Same Space property. ④ An accompanying bait AR object. After the user clicks on the bait object ③, the interaction goes into the advertisement object ① and generates the prompt ②.

interaction.

Attack Design We implement this attack in iPhone 13 using the ARKit SDK. Our intuition is that the AR platform will handle the rendering sequence and interaction sequence differently when two objects are placed in the same 3D coordinates. For example, when we place two AR objects in ARKit using the same anchor ((`object.setParent(AnchorEntity)`)), the object placed second will always render over the first object, but the user's interaction will always trigger the functionality of the first, now hidden object. The attacker here is a revenue-hungry developer. The ad revenue will go to the developer as the user's interaction with the bait object goes into the ad object. The preconditions for this attack are (1) the location of the targeted object, and (2) the ability to generate virtual content interleaved with the victim's content.

Figure 4.1 illustrates the attack. When the victim/user launches the application, an advertising platform places a third-party ad ① in a certain bounded region of the main app, and a revenue-hungry developer/attacker then places a new interactive bait object ③ in the same space as the advertisement. This component displays the message "click here to win your free cookie" to bait user clicks. However, the user's interaction with the bait object actually triggers the underlying ad ① even as the attacker 'steals' revenue from the click.

4.3.3 Denial-of-Service Proof-of-Concept: Leverage Invisibility



(a) User's view of the victim object. User is able to interact with the object.

(b) Demonstration of the first denial of service attack. "Cage" covers the object.

(c) Demonstration of the second denial of service attack. "Cage" covers the entire space.

Figure 4.2: **Denial-of-user-input attack on Hololens.** ① The user can select and interact with the victim object (red). ② The attacker overlays a fully transparent object over the victim object (red). For demonstration, we make the transparent object ("cage") slightly visible. ③ The invisible "cage" blocks user interaction with the victim object (red). ④ In addition, the attacker can surround the user in a fully transparent "cage". ⑤ The invisible "cage" blocks the user from interacting with *any* AR objects, in this case, the red and blue AR objects.

Attack Motivation: Block User Interaction A denial-of-service attack refers to an explicit attempt by a malicious entity to deny legitimate users access to an object/service. In this context, the attacker's goal is to stealthily prevent the user from interacting with the target AR object, e.g., preventing the user from engaging with a competitor's content or disrupting the user from properly engaging with *any* AR object.

Attack Design We implement two variant of the denial-of-service attacks in Hololens 2 using the MRTK. The attack insight is that the Hololens default raycast implementation, i.e., `private static RaycastResult PrioritizeRaycastResult`, would return the closest hit object. Based on our findings in Section 4.4, we find that an attacker can construct a completely invisible object that captures the user's input. Furthermore, the 3D nature of the AR virtual world lets the attacker envelop the user within this invisible 'object cage,' causing the invisible object to intercept, initially and always, the user's interactions. The preconditions for this attack are (1) the location of the targeted object, and (2) the ability to generate virtual content interleaved with the victim's content.

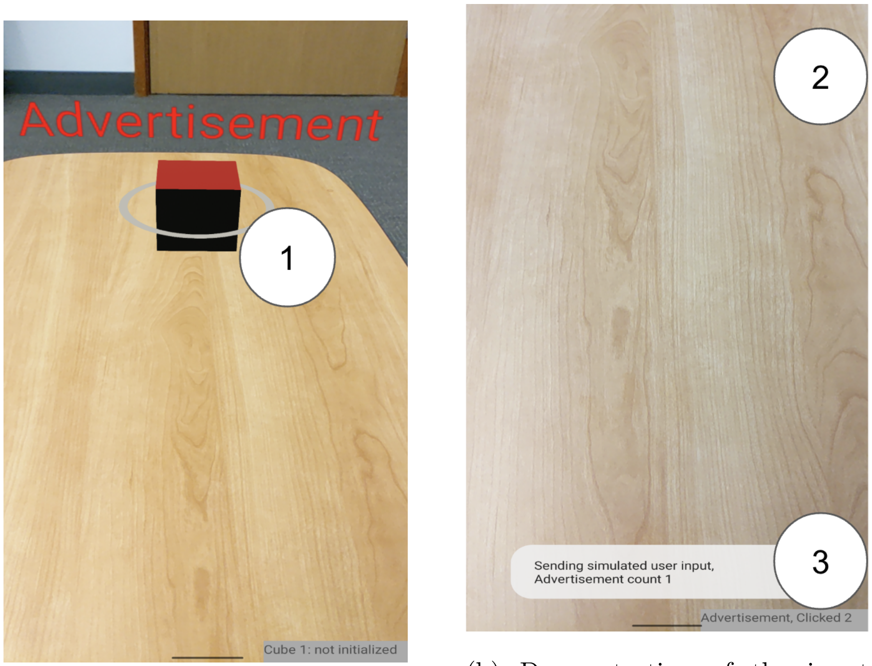
Figure 4.2 illustrates this attack. The victim/user launches the application and intends to select an AR object ① using a hand ray as the input mechanism. However, the attacker can either overlay an invisible object ② on the targetted AR object to block user from interacting with it ③. Furthermore, the attacker can overlay a large invisible object ④ that encapsulates the user such that the user is always physically inside of the object. The user's interaction towards *any* AR object is thus blocked ⑤ regardless of their movements within the physical space.

4.3.4 Input Forgery: Leveraging Synthetic User Input

Attack Motivation: Impersonate User Interaction Similar to the motivation in the clickjacking attack, here the adversary's goal is to maximize advertisement engagement. The attacker places an advertisement object outside of the user's view and generates synthetic user input to increase the number of ad interactions. Placing it outside of the user's peripheral is not a requirement though it will make the attack more stealthy.

Attack Design We implement this attack in Pixel 5 using the ARCore SDK. The attack insight here is that a programmable click can interact with objects outside of user's view by exploring the limited display of field-of-view. The precondition for this attack is the execution of synthetic input.

Figure 4.3 illustrates the attack. When the victim/user launches the application, the malicious application developer will place the third-party ad ① in the user's view. When the ads' location is outside of the user's view ②, the attacker will then generate synthetic input ③ to trigger interactions on the ads, increasing its interaction count, and later charging the respective advertisers for this inflated number of ad views.



Sending simulated user input,
Advertisement count 1

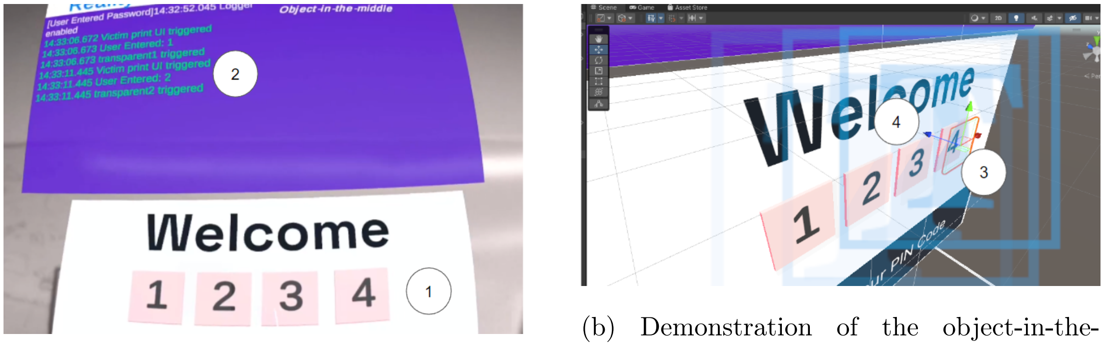
(a) User's view of the victim app.

(b) Demonstration of the input forgery attack

Figure 4.3: **Input forgery attack on ARCore.** ① The advertisement object displayed in the AR world. ② When the advertisement object is outside of the user's view, ③ the attacker launches a synthetic input to interact with the object. The prompt demonstrates the attack is successful.

4.3.5 Intercepting User Inputs: Combining Invisible Objects and Synthetic User Input

Attack Motivation: Hijack User's Interaction Object-in-the-middle is an attack in which a third party object gains access to (or "intercepts") the communication between two other objects. As AR applications continue to grow in domains beyond entertainment, users may enter sensitive information—such as PIN codes, passwords, or private messages—by interacting with a virtual keyboard rendered in the AR space. Each virtual key is a `collider` that provides collision detection. When the user either presses a key or casts a ray, the collider detects the intersection and outputs the corresponding value.



```
[User Entered Password]14:32:52.045 Logger
enabled
14:33:06.672 Victim print UI triggered
14:33:06.673 User Entered: 1
14:33:06.673 transparent1 triggered
14:33:11.445 Victim print UI triggered
14:33:11.445 User Entered: 2
14:33:11.445 transparent2 triggered
```

(a) User's view of the victim app.

(b) Demonstration of the object-in-the-middle attack.

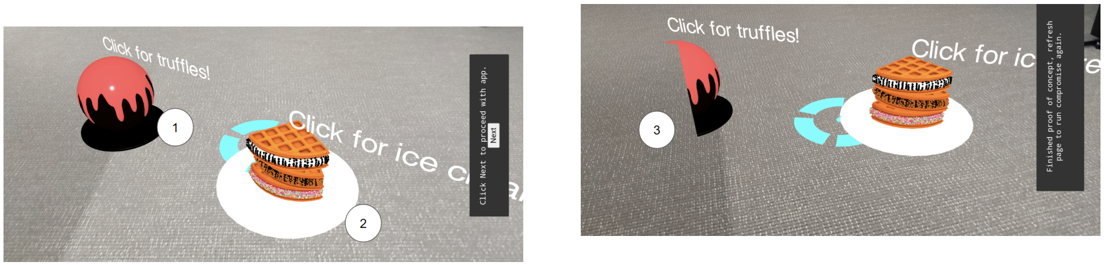
Figure 4.4: **Object-in-the-middle attack on Oculus.** ① The interface for authentication. ② The logger for the execution result. ③ The invisible object the attacker places over the pin pad object. ④ The blue arrow suggests the direction of the synthetic input to trigger the pin pad object.

Prior work has shown the possibility to sniff user text input through various side-channels [203,249]. Here, we demonstrate in AR that not only can an attacker surreptitiously steal the user's input, but it can also impersonate the user or even modify the original input, similar to a man-in-the-middle attack.

Attack Design We implement this attack in the passthrough mode of Oculus Quest 2. Our attack intuitions here are twofold. First, similar to the denial-of-service attack, it is possible to block the user's original input by overlaying a transparent object on top of the keyboard. Second, we can cast a synthetic ray to impersonate the user's interaction since the current platform does not provide input provenance.

Given our observations, we implemented an object-in-the-middle attack on Oculus. The preconditions for this attack are (1) the location of the targeted object, (2) the ability to generate virtual content interleaved with the victim's content, and (3) the execution of synthetic input.

Figure 4.4 illustrates the attack. When the victim/user launches the application, it shows a mock authentication interface ①, part of the main app that consists of a pin pad for the


Click for truffles!
1
Click for ice c
2
Click Next to proceed with app.
Next
Click for truffles!
3
Click for ic
Finished proof of concept, refresh page to run compromise again.

(a) User's view of victim app.

(b) Demonstration of the object erasure attack.

Figure 4.5: **Object erasure attack on WebXR.** (①,② Two competing advertisements. ③ Attempt to erase the competitor advertisement using transparent mesh. (We partially offset the invisible object for attack visibility.

user to enter their password. We present a logger ② to illustrate the efficacy of the attack. A malicious third-party component places several transparent meshes in front of the pin pad entity to intercept the user's input ③. Once the transparent meshes detect the user's interaction, the malicious component casts a synthetic input ④ to the pin pad, passing the user's input on to its intended destination.

The result is that the attacker has intercepted the password, but the user has not noticed anything amiss. Following the same attack logic, the attacker could also modify the user's input if needed since they can arbitrarily define the synthetic input destination.

4.3.6 Object Erasure: Leveraging Invisible Meshes

Attack Motivation: Erase Other AR Objects Echoing our attack focus, here we consider a scenario where the attacker aims to interfere with another party's virtual content by manipulating targeted objects. Motivations for such actions can vary greatly. For instance, one potential objective could be to erase a competitor's AR content, such as advertisements or promotional material. Another might be the desire to manipulate the user's perception: by altering or erasing certain AR objects, e.g., vital information or safety warnings, an attacker could significantly distort the user's view of reality [119, 180]. Such manipulation of AR content could also be deployed as a form of digital vandalism [166], comparable to defacing a physical sign or billboard.

Attack Design We implement this attack in the Chrome browser using the WebXR SDK and Three.js library. When we assign a fully transparent mesh (png format) as the material of an AR object using `new THREE.MeshBasicMaterial( map: THREE.TextureLoader().load('transparent.png' ))`, we find that not only does the AR object become fully invisible, but it also "erases" the rendering of other AR objects positioned behind it. The preconditions for this attack are (1) the location of the targeted object and (2) the ability to generate virtual content interleaved with the victim's content.

Figure 4.5 illustrates the attack. When the user launches the application, it presents two competing third-party ad libraries ① and ②. Code from one mock ad library then places a transparent mesh in the same space as the competing advertisement to erase it ③.

4.4 Empirical Investigation

Stepping back from individual proof-of-concept attacks, we now turn to a more systematic investigation of current AR platforms. We evaluate how five leading AR platforms — ARCore (Google), ARKit (Apple), Oculus (Meta), Hololens (Microsoft), and WebXR (browser) — implement the AR UI properties identified in Section 4.2. In this section, we first describe the infrastructure for our experiments, detail the platforms and configurations we used, and then present results of the experiments. Finally, we discuss the security implications of our results (Section 4.4.4).

4.4.1 Overall Experiment Infrastructure

We systematically developed a repeatable set of experiments for each property that we identified in the previous section (i.e., Same Space, Invisibility, and Synthetic Input). The experiments we conducted varied depending on the evaluation metrics for that property.

A human experimenter ran each experiment and another researcher recorded the experiment results, as demonstrated in Figure 4.6. Each experiment follows the procedure described in Figure 4.7. The central component of each experiment is a harness launched at the procedure's beginning. Corresponding to our threat model, each experiment further

Figure 4.6: Researchers conducting the experiments.

involves two AR application components, ComponentA and ComponentB, representing two pieces of code from different entities (e.g., a main app and an embedded third-party library).

With the experimenter's input, the harness (1) launches ComponentA, then (after the experimenter made necessary observations) (2) launches ComponentB, and finally (3) steps through the experiment in four different locations, which we annotated as Location N, S, E, W, interacting with the AR object in different spatial locations and providing input to the harness to proceed.

To account for potential non-determinism, we conducted each test case experiment $M = 5$ times. Further, each test can have up to $N = 5$ sub-experiments. Details about ComponentA and ComponentB depend on the nature of each test case.

4.4.2 Experiment Platforms and Configurations

We describe here the specific experimental configurations and versions we used for each platform. An experimental setup requires not only a choice of platform (e.g., ARKit) but

```
**Experimenter's Procedure**($M, N$)
  For $j$ = 1 to $M$ do
    Launch Harness($N$)
    Launch and provided any user input to start *experiment*
    Launch and provided any user input to start
      ComponentA.launch()
    Launch and provided any user input to start
      ComponentB.launch()
    For $k$ = 1 to $N$ do
      click "Next" button
      Start ComponentA.next()
      Start ComponentB.next()
      Perform sub-experiment observations and interactions
    Exit Harness($N$)
```

Figure 4.7: The parameter M denotes the number of experiments that the experimenter (a person) should run. Throughout the procedure's pseudocode is the assumption that the experimenter records detailed notes of observations during each experiment.

also additional choices for the entire tech stack (e.g., which SDK to use). Different choices for the full tech stack might have led to different experimental results, and we will release our experiment code to allow future work to adapt it for other contexts — including future technologies, like Apple's recently announced Vision Pro headset [56].

ARCore ARCore [4] is Google's Android core SDK for augmented reality. We built our test cases using ARCore v1.32.0 [4] for AR functionalities and Sceneform SDK [12] for 3D content rendering (version 1.20.5). The test cases were implemented in Java and tested on the Pixel 5a.

ARKit ARKit [8] is Apple's main AR SDK and includes multiple iOS AR frameworks, such as RealityKit [1] and RoomPlan [2]. We built our test cases using ARKit and RealityKit for necessary AR functionalities. The three test cases were implemented mainly in Swift and tested on the iPhone 13 Pro.

Oculus We use Quest 2 with the experimental Passthrough API [113] and spatial anchors to enable an AR experience. We built our test cases using Oculus Integration SDK v44.0 [23]. They were implemented mainly in C# (Unity) and tested on the Oculus Quest 2.

WebXR The WebXR Device API [44], led by the W3C Immersive Web Community Group, provides a uniform abstraction layer to host immersive web content. We built our test cases using the WebXR API-20220601 [45] for necessary AR functionalities and Three.js [37] for 3D content rendering. The three test cases were implemented mainly in JavaScript and tested on the Chrome Browser.

Hololens We deployed a Hololens 2 application using the Mixed Reality Toolkit (MRTK) 2.0 [46] and Unity 2021.3.16f1 [40]. MRTK is an authorized Microsoft project that provides components to facilitate MR development in Unity. We employed numerous features of MRTK in our test cases, including spatial anchors, gesture recognition, and object manipulation.

4.4.3 Experiments and Results

We now examine our experiments and results, per property, based on the evaluation metrics presented in Section 4.2. Table 4.1 highlights all our results.

Property	Condition	Metrics	ARCore	ARKit	Hololens	Oculus	WebXR
Same Space	Location N	Rendering Order	Unstable	②	①	Unstable*	Unstable
		Interaction Order	②	①	②	Unstable*	Unstable
	Location S	Rendering Order	Unstable	②	①	Unstable*	Unstable
		Interaction Order	②	①	②	Unstable*	Unstable
	Location E	Rendering Order	Unstable	②	①	Unstable*	Unstable
		Interaction Order	②	①	②	Unstable*	Unstable
	Location W	Rendering Order	Unstable	②	①	Unstable*	Unstable
		Interaction Order	②	①	②	Unstable*	Unstable
		Rendering Flicker-Free	○	●	●	●	○
		Rendering Consistency	○	●	●	○	○
		Interaction Consistency	●	●	●	○	○
		Rendering-Interaction Consistency	○	○	○	●	○
Invisibility	Alpha Value = 0	Create Invisible Object	○	●	○	●	○
		Invisible User Interaction	N/A	●	N/A	●	N/A
	Disable Renderer	Create Invisible Object	●	●	●	●	●
		Invisible User Interaction	○	○	●	●	●
	Customized Material	Create Invisible Object	●	○	○	○	●
		Invisible User Interaction	●	N/A	N/A	N/A	●
	Null Material	Create Invisible Object	○	○	○	○	●
		Invisible User Interaction	N/A	N/A	N/A	N/A	●
		Composes as Expected w/ Virtual Objects	●	●	●	●	○
Synthetic Input	Inside Field-of-view	Create Synthetic Input	●	●	●	●	●
		Invisible Synthetic Input	●	●	●	●	●
	Outside Field-of-view	Create Synthetic Input	●	●	●	●	●
		Invisible Synthetic Input	●	●	●	●	●
		Input Provenance	○	○	○	○	○

Table 4.1: Overview of experiment results. Filled circles indicate the metric is satisfied, and empty circles indicate they are not. For the Same Space experiment, circled 1 indicates that the object is created by the first (victim) principal; circled 2 indicates that the object is created by the second (adversarial) principal. "Unstable" means that the results are inconsistent during a single trial. "Unstable*" means that the results are inconsistent between multiple trials.

Same Space Experiments and Results

Experiment Design In the SameSpace experiment, ComponentA creates a virtual Cube1 object at one coordinate in the physical world, and ComponentB creates a second Cube2 placed in the same physical location. Both Cubes use the same spatial anchor, which is a specific 6-degree-of-freedom pose (position and orientation) of the physical world, as the physical location reference and are registered with a user input handler. The experimenter moves to four different locations (noted as North (N), South (S), East (E), and West (W)). At each location, the experimenter observes the rendering sequence of the overlapping cubes and interacts with the Cubes by sending a virtual ray or tapping on the overlapping region to register the returning result. To verify our code is operating correctly, each Component displays additional cubes that the experimenter observes and verifies as part of the procedure. We omit further details about such checks because in all cases the checks were made and passed.

From these activities, the experimenter evaluates the metrics specified in Section 4.2. Specifically, the experimenter observes: which cube is visible (based on which color is visible), which cube takes user input (based on log output from the cube's input handler), and whether these observations are consistent across multiple conditions and trials.

Experiment Results The top section of Table 4.1 presents the results. We find that platforms are inconsistent about which object is visible, which object receives input, and whether the visible object is the one receiving input. Significantly, these inconsistencies appear between platforms, across multiple trials of a single platform, and even within a single trial (i.e., flickering cubes). For example, for ARKit, Cube2 is always rendered, but Cube1 receives the user input; the opposite is true for HoloLens. For WebXR, there was significant inconsistency across multiple experiments as the overlapping Cube flickers and either cube could register the user's input. ARCore flickers in a slower way, but all of the user's input goes into Cube2.

We find that Oculus is the only platform that handles rendering-interaction consistently, with a caveat that 15% (15 out of 100) of the trials fail to maintain this consistency. While Oculus prevents flickering from the overlapping Cube, we find that the rendering sequence and the corresponding interaction sequence flip as the user moves to the next testing location. In addition, we also notice an inconsistency across multiple trails in terms of which Cube appears on top. These findings suggest that current AR platforms have not systematically considered the Same Space property or these types of corner cases.

Invisibility Experiments and Results

Experiment Design In the Invisibilityexperiment, ComponentA creates a visible Cube1 object at one coordinate, and ComponentB creates a visible Cube2 in front of Cube1. Cube2 is significantly larger than Cube1. The harness asks the experimenter to confirm if Cube2 completely occludes Cube1 and, upon confirmation, ComponentB deletes Cube2 and creates a new invisible Cube2' at the same location and with the same geometry; it does this by attempting four different mechanisms: (1) setting the alpha value to zero, (2) providing a null material, (3) disabling rendering, and (4) uploading a transparent texture as the object's material. Both Cubes are registered with a user input handler. The process with Cube2 size and placement verification and then Cube2' instantiation ensures that input makes sure any interaction will go into Cube2' first. The experimenter then evaluates the metrics from Section 4.2, observing whether the invisible cube is fully invisible and dispatching user input to the invisible cube to observe if its input handler is triggered.

Experiment Results The middle section of Table 4.1 presents the results. We find that all five AR platforms can create invisible objects that take the user's input. However, the detailed implementation condition for each platform differs slightly. For example, by setting Cube2's alpha value to zero, ARKit and Oculus both generate a completely transparent cube while registering the user's input. When disabling the rendering, ARCore and ARKit completely occlude Cube2', meaning that the cube can neither be seen nor receive input; Cube2' in WebXR, Hololens, and Oculus instead take input while being fully invisible. For Hololens and Oculus, this occurs because the rendering and collision engines are separate, which means objects can still possess physical characteristics without being rendered.

When uploading the customized transparent texture, we find that ARCore and WebXR produce a fully invisible Cube2 that takes input, while the other three platforms generate a slightly visible Cube2. All platforms except WebXR handle the null material edge cases by generating a solid color Cube2. Surprisingly, we found that the AR object generated by one principal in WebXR can affect the appearance of *other* AR objects. Specifically, whenCube2' is created with an uploaded transparent texture, we observe that the visible object located behind the invisible object (based on the user's viewing position) seems to disappear.

Synthetic Input Experiments and Results

Experiment Design In the Synthetic Input experiment, ComponentA creates a virtual Cube1 object at one coordinate with an input handler. When the experimenter presses a button, ComponentB creates synthetic user input — in the form of a simulated raycast — to interact in the direction of Cube1. The direction of the simulated raycast is calculated by retrieving the location of Cube1 and generating the corresponding direction vector. In addition, we test if input visualization is required to generate synthetic input. The experimenter observes whether the input handler of Cube1 was triggered. This experiment is repeated with the target object (Cube1) inside the user's field of view and outside the user's field of view (e.g., behind the user).

Experiment Results The bottom section of Table 4.1 presents the results. We find that all platforms allow synthetic user input, which is moreover invisible *and* can interact with the target AR object (Cube1). One consistent observation across all platforms is the absence of effective input provenance verification. This is a significant shortfall as the raycast API does not supply sufficient information to distinguish between genuine and synthetic user input. In addition, we discovered that for synthetic user input to be functional, the target objects need not be within the user's field of view. This leverages users' limited visual awareness when they are physically present within the 360-degree immersive AR user interface when the AR objects themselves remain rendered within the scene, exploiting this gap in user perception.

Table 4.2: Analysis of which attacks each platform (in our testing configuration) enables. ★ indicates this attack is possible to implement based on our experimental results; * indicates the attack might fail due to the AR platform's inconsistent behavior.

Proposed Attack	Property	Attack Precondition	ARCore	ARKit	Hololens	Oculus	WebXR
Denial-of-service	Invisibility Invisibility	Create Invisible Object: ● Invisible User Interaction: ●	★	★	★	★	★
Object Erasure	Invisibility	Composes as Expected w/ Virtual Obj.: ○					★
Input Forgery	Synthetic Input	Input Provenance: ○	★	★	★	★	★
Clickjacking	Same Space Same Space	Interaction Consistency: ● Rendering-Interaction Consistency: ○		★	★	★*	
Object-in-the- Middle	Invisibility Synthetic Input Synthetic Input	Invisible Interaction: ● Input Provenance: ○ Invisible Synthetic Input: ●	★	★	★	★	★

4.4.4 Security Implications

Returning to our proof-of-concept attacks from Section 4.3, we can now directly connect our empirical evaluation results with those attacks. In Table 4.2, we connect each attack to its preconditions — that is, to the necessary experiment result(s) that would enable the attack. For example, our denial-of-service attack requires that a platform be able to create invisible cubes and that those cubes be able to receive user input (i.e., "Invisible Cube Input Registration: ●"). Based on our experiment results in Table 4.1 and each of the attack preconditions, we can thus summarize in Table 4.2 which of our tested SDKs are vulnerable to which attack. We demonstrate all five proof-of-concept attacks in current AR platforms in Section 4.3.

4.5 Discussion

As with all emerging technologies — from smartphones (in the early 2000s) to computerized automobiles (also in the early 2000s) — many defensive strategies can be adopted from earlier technologies and new challenges must be overcome. We view this work as one component in the evolution of computer security and privacy for AR systems. While it is impossible to predict the future, we reflect on our work's possible place in this evolution here.

4.5.1 Problem Formulation

There are often vulnerabilities in emerging technologies that, from a purely abstract technical perspective, are unsurprising. Consider, for example, the discovery of `strcpy` vulnerabilities in automobiles in 2011 [115], over two decades after Aleph One's classic "Smashing the Stack for Fun and Profit" [217]. The contribution of such results is not in finding yet another vulnerability but in assessing *how* an emerging technology — one that has not yet seen significant security analysis — *might* be vulnerable. We consider our work — our identification of properties to assess as well as our exploration of case study attacks — to be of this lineage. Moreover, we consider not only how AR systems might be vulnerable, but we empirically find that the designs of *all* five instantiations of the AR technologies that we study *do* expose themselves to attacks.

Today's AR systems may not need to be secure against the types of attacks we study.

They are still emerging and predominantly single-principal. But, under the assumptions that future systems might be multi-principal, more widely used, and make design decisions that often persist in future technologies even as the threat landscape changes, we believe that it is imperative for AR platform designers to consider mitigation strategies now.

4.5.2 Knowledge of Defenses for 2D UIs Will Help

Applying existing (known) defensive strategies from other domains is a natural and appropriate first line of defense when considering newly discovered vulnerabilities in emerging technologies. For example, in the automotive domain, early research suggested the use of (even then) standard defenses, such as application-level authentication and encryption and the avoidance of unsafe code like `strcpy` [115]. Likewise, there is already an existing wealth of knowledge on UI security for 2D interfaces (desktop, mobile, web), and we encourage the adoption (or extension) of those defensive techniques to AR systems.

For example, an invariant that can provide resilience to clickjacking on the web is the following: a user can interact with a web object if and only if the web object is visible and has been visible for at least a minimum amount of time [157]. That defensive strategy, if implemented and fully instantiated, would serve to strengthen AR systems. However, is it possible to fully instantiate that defensive strategy? We elaborate on this question below.

Other known techniques that could be adapted for AR UI security — once the need to adapt them has been identified, as through our work — include: (1) input provenance to help the application distinguish real user input from synthetic input, as well as synthetic input from different sources [10, 142]; (2) "Z-fighting" (same space) mitigations for same space conflicts (such as slightly offsetting multiple objects [32] or higher resolution buffers [169]); and (3) the isolation of different application components (i.e., a "same origin policy" for AR applications).

However, applying these techniques for AR may not always be straightforward or sufficient, as we elaborate in the next subsection.

4.5.3 Knowledge of Defenses for 2D UIs Is Not Sufficient

Though it is impossible to fully predict all the challenges with future 3D interfaces, we observe several potential differences here.

For example, consider known defenses for clickjacking. From our assessment of 3D gaming environments, there *are* reasons for which developers might intentionally create invisible yet interactable objects. Thus, the anti-clickjacking invariant mentioned above may not be directly applicable in all immersive 3D environments and use cases. Further, unlike desktop and web environments, where one object is clearly "on top," the visibility of an object might be impacted by the position of the viewer, who could be moving around the objects, or the objects could be moving around the viewer. This "on top" nature becomes even more complicated if the same set of objects are being viewed by multiple people (different viewers of the same scene might see different objects on top). Certainly, computational methods could be used to determine which object is on top of each user, though platform designers must account for the different architecture of AR systems and the different roles of the physics and rendering engines. Even then, some invariants, such as minimum time of visibility before being clickable, may be incompatible with some use cases, like playing a game with fast-moving objects while simultaneously using another leisure application.

As another example, consider that our attacks depend on a threat model in which multiple application components — or multiple applications — are mutually distrusting. While current AR platforms do not (yet) support rich multi-application interactions (except with applications confined to 2D windows), we can anticipate that future platforms will evolve to fully support two (or more) immersive augmented reality applications at the same time: for example, walking directions interspersed with game content from another application. Securely managing the integration of content from multiple applications in the AR context, or even isolating content from different sources within the same application (like Site Isolation for iframes on the web [229]), will be a large technical and research challenge [179].

4.5.4 Potential Defensive Techniques

Work has already begun to emerge within the security research community that provides potential defensive directions against the issues we raise in this chapter. For example, Lee et al. [184] introduced AdCube, a defensive system to counter several WebXR attacks. The attacks they consider have some similarity (and some differences) with the ones we discuss here. The blind spot tracking attack from AdCube places the ad entity outside of the user's peripheral to increase the number of ad appearances, while the input forgery attack in this chapter maximizes the number of ad clicks through synthetic input. The cursor-jacking attacks from AdCube exploit the user's perception to hijack authentic clicks whereas our clickjacking attack utilizes the rendering-interaction inconsistency of virtual objects placed in the same space. Despite these differences, the AdCube idea of confining untrustworthy third-party libraries and preventing them from placing virtual objects could mitigate attacks like those we explore as well: specifically, attacks that rely on the ability to generate virtual content interleaved with the victim's content. Future researchers could build upon AdCube and our findings to develop a cross-platform AR defensive toolkit. More generally, future work should also explore other possible approaches that isolate UI components from multiple principals in future AR platforms.

4.5.5 Stepping Back

The preceding observations do not imply that the types of attacks we describe here are insurmountable. Rather, this discussion reflects our view of how this work might fit into the broader evolution of security and privacy for AR systems. At the highest level, and returning to Section 4.5.1, we believe that our first and most important contribution is the knowledge of *what* vulnerabilities might exist and how they presently manifest in different systems. With that knowledge, it is possible to defend systems by leveraging existing knowledge from 2D contexts (Section 4.5.2), by creating new knowledge (Section 4.5.3), and by extending existing defensive directions from AR contexts (Section 4.5.4). Toward the creation of new knowledge and new defenses, a critical step will be the assessment of what types of applications and use cases the designers intend to support and the management of tensions that arise when the exact feature exploited by an adversarial UI application is also needed by a desirable application.

4.6 Conclusion

We presented an empirical analysis of five current AR platforms, systematically investigating how they handle three UI security related properties that directly impact the output and interaction of virtual content in multi-principal environments: Same Space, Invisibility, and Synthetic Input. We demonstrated five proof-of-concept attacks — one implemented for each of our test platforms — that leverage different design choices in the context of these AR UI security properties. We found that these current AR platforms, including Apple's ARKit, Google's ARCore, Meta's Oculus, Microsoft's HoloLens, and WebXR, are all designed and implemented in ways that enable our AR UI attacks to succeed. Our findings lay the groundwork for future research and design work to consider and address AR UI security to mitigate the risks of such attacks, either through directly applying past lessons from 2D UI security or by developing new, AR-specific output isolation mechanisms that account for the unique challenges of 3D, immersive interfaces.

All code for our experiments and the video demonstrations are available online at `https://ar-sec.cs.washington.edu/ar_ui/` to support future researchers in extending our analyses to other and future AR platforms.

Chapter 5

Interaction: Perceptual Manipulation Attacks

This chapter considers user interaction with AR content, the third phase of the AR system data flow highlighted in Chapter 1. Specifically, we examine how users interact with and respond to potentially malicious AR content, and how these interactions can be manipulated to influence user behavior and decision-making. We define such interaction as *Perceptual Manipulation Attacks* (PMA). While current AR technology is sufficient to create PMA, little research has been done to understand how users perceive, interact with, and defend against such potential manipulations. To provide a foundation for understanding and addressing PMA in AR, we conducted an in-person study with 21 participants and analyzed both qualitative and quantitative results to understand user experiences during these malicious interactions. This research contributes to our understanding of how users' interactions with AR content can be exploited and provides insights for developing defensive mechanisms that protect users during their AR interactions. The work in this chapter, Exploring User Reactions and Mental Models Towards Perceptual Manipulation Attacks in Mixed Reality, was first published and presented at the 32nd USENIX Security Symposium in 2023 [119] [1]

5.1 Introduction

Mixed reality (MR) technologies — technologies that place virtual content in users' real-world environment — are poised to dramatically alter how people interact with the physical world, the digital world, and each other. Once inaccessible to the general public, MR devices

[1] Throughout this chapter, we primarily use the term "mixed reality", consistent with the terminology used in our 2023 USENIX publication. All references to MR can be interpreted as AR.

are becoming more available and affordable. Technologies like Microsoft's Hololens 2 [16], Meta's Oculus Quest 2 [25], Project Cambria [18], Snapchat's Spectacles [34], and Apple's incoming MR headset [56] are transforming previous visions for MR into reality.

Despite the benefit MR technologies could deliver, a growing body of research in the computer security and MR communities has looked at perceptual issues in MR [129, 175] and how users could be manipulated by content created by MR applications. One class of potential attacks, termed *Perceptual Manipulation Attacks* (PMA) by Tseng et al. [269], aims to manipulate the human multi-sensory perceptions of the physical world to influence users' decision-making, interaction, and even lead to physical harm through the presented MR output stimuli. Unlike attacks targeting vulnerable hardware or platforms, here the attack is impacting the perception and/or cognition of a person in an immersive way using the MR system.

While PMA also exist on traditional platforms (e.g., phishing, distracting pop-ups), the MR experience is fundamentally more immersive: for example, previous studies in gaming settings [101,220,279] suggested that MR might produce meaningfully different experiences, such as higher presence, more real and personal involvement, and higher affective responses.

PMA in MR are not only a theoretical threat; precursors are starting to manifest in practice. Although not adversarial in nature, recent research [164, 274] documented severe negative psychological impacts and physical injuries, including even death on users from using real-world MR applications (such as Pokémon Go), which indicates that even benignly-designed MR applications can unintentionally "hack" user perceptions and introduce critical risks. Prior work also demonstrates different types of PMA in a laboratory setting, such as obscuring important real-world content [180], creating audio indistinguishable from reality [17], alternating perceived haptic softness level [223, 224, 264], affecting users' gustatory sensations [211], and disrupting users physiologically [98].

Thus, while it is clear that PMA are possible, both with today's technologies and in the future, what is not known are the human experiences while undergoing such attacks, where "experiences" include both behaviors (e.g., actions) and thoughts (e.g., impressions, interpretations). As our community seeks to develop defenses against PMA in MR environments, we argue that it is essential to understand users' experiences with PMA. Such an understanding can form the basis of future risk assessments and defensive approaches.

Research Questions Motivated by the above, we formulate the following two key research questions:

1. *RQ1:* What physical or behavioral reactions and responses do users have during their interaction with perceptual manipulation attacks (PMA) in MR?

2. *RQ2:* What are user-reported reflections, reactions, and defensive strategies to PMA in MR during or shortly after they occur? For example: Do users rationalize their behaviors and responses to attacks? Do they perceive that they are under attack during the attack? How do users defend against such attacks?

Answering the above questions enables us to suggest key strategies for researchers and industry to reduce the harms of PMA in MR in the wild. These strategies include both preventative measures and approaches for resiliency and harm reduction if attacks manifest.

Methodology To answer RQ1 and RQ2, we develop a methodology with the following approach: we subject participants to PMA and (1) observe their physical and behavioral reactions and responses during the attacks and (2) listen to any thoughts they might express during the attacks and interview them after.

We highlight here several key elements of our methodology. First, because we would be subjecting participants to PMA in a laboratory setting, it was essential to design a safe testing procedure, including but not limited to receiving IRB approval. Second, it was essential to create an experimental apparatus that (A) exposed participants to programmatic MR content (virtual content placed in the physical world), (B) allowed that content to sometimes (but not always) be adversarial, and (C) control the physical environment such that the experiments would be repeatable. Elaborating on (C), it was essential for the physical environment and the MR content (including adversarial content) to be synchronized.

To meet our experimental objectives, we designed an experimental harness with the following properties: The physical world environment included a computer monitor and mouse. The participants wore a mixed-reality headset. When the participants looked at the real-world computer monitor while wearing the headset, they saw that monitor. Our

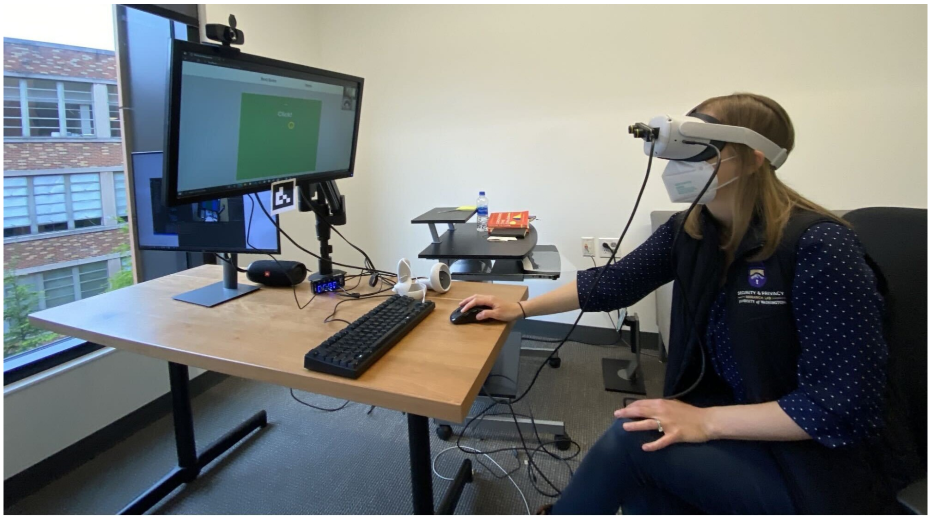


Figure 5.1: Researcher testing our experimental harness.

experimental harness advanced both the content displayed on the computer monitor and the virtual content displayed within the mixed-reality headset. See Figure 5.1.

We experimented with three controlled scenarios in which we asked participants to do tasks where their performance was measured using three cognitive metrics: reaction speed, sustained attention, and focus. Our study was a deception study: participants did not initially know that we would be subjecting them to PMA. Within these scenarios, we then subjected participants to three different PMA, each with the intention to attack different perceptions: visual, auditory, and situational awareness.

Contributions In summary, our contributions include:

1. *End User Behavioral Reactions:* We focus on end users' *experiences* when they encounter adversarial MR content. Our results suggest perceptual manipulation attacks (PMA) successfully disrupt user performance and evoke the adversarially intended reaction from users — as well as secondary effects, such as slowing down on non-attack settings or amplifying subsequent PMA impacts.

2. *End User Self-Reported Reflections:* Through follow-up interviews, we provide rich qualitative insights about how people assess, reason about, and defend against PMA

in MR in practice.

3. *MR PMA Experimental Harness:* We implemented a harness to capture the impact of PMA in MR on end users. We have publicly released our experimental harness[2] to facilitate open science.

4. *Foundation for Future Defenses:* Stepping back and reflecting on our studies of PMA with real users, we provide suggestions for researchers and industry to build the next generation of MR defenses and strategies to reduce the harms of attacks on the user-MR interaction in the wild.

5.2 Methodology

To empirically explore user reactions to PMA, we designed an in-lab study in a user study room. As discussed in Section 5.1, ours was a deception study in which participants were told that we were evaluating the impact of wearing a MR headset while conducting tasks. We mounted PMA on participants as they performed these tasks, interviewed them, and then debriefed them after completing all tasks. Each study lasted for around 60 minutes.

5.2.1 Study Procedure

Warm-up Phase We started by having participants familiarize themselves with the MR world. We helped participants, by non-contact demonstration, adjust the headset and tune the interpupillary distance to minimize their discomfort level.

Experiment Phase Participants completed specified tasks in a benchmark setting and under the influence of PMA. We intentionally did not mention security to them until a debriefing at the end of the study, to minimize priming participants to the possibility of attacks. Then, participants were instructed to complete three tasks (see Section 5.3 for details). Most tasks consisted of three rounds: (1) an initial *task-training* round without the headset, followed by (2) a *benchmark* round where they completed the task while wearing the headset (though no MR content was shown), followed by (3) a *withAttack* round during which we mounted PMA while they completed the task. We measured and recorded participants'

[2] `https://github.com/UWCSESecurityLab/MR-PMA-Harness`

performance on the tasks under each condition. After the benchmark round, we did not give instructions about any virtual content in order to observe the participants' organic mental models; the task goal remained the same after the benchmark round.

We encouraged participants to talk aloud throughout the experiment to capture in-the-moment reactions. For all experiments, researchers could see participants' view through the headset streamed to a separate desktop computer display. With participant consent, we recorded the entire study, including this captured first-person view; we present screenshots of this view in figures throughout this chapter.

We emphasize that our *benchmark* round includes *no* MR content at all, rather than attempting to compare with a condition containing benign MR output. There are many ways that a benign or intending-to-be helpful MR application could be designed, and a benign application might accidentally influence user perception as well (as in Pokémon Go). Thus, in our work, we focus on exploring user reactions to an *intentionally* manipulative MR application, compared to no MR content at all; future experiments could explore the spectrum of reactions user might have to non-adversarial MR content as well.

Post-Task Interview Phase We conducted an in-depth, qualitative interview with participants. We asked questions about their experiences doing the tasks, beginning with more open-ended questions to avoid priming them. We then asked follow-up questions if participants discussed the adversarial MR outputs, and we eventually debriefed participants and disclosed the purpose of the study. We continued to talk to participants for about 15 minutes about their reflections on the study in particular and PMA in general (see Appendix B.2). We do not include any information from these post-debrief conversations in the results because participants had been primed about PMA at this point, but we believe that this post-study session helped participants process and understand the study. Before participants left the room, we reminded them that they could reach out to us if they felt discomfort or experienced any negative impact from the study. No one mentioned, nor did we observe, any concern or discomfort. We include our interview script as well as our debrief script in Appendix B.2.

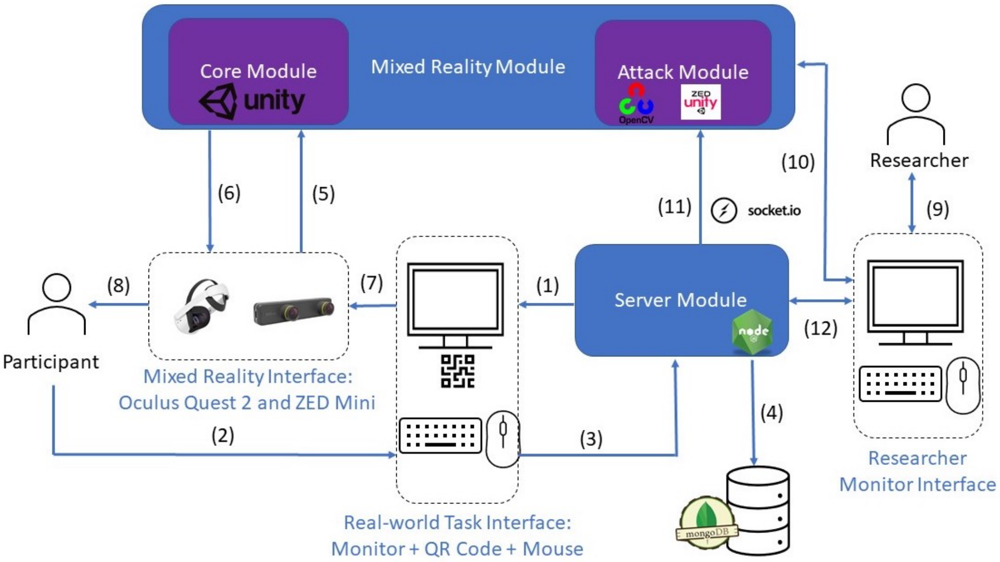


Figure 5.2: Diagram of our experimental harness.

5.2.2 Experimental Harness

Figure 5.2 presents an overview of our experimental harness. The computer and mouse correspond to the real-world tasks presented to users. Affixed to the monitor is a QR code, which enables the localization of the PMA output. In the following, numbers refer to the arrows in Figure 5.2.

The Server Module controls the benchmark experiment. It (1) determines what content to display on the monitor (content corresponding to the real-world task that the user is performing). It also receives user input (2), in the form of mouse clicks (3), and saves user's performance in our MongoDB database (4). The Server Module is implemented with Nodejs. The Server Module, combined with the monitor and the mouse, is the entirety of the participant interface during the *task-training* round of the experimental phase.

The Mixed Reality Interface uses the Oculus Quest 2 head-mounted display with a ZED Mini camera. We attached the ZED Mini to the Oculus headset with adhesive tape. The Unity-based Mixed Reality Module renders the ZED Mini camera stereo view (5) in the Oculus (6), and outputs it with the experiment (7) to user (8) and researcher (9). This part of the Mixed Reality Module, along with the Server Module, constitute the entirety of the participant interface during the *benchmark* round of the experimental phase.

During the *withAttack* round of each scenario, the researchers start the Mixed Reality Module, as they did during the *benchmark* round (9 and 10). The Server Module sends the trigger via Socket.io to the Attack Module (11). The Attack Module leverages the OpenCV library to detect the QR code (via (5)) on the Task Interface. It calculates the adversarial output's placement and generates it. The MR module then mixes the adversarial MR output with the ZED Mini input (5) to render visual and auditory output (6) and displays it to user (8), and to researchers (10 and 9).

5.2.3 Recruitment and Screening

Due to COVID-19, university safety protocols, as well as the nature of MR requiring participants to be in person, we recruited participants with access to school buildings via department Slack and personal contacts. A study session took around 80 minutes including sanitizing the equipment.

We advertised the study goal as evaluating the impact of wearing a MR headset while conducting a primary task: we did not advertise it as a MR attack study to minimize priming participants and potentially affecting their behaviors.

Candidates completed a screening survey (Appendix B.1), indicating any previous MR/VR experience, demographics, and contact information (name and email). We did not consider for the full study anyone who indicated dizziness or nausea during previous MR/VR use. Ultimately, we recruited 21 participants (10 men, 11 women; age: M = 22.12, SD = 4.31). Overall, most participants (around 85%) had "some" experience with MR/VR, two participants were regular users, and one participant had never tried MR/VR. Our participants' ages ranged from 20 to 28; the majority are in the technology industry.

Participants who completed the full study were compensated with a $30 Amazon gift card. The compensation was based approximately on the hourly minimum wage in our area ($15) and accounting for additional time that participants might need to commute to our lab. The compensation method and amount were approved as part of our full IRB protocol. We note that compensation may influence a participant's decision to participate in the study in the first place.

5.2.4 Ethical Considerations

Our study, reviewed and approved by our university's IRB, raised several ethical considerations. First, it was designed as a deception study: we did not initially disclose its true goals so we could avoid biasing reactions. We designed the study to minimize any potential risks beyond task performance impairments: participants completed the tasks while seated, and the tasks did not involve moving around the physical space while wearing the MR headset. We informed participants that the MR headset could cause discomfort (such as nausea or dizziness) and that they could stop at any time during the study with no loss of promised compensation. (No participants discontinued the study before completion, and none expressed or appeared to experience discomfort during the study.) Since attacks could affect participant performance on tasks, we framed the study as an evaluation of the MR technology rather than participant performance. We debriefed all participants about the true nature of the study at the end.

Additionally, we did not ask participants to reveal sensitive information. We sent the consent form to participants days before the study and asked for their physical signature when they arrived at the user study room to participate in the study and to be video recorded. We stored all recordings on password-protected drives and removed any personally identifying information from notes and transcripts.

5.2.5 Data Analysis

We analyzed our data using both quantitative and qualitative methods. In the experiment phase, we evaluated participants' performance for each task under various conditions using the metrics we describe in Section 5.3. For the qualitative data, we transcribed the interview audio using Rev [30]. All four researchers independently developed preliminary codebooks based on three interviews. These researchers then iteratively resolved disagreements and developed a full version of the codebook collaboratively. The first author applied the codebook to all interview transcripts. All researchers discussed when new codes emerged, and resolved any further disagreements. The first author kept the codebook updated, and applied the final codebook to all interview transcripts. No new theme was found. We conducted thematic analyses from a broad family of methods [109, 202], combining a deductive approach (applying a security threat modeling framework to our interview data to identify sub-themes related to attack attribution and defensive strategy) and an inductive approach (generating additional themes/codes from the interview data). We provide the full codebook in Appendix B.3.

5.2.6 Limitations

First, though we report experimental data for participants' performance, we do not aim to make causal or generalizable claims. We did not conduct a large-scale randomized trial with many participants and, for example, our experiments do not account for possible ordering effects. While we asked participants to randomly select either Scenario A or Scenario B to start (see Section 5.3), Scenario C always came last because it most clearly gave away the adversarial nature of the study. Our work is the first step towards deeply understanding user perceptions and defensive approaches to PMAs, and lays a foundation for future work to conduct larger-scale studies to test ordering effects or the generality of our findings.

Second, our participant sample has the following limitations: Most of our participants were predominantly young adults with STEM education backgrounds. Due to these limitations, the findings of our study should not be generalized. Future work could explore a broader population or more directly focus on specific populations.

Third, because participants were in a lab setting and in some cases previously knew the researchers, they may have either trusted the MR content more (assuming good intentions of the researchers) or less (if participants happened to know that the researchers work in a security-focused group). To mitigate these potential impacts, we designed the study to avoid priming participants about security. We found that participants *were* impacted by the adversarial MR content in our experiments, even in cases where they assumed good intentions (e.g., attributing attacks to glitches) or knew the researchers. Likewise, we expect that users in real MR scenarios will bring a variety of preconceptions to the situation.

Fourth, we did not ask participants about color blindness during our screening (though we should have). No participants mentioned color-blindness during the tasks, and their task performance suggests that they could see the colors we used.

Finally, we conducted our experiment with one particulMR hardware setup. We chose state-of-the-art hardware and software, but our results may not generalize to other setups or future technologies. Despite the imperfections of the MR setup (e.g., virtual content did not necessarily believably blend into the real world), participants were impacted by our attacks.

5.3 Scenarios and Attacks

Our high-level goal is to explore how users are impacted by and respond to different PMA in MR. For this investigation, we thus design and develop controllable, repeatable in-lab experiments that were modeled after both known psychological experiments and prior concern about MR in real environments. In choosing our attacks, we focus on exploring different types of perceptions to attack: visual, auditory, and situational awareness. Note that the attacks we present are prototypes; future attackers might mount far more sophisticated and powerful attacks with the help of next generation MR headsets and external devices such as eye tracking hardware [38] or electromyography wearable wristbands [11].

The following three subsections describe our three scenario case studies. We summarize all of our attacks in Table 5.1.

5.3.1 Scenario A: Reaction Time Task, Visual Attacks

Reaction Time Task Reaction time is the duration of the interval between presentation of a stimulus and response to the stimulus. There are concerns about how adversarial MR content might manipulate user visual perception to impact those reaction times rooted in classical psychology literature [148, 219]. For example, for people using MR while driving, attackers could overlay virtual objects on real traffic lights, causing the driver to react slowly or to misinterpret those lights [145, 146].

To evaluate the effect of adversarial MR output on reaction time, we created an experiment in which participants were asked to respond quickly to a stimulus, modeled after an existing cognitive study game [33]. The real-world task consisted of red and green boxes shown on a computer screen: participants were asked to wait while a red box was visible, and click a mouse button as quickly as possible after a green box appeared (see Figure 5.3). Success metrics for participants on this task are (1) clicking correctly, i.e., only when a green box appears, and (2) fast reaction time in clicking when a green box appears. As introduced

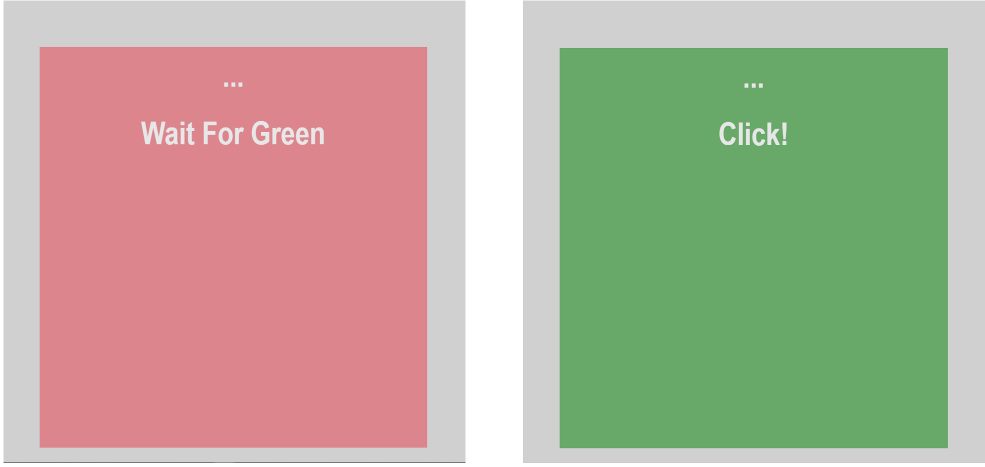
Table 5.1: Summary of all attacks.

Name	**Attack Description**	**Attack Goal**	**Perception**
Reaction–FalseGreen	Overlay a virtual green object while the real-world box is still red.	Make participants click incorrectly Slow their reaction significantly	Visual
Reaction–DoubleGreen	Overlay a virtual green object at the time the real-world box turns green.	Slow their reaction significantly	Visual
Reaction–DoubleRed	Overlay a virtual red object while the real-world box is still red.	Make participants click incorrectly Slow their reaction significantly	Visual
Reaction–FalseRed	Overlay a virtual red object at the time the real-world box turns green.	Slow their reaction significantly	Visual
Sustained Attention–NotificationSound	Play a sequence of notification sounds during level 2.	Make participants click incorrectly	Auditory
Sustained Attention–RingtoneSound	Play a sequence of ringtone sounds during level 4.	Make participants click incorrectly	Auditory
Focus–CountingCard	Display a sequence of playing cards.	Prevent participants from noticing important context	Situational Awareness

in Section 5.2.1, this experiment consists of three rounds: first, the *task-training* round without an MR headset; second, the *benchmark* round with the MR headset; third, the *withAttack* round with the headset. Each round consists of eight levels, i.e., eight instances of a green box appearing and the user needing to click.

Color Attacks In the context of this task, we craft visual manipulation attacks named "Color Attacks" with two types of goals: (1) **Induce an incorrect reaction** and (2) **Delay a correct reaction**. We implement a total of four attacks in the *withAttack* round. These attacks are described in Figure 5.4, and the participant view is shown in Figure 5.5.

Two of our attacks involve attempting to fool the user about the color of the real-world box, by overlaying a virtual box with the wrong color. (Note that we slightly misaligned the virtual box and made it partially transparent, so that participants could still see the actual real-world stimulus.)



(a) Waiting Page

(b) Reaction Page

Figure 5.3: Participant view of the real-world Reaction Task.

1. **Reaction-FalseGreen** attack: Overlay a virtual green object while the actual box is still red. This attack aims to cause participants to click incorrectly.

2. **Reaction-FalseRed** attack: Overlay a virtual red object when the actual box turns green. This attack aims to mislead the participants and delay their reaction or cause them to fail to click entirely.

We crafted two other attacks, in which the virtual box color matches that of the real-world box. These attacks allow us to investigate the impact of PMA that may be startling or distracting, but is not directly misleading.

3. **Reaction-DoubleGreen** attack: Overlay a virtual green object when the actual green box appears. This attack may cause users to delay or avoid clicking as they focus on interpreting the adversarial content.

4. **Reaction-DoubleRed** attack: Overlay a virtual red object while the actual box is still red. This attack may induce users to click incorrectly, e.g., if they attempt to overcompensate for the adversarial content.

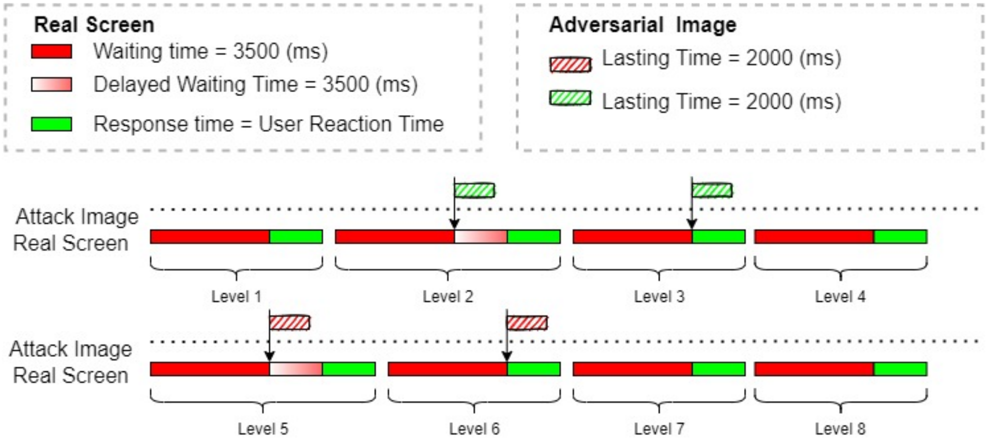


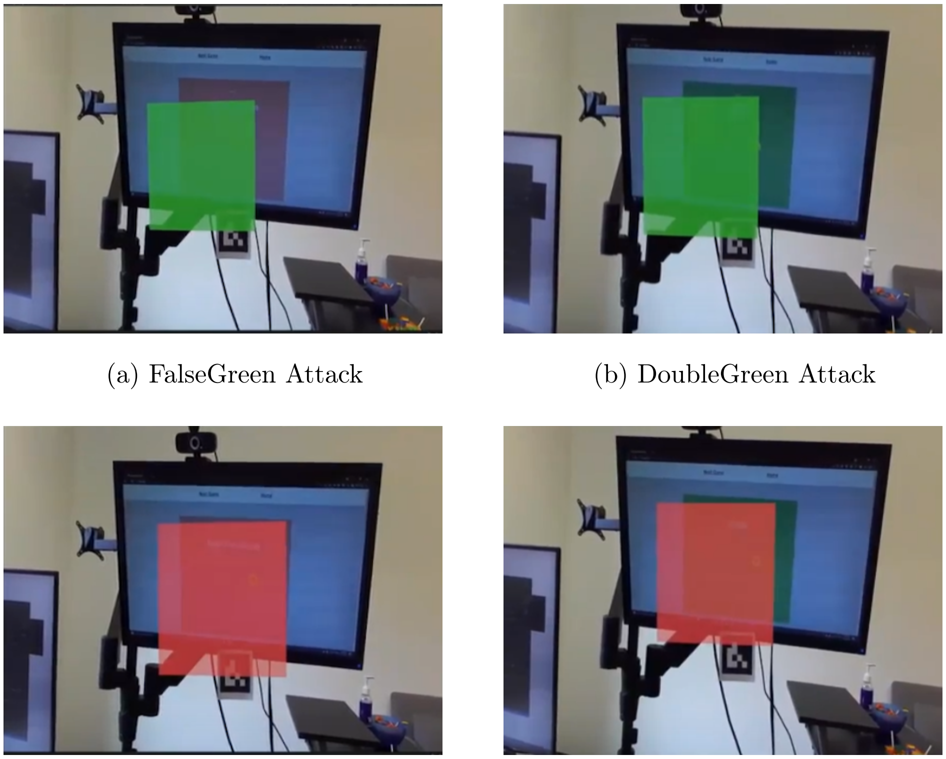
Figure 5.4: Timeline of attacks on the Reaction Task during the withAttack round (round 3).

We expose participants to these attacks in the following order (see Figure 5.4): the FalseGreen attack on level 2, the DoubleGreen attack on level 3, the DoubleRed attack on level 5, and the FalseRed attack on level 6. For this exploratory study, we did not randomize the order in which these attacks were presented (which would have required an infeasibly large number of participants and was not our goal).

If participants click incorrectly on level 2 (Figure 5.5(a), FalseGreen attack) or level 5 (Figure 5.5(c), DoubleRed attack), when the actual box is still red, we conclude the attack is successful in manipulating user visual perception and inducing an incorrect reaction. If the participants click correctly on level 3 (Figure 5.5(b), DoubleGreen attack) or level 6 (Figure 5.5(d), FalseRed attack), we compare their reaction time with group's average performance on the MR benchmark round. If the reaction time on valid clicks under these two attacks falls outside two standard deviations of the group's average performance, we can conclude that the attack is successful in manipulating user visual perception and slowing reaction time.

5.3.2 Scenario B: Sustained Attention Task, Auditory PMA

Sustained Attention Task Sustained attention is the ability to concentrate on an activity. While naturally occurring stimuli may also disrupt users' controlled mental processing



(a) FalseGreen Attack

(b) DoubleGreen Attack

(c) DoubleRed Attack

(d) FalseRed Attack

Figure 5.5: Participant views of attacks in the Reaction Task — The floated boxes are generated by Color Attacks and blended into the user's view to *intentionally* mislead them.

and lead to focused-attention deficit [245], we are especially interested in the capability of immersive MR audio to intentionally distract a user from another task. To evaluate the effect of auditory PMA on sustained attention, we create a scenario in which participants are asked to memorize a sequence of real-world stimuli, modeled after an existing cognitive study game [31]. For the purposes of our experiment, that real-world stimulus consists of increasingly long sequences of flashing buttons. The sequences do not build on each other but are newly random at each length. Participants are asked to memorize the sequence, and then click each button at the correct location, as shown in Figure 5.6. The success metric for participants on this task is to recall as many sequences correctly as possible. This experiment also consists of three rounds (*task-training*, *benchmark*, and *withAttack* rounds).

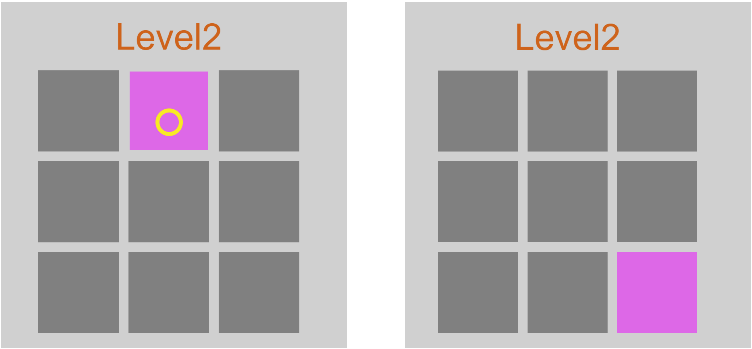


Figure 5.6: Two-item sequence on level 2 of the Attention Task.

Auditory Attacks Towards our goal of exploring different types of PMA, in this scenario we consider audio instead of visual adversarial content. Given the immersive nature of MR audio, participants might treat it as a real-world stimulus (e.g., think an actual phone is ringing). MR attackers (unlike other distractions in the user's environment) can also precisely and stealthily inject audio when participants are under high cognitive load based on MR device sensor data. Given the above task, we crafted two auditory attacks with one goal: (1) **Distract users at a specific point in the task**.

1. **Sustained Attention-NotificationSound** attack: Play a sequence of notification sounds during all of level 2.

2. **Sustained Attention-RingtoneSound** attack: Play a sequence of ring-tone sounds during all of level 4.

We use the *Audio Spatializer SDK* to create an immersive 3D sound effect on both audio data. The sound is played from the Oculus built-in speakers.

Figure 5.7 demonstrates the timeline of the *withAttack* round. If more participants fail at recalling the memorized sequence on level 2 or level 4 during *withAttack* round than *benchmark* round, we conclude that the attack is successful at affecting auditory perception and distracting participants from the primary task at the chosen times.

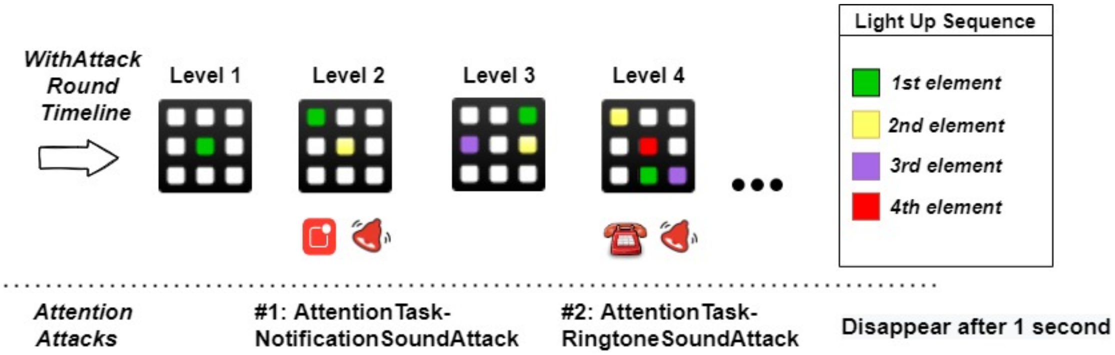


Figure 5.7: Timeline of the *withAttack* round (round 3) of the Attention Task. In the actual task, all elements lit up with the same color; here we use different colors to illustrate time.

5.3.3 Scenario C: Focus Task, Situational Awareness Attack

Focus Task Focus is defined here as the ability to direct your attention to a particular idea [226]. As people are exploring MR usage in critical operation settings such as surgery [239], construction [207], and driving [145], researchers have raised concerns about users focusing on MR content to the detriment of other stimulus, and thus fail to notice fully-visible yet unexpected important notices. Such distraction due to MR can lead to serious danger such as falling down cliffs [39] or wandering into the street without noticing incoming vehicles [164].

To evaluate the effect of situational awareness attack to manipulate user focus, we create a scenario, parallel with the classic "Gorilla experiment" [247], in which participants' task is to notice real-world context on the monitor (though they do not know that this is their task when they begin the scenario). The text on the monitor appears while the participants are doing a decoy activity (i.e., the MR attack) with virtual content in the MR headset. The monitor text says: “If you see this message, raise your hand immediately”. This real-world content becomes increasingly visible in four phases, described in the caption of Figure 5.8. Participants' metric for success is: notice the change in the real world as quickly as possible.

We emphasize that this task is different from the previous tasks, in that the *actual* task and success metrics are not given to participants, but rather they are given an MR activity that turns out to be the attack.

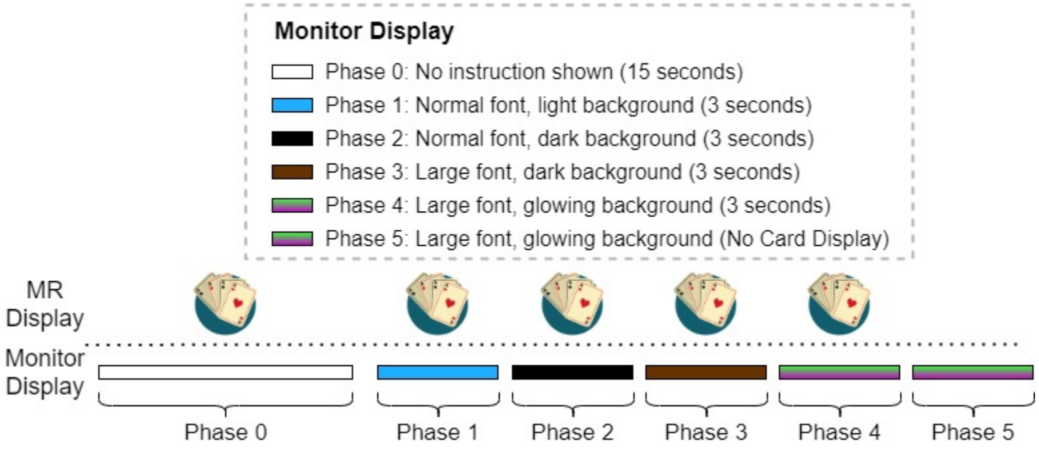
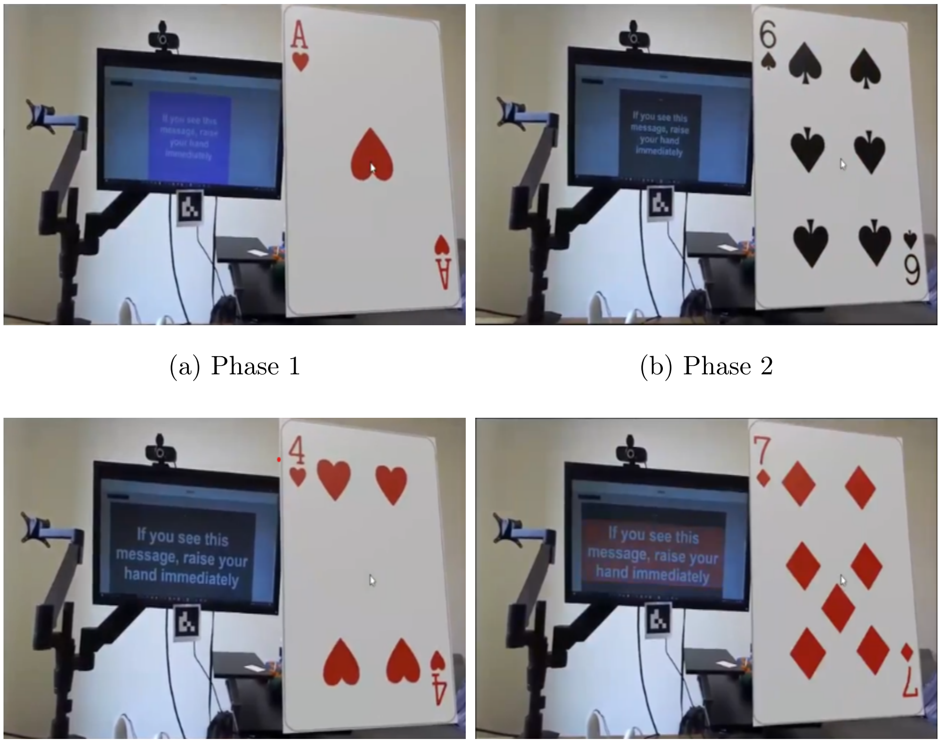

Monitor Display
Phase 0: No instruction shown (15 seconds)
Phase 1: Normal font, light background (3 seconds)
Phase 2: Normal font, dark background (3 seconds)
Phase 3: Large font, dark background (3 seconds)
Phase 4: Large font, glowing background (3 seconds)
Phase 5: Large font, glowing background (No Card Display)
MR Display
Monitor Display
Phase 0
Phase 1
Phase 2
Phase 3
Phase 4
Phase 5

Figure 5.8: Timeline of the Focus Task and corresponding Situational Awareness Attack. In phase 1, the screen changes from no text displayed to our instruction. In phase 2, the background color changed from blue to black. In phase 3, the font size and the window width increased to maximum, and in phase 4, the background start to blink with different colors.

Situational Awareness Attack Given the above task, we crafted one PMA targeting situational awareness, which refers to the perception of what is around us [132]. This attack has one goal: (1) **Prevent participants from noticing real-world instructions**. A more successful attack will keep users from noticing the real-world changes in the Focus Task for longer.

1. **Focus-CardCounting** attack: Display a sequence of playing cards to prevent participants from perceiving the real-world target.

Unlike previous setups, this experiment only has one round, as a non-attack round would give away the nature of the task. The attack consists of 28 virtual playing cards, displayed one at a time, for one second each. Participants are asked to count the number of red cards that appeared. Starting after 15 seconds of card counting, we showed the instruction corresponding to the actual Focus Task on the computer screen. Figure 5.8 shows the attack and task timeline, and Figure 5.9 shows the participant's view. To explore user reactions



(a) Phase 1

(b) Phase 2

(c) Phase 3

(d) Phase 4

Figure 5.9: Participant views of the Situational Awareness Attack during the Focus Task. The floating cards are generated by the Situational Awareness Attack. Four increasingly visible phases of the real-world notification are shown on the monitor. Omitted from this figure are Phase 0 (no text on the screen) and Phase 5 (no cards in the foreground).

under different degrees of awareness conditions, starting from the 6th participant, we advised them at the beginning of this task to stay aware of their real-world surroundings.

5.4 Results: Behavioral Reactions

We begin with an analysis of our experimental data to study users' reactions when encountering MR PMA (RQ1).

Impacts of Visual Attacks on Reaction Task We use two metrics to evaluate visual PMA effectiveness: *(1) invalid click rate*, which measures the number of participants out of 21 who clicked incorrectly (before the real-world box shown on the computer monitor

Table 5.2: Experimental results for the Reaction Task, comparing different attacks, non-attack, and benchmark conditions. * In the benchmark round, 5% of clicks are delayed by definition (since we defined delayed clicks as those outside of 95% of the benchmark data). We give percentages for ease of interpretation, not to imply generalizability to a broader population.

Color Attacks	Metric (1): Invalid Click Rate			Metric (2): Delayed Click Rate			Combined
	Invalid	Total	Percentage	Delayed	Valid	Percentage	All
FalseGreen (R3)	15	21	71.42%	5	6	83.33%	95.24% (20/21)
DoubleGreen (R3)	0	21	0%	14	21	66.67%	66.67% (14/21)
DoubleRed (R3)	6	21	28.57%	13	15	86.66%	90.48% (19/21)
FalseRed (R3)	0	21	0%	16	21	76.19%	76.19% (16/21)
No Attack (R3)	0	84	0%	33	84	39.29%	39.29% (33/84)
Benchmark (R2)	0	168	0%	10*	168	5.95%*	5.95% (10/168)*

turns green) for a given attack, and *(2) delayed click rate*, which measures the number of participants out of 21 who took a significantly longer time to click on the real-world stimulus. Here, we define "significantly" as outside two standard deviations of the group's average performance in the previous *benchmark* round. Table 5.2 summarizes the attack efficacy of all four Color Attacks.

For the DoubleGreen and FalseRed attacks, recall that the real-world box turned green at the time of the attack, so Metric (1) (invalid click rate) does not apply (i.e., all clicks were valid); thus, Metric (1) is reported only for the FalseGreen and DoubleRed attacks. For the FalseGreen and DoubleRed attacks, some participants avoided the initial attack and *did* manage a valid click after the real-world box later turned green; in those cases, we evaluated their performance under Metric (2) (delayed clicks rate of valid clicks).

Figure 5.10 details individual participants' performance under each attack. The top of this figure shows reaction time performance for valid clicks under each attack, compared with the participant's benchmark performance. The bottom part of the figure counts the participant's invalid clicks.

Participants were susceptible to manipulative MR content that tried to evoke the target re-

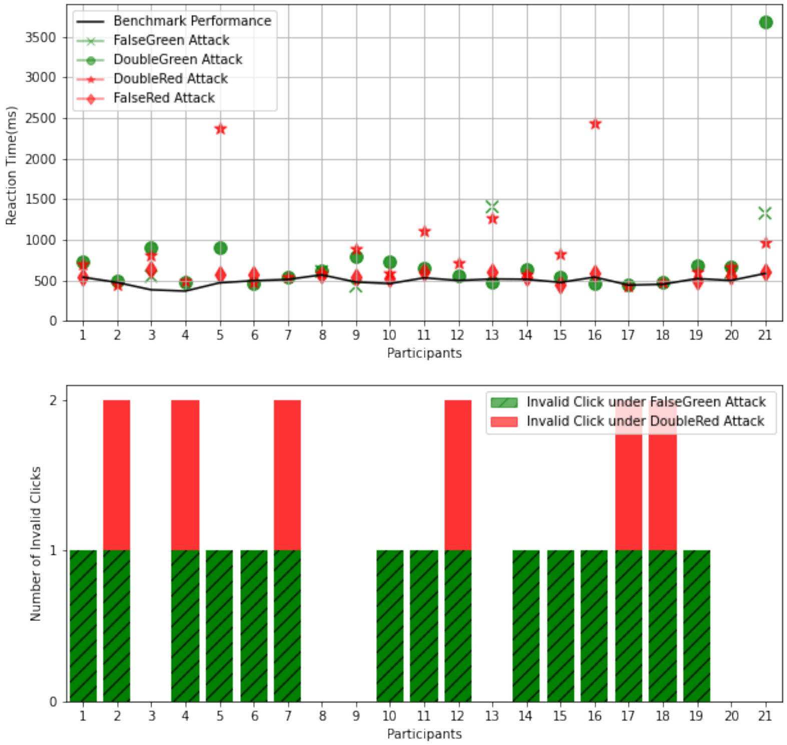


Figure 5.10: Per-participant performance on different Reaction Task attacks. The top graph captures every valid click, and the bottom bar chart captures the number of invalid clicks. For visual clarity we connect the benchmark dots with a line, not to imply points between participants on the line.

action (i.e., clicks). As the first row in Table 5.2 shows, almost all of our participants were affected by the FalseGreen attack. That is, most participants were fooled by the adversarial virtual green box and invalidly clicked in response (15/21). While our study allowed us to observe this only in our specific experimental setting, it provides a proof-of-concept demonstration, suggesting one type of user impact as a result of perceptual manipulation. This observation allows us to hypothesize that this finding would generalize to other settings where people perform tasks requiring quick reaction times while viewing MR content.

Participant reactions were slowed by manipulative MR content. As the Metric (2) (Delayed

Click Rate) column in Table 5.2 shows, all four of our attacks delayed participants' reaction times on valid clicks. We can consider two cases here: first, attacks that occurred when the real-world box turned green, i.e., the DoubleGreen and FalseRed attacks. In both cases, the attack caused significant delays in participants responding to the real-world green box (14/21 and 16/21 participants respectively). In the second case, with the FalseGreen and DoubleRed attacks, a click at the initial time of the attack would have been invalid (a successful attack per Metric (1)); some participants avoided clicking falsely then, and made it to the point in time when the real-world box turned green (refer to the timeline of the attacks in Figure 5.4). However, even in these cases, participant responses were often delayed (5/6 and 13/15 participants respectively).

We observed a few cases of extremely delayed responses — in particular, a few participants (P5, P16, P21) took over 2000 ms to click on some attacks (see Figure 5.10). Because the manipulative MR content was programmed to disappear after two seconds, this means that these participants waited until after the virtual box had disappeared to click. We cannot determine from our experimental results *why* participants were slowed by these attacks: whether they were distracted or confused by the manipulative MR content, whether they were attempting to avoid manipulative content, or whether they believed it was real-world content. We return to these questions in our qualitative analysis in Section 5.5 later.

Participant reactions were triggered by non-target manipulative MR content.

We saw above that in the FalseGreen attack, participants were induced to click even though the real-world box was not green. As the third row in Table 5.2 shows, we also find that some (6/21) participants clicked invalidly under the DoubleRed attack, where the real-world box was also not green — even though the manipulative MR content was red. By contrast, in the *benchmark* round (with the MR headset but without any attack) in Table 5.2, we see that no participant *ever* clicked while the real-world box was red. In this case, we hypothesize that participants were induced to click not because they thought the manipulative MR content was real, but because they were surprised or distracted by it, or because (having encountered some manipulative content already) they tried incorrectly to compensate for it. We again return to these questions in our qualitative analysis in

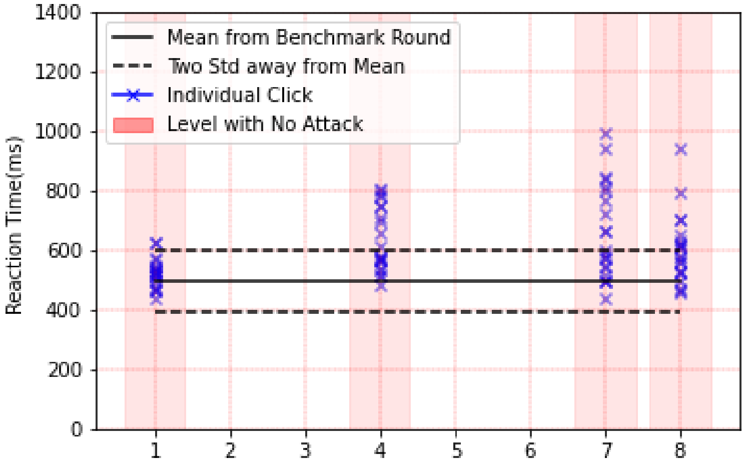


Figure 5.11: For the Reaction Task, comparing participants' performance on *non-attack* levels in the *withAttack* round with the group's average performance in the *benchmark* round.

Section 5.5 later.

Secondary impacts: reduced reaction time performance in non-attack settings. We have thus far discussed only *direct* impacts of the Color Attacks. However, we also observed an *indirect/secondary* impact. Specifically, when no attack occurred, participants still took a significantly longer time to click *after* having encountered at least one attack. Figure 5.11 summarizes the performance on each non-attack level during the *withAttack* round and compares it with the group's average performance during the *benchmark* round. At level 1, no attack had yet been encountered. In level 4, when participants just experienced FalseGreen and DoubleGreen attacks, we noticed a significant increase in average reaction times for nearly half of the participants. In level 7, after the FalseRed and DoubleRed attacks, we notice similar numbers of impacted participants, but with a more scattered distribution.

This result suggests that even when attacks do not appear, past attacks can still impact participants' reaction process.

In our setting, we saw participants become more cautious and slow down after attacks manifested, and they became even more cautious after experiencing different types of attacks.

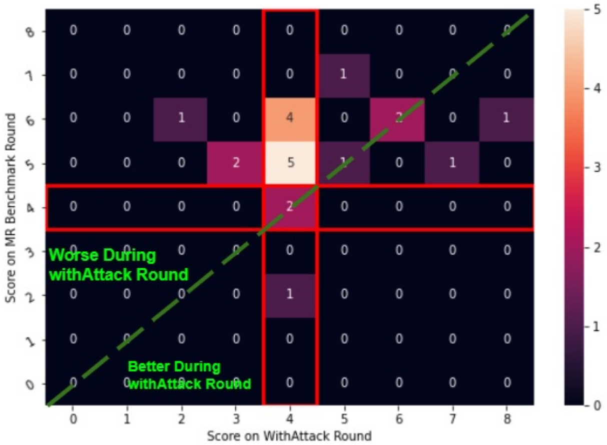


Figure 5.12: Results from the Attention Task. The x-axis shows scores during the *withAttack* round, and the y-axis shows scores during the benchmark round. Each box contains the number of participants who received that combination of scores. The red lines highlight that most people achieve scores of 5 or 6 in the benchmark round, while most people only reach 4 in the *withAttack* round.

Impacts of Auditory PMA on Sustained Attention Task We use one metric to evaluate audio attack effectiveness, i.e., *failure rate*, which measures the number of participants out of 21 who failed at correctly recalling the provided sequence on a level when an audio attack was played. Recall that we experimented with two audio attacks: the NotificationSound at level 2, and the RingtoneSound at level 4.

Manipulative MR audio content impacted participants' performance on the Sustained Attention task.

We find that the NotificationSound attack on level 4 impacts many participants. The heatmap in Figure 5.12 shows the performance of each participant on the *benchmark* round and the *withAttack* round. Each cell in the heatmap represents the number of participants who finished at level [y] on the MR benchmark round, and at level [x] on the *withAttack* round. Overall, we find that significantly more participants (12) failed on level 4 in the *withAttack* round compared with the previous *benchmark* round (2). As above, we cannot say from our experimental results why this attack was effective, but we provide participants'

Table 5.3: Experimental results for the Focus Task under the Card Attack, with original and updated instructions (where participants were told to pay attention to real-world context).

	Original Instruction		Updated Instruction	
	Raise	Total	Raise	Total
Phase 1	1	5	1	16
Phase 2	0	5	4	16
Phase 3	1	5	0	16
Phase 4	0	5	1	16
Phase 5	3	5	10	16

self-described reflections in Section 5.5.

Participants were resilient to auditory PMA under some conditions. While the NotificationSound attack on level 4 was effective, we found that the RingtoneSound attack was much less effective. Referring again to Figure 5.12, we see that most participants progressed past level 2 and the RingtoneSound attack during the *withAttack* round. We suspect (and some participants mentioned) that the task was sufficiently simple at earlier levels.

Impacts of Card Attack on Focus Task We use one metric to evaluate the Card Attack's effectiveness: *phase number* (1-5), which measures when participants saw and reacted to instructions in the real world (i.e., when they raised their hands). Higher phase numbers mean that participants reacted later, that is, the attack was more effective.

Manipulative MR content prevented participants from reacting to the real-world instruction.

Of our 21 participants, only two immediately noticed the real-world instruction and raised their hands (phase 1). In contrast, 13 participants reacted to the instruction only *after* the Card Attack was completely gone (phase 5). Despite the increasing visibility of the real-world instruction, only six participants noticed it during phases 2-4 (see Table 5.3).

Manipulative MR content still distracts participants even when instructed otherwise.

Of the 16 participants to whom we gave updated instructions to pay attention to the real world during the scenario, only six reacted before phase 5. This suggests the unexpectedness of manipulative MR content, while slightly remedied, can still leave many participants vulnerable.

5.5 Results: User-Reported Reflections

From our observations of participants performing tasks and for our subsequent interview, we identified several points during our evaluation when significant human-AR interactions or tensions arose about which we gathered participant reflections (RQ2): (1) reactions to attacks, (2) mitigating the onset of attacks, and (3) making sense of the attacks. We organize our results around these points with mappings to the corresponding scenarios and attacks.

We lost the transcript for P3 because we failed to save the Zoom recording, but we did capture P3's task performance data. Here we thus report qualitative results only for 20 participants. We slightly edited some quotes for readability.

(1) Self-Reported Impacts of PMA In Section 5.4, we observed attack impacts by comparing participant performance on several key metrics. Here, we turn to participants' self-reported reactions, asking them to walk us through what went through their mind when attacks occurred, helping us better understand why attacks may have been successful (or not).

Attack impact: Not knowing how to proceed . During the Reaction Time Task and Sustained Attention Task when participants first experienced PMA, many (9 of 20) were not sure what it was or how to respond. For example, P16 justified her delayed response time (over 2000 ms) when she first encountered the FalseGreen attack:

> *Because like I knew that I was supposed to follow like the color, like when it turns green on the screen. But like if it turns green outside of the screen, I was like, should I follow that? Or should I keep following the screen? (P16)*

Participants also described trying to make sense of the FalseRed attack:

> *If there is a red screen coming in front of your eyes when there is actually a green screen in the background, that's more conflicting than what I would expect it to be. (P7)*

Attack impact: Inability to focus on the primary task. Another commonly mentioned (9 of 20) impact from the Reaction Time Task and Sustained Attention Task was that the PMA distracted participants from focusing on the primary task. For example, P8 related his initial impression when experiencing one of the Color Attack:

> *It was like a lot slower because I was distracted by the green square that was popping up. (P8)*

P14 suggested the Card Attack prevented her from noticing the primary task:

> *I can't multitask, so you're like, "Pay attention to your surroundings," and I'm like, "I got to pick one, so I pick the cards. (P14)*

Attack impact: Inability to distinguish between virtual and real. Though our manipulative MR content was not close to full fidelity, and the Color Attack objects were even misaligned, we found that some participants on first impression were unable to identify the manipulative MR content as being virtual. For example, P6 later described not realizing the stimulus was virtual in Color Attack even after clicking on it multiple times:

> *I think the first color was green, and when it popped up, I clicked it, and I was like, it didn't work so I kept clicking it because it should work. (P6)*

We found that more participants (7 of 20) initially treated the audio attack as a real-world sound. For example, when P14 first heard the RingtoneSound attack, he described believing it was actually coming from a physical phone:

> *What is going on? Somebody's calling? (P14)*

Attack Impact: Entangling manipulative audio output with primary task. Six participants discussed how auditory attacks impacted their decision-making process in the Sustained Attention Task. P11 mentioned that:

> *I heard sound when I was doing the clicking and because the rhythm of the sound is different from the box changing colors, I got distracted I don't know how many times. (P11)*

P1 discussed the mental overload of handling video and audio at the same time:

> *Some of [the attacks] don't line up with what you're seeing, extra processing that you're having to do, or extra filtering to do those things... Maybe those are different parts of the brain, and those parts of the process might not really overlap. (P1)*

Though visual virtual content remains predominant in today's MR platforms and applications, our finding nevertheless suggests potential risks as MR increasingly incorporates *different* output modalities (e.g., auditory, haptic). We hypothesize that future PMA may be able to leverage multiple sensory modalities to be particularly effective.

(2) Mitigating Onset of Attacks We now turn to participants' adaptation or defensive strategies when experiencing PMA and the subsequent impacts of their chosen strategies. Though participants did not typically interpret the attack outputs as being malicious, we can still learn from how they attempted to avoid the manipulative MR content.

Defensive technique: Mentally filtering out attack content. Many participants (8 of 20) tried to filter the attack content out of their awareness and concentrate on the non-affected area during the Reaction Time Task and Sustained Attention Task. P13 described that:

> *I think instead of noticing the [visual attack], I try to concentrate on what's behind it. (P13)*

Defensive technique: Learning from past attacks. Once participants realized that attacks were occurring (even if they did not think of them as malicious but rather as glitches), some participants adapted their behavior, anticipating and reacting more quickly to subsequent Color and Audio Attacks. For example, P6 explained:

> *At one point I kind of got used to it, and then when it flips colors, it took less time to get used to. (P6)*

Defensive technique: Physically swipe it away. We noticed that when the FalseGreen attack appeared, two participants instinctively raised their hand and tried to swat away the virtual green box. While participants did not discuss this approach, we think this natural reaction suggests potential avenues for future MR systems to detect manipulative content.

When defensive techniques fail under changing attacks. Once some participants developed a particular defensive strategy and/or adapted to a given attack, they often expected that similar attacks would occur. When the Color Attack instead changed, for example, P18 explained how he found himself newly impacted:

> *The red object tripped me up because the green object was in sync with the span. So I thought, "Oh, if I see the objects, I can just click on the feedback loop." Then when I saw the red object, I clicked on it. Obviously, I wasn't supposed to. (P18)*

Side effects from defensive techniques. Participants reported that attempting to avoid manipulative MR content was challenging. And though we found that while defensive strategies sometimes helped avoid attacks, they also caused participants to become more cautious and slower, as P1 described:

> *I think it takes a little bit more mental effort to like filter those out. (P1)*

This finding supports our experimental results from Figure 5.11, which showed that participants' reactions times were slowed even under non-attack conditions, after experiencing attacks.

(3) Attribution and Interpretation of PMA Participants attributed the attacks they experienced to different causes and/or interpreted them in different ways. Before the debrief, we asked participants to share their thoughts and feelings about the experiment, and describe anything that impacted their performance. If they responded by asking about glitches, for example, we did not directly debrief with our research goal, but asked them to first elaborate on their thoughts.

Thought the attack outputs were glitches. We found that the majority of participants (14 of 20) initially assumed that the unexpected outputs in the Reaction Time Task and Sustained Attention Task were glitches, sometimes thinking back to their previous MR/VR experiences. For example, P1 with VR gaming experience recalled:

> *This is absolutely similar to some of my experiences, like when you're playing a game, and the game glitches out a little bit. (P1)*

The fact that participants often assumed (at least at first) that the attacks were glitches could in part reflect the experimental setting: we intentionally did not prime participants about security or the possible presence of attacks, and they may have given the study and the researchers the benefit of the doubt by not jumping to conclusions about malicious intentions. Still, we consider this finding to be meaningful. First, we stress that even participants who attributed the attacks to glitches were impacted by the attacks in practice. Moreover, in real MR settings, users may also be disinclined to assume the presence of malicious adversaries, and the fact that MR software glitches are already common experiences may allow MR attacks to "hide" under the cover of such glitches.

Thought the attack output was supposed to help them. In other instances, participants assumed that the manipulative MR content in the Reaction Time Task and Sustained Attention Task was actually intended to support the primary task. For example, P8 speaks about the Color Attacks:

> *Oh it's here to like maybe help me with the task and like actually performed better. (P8)*

In the Sustained Attention Task, some participants also tried to link the audio output with the visual task. P9 assumed it was aimed to help them perform better:

> *I think I started hearing some like beeping noises... I wondered at first if maybe that was a way to give me a hint.(P9)*

This assumption was again perhaps the result of the experimental setting — expecting that the study was about testing how MR content might help someone perform a task — but we emphasize that people may make such assumptions in real MR settings, as well. Indeed, we observed that participants' trust levels towards MR were relatively high in general and that they had not previously experienced or even considered attacks in MR. Such an assumption presents a possible opportunity for attackers to either make their attacks more stealthy and/or more directly influence people's behaviors on a task.

Attributed attack output to part of the study. Ten participants noticed something was off and guessed (correctly) that it might be a deliberate part of the study, as P13 suggested:

> *I would assume if you guys are conducting the experiment, you would have it done correctly. You would stop the game if it was going awry. (P13)*

Two participants successfully identified the full purpose of the study, with one participant even spontaneously bringing up the gorilla experiment on which our Selective Attention Task was based. We stress that even these participants were still *impacted* by attacks, suggesting that suspecting attacks is not enough to protect users.

5.6 Discussion

While a growing body of prior work has contended with PMA in MR, our work is the first, to our knowledge, to experimentally understand end users' reactions, interpretation, and defensive strategies when experiencing PMA. We highlight several key lessons from our work, and we reflect on implications for MR designers and paint a future research vision.

5.6.1 Key Lessons

1. User can be manipulated by adversarial MR content even despite today's technical limitations. PMA has the ability to manipulate a user's perception of the real world (e.g., treating PMA as if it originated from the real world) and jeopardize their performance on main tasks.

2. In addition to the direct impacts of PMA, we also documented *secondary* effects that manifested on subsequent tasks or task instances — for example, participants becoming more cautious and slow on non-attack tasks after experiencing PMA.

3. Upon experiencing attacks, we observed participants developing a variety of hypotheses, including that the adversarial MR outputs were glitches, outputs were real, or outputs were supposed to help them, to explain the adversarial MR content — but

participants were nevertheless impacted by them. Such expectations can be leveraged by real attackers to either make their attack more stealthy and/or more manipulative.

4. We observed cases of participants successfully adapting to the potential presence of attacks and performing better to subsequent attacks. Meanwhile, there were also examples of participants' adaptive or defensive strategies backfiring — particularly when the attack goal changed.

5.6.2 Implications for MR Defenses

We re-emphasize the call from prior work: MR system and application designers *must* take into account the possibility of adversarial content. Our work, along with others', experimentally demonstrates that such attacks can have real impacts on people using MR systems, and we expect that the impacts in more critical applications and/or with more finely-tuned attacks may be substantially worse. In terms of *how* to take these concerns into account, our findings with real participants position us to make several recommendations:

Contextual focus mode. As our results and previous research indicated, user can be intentionally manipulated or unintentionally distracted by MR stimuli, which can affect their performance on critical tasks. Future MR systems should take that into consideration and incorporate different levels of engagement. For example, inspired by users' defensive strategy in Section 5.5, when a user is on high cognitive load, future MR systems could minimize the amount of displayed information or filter other MR content out of the user's field of view.

Escape to reality. When buggy or malicious content inevitably occurs, user should be able to safely exit back to reality. Similar to the "control-alt-delete" concept for PCs, future MR systems should allow the users to easily and reliably exit the MR view when they wish, i.e., with all MR outputs verifiably disabled (as also suggested in [234]). This mechanism could also be used by users to verify whether something they perceive is MR content or part of the physical world.

Build human resilience against attacks. We observed that some participants were able to perform better on subsequent attacks after they had been exposed to other attacks (though this was not uniformly the case). We encourage future work to further study the potential impact of prior exposure to adversarial MR content on future attack resilience, and to explore how to best take advantage of such resilience. For example, can people be trained or "inoculated" against some types of manipulative MR content through periodic exercises?

Leverage our experimental methodology to evaluate the effectiveness of proposed defenses. Besides enabling the evaluation of PMA, our methodology, which involves exposing participants to PMA in a controlled real-world environment, can be used to measure the impacts of proposed defenses above. As mentioned in Section 5.1, we have made publicly available our experimental testbed implementation.

Attribution of MR content. Given the rising integration of various third-party tools in the MR development cycle, we hypothesize future PMA are likely to manifest in the wild through imported third-party malicious code. Current MR systems render first-party content and third-party content in a similar format, which makes it hard for the user to distinguish if the rendered content is originated from a trusted source. We believe future MR systems should explore ways of providing trusted indicators about the source of content.

5.6.3 Future Directions

Anticipating future PMA in MR. Based on the interviews with participants, we speculate the possibility of even more effective PMA. For example, attacks that simultaneously combine adversarial visual and auditory outputs, or attacks that shift strategies over time to undermine users' defensive adaptations. While our work is early in the evolution of MR technologies (and hence early in the evolution of PMA), it would seem reasonable to assume that adversaries — once they manifest — may conduct their own experiments to maximize the impact of their attacks. Thus, future studies must also attempt to anticipate and protect against such threats.

Evaluating PMA in real-world settings. While we chose to conduct our experiment in a lab setting given safety concerns, a number of related works have already started to implement MR in real life scenarios such as driving [145] and walking [285]. While these works focus more on exploring the technical possibility with MR, future security researchers could apply a similar methodology to implement specialized PMA and investigate their impact on users with proper safety precautions.

Exploring PMA in a multi-user setting. Adversarial MR content might come — as in our study — from a malicious application, but it might also come from other users. As online metaverse platforms start to emerge, toxic and abusive behavior has already been observed in a multi-user settings [29], and some research has begun to explore security and privacy for multi-user MR content sharing [225, 237]. Future studies should also investigate user perception of and reaction to PMA under different multi-user dynamics.

5.7 Conclusion

Our goal in this work has been to explore experimentally the spectrum of end-user reaction, perceptions, and defensive strategies as a result of MR-based perceptual manipulation attacks (PMA). In order to do so, we created a variety of tasks and attack scenarios and observed how users responded, adapted to, and reasoned about them. We view our contribution as laying the groundwork for continued study of PMA. To that end, our work presents a PMA evaluation framework, surfaces several key lessons from user reactions, and proposes directions for future defenses. By constructing PMA targeting different perceptions, and conducting in-depth interviews learning about user perception now, we are taking concrete steps toward securing future human–MR interactions.

Chapter 6

Conclusion

6.1 Summary of Contributions

This thesis adds to a growing body of work from the computer security and privacy community, which has been addressing security, privacy, and safety risks in AR for over a decade [233]. While prior work provides valuable contributions in hypothesizing risks with AR systems, my research is situated at the cusp of significant AR technology advancement, allowing me to identify new examples of attack vectors that are actively exploitable on current AR platforms. To accomplish this, I conducted comprehensive security and privacy analyses of the latest AR ecosystems, examining the core phases of the AR system data flow: input, output, and interaction.

Permission Design for AR Input. Novel inputs such as eye-tracking and hand-tracking enable exciting functionality, enhance understanding of user intention, and greatly improve ergonomic ease when interacting with AR systems. However, existing research has highlighted significant privacy concerns. Our work in Chapter 3 investigates best practices for designing permission frameworks and empirically evaluates existing designs on modern AR platforms. Key guiding principles from our study include clear articulation of privacy mechanisms, ensure sensitive data is processed only on-device, and provide fine-grained access control to users.

Security Vulnerabilities with AR Output. Extensive past research and practice have considered user interface (UI) security for two-dimensional output. Our work in Chapter 4 systematically investigates how AR platforms handle three-dimensional UI output security-related properties: Same Space, Invisibility, and Synthetic Input. We demonstrated five proof-of-concept attacks, with one implemented for each of our test platforms, that leverage different design choices in the context of these AR UI security properties. We found that current AR platforms are all designed and implemented in ways that enable our AR UI attacks to succeed. We discuss potential future defenses, including applying lessons from 2D UI security and identifying new directions for AR UI security. Our findings establish groundwork for future research and design work to consider and address AR UI security challenges.

Safety Risks from AR Interaction. When interacting with overlaid AR content, users may face safety risks if the presented content influences their decision-making in ways that could lead to incorrect perception, cognition, or resulting reaction. The goal of our work in Chapter 5 is to experimentally explore the spectrum of end-user reactions, perceptions, and defensive strategies in response to AR-based perceptual manipulation attacks (PMA). To that end, our work presents a comprehensive PMA evaluation framework, surfaces several key insights from user reactions, and proposes directions for future defenses.

6.2 Looking Forward

The future of AR security and privacy is intrinsically tied to the evolution of AR technology itself. The current north star for AR development is to build everyday-wear glasses that are contextually intelligent, delivering not just immersive experiences but instructive information at the right time. As these technologies continue to develop, the framework presented in this thesis can be used to systematically identify and proactively apply relevant mitigation to security and privacy threats embedded in current technical trajectories.

Input Evolution. Emerging biometric controllers such as electromyography (EMG) wrist bands [35] and brain-computer interface (BCI) controllers [26] are in early development stages but represent the next frontier of AR input modalities. These technologies will bring new opportunities but also pose serious privacy threats as they capture intimate physiological and neural data. Our findings in Chapter 3 provide a foundation for addressing these challenges: when these technologies enter mainstream markets, system developers can leverage our findings to inform permission design, more clearly communicate their privacy-preserving techniques, and build adoption-friendly permission flows.

Holographic Output. Recent breakthroughs in near-eye holographic displays [147, 161] provide enhanced visual fidelity and immersion in AR experiences. As 3D content becomes more realistic and spatially integrated, proper content isolation mechanisms become critical to prevent third-party applications from manipulating output for malicious purposes as shown in Chapter 4. In addition, drawing from our findings in Chapter 5, future AR platforms must implement robust authentication mechanisms that can distinguish between trusted and untrusted content sources while maintaining seamless user experiences.

Interaction with Embedded AI. Existing AR glasses prototypes from Google and Meta already incorporate advanced AI features that deliver contextually intelligent experiences [27, 154]. However, to provide contextual information, these AI models require access to sheer amounts of personal data captured from the user's first-person perspective, including visual scenes, audio conversations, bystander activity, and environmental context. Based on our findings in Chapter 3, we recommend processing sensitive data on-device or implementing opt-in/opt-out features for users to choose whether they wish to share their data with AI services. Beyond data collection concerns, another challenge is how to safely integrate AI-generated outputs into users' field of view. The overlays could inadvertently manipulate user perception and decision-making in ways similar to the PMA we demonstrated. Our framework in Chapter 5 can serve as an evaluation benchmark to systematically assess how AI-generated AR content affects human performance and to develop mitigation mechanisms that ensure user safety while enabling intelligent assistance.

6.3 Final Remarks

While we cannot fully predict the future trajectory of AR technology, this combination of active threat identification and proactive mitigation strategies provides a robust foundation for future defense. As AR technology continues its rapid advancement toward ubiquitous adoption, the security and privacy challenges identified in this thesis will only grow in importance. The frameworks, evaluations, and solutions presented here provide essential foundations to pave the way for a secure and trustworthy AR future.

Bibliography

[1] Apple RealityKit. https://developer.apple.com/documentation/realitykit/.

[2] Apple RoomPlan. https://developer.apple.com/augmented-reality/roomplan/.

[3] AR Advertising Market Insights. https://www.statista.com/outlook/amo/ar-vr/ar-advertising/united-states.

[4] ARCore. https://developers.google.com/ar/develop.

[5] ARCore – Depth adds realism. https://developers.google.com/ar/develop/depth.

[6] ARCore – Geospatial. https://developers.google.com/ar/develop/geospatial.

[7] ARCore – Light Estimation. https://developers.google.com/ar/develop/lighting-estimation.

[8] ARKit. https://developer.apple.com/augmented-reality/arkit/.

[9] Babylon.js. https://www.babylonjs.com/.

[10] Event: isTrusted property. https://developer.mozilla.org/en-US/docs/Web/API/Event/isTrusted.

[11] Facebook Reality Labs: Wristband for AR. https://tech.fb.com/ar-vr/2021/03/inside-facebook-reality-labs-wrist-based-interaction-for-the-next-computing-platform/.

[12] Google Sceneform. https://developers.google.com/sceneform/develop.

[13] Headers/iframe. https://developer.mozilla.org/en-US/docs/Web/HTML/Element/iframe.

[14] Headers/X-Frame-Options. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options.

[15] Hololens – Scene understanding. https://learn.microsoft.com/en-us/windows/mixed-reality/develop/unity/scene-understanding-SDK.

[16] Hololens 2. https://www.microsoft.com/en-us/hololens.

[17] Inside Facebook Reality Labs Research: The Future of Audio. https://about.fb.com/news/2020/09/facebook-reality-labs-research-future-of-audio/.

[18] Mark Zuckerberg demonstrating Meta's high-end Project Cambria VR headset. https://www.theverge.com/2022/5/12/23068536/meta-project-cambria-vr-ar-demo-mark-zuckerberg.

[19] Meta Quest 3. https://about.fb.com/news/2023/06/meta-quest-3-coming-this-fall/.

[20] Metaguard++: Deep motion masking for anonymizing virtual reality motion data. https://github.com/MetaGuard/MetaGuardPlus.

[21] Mojo Vision's Patented Smart Contact Lenses: A New Era in Augmented Reality — insights.greyb.com. https://insights.greyb.com/mojo-visions-patented-smart-contact-lenses/. [Accessed 07-06-2025].

[22] Oculus – Spatial Anchor. https://developer.oculus.com/documentation/unity/unity-spatial-anchors-overview/.

[23] Oculus Integration with Unity. https://assetstore.unity.com/packages/tools/integration/oculus-integration-82022.

[24] Oculus Passthrough. https://developer.oculus.com/documentation/unity/unity-passthrough/.

[25] Oculus Quest 2. https://www.oculus.com/quest-2/.

[26] Open Source Tools for Neuroscience — openbci.com. https://openbci.com/. [Accessed 09-06-2025].

[27] Project Astra — deepmind.google. https://deepmind.google/models/project-astra/. [Accessed 09-06-2025].

[28] RealityKit AnchorEntity. https://developer.apple.com/documentation/realitykit/anchorentity.

[29] A researcher's avatar was sexually assaulted on a metaverse platform. https://www.businessinsider.com/researcher-claims-her-avatar-was-raped-on-metas-metaverse-platform-2022-5.

[30] Rev speech-to-text services. https://www.rev.com/.

[31] Sequence memory test. https://humanbenchmark.com/tests/sequence.

[32] Shaderlab command: Offset. https://docs.unity3d.com/2021.1/Documentation/Manual/SL-Offset.html.

[33] A simple tool to measure your reaction time. http://humanbenchmark.com/tests/reactiontime.

[34] Spectacles from Snapchat. https://www.spectacles.com/.

[35] Surface EMG: An Exciting New Form of HCI That Considers Everyone — meta.com. https://www.meta.com/blog/surface-emg-wristband-electromyography-human-computer-interaction-hci/?srsltid=AfmBOopAXq3Wb1ZZ5WOhyGRilWCqhGJ2JDSDw0Z6hm6GvxfBQvS83Wd3. [Accessed 09-06-2025].

[36] The Touring Machine — graphics.cs.columbia.edu. https://graphics.cs.columbia.edu/projects/mars/touring.html. [Accessed 07-06-2025].

[37] Three.js. https://threejs.org/.

[38] Tobii XR eye tracking system. https://vr.tobii.com/oem/.

[39] Two men fall off a cliff playing Pokémon Go. https://www.latimes.com/local/lanow/la-me-ln-pokemon-go-players-stabbed-fall-off-cliff-20160714-snap-story.html.

[40] Unity 2021.3.16. https://unity.com/releases/editor/whats-new/2021.3.16l.

[41] Unity AR Foundation. https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@2.2/manual/index.html.

[42] Unreal Game Engine. https://www.unrealengine.com/en-US.

[43] User Interface Security and the Visibility API – Privacy Consideration. https://www.w3.org/TR/UISecurity/#privacy-considerations.

[44] WebXR Device API. https://www.w3.org/TR/webxr/.

[45] WebXR Device API Version 20220601. https://www.w3.org/TR/2022/CRD-webxr-20220601/.

[46] What is Mixed Reality Toolkit 2? https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/.

[47] What Is The Invisible Obstacle In Pokémon Go? https://www.ginx.tv/en/pokemon-go/invisible-obstacle.

[48] XR | kauaimuseum — kauaimuseum.org. https://www.kauaimuseum.org/xr. [Accessed 20-05-2025].

[49] Inside Facebook Reality Labs: Wrist-based interaction for the next computing platform — tech.facebook.com. https://tech.facebook.com/reality-labs/2021/3/inside-facebook-reality-labs-wrist-based-interaction-for-the-next-computing-platform/, 2021. [Accessed 21-05-2025].

[50] Eye tracking on meta quest pro. https://www.meta.com/help/quest/articles/getting-started/getting-started-with-quest-pro/eye-tracking/, 2022.

[51] Hand tracking gestures - hololens. https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/features/input/gestures?view=mrtkunity-2022-05, 2022.

[52] Hand tracking — mrtk2. https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/features/input/hand-tracking?view=mrtkunity-2022-05, 2022.

[53] Locatable camera overview. https://learn.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/locatable-camera-overview, 2022.

[54] Microsoft has sold 300,000 hololens units according to analysts. https://www.thurrott.com/mobile/275228/microsoft-hololens-300000-units-sold, 2022.

[55] Adopting best practices for privacy and user preferences - vision pro. https://developer.apple.com/documentation/visionos/adopting-best-practices-for-privacy, 2023.

[56] Apple vision pro. https://www.apple.com/apple-vision-pro/, 2023.

[57] Building Eye Tracking on Meta Quest Pro Responsibly . https://scontent-sea1-1.xx.fbcdn.net/v/t39.8562-6/312898144_1269308143870038_8244941952542354869_n.pdf?_nc_cat=111&ccb=1-7&_nc_sid=b8d81d&_nc_ohc=t8ttpL9ReZ0AX-ZuIfQ&_nc_ht=scontent-sea1-1.xx&oh=00_AfAiD1zsX1vS-jlTgFodXOrETENsUSz9VXDDh4aTDIxdPQ&oe=654F5076, 2023.

[58] Collect and use diagnostic information from hololens devices. https://learn.microsoft.com/en-us/hololens/hololens-diagnostic-logs, 2023.

[59] Enable hand tracking on oculus. https://developer.oculus.com/documentation/native/android/mobile-hand-tracking/, 2023.

[60] Extended eye tracking in native hololens engine. https://learn.microsoft.com/en-us/windows/mixed-reality/develop/native/extended-eye-tracking-native, 2023.

[61] Eye tracked foveated rendering. https://developer.oculus.com/documentation/unity/unity-eye-tracked-foveated-rendering/, 2023.

[62] Eye tracking on hololens 2. https://learn.microsoft.com/en-us/windows/mixed-reality/design/eye-tracking, 2023.

[63] Eye tracking privacy notice for oculus. https://www.meta.com/help/quest/articles/accounts/privacy-information-and-settings/eye-tracking-privacy-notice/, 2023.

[64] Hand and body privacy notice - oculus. https://www.meta.com/help/quest/articles/accounts/privacy-information-and-settings/hand-tracking-privacy-notice/, 2023.

[65] Hand tracking gestures - oculus. https://www.meta.com/help/quest/articles/headsets-and-accessories/controllers-and-hand-tracking/hand-tracking/, 2023.

[66] Hand tracking gestures detection - vsion pro. https://developer.apple.com/documentation/vision/detecting_hand_poses_with_vision, 2023.

[67] Hand tracking on meta quest. https://www.meta.com/help/quest/articles/headsets-and-accessories/controllers-and-hand-tracking/hand-tracking/, 2023.

[68] Handtrackingprovider from visionos. https://developer.apple.com/documentation/arkit/handtrackingprovider, 2023.

[69] Hololens 2 extended eye tracking in unity. https://learn.microsoft.com/en-us/windows/mixed-reality/develop/unity/extended-eye-tracking-unity, 2023.

[70] Hololens 2 native eye-tracking api. https://learn.microsoft.com/en-us/uwp/api/windows.perception.people.eyespose?view=winrt-22621, 2023.

[71] Hololens 2 technical specifications. https://www.apple.com/newsroom/2023/06/introducing-apple-vision-pro/, 2023.

[72] How many vr headsets did meta sell in q1 2023? https://arinsider.co/2023/05/01/how-many-vr-headsets-did-meta-sell-in-q1/, 2023.

[73] How many vr headsets did meta sell in q2 2023? https://arinsider.co/2023/07/31/how-many-vr-headsets-did-meta-sell-in-q2/, 2023.

[74] How many vr headsets did meta sell in q3 2023? https://arinsider.co/2023/10/30/how-many-vr-headsets-did-meta-sell-in-q3-2/, 2023.

[75] How many vr headsets did meta sell in q4 2022? https://arinsider.co/2023/02/06/how-many-vr-headsets-did-meta-sell-in-q4/, 2023.

[76] Improve visual quality and comfort - hololens. https://learn.microsoft.com/en-us/hololens/hololens-calibration, 2023.

[77] Lightship vps from niantic. https://lightship.dev/products/vps, 2023.

[78] Location-based ar experiences with the arcore geospatial api. https://developers.google.com/ar/develop/geospatial, 2023.

[79] Manage user identity and login for hololens. https://learn.microsoft.com/en-us/hololens/hololens-identity, 2023.

[80] Permissions required to use the dynamics 365 guides hololens app. https://learn.microsoft.com/en-us/dynamics365/mixed-reality/guides/hololens-permissions, 2023.

[81] Vision pro eye interaction guidelines. https://developer.apple.com/design/human-interface-guidelines/eyes/, 2023.

[82] Apple has sold approximately 200,000 vision pro headsets. https://www.macrumors.com/2024/01/29/apple-vision-pro-headset-sales/, 2024.

[83] Apple vision pro privacy overview. https://www.apple.com/privacy/docs/Apple_Vision_Pro_Privacy_Overview.pdf, 2024.

[84] How many vr headsets did meta sell in q1 2024? https://arinsider.co/2024/04/29/how-many-headsets-did-meta-sell-in-q1, 2024.

[85] How many vr headsets did meta sell in q4 2023? https://arinsider.co/2024/02/12/how-many-headsets-did-meta-sell-in-q4/, 2024.

[86] Microsoft privacy statement. https://privacy.microsoft.com/en-us/privacystatement, 2024.

[87] Openxr. https://registry.khronos.org/OpenXR/specs/1.0/html/xrspec.html, 2024.

[88] Swift UI. https://developer.apple.com/xcode/swiftui/, 2024.

[89] UIKit. https://developer.apple.com/documentation/uikit, 2024.

[90] Alper Açık, Adjmal Sarwary, Rafael Schultze-Kraft, Selim Onat, and Peter König. Developmental changes in natural viewing behavior: bottom-up and top-down differences between children, young adults and older adults. *Frontiers in psychology*, 1:7198, 2010.

[91] Surin Ahn, Maria Gorlatova, Parinaz Naghizadeh, Mung Chiang, and Prateek Mittal. Adaptive fog-based output security for augmented reality. In *Proceedings of the Morning Workshop on Virtual Reality and Augmented Reality Network*, pages 1–6, 2018.

[92] Devdatta Akhawe, Warren He, Zhiwei Li, Reza Moazzezi, and Dawn Song. Clickjacking revisited: A perceptual view of UI security. In *8th USENIX Workshop on Offensive Technologies WOOT 14)*, 2014.

[93] Dhiraj Amin and Sharvari Govilkar. Comparative study of augmented reality SDKs. *International Journal on Computational Science & Applications*, 5(1):11–26, 2015.

[94] Simone Aonzo, Alessio Merlo, Giulio Tavella, and Yanick Fratantonio. Phishing attacks on modern android. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1788–1801, 2018.

[95] Apple. RoomPlan - Augmented Reality, 2022.

[96] Sajjad Arshad, Amin Kharraz, and William Robertson. Include me out: In-browser detection of malicious third-party content inclusions. In *Financial Cryptography and Data Security: 20th International Conference, FC 2016, Christ Church, Barbados, February 22–26, 2016, Revised Selected Papers 20*, pages 441–459. Springer, 2017.

[97] Benjamin Bach, Ronell Sicat, Johanna Beyer, Maxime Cordeil, and Hanspeter Pfister. The hologram in my hand: How effective is interactive exploration of 3d visualizations in immersive tangible augmented reality? *IEEE transactions on visualization and computer graphics*, 24(1):457–467, 2017.

[98] Stefano Baldassi, Tadayoshi Kohno, Franziska Roesner, and Moqian Tian. Challenges and new directions in augmented reality, computer security, and neuroscience–part 1: Risks to sensation and perception. *arXiv preprint arXiv:1806.10557*, 2018.

[99] Marco Balduzzi, Manuel Egele, Engin Kirda, Davide Balzarotti, and Christopher Kruegel. A solution for the automated detection of clickjacking attacks. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pages 135–144, 2010.

[100] France Bélanger and Robert E Crossler. Privacy in the digital age: a review of information privacy research in information systems. *MIS quarterly*, pages 1017–1041, 2011.

[101] Stuart Bender and Billy Sung. Fright, attention, and joy while killing zombies in virtual reality: A psychophysiological analysis of VR user experience. *Psychology & Marketing*, 2021.

[102] Kevin Benton, L Jean Camp, and Vaibhav Garg. Studying the effectiveness of android application permissions requests. In *2013 IEEE international conference on pervasive computing and communications workshops (PERCOM Workshops)*, pages 291–296. IEEE, 2013.

[103] Antonio Bianchi, Jacopo Corbetta, Luca Invernizzi, Yanick Fratantonio, Christopher Kruegel, and Giovanni Vigna. What the app is that? deception and countermeasures in the android user interface. In *2015 IEEE Symposium on Security and Privacy*, pages 931–948. IEEE, 2015.

[104] Christoph Bichlmeier, Felix Wimmer, Sandro Michael Heining, and Nassir Navab. Contextual anatomic mimesis hybrid in-situ visualization method for improving multi-sensory depth perception in medical augmented reality. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 129–138. IEEE, 2007.

[105] María José Blanca, Jaume Arnau, Javier García-Castro, Rafael Alarcón, and Roser Bono. Non-normal data in repeated measures anova: impact on type i error and power. *Psicothema*, pages 21–29, 2023.

[106] Jonas Blattgerste, Patrick Renner, and Thies Pfeiffer. Advantages of eye-gaze over head-gaze-based selection in virtual and augmented reality under varying field of views. In *Proceedings of the Workshop on Communication by Gaze Interaction*, pages 1–9, 2018.

[107] Rainer Böhme and Jens Grossklags. The security cost of cheap user interaction. In *Proceedings of the 2011 New Security Paradigms Workshop*, pages 67–82, 2011.

[108] Rainer Böhme and Stefan Köpsell. Trained to accept? a field experiment on consent dialogs. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2403–2406, 2010.

[109] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2), 2006.

[110] Peter Casey, Ibrahim Baggili, and Ananya Yarramreddy. Immersive virtual reality attacks and the human joystick. *IEEE Transactions on Dependable and Secure Computing*, 2019.

[111] Sylvain Castagnos, Nicolas Jones, and Pearl Pu. Eye-tracking product recommenders' usage. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 29–36, 2010.

[112] Ruei-Che Chang, Chia-Sheng Hung, Bing-Yu Chen, Dhruv Jain, and Anhong Guo. Soundshift: Exploring sound manipulations for accessible mixed-reality awareness. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference*, pages 116–132, 2024.

[113] Gaurav Chaurasia, Arthur Nieuwoudt, Alexandru-Eugen Ichim, Richard Szeliski, and Alexander Sorkine-Hornung. Passthrough+ real-time stereoscopic view synthesis for mobile mixed reality. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 3(1):1–17, 2020.

[114] Gaurav Chaurasia, Arthur Nieuwoudt, Alexandru-Eugen Ichim, Richard Szeliski, and Alexander Sorkine-Hornung. Passthrough+ real-time stereoscopic view synthesis for

mobile mixed reality. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 3(1):1–17, 2020.

[115] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. Comprehensive experimental analyses of automotive attack surfaces. In *20th USENIX security symposium (USENIX Security 11)*, 2011.

[116] Qi Alfred Chen, Zhiyun Qian, and Z Morley Mao. Peeking into your app without actually seeing it: UI state inference and novel android attacks. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 1037–1052, 2014.

[117] Kaiming Cheng, Arkaprabha Bhattacharya, Michelle Lin, Jaewook Lee, Aroosh Kumar, Jeffery F Tian, Tadayoshi Kohno, and Franziska Roesner. When the user is inside the user interface: An empirical study of ui security properties in augmented reality. In *USENIX Security*, 2024.

[118] Kaiming Cheng, Mattea Sim, Tadayoshi Kohno, and Franziska Roesner. User comprehension and comfort with eye-tracking and hand-tracking permissions in augmented reality. In *Symposium on Usable Security and Privacy (USEC)*, 2025.

[119] Kaiming Cheng, Jeffery F Tian, Tadayoshi Kohno, and Franziska Roesner. Exploring user reactions and mental models towards perceptual manipulation attacks in mixed reality. In *USENIX Security*, volume 18, 2023.

[120] Kang Leng Chiew, Kelvin Sheng Chek Yong, and Choon Lin Tan. A survey of phishing attacks: Their types, vectors and technical approaches. *Expert Systems with Applications*, 106:1–20, 2018.

[121] Koen Claessen and John Hughes. Quickcheck: a lightweight tool for random testing of haskell programs. In *Proceedings of the fifth ACM SIGPLAN international conference on Functional programming*, pages 268–279, 2000.

[122] Matthew Corbett, Brendan David-John, Jiacheng Shang, Y Charlie Hu, and Bo Ji. Bystandar: Protecting bystander visual data in augmented reality systems. In *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*, pages 370–382, 2023.

[123] Eleanor E Cranmer, M Claudia tom Dieck, and Paraskevi Fountoulaki. Exploring the value of augmented reality for tourism. *Tourism Management Perspectives*, 35:100672, 2020.

[124] James Crowley, François Berard, Joelle Coutaz, et al. Finger tracking as an input device for augmented reality. In *International Workshop on Gesture and Face Recognition*, pages 195–200, 1995.

[125] Brendan David-John, Kevin Butler, and Eakta Jain. Privacy-preserving datasets of eye-tracking samples with applications in xr. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2774–2784, 2023.

[126] Brendan David-John, Diane Hosfelt, Kevin Butler, and Eakta Jain. A privacy-preserving approach to streaming eye-tracking data. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2555–2565, 2021.

[127] Jaybie A De Guzman, Kanchana Thilakarathna, and Aruna Seneviratne. Security and privacy approaches in mixed reality: A literature survey. *ACM Computing Surveys (CSUR)*, 52(6):1–37, 2019.

[128] Tamara Denning, Zakariya Dehlawi, and Tadayoshi Kohno. In situ with bystanders of augmented reality glasses: Perspectives on recording and privacy-mediating technologies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2377–2386, 2014.

[129] David Drascic and Paul Milgram. Perceptual issues in augmented reality. In *Stereoscopic Displays and Virtual Reality Systems III*, volume 2653, pages 123–134. Spie, 1996.

[130] Ruofei Du, Eric Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte, Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces, et al. DepthLab: Real-time 3D interaction with depth maps for mobile augmented reality. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 829–843, 2020.

[131] Andrew T Duchowski. Gaze-based interaction: A 30 year retrospective. *Computers & Graphics*, 73:59–69, 2018.

[132] Mica R Endsley. A taxonomy of situation awareness errors. *Human Factors in Aviation Operations*, 3(2):287–292, 1995.

[133] J. Epstein, J. McHugh, R. Pascale, C. Martin, D. Rothnie, H. Orman, A. Marmor-Squires, M. Branstad, and B. Danner. Evolution of a trusted B3 window system prototype. In *IEEE Symposium on Security and Privacy*, 1992.

[134] Sukru Eraslan, Yeliz Yesilada, and Simon Harper. Scanpath trend analysis on web pages: Clustering eye tracking scanpaths. *ACM Transactions on the Web (TWEB)*, 10(4):1–35, 2016.

[135] Habiba Farrukh, Reham Mohamed, Aniket Nare, Antonio Bianchi, and Z Berkay Celik. {LocIn}: Inferring semantic location from spatial maps in mixed reality. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 877–894, 2023.

[136] Elizabeth Fehrer and David Raab. Reaction time to stimuli masked by metacontrast. *Journal of Experimental Psychology*, 63(2):143, 1962.

[137] Adrienne Porter Felt, Serge Egelman, Matthew Finifter, Devdatta Akhawe, David A Wagner, et al. How to ask for permission. *HotSec*, 12:7–7, 2012.

[138] Adrienne Porter Felt, Serge Egelman, and David Wagner. I've got 99 problems, but vibration ain't one: a survey of smartphone users' concerns. In *Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices*, pages 33–44, 2012.

[139] Adrienne Porter Felt, Kate Greenwood, and David Wagner. The effectiveness of application permissions. In *2nd USENIX Conference on Web Application Development (WebApps 11)*, 2011.

[140] Adrienne Porter Felt, Elizabeth Ha, Serge Egelman, Ariel Haney, Erika Chin, and David Wagner. Android permissions: User attention, comprehension, and behavior. In *Proceedings of the eighth symposium on usable privacy and security*, pages 1–14, 2012.

[141] Adrienne Porter Felt, Robert W Reeder, Hazim Almuhimedi, and Sunny Consolvo. Experimenting at scale with google chrome's ssl warning. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2667–2670, 2014.

[142] Adrienne Porter Felt, Helen J. Wang, Alexander Moshchuk, Steve Hanna, and Erika Chin. Permission Re-Delegation: Attacks and defenses. In *USENIX Security Symposium*, 2011.

[143] George Fink and Matt Bishop. Property-based testing: a new approach to testing for assurance. *ACM SIGSOFT Software Engineering howpublisheds*, 22(4):74–80, 1997.

[144] Andrea Gallardo, Chris Choy, Jaideep Juneja, Efe Bozkir, Camille Cobb, Lujo Bauer, and Lorrie Cranor. Speculative privacy concerns about ar glasses data collection. *Proceedings on Privacy Enhancing Technologies*, 4:416–435, 2023.

[145] Florin-Timotei Ghiurău, Mehmet Aydın Baytaş, and Casper Wickman. ARCAR: On-Road Driving in Mixed Reality by Volvo Cars. In *Adjunct Publication of the 33rd Annual ACM Symposium on User Interface Software and Technology*, 2020.

[146] David Goedicke, Alexandra WD Bremers, Hiroshi Yasuda, and Wendy Ju. Xr-oom: Mixing virtual driving simulation with real cars and environments safely. In *13th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, 2021.

[147] Manu Gopakumar, Gun-Yeal Lee, Suyeon Choi, Brian Chao, Yifan Peng, Jonghyun Kim, and Gordon Wetzstein. Full-colour 3d holographic augmented-reality displays with metasurface waveguides. *Nature*, 629(8013):791–797, 2024.

[148] Anthony G Greenwald, Debbie E McGhee, and Jordan LK Schwartz. Measuring individual differences in implicit cognition: the implicit association test. *Journal of Personality and Social Psychology*, 74(6):1464, 1998.

[149] Jaybie Agullo de Guzman, Aruna Seneviratne, and Kanchana Thilakarathna. Unravelling spatial privacy risks of mobile mixed reality data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(1):1–26, 2021.

[150] Jassim Happa, Mashhuda Glencross, and Anthony Steed. Cyber security threats and challenges in collaborative mixed-reality. *Frontiers in ICT*, 6:5, 2019.

[151] David Harborth and Alisa Frik. Evaluating and redefining smartphone permissions with contextualized justifications for mobile augmented reality apps. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*, pages 513–534, 2021.

[152] Eric Dean Haugh and Matt Bishop. *Testing C programs for buffer overflow vulnerabilities.* PhD thesis, Citeseer, 2002.

[153] Scott Hayden. Magic Leap Finally Unveils its First AR Product 'Magic Leap One', Shipping Starts in 2018 — roadtovr.com. https://www.roadtovr.com/magic-leap-finally-unveils-first-ar-product-magic-leap-one-shipping-starts-2018/. [Accessed 07-06-2025].

[154] heathera. Introducing Orion, Our First True Augmented Reality Glasses — about.fb.com. https://about.fb.com/news/2024/09/introducing-orion-our-first-true-augmented-reality-glasses/. [Accessed 09-06-2025].

[155] Harry Helson and Joseph A Steger. On the inhibitory effects of a second stimulus following the primary stimulus to react. *Journal of Experimental Psychology*, 64(3):201, 1962.

[156] Jinhan Hu, Andrei Iosifescu, and Robert LiKamWa. Lenscap: split-process framework for fine-grained visual privacy control for augmented reality apps. In *Proceedings of the 19th annual international conference on mobile systems, applications, and services*, pages 14–27, 2021.

[157] Lin-Shung Huang, Alexander Moshchuk, Helen J Wang, Stuart Schechter, and Collin Jackson. Clickjacking: Attacks and defenses. In *USENIX security symposium*, pages 413–428, 2012.

[158] Duha Ibdah, Nada Lachtar, Satya Meenakshi Raparthi, and Anys Bacha. “why should i read the privacy policy, i just need the service”: A study on attitudes and perceptions toward privacy policies. *IEEE access*, 9:166465–166487, 2021.

[159] Suman Jana, David Molnar, Alexander Moshchuk, Alan Dunn, Benjamin Livshits, Helen J Wang, and Eyal Ofek. Enabling fine-grained permissions for augmented reality applications with recognizers. In *22nd USENIX Security Symposium*, pages 415–430, 2013.

[160] Suman Jana, Arvind Narayanan, and Vitaly Shmatikov. A scanner darkly: Protecting user privacy from perceptual applications. In *IEEE Symposium on Security and Privacy*, pages 349–363, 2013.

[161] Changwon Jang, Kiseung Bang, Minseok Chae, Byoungho Lee, and Douglas Lanman. Waveguide holography for 3d augmented reality glasses. *Nature Communications*, 15(1):66, 2024.

[162] Adam L Janin, David W Mizell, and Thomas P Caudell. Calibration of head-mounted displays for augmented reality applications. In *Proceedings of ieee virtual reality annual international symposium*, pages 246–255. IEEE, 1993.

[163] Jk Jensen, Jinhan Hu, Amir Rahmati, and Robert LiKamWa. Protecting visual information in augmented reality from malicious application developers. In *The 5th ACM Workshop on Wearable Systems and Applications*, pages 23–28, 2019.

[164] Bellal Joseph and David G Armstrong. Potential perils of peri-Pokémon perambulation: the dark reality of augmented reality? *Oxford Medical Case Reports*, 2016(10), 2016.

[165] Nikhita Joshi, Parastoo Abtahi, Raj Sodhi, Nitzan Bartov, Jackson Rushing, Christopher Collins, Daniel Vogel, and Michael Glueck. Transferable microgestures across hand posture and location constraints: Leveraging the middle, ring, and pinky fingers. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–17, 2023.

[166] Levente Juhász, Tessio Novack, Hartwig H Hochmair, and Sen Qiao. Cartographic vandalism in the era of location-based games—the case of openstreetmap and pokémon go. *ISPRS International Journal of Geo-Information*, 9.

[167] Seokbin Kang, Ekta Shokeen, Virginia L Byrne, Leyla Norooz, Elizabeth Bonsignore, Caro Williams-Pierce, and Jon E Froehlich. Armath: augmenting everyday life with math learning. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–15, 2020.

[168] Farzaneh Karegar, John Sören Pettersson, and Simone Fischer-Hübner. The dilemma of user engagement in privacy notices: Effects of interaction modes and habituation on user attention. *ACM Transactions on Privacy and Security (TOPS)*, 23(1):1–38, 2020.

[169] Mark Kilgard. Creating reflections and shadows using stencil buffers. In *At Game Developers Conference*, volume 7, 1999.

[170] Yoonsang Kim, Sanket Goutam, Amir Rahmati, and Arie Kaufman. Erebus: Access control for augmented reality systems. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 929–946, 2023.

[171] Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage. Experimental security analysis of a modern automobile. In *IEEE Symposium on Security and Privacy*, 2010.

[172] George Alex Koulieris, Kaan Akşit, Michael Stengel, Rafał K Mantiuk, Katerina Mania, and Christian Richardt. Near-eye display and tracking technologies for virtual and augmented reality. In *Computer Graphics Forum*, volume 38, pages 493–519. Wiley Online Library, 2019.

[173] Gregory Kramida. Resolving the vergence-accommodation conflict in head-mounted displays. *IEEE transactions on visualization and computer graphics*, 22(7):1912–1931, 2015.

[174] Jacob Leon Kröger, Otto Hans-Martin Lutz, and Florian Müller. What does your gaze reveal about you? on the privacy implications of eye tracking. *Privacy and Identity Management. Data for Better Living: AI and Privacy: 14th IFIP WG 9.2, 9.6/11.7, 11.6/SIG 9.2. 2 International Summer School, Windisch, Switzerland, August 19–23, 2019, Revised Selected Papers 14*, pages 226–241, 2020.

[175] Ernst Kruijff, J Edward Swan, and Steven Feiner. Perceptual issues in augmented reality revisited. In *IEEE International Symposium on Mixed and Augmented Reality*, 2010.

[176] Eike Langbehn, Frank Steinicke, Markus Lappe, Gregory F Welch, and Gerd Bruder. In the blink of an eye: leveraging blink-induced suppression for imperceptible position and orientation redirection in virtual reality. *ACM Transactions on Graphics (TOG)*, 37(4):1–11, 2018.

[177] Joseph S Lappin and Charles W Eriksen. Use of a delayed signal to stop a visual reaction-time response. *Journal of Experimental Psychology*, 72(6):805, 1966.

[178] Kiron Lebeck, Tadayoshi Kohno, and Franziska Roesner. How to safely augment reality: Challenges and directions. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*, pages 45–50, 2016.

[179] Kiron Lebeck, Tadayoshi Kohno, and Franziska Roesner. Enabling multiple applications to simultaneously augment reality: Challenges and directions. In *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications*, pages 81–86, 2019.

[180] Kiron Lebeck, Kimberly Ruth, Tadayoshi Kohno, and Franziska Roesner. Securing augmented reality output. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 320–337. IEEE, 2017.

[181] Kiron Lebeck, Kimberly Ruth, Tadayoshi Kohno, and Franziska Roesner. Arya: Operating system support for securely augmenting reality. *IEEE Security & Privacy*, 16(1):44–53, 2018.

[182] Kiron Lebeck, Kimberly Ruth, Tadayoshi Kohno, and Franziska Roesner. Towards security and privacy for multi-user augmented reality: Foundations with end users. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 392–408. IEEE, 2018.

[183] Daehyeon Lee, Woosung Shim, Munyong Lee, Seunghyun Lee, Kye-Dong Jung, and Soonchul Kwon. Performance Evaluation of Ground AR Anchor with WebXR Device API. *Applied Sciences*, 11(17):7877, 2021.

[184] Hyunjoo Lee, Jiyeon Lee, Daejun Kim, Suman Jana, Insik Shin, and Sooel Son. AdCube: WebVR Ad Fraud and Practical Confinement of Third-Party Ads. In *USENIX Security Symposium*, pages 2543–2560, 2021.

[185] Jaewook Lee, Andrew D Tjahjadi, Jiho Kim, Junpu Yu, Minji Park, Jiawen Zhang, Jon E Froehlich, Yapeng Tian, and Yuhang Zhao. Cookar: Affordance augmentations in wearable ar to support kitchen tool interactions for people with low vision. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–16, 2024.

[186] Jaewook Lee, Jun Wang, Elizabeth Brown, Liam Chu, Sebastian S Rodriguez, and Jon E Froehlich. Gazepointar: A context-aware multimodal voice assistant for pronoun disambiguation in wearable augmented reality. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–20, 2024.

[187] Yeonjoon Lee, Xueqiang Wang, Kwangwuk Lee, Xiaojing Liao, XiaoFeng Wang, Tongxin Li, and Xianghang Mi. Understanding iOS-based Crowdturfing Through Hidden UI Analysis. In *USENIX Security Symposium*, pages 765–781, 2019.

[188] Sarah Lehman, Semir Elezovikj, Haibin Ling, and Chiu Tan. ARCHIE++: A cloud-enabled framework for conducting AR system testing in the wild. *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[189] Sarah M Lehman, Haibin Ling, and Chiu C Tan. Archie: A user-focused framework for testing augmented reality applications in the wild. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 903–912. IEEE, 2020.

[190] Jiachun Li, Yan Meng, Yuxia Zhan, Le Zhang, and Haojin Zhu. Dangers behind charging vr devices: Hidden side channel attacks via charging cables. *IEEE Transactions on Information Forensics and Security*, 2024.

[191] Jingjie Li, Amrita Roy Chowdhury, Kassem Fawaz, and Younghyun Kim. {Kalεido}:{Real-Time} privacy control for {Eye-Tracking} systems. In *30th USENIX security symposium (USENIX security 21)*, pages 1793–1810, 2021.

[192] Shuqing Li, Yechang Wu, Yi Liu, Dinghua Wang, Ming Wen, Yida Tao, Yulei Sui, and Yepang Liu. An exploratory study of bugs in extended reality applications on the web. In *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*, pages 172–183. IEEE, 2020.

[193] Jonathan Liebers, Sascha Brockel, Uwe Gruenefeld, and Stefan Schneegass. Identifying users by their hand tracking data in augmented and virtual reality. *International Journal of Human–Computer Interaction*, pages 1–16, 2022.

[194] Daniel J Liebling and Sören Preibusch. Privacy considerations for a pervasive eye tracking world. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 1169–1177, 2014.

[195] Ao Liu, Lirong Xia, Andrew Duchowski, Reynold Bailey, Kenneth Holmqvist, and Eakta Jain. Differential privacy for eye-tracking data. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, pages 1–10, 2019.

[196] Xueqing Liu, Yue Leng, Wei Yang, Wenyu Wang, Chengxiang Zhai, and Tao Xie. A large-scale empirical study on android runtime-permission rationale messages. In *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 137–146. IEEE, 2018.

[197] Meng Luo, Oleksii Starov, Nima Honarmand, and Nick Nikiforakis. Hindsight: Understanding the evolution of UI vulnerabilities in mobile browsers. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 149–162, 2017.

[198] Tongbo Luo, Xing Jin, Ajai Ananthanarayanan, and Wenliang Du. Touchjacking attacks on web in android, ios, and windows phone. In *International Symposium on Foundations and Practice of Security*, pages 227–243. Springer, 2012.

[199] Shahzad Malik, Chris McDonald, and Gerhard Roth. Hand tracking for interactive pattern-based augmented reality. In *Proceedings. International Symposium on Mixed and Augmented Reality*, pages 117–126. IEEE, 2002.

[200] Joel Martin and David Levine. Property-based testing of browser rendering engines with a consensus oracle. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 424–429. IEEE, 2018.

[201] Marta Matamala-Gomez, Tony Donegan, Sara Bottiroli, Giorgio Sandrini, Maria V Sanchez-Vives, and Cristina Tassorelli. Immersive virtual reality and virtual embodiment for pain relief. *Frontiers in human neuroscience*, page 279, 2019.

[202] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. Reliability and inter-rater reliability in qualitative research: Norms and guidelines for CSCW and HCI practice. *Proc. of the ACM on Human-Computer Interaction*, 3(CSCW):1–23, 2019.

[203] Ülkü Meteriz-Yıldıran, Necip Fazıl Yıldıran, Amro Awad, and David Mohaisen. A keylogging inference attack on air-tapping keyboards in virtual environments. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 765–774. IEEE, 2022.

[204] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.

[205] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. In *Telemanipulator and telepresence technologies*, volume 2351, pages 282–292. Spie, 1995.

[206] Geoffrey A Moore and Regis McKenna. Crossing the chasm. 1999.

[207] H Frank Moore and Masoud Gheisari. A review of virtual and mixed reality applications in construction safety literature. *Safety*, 2019.

[208] Christian Moro, Charlotte Phelps, Petrea Redmond, and Zane Stromberga. Hololens and mobile augmented reality in medical and health science education: A randomised controlled trial. *British Journal of Educational Technology*, 52(2):680–694, 2021.

[209] Alexios Mylonas, Marianthi Theoharidou, and Dimitris Gritzalis. Assessing privacy risks in android: A user-centric approach. In *Risk Assessment and Risk-Driven Testing: First International Workshop, RISK 2013, Held in Conjunction with ICTSS 2013,*

Istanbul, Turkey, November 12, 2013. Revised Selected Papers 1, pages 21–37. Springer, 2014.

[210] Vivek Nair, Wenbo Guo, Justus Mattern, Rui Wang, James F O'Brien, Louis Rosenberg, and Dawn Song. Unique identification of 50,000+ virtual reality users from head & hand motion data. *arXiv preprint arXiv:2302.08927*, 2023.

[211] Kizashi Nakano, Daichi Horita, Nobuchika Sakata, Kiyoshi Kiyokawa, Keiji Yanai, and Takuji Narumi. DeepTaste: Augmented reality gustatory manipulation with GAN-based real-time food-to-food translation. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 212–223. IEEE, 2019.

[212] Arpit Nama, Amaya Dharmasiri, Kanchana Thilakarathna, Albert Zomaya, and Jaybie Agullo de Guzman. User configurable 3d object regeneration for spatial privacy. *arXiv preprint arXiv:2108.08273*, 2021.

[213] David M Neyens and Linda Ng Boyle. The effect of distractions on the crash types of teenage drivers. *Accident Analysis & Prevention*, 39(1):206–212, 2007.

[214] Paweł Nowacki and Marek Woda. Capabilities of ARcore and ARkit platforms for AR/VR applications. In *International Conference on Dependability and Complex Systems*, pages 358–370. Springer, 2019.

[215] Blessing Odeleye, George Loukas, Ryan Heartfield, and Fotios Spyridonis. Detecting framerate-oriented cyber attacks on user experience in virtual reality. 2021.

[216] Joseph O'Hagan, Pejman Saeghe, Jan Gugenheimer, Daniel Medeiros, Karola Marky, Mohamed Khamis, and Mark McGill. Privacy-enhancing technology and everyday augmented reality: Understanding bystanders' varying needs for awareness and consent. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(4):1–35, 2023.

[217] Aleph One. Smashing the stack for fun and profit. *Phrack*, 7(49), November 1996.

[218] Zainab Oufqir, Abdellatif El Abderrahmani, and Khalid Satori. ARKit and ARCore in serve to augmented reality. In *2020 International Conference on Intelligent Systems and Computer Vision (ISCV)*, pages 1–7. IEEE, 2020.

[219] B Keith Payne. Prejudice and perception: the role of automatic and controlled processes in misperceiving a weapon. *Journal of personality and social psychology*, 81(2):181, 2001.

[220] Xiaolan Peng, Jin Huang, Linghan Li, Chen Gao, Hui Chen, Feng Tian, and Hongan Wang. Beyond horror and fear: Exploring player experience invoked by emotional challenge in vr games. In *Extended abstracts of the CHI Conference on Human Factors in Computing Systems*, 2019.

[221] Ken Pfeuffer, Matthias J Geiger, Sarah Prange, Lukas Mecke, Daniel Buschek, and Florian Alt. 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*, pages 1–12, 2019.

[222] Joe Gibbs Politz, Spiridon Eliopoulos, Arjun Guha, and Shriram Krishnamurthi. Adsafety: Type-based verification of javascript sandboxing. *arXiv preprint arXiv:1506.07813*, 2015.

[223] Parinya Punpongsanon, Daisuke Iwai, and Kosuke Sato. Softar: Visually manipulating haptic softness perception in spatial augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 21(11):1279–1288, 2015.

[224] Parinya Punpongsanon, Daisuke Iwai, and Kosuke Sato. Flexeen: Visually manipulating perceived fabric bending stiffness in spatial augmented reality. *IEEE transactions on visualization and computer graphics*, 26(2):1433–1439, 2018.

[225] Shwetha Rajaram, Franziska Roesner, and Michael Nebeling. Designing privacy-informed sharing techniques for multi-user ar experiences. In *VR4Sec: 1st International Workshop on Security for XR and XR for Security*, 2021.

[226] Vilayanur S Ramachandran. *Encyclopedia of Human Behavior*. Academic Press, 2012.

[227] Philipp A Rauschnabel, Jun He, and Young K Ro. Antecedents to the adoption of augmented reality smart glasses: A closer look at privacy risks. *Journal of Business Research*, 92:374–384, 2018.

[228] Keith Rayner, Monica S Castelhano, and Jinmian Yang. Eye movements when looking at unusual/weird scenes: Are there cultural differences? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 35(1):254, 2009.

[229] Charles Reis, Alexander Moshchuk, and Nasko Oskov. Site isolation: Process separation for web sites within the browser. In *USENIX Security Symposium*, 2019.

[230] Olle Renius. A Technical Evaluation of the WebXR Device API for Developing Augmented Reality Web Applications, 2019.

[231] Franziska Roesner, James Fogarty, and Tadayoshi Kohno. User interface toolkit mechanisms for securing interface elements. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 239–250, 2012.

[232] Franziska Roesner and Tadayoshi Kohno. Securing embedded user interfaces: Android and beyond. In *22nd USENIX Security Symposium (USENIX Security 13)*, pages 97–112, 2013.

[233] Franziska Roesner and Tadayoshi Kohno. Security and privacy for augmented reality: Our 10-year retrospective. In *VR4Sec: 1st International Workshop on Security for XR and XR for Security*, 2021.

[234] Franziska Roesner, Tadayoshi Kohno, and David Molnar. Security and privacy for augmented reality systems. *Communications of the ACM*, 57(4):88–96, 2014.

[235] Franziska Roesner, David Molnar, Alexander Moshchuk, Tadayoshi Kohno, and Helen J Wang. World-driven access control for continuous sensing. In *ACM Conference on Computer and Communications Security*, pages 1169–1181, 2014.

[236] Manuel Rudolph, Denis Feth, and Svenja Polst. Why users ignore privacy policies–a survey and intention model for explaining user privacy behavior. In *Human-Computer Interaction. Theories, Methods, and Human Issues: 20th International Conference, HCI International 2018, Las Vegas, NV, USA, July 15–20, 2018, Proceedings, Part I 20*, pages 587–598. Springer, 2018.

[237] Kimberly Ruth, Tadayoshi Kohno, and Franziska Roesner. Secure Multi-Sser content sharing for augmented reality applications. In *28th USENIX Security Symposium*, pages 141–158, 2019.

[238] Gustav Rydstedt, Elie Bursztein, Dan Boneh, and Collin Jackson. Busting frame busting: a study of clickjacking vulnerabilities at popular sites. *IEEE Oakland Web*, 2(6):24, 2010.

[239] Yu Saito, Maki Sugimoto, Satoru Imura, Yuji Morine, Tetsuya Ikemoto, Shuichi Iwahashi, Shinichiro Yamada, and Mitsuo Shimada. Intraoperative 3D hologram support with mixed reality techniques in liver surgery. *Annals of surgery*, 2020.

[240] Tim Scargill, Gopika Premsankar, Jiasi Chen, and Maria Gorlatova. Here To Stay: A Quantitative Comparison of Virtual Object Stability in Markerless Mobile AR. In *Proc. IEEE/ACM Workshop on Cyber-Physical-Human System Design and Implementation*, 2022.

[241] Susanne Schmidt, Gerd Bruder, and Frank Steinicke. Depth perception and manipulation in projection-based spatial augmented reality. *PRESENCE: Virtual and Augmented Reality*, 27(2):242–256, 2018.

[242] Jonathan S. Shapiro, John Vanderburgh, Eric Northup, and David Chizmadia. Design of the EROS trusted window system. In *USENIX Security Symposium*, 2004.

[243] Bingyu Shen, Lili Wei, Chengcheng Xiang, Yudong Wu, Mingyao Shen, Yuanyuan Zhou, and Xinxin Jin. Can systems explain permissions better? understanding users' misperceptions under smartphone runtime permission model. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 751–768. USENIX Association, August 2021.

[244] Yan Shen, Soh-Khim Ong, and Andrew YC Nee. Vision-based hand interaction in augmented reality environment. *Intl. Journal of Human–Computer Interaction*, 27(6):523–544, 2011.

[245] Richard M Shiffrin and Walter Schneider. Controlled and automatic human information processing: II. perceptual learning, automatic attending and a general theory. *Psychological Review*, 84(2):127, 1977.

[246] Gulshan Shrivastava, Prabhat Kumar, Deepak Gupta, and Joel JPC Rodrigues. Privacy issues of android application permissions: A literature review. *Transactions on Emerging Telecommunications Technologies*, 31(12):e3773, 2020.

[247] Daniel J Simons and Christopher F Chabris. Gorillas in our midst: Sustained inattentional blindness for dynamic events. *Perception*, 28(9):1059–1074, 1999.

[248] Carter Slocum, Xukan Ran, and Jiasi Chen. RealityCheck: A tool to evaluate spatial inconsistency in augmented reality. In *2021 IEEE International Symposium on Multimedia (ISM)*, pages 58–65. IEEE, 2021.

[249] Carter Slocum, Yicheng Zhang, Nael Abu-Ghazaleh, and Jiasi Chen. Going through the motions:AR/VR keylogging from user head motions. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 159–174, 2023.

[250] Carter Slocum, Yicheng Zhang, Erfan Shayegani, Pedram Zaree, Nael Abu-Ghazaleh, and Jiasi Chen. That doesn't go there: Attacks on shared state in {Multi-User} augmented reality applications. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 2761–2778, 2024.

[251] Victoria Song. A nerdy dive into Meta's Orion glasses. — theverge.com. `https://www.theverge.com/news/625828/meta-orion-ar-glasses-components`. [Accessed 07-06-2025].

[252] Maximilian Speicher, Brian D Hall, and Michael Nebeling. What is mixed reality? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2019.

[253] Sruti Srinidhi, Edward Lu, and Anthony Rowe. Xair: An xr platform that integrates large language models with the physical world. In *2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 759–767. IEEE, 2024.

[254] Paula C Stacey, Stephanie Walker, and Jean DM Underwood. Face processing and familiarity: Evidence from eye-movement data. *British Journal of Psychology*, 96(4):407–422, 2005.

[255] Nick Statt. Google opens its latest Google Glass AR headset for direct purchase — theverge.com. `https://www.theverge.com/2020/2/4/21121472/google-glass-2-enterprise-edition-for-sale-directly-online`. [Accessed 07-06-2025].

[256] Julian Steil, Inken Hagestedt, Michael Xuelin Huang, and Andreas Bulling. Privacy-aware eye tracking using differential privacy. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, pages 1–9, 2019.

[257] Sophie Stellmach and Raimund Dachselt. Still looking: Investigating seamless gaze-supported selection, positioning, and manipulation of distant targets. In *Proceedings of the sigchi conference on human factors in computing systems*, pages 285–294, 2013.

[258] William Steptoe, Anthony Steed, Aitor Rovira, and John Rae. Lie tracking: social presence, truth and deception in avatar-mediated telecommunication. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1039–1048, 2010.

[259] Xia Su, Han Zhang, Kaiming Cheng, Jaewook Lee, Qiaochu Liu, Wyatt Olson, and Jon E Froehlich. Rassar: Room accessibility and safety scanning in augmented reality. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2024.

[260] Zihao Su, Kunlin Cai, Reuben Beeler, Lukas Dresel, Allan Garcia, Ilya Grishchenko, Yuan Tian, Christopher Kruegel, and Giovanni Vigna. Remote keylogging attacks in multi-user {VR} applications. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 2743–2760, 2024.

[261] Qi Sun, Anjul Patney, Li-Yi Wei, Omer Shapira, Jingwan Lu, Paul Asente, Suwen Zhu, Morgan McGuire, David Luebke, and Arie Kaufman. Towards virtual reality infinite walking: dynamic saccadic redirection. *ACM Transactions on Graphics (TOG)*, 37(4):1–13, 2018.

[262] Ivan E. Sutherland. A head-mounted three-dimensional display. In *Fall Joint Computer Conference, American Federation of Information Processing Societies*, 1968.

[263] Joshua Tan, Khanh Nguyen, Michael Theodorides, Heidi Negrón-Arroyo, Christopher Thompson, Serge Egelman, and David Wagner. The effect of developer-specified explanations for permission requests on smartphone user behavior. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 91–100, 2014.

[264] Yujie Tao, Shan-Yuan Teng, and Pedro Lopes. Altering perceived softness of real rigid objects by restricting fingerpad deformation. In *ACM Symposium on User Interface Software and Technology*, 2021.

[265] Jan Theeuwes. Exogenous and endogenous control of attention: The effect of visual onsets and offsets. *Perception & psychophysics*, 49(1):83–90, 1991.

[266] P CAUDELL Thomas and WM David. Augmented reality: An application of heads-up display technology to manual manufacturing processes. In *Hawaii international conference on system sciences*, volume 2, pages 659–669. ACM SIGCHI Bulletin, 1992.

[267] Porfirio Tramontana, Marco De Luca, and Anna Rita Fasolino. An Approach for Model Based Testing of Augmented Reality Applications. In *RCIS Workshops*, 2022.

[268] Rahmadi Trimananda, Hieu Le, Hao Cui, Janice Tran Ho, Anastasia Shuba, and Athina Markopoulou. OVRseen: Auditing Network Traffic and Privacy Policies in Oculus VR. In *31st USENIX security symposium (USENIX security 22)*, pages 3789–3806, 2022.

[269] Wen-Jie Tseng, Elise Bonnail, Mark Mcgill, Mohamed Khamis, Eric Lecolinet, Samuel Huron, and Jan Gugenheimer. The dark side of perceptual manipulations in virtual reality. In *CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2022.

[270] Tetiana A Vakaliuk and Svitlana I Pochtoviuk. Analysis of tools for the development of augmented reality technologies. CEUR Workshop Proceedings, 2021.

[271] Steven Van Acker, Philippe De Ryck, Lieven Desmet, Frank Piessens, and Wouter Joosen. Webjail: least-privilege integration of third-party components in web mashups. In *Proceedings of the 27th Annual Computer Security Applications Conference*, pages 307–316, 2011.

[272] John Vilk, David Molnar, Benjamin Livshits, Eyal Ofek, Chris Rossbach, Alexander Moshchuk, Helen J Wang, and Ran Gal. SurroundWeb: Mitigating privacy concerns in a 3D web browser. In *IEEE Symposium on Security and Privacy*, pages 431–446, 2015.

[273] Vinoba Vinayagamoorthy, Andrea Brogni, Marco Gillies, Mel Slater, and Anthony Steed. An investigation of presence response across variations in visual realism. In *The 7th Annual International Presence Workshop*, pages 148–155, 2004.

[274] Victoria R Wagner-Greene, Amy J Wotring, Thomas Castor, Jessica Kruger, Sarah Mortemore, and Joseph A Dake. Pokémon go: Healthy or harmful? *American Journal of Public Health*, 107(1):35, 2017.

[275] Hanqiu Wang, Zihao Zhan, Haoqi Shan, Siqi Dai, Maximilian Panoff, and Shuo Wang. Gazeploit: Remote keystroke inference attack by gaze estimation from avatar views in vr/mr devices. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 1731–1745, 2024.

[276] Xiaoyin Wang. Vrtest: an extensible framework for automatic testing of virtual reality scenes. In *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings*, pages 232–236, 2022.

[277] Frederike Wenzlaff, Peer Briken, and Arne Dekker. Video-based eye tracking in sex research: a systematic literature review. *The Journal of Sex Research*, 53(8):1008–1019, 2016.

[278] Primal Wijesekera, Arjun Baokar, Ashkan Hosseini, Serge Egelman, David Wagner, and Konstantin Beznosov. Android permissions remystified: A field study on contextual integrity. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 499–514, 2015.

[279] Graham Wilson and Mark McGill. Violent video games in virtual reality: Re-evaluating the impact and rating of interactive experiences. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play*, 2018.

[280] Yi Wu, Cong Shi, Tianfang Zhang, Payton Walker, Jian Liu, Nitesh Saxena, and Yingying Chen. Privacy leakage via unrestricted motion-position sensors in the age of virtual reality: A study of snooping typed input on virtual keyboards. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 3382–3398. IEEE Computer Society, 2023.

[281] Aiping Xiong, Tianhao Wang, Ninghui Li, and Somesh Jha. Towards effective differential privacy communication for users' data sharing decision and comprehension. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 392–410. IEEE, 2020.

[282] Yanming Xiu, Tim Scargill, and Maria Gorlatova. Viddar: Vision language model-based task-detrimental content detection for augmented reality. *arXiv preprint arXiv:2501.12553*, 2025.

[283] Chengyuan Xu, Radha Kumaran, Noah Stier, Kangyou Yu, and Tobias Höllerer. Multimodal 3d fusion and in-situ learning for spatially aware ai. In *2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 485–494. IEEE, 2024.

[284] Songhua Xu, Hao Jiang, and Francis CM Lau. Personalized online document, image and video recommendation via commodity eye-tracking. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 83–90, 2008.

[285] Jackie Yang, Christian Holz, Eyal Ofek, and Andrew D Wilson. Dreamwalker: Substituting real-world walking experiences with a virtual reality. In *ACM Symposium on User Interface Software and Technology*, 2019.

[286] Zhuolin Yang, Zain Sarwar, Iris Hwang, Ronik Bhaskar, Ben Y Zhao, and Haitao Zheng. Can virtual reality protect users from keystroke inference attacks? In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 2725–2742, 2024.

[287] Steven Yantis and John Jonides. Abrupt visual onsets and selective attention: evidence from visual search. *Journal of Experimental Psychology: Human Perception and Performance.*

[288] Xian Zhan, Lingling Fan, Sen Chen, Feng We, Tianming Liu, Xiapu Luo, and Yang Liu. Atvhunter: Reliable version detection of third-party libraries for vulnerability identification in android applications. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pages 1695–1707. IEEE, 2021.

[289] Mingxue Zhang, Wei Meng, Sangho Lee, Byoungyoung Lee, and Xinyu Xing. All your clicks belong to me: Investigating click interception on the web. In *USENIX Security Symposium*, pages 941–957, 2019.

[290] Tianfang Zhang, Zhengkun Ye, Ahmed Tanvir Mahdad, Md Mojibur Rahman Redoy Akanda, Cong Shi, Yan Wang, Nitesh Saxena, and Yingying Chen. Facereader: unobtrusively mining vital signs and vital sign embedded sensitive info via ar/vr motion sensors. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 446–459, 2023.

[291] Yicheng Zhang, Carter Slocum, Jiasi Chen, and Nael Abu-Ghazaleh. It's all in your head (set): Side-channel attacks on ar/vr systems. In *USENIX Security*, 2023.

[292] Zicheng Zhang, Wenrui Diao, Chengyu Hu, Shanqing Guo, Chaoshun Zuo, and Li Li. An empirical study of potentially malicious third-party libraries in android apps. In *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 144–154, 2020.

[293] Yiqin Zhao, Sheng Wei, and Tian Guo. Privacy-preserving reflection rendering for augmented reality. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2909–2918, 2022.

[294] Yuting Zhou, Juanjuan Chen, and Minhong Wang. A meta-analytic review on incorporating virtual and augmented reality in museum learning. *Educational Research Review*, 36:100454, 2022.

Appendix A

Additional material for: User Comprehension and Comfort with Eye-Tracking and Hand-Tracking Permissions in Augmented Reality

A.1 Recruitment & Survey

In our study, we asked the same set of questions across three different devices. The only difference was the corresponding information about the permission UI, such as permission dialog screenshots and whether the device prompted for permission. We provided alt-text for all screenshots in our survey.

[Recruiting Message] In this study, we are hoping to evaluate the permission-granting process of current Augmented/Mixed Reality Headsets (Apple Vision Pro, Hololens 2, and Oculus Quest Pro). You will be asked to complete a questionnaire which will take around 13 minutes. We are looking for participants who have little or no experience with Augmented/Mixed Reality headsets. When taking the survey, simply answer the questions as honestly as you can. Thank you for your interest in this research.

[Consent Form] Thank you for taking the survey! We are a group of researchers from the University of Washington, and we are hoping to evaluate the permission-granting process of current Augmented/Mixed Reality Headsets (Apple Vision Pro, Hololens 2, and Oculus Quest Pro). You will be asked to complete a questionnaire which will take around 10 minutes. This study was reviewed by the UW Institutional Review Board (IRB) and deemed exempt because it involves no more than minimal risk and meets other criteria. Your responses to this survey will be anonymized. Data from this survey will be stored securely and kept confidential. Your participation in this survey is voluntary, and you may withdraw anytime. If you have questions about this study, please contact Kaiming Cheng (Ph.D. candidate at UW) at kaimingc@cs.washington.edu. You may also contact the UW Human Subjects Division (HSD), which manages IRB review, at hsdinfo@uw.edu. Thank you for taking our survey!

[Filtering] Do you consent to participate in this study?

(i) I am at least 18 years old, I have read and understood this consent form, and I agree to participate in this online research study.

(ii) I do not wish to participate in this study.

[Context] Welcome to the study! We are investigating user perceptions and comfort with the permission-granting process in Augmented and Mixed Reality technologies. Augmented Reality/Mixed Reality (AR/MR) is a technology that overlays digital information onto a user's view of the real world. One common Augmented/Mixed Reality device is a Head Mounted Display, or a headset. AR/MR headsets come in various forms - from looking like regular glasses to looking more like helmets. For example, here are some existing AR/MR headsets on the market today (Figure A.1)
[Tech Background]Do you have a background in technology through education or professional experience?

(i) Yes
(ii) No

[AR Familiarity] Have you heard of Augmented Reality/Mixed Reality (AR/MR) before this study?

(i) Yes
(ii) No

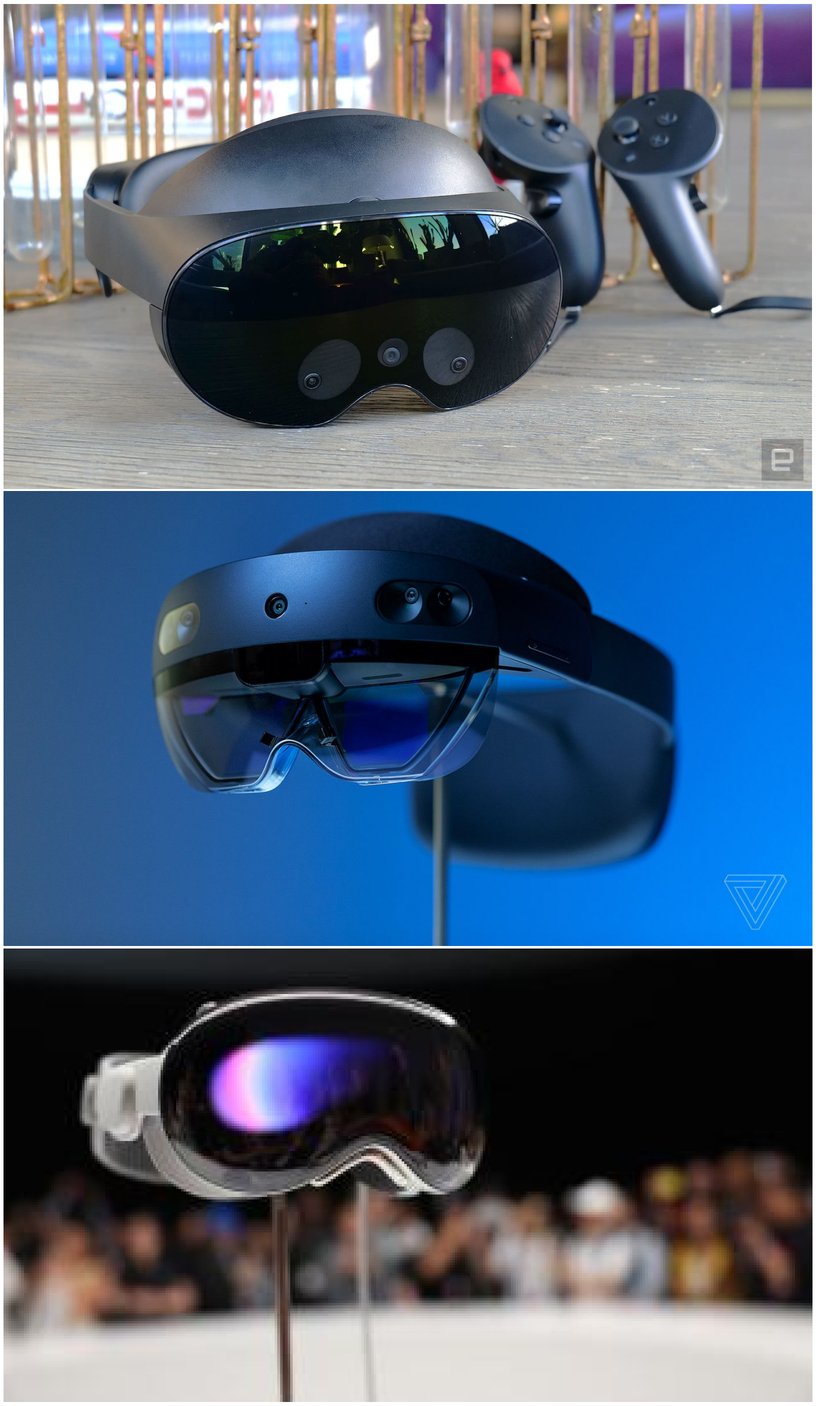


Figure A.1: Images of AR headsets (Meta's Quest Pro, Microsoft's HoloLens 2, and Apple's Vision Pro)

[AR Experience] What is your experience level with Augmented Reality/Mixed Reality (AR/MR) headsets?

(i) I have never used any AR/MR headset.

(ii) I have used an AR/MR headset a few times.

(iii) I am an active user of AR/MR headsets.

[AR Headset Usage] If you have used any of the following AR/MR headsets: Microsoft Hololens 2, Apple Vision Pro, or Meta Quest 3, please select those devices below.

(i) Microsoft Hololens 2

(ii) Apple Vision Pro

(iii) Meta Quest 3

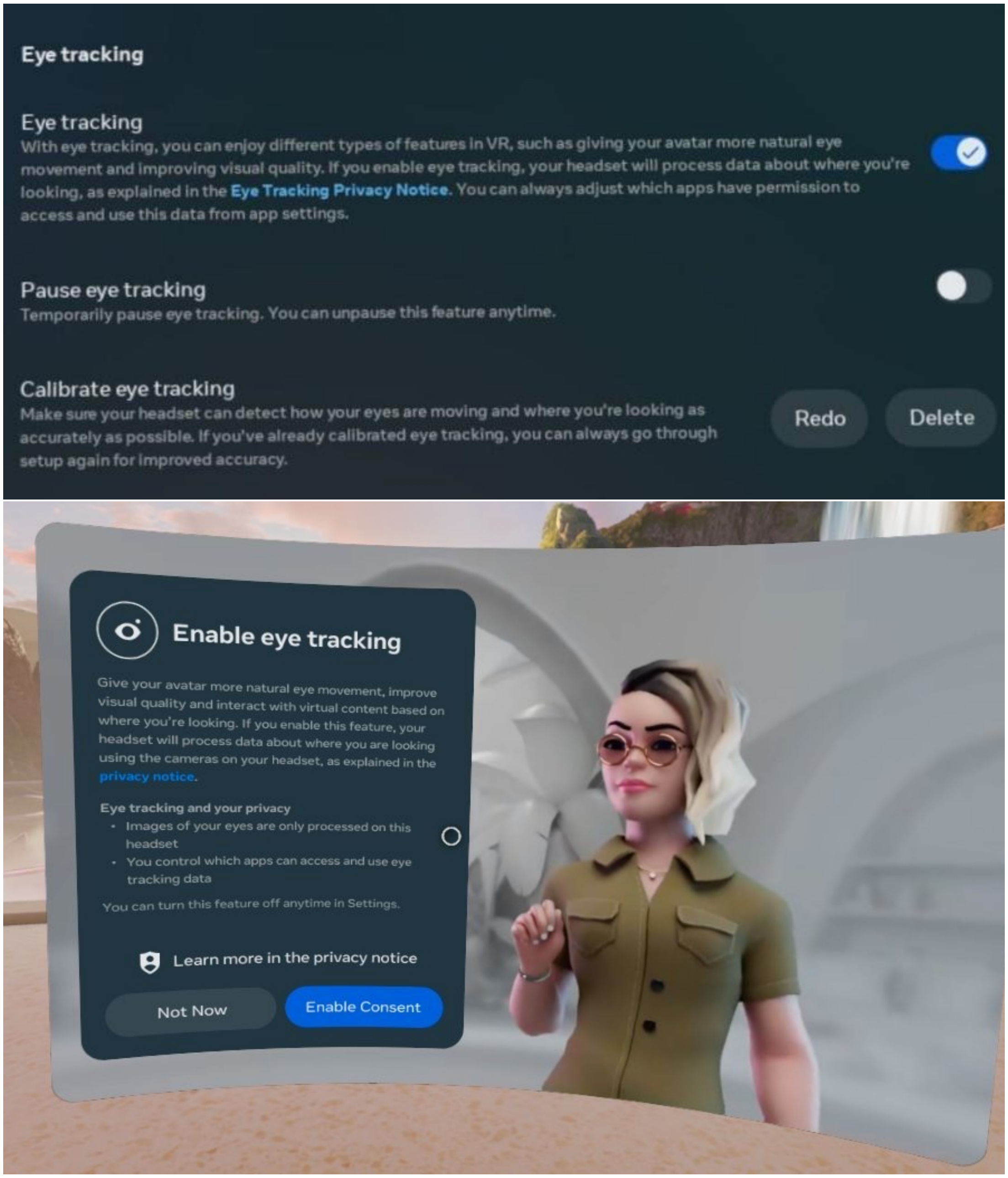
(iv) I have not used any of the above devices.

A.1.1 Survey for Eye-tracking on Oculus

[Introduction] Augmented and Mixed Reality headsets have a variety of sensors recording data while the headset is in use. Users of these headsets typically view permission dialogs to let you allow or deny this request to access your data for different sensors. In this survey, we will present permission dialogs for two different types of sensors and ask for your impressions of each set of dialogs. When you continue, you will see the first sensor.

[Instruction] Suppose you want to use an AR/MR headset with an eye-tracking feature. Below is what you see in the process of granting permission for eye-tracking. We would like to ask you about your comfort levels and how informed you feel during this permission-granting flow. You will first navigate the system-level permission settings for eye tracking. You can enable eye-tracking permission, pause eye tracking, and control eye calibration data for the system in this dialog from the system setting.After you toggle the button, the following dialog appears (Figure A.2):

After you enable the eye-tracking feature, you will be asked to perform a calibration process. You can control which application has access to your eye-tracking data in the system setting (Figure A.3):



Eye tracking

Eye tracking
With eye tracking, you can enjoy different types of features in VR, such as giving your avatar more natural eye movement and improving visual quality. If you enable eye tracking, your headset will process data about where you're looking, as explained in the **Eye Tracking Privacy Notice**. You can always adjust which apps have permission to access and use this data from app settings.

Pause eye tracking
Temporarily pause eye tracking. You can unpause this feature anytime.

Calibrate eye tracking
Make sure your headset can detect how your eyes are moving and where you're looking as accurately as possible. If you've already calibrated eye tracking, you can always go through setup again for improved accuracy.

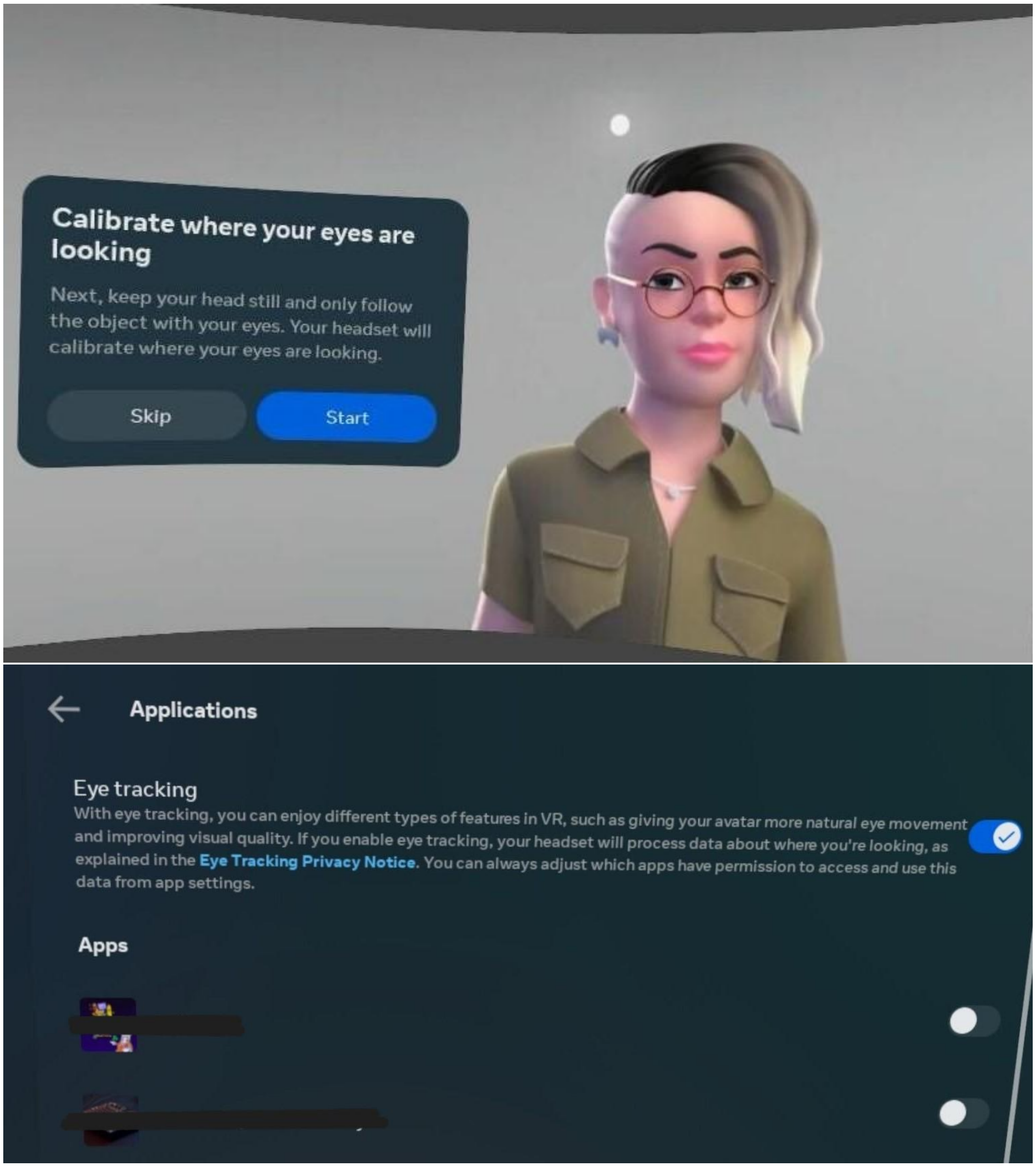
Enable eye tracking

Give your avatar more natural eye movement, improve visual quality and interact with virtual content based on where you're looking. If you enable this feature, your headset will process data about where you are looking using the cameras on your headset, as explained in the privacy notice.

Eye tracking and your privacy
- Images of your eyes are only processed on this headset
- You control which apps can access and use eye tracking data

You can turn this feature off anytime in Settings.

Figure A.2: System-level eye-tracking permission dialog (Oculus)



Calibrate where your eyes are looking

Next, keep your head still and only follow the object with your eyes. Your headset will calibrate where your eyes are looking.

Eye tracking ☑

With eye tracking, you can enjoy different types of features in VR, such as giving your avatar more natural eye movement and improving visual quality. If you enable eye tracking, your headset will process data about where you're looking, as explained in the Eye Tracking Privacy Notice. You can always adjust which apps have permission to access and use this data from app settings.

Apps

[redacted] ○

[redacted] ○

Figure A.3: Eye-tracking calibration and app permission control (Oculus)

[Q1]: I feel informed about the utility of this permission. (5pt Likert scale from "Strong disagree" to "Strong agree")

[Q2]: I feel informed about the associated privacy risk of this permission. (5pt Likert scale from "Strong disagree" to "Strong agree")

[Q3]: I feel confident that this AR/MR system will securely store my eye-tracking data. (5pt Likert scale from "Strong disagree" to "Strong agree")

[Q4]: I know exactly what type data will be collected, how it will be used, and who will have access to it based on the information presented in the above permission screenshots. (5pt Likert scale from "Strong disagree" to "Strong agree")

[Q5]: I feel comfortable using the device knowing the level of access it has to my eye tracking data. (5pt Likert scale from "Strong disagree" to "Strong agree")

[Instruction] Now, we are interested in the degree to which you understand what the system (i.e., the headset) can do with your data once you grant permission. Answer the following true or false questions regarding the **sensor capability**. This is not an evaluation of you; rather, we are attempting to evaluate the efficacy of these dialogs.

[Q6]: The system can understand where your eyes look to indicate which virtual object to select. 1. True 2. False 3. I don't know

[Q7]: The system can identify which real-world objects you are looking at. 1. True 2. False 3. I don't know

[Q8]: The system can simulate your eye movement for your virtual avatar. 1. True 2. False 3. I don't know

[Q9]: The system can authenticate your identity from the unique aspect of your eye (i.e., iris). 1. True 2. False 3. I don't know

[Q10]: The system can adjust eye calibration for new users. 1. True 2. False 3. I don't know

[Instruction] Answer the following true or false questions regarding the **sensor privacy**. This is not an evaluation of you; rather, we are attempting to evaluate the efficacy of these dialogs.

[Q11]: The system requires your permission to access your eye tracking data. 1. True 2. False 3. I don't know

[Q12]: The system allows you to control which application has access to your eye tracking. 1. True 2. False 3. I don't know

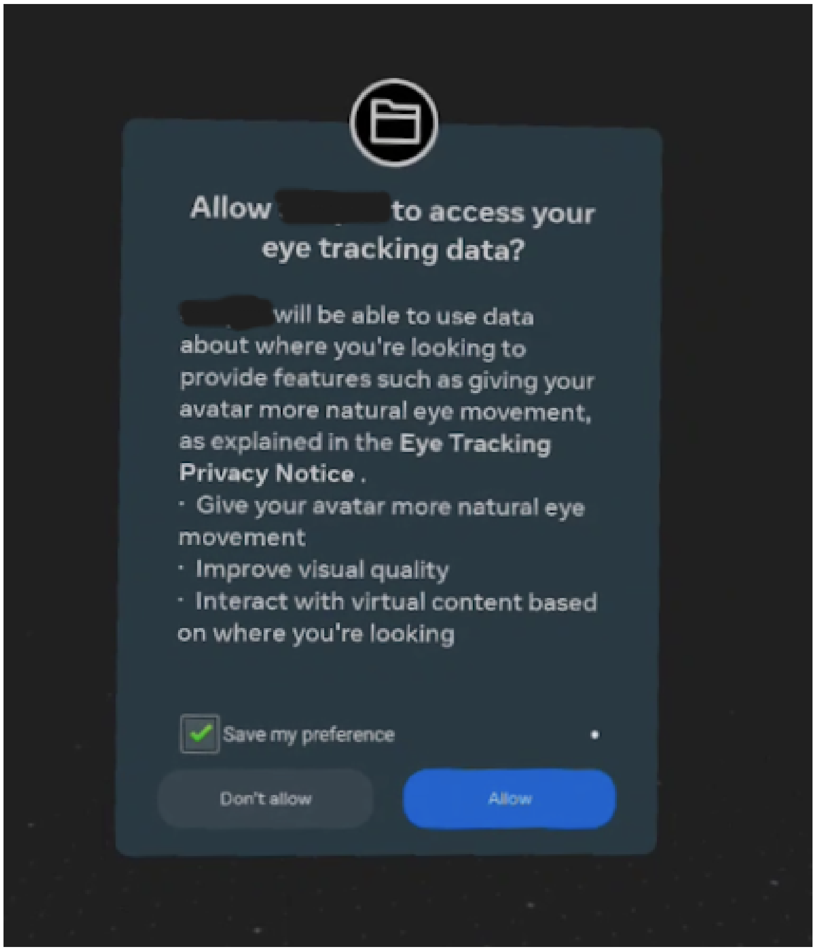
[Q13]: The system can transfer your eye tracking data to an external device (e.g., a company server). 1. True 2. False 3. I don't know

[Q14]: The system can retain the unprocessed image of your eye on the AR/MR headset. 1. True 2. False 3. I don't know

[Q15]: The system only collects your final selection (instead of your eye movements) from the eye tracking data. 1. True 2. False 3. I don't know

[Instruction] Now, you open an app on the headset, which has its own app-level permission settings for eye tracking. The following app dialog appears after you open the application for the first time (Figure A.4):

[Q16]: I feel informed about the utility of this permission. (5pt Likert scale from "Strong



Allow [redacted] to access your eye tracking data?

[redacted] will be able to use data about where you're looking to provide features such as giving your avatar more natural eye movement, as explained in the **Eye Tracking Privacy Notice** .

- Give your avatar more natural eye movement
- Improve visual quality
- Interact with virtual content based on where you're looking

☑ Save my preference

Figure A.4: App-level eye-tracking permission dialog (Oculus)

disagree” to “Strong agree”)

[Q17]: I feel informed about the associated privacy risk of this permission. (5pt Likert scale from “Strong disagree” to “Strong agree”)

[Q18]: I feel confident that this AR/MR application will securely store my eye-tracking data. (5pt Likert scale from “Strong disagree” to “Strong agree”)

[Q19]: I know exactly what type data will be collected, how it will be used, and who will have access to it based on the information presented in the above permission screenshots. (5pt Likert scale from “Strong disagree” to “Strong agree”)

[Q20]: I feel comfortable using this AR/MR application knowing the level of access it has to my eye-tracking data. (5pt Likert scale from “Strong disagree” to “Strong agree”)

[Instruction] Now, we are interested in the degree to which you understand what the application can do with your data once you grant permission. Answer the following true or false questions regarding the **sensor capability**. This is not an evaluation of you; rather, we are attempting to evaluate the efficacy of these dialogs.

[Q21]: The application can understand where your eyes look to indicate which virtual object to select. 1. True 2. False 3. I don’t know

[Q22]: The application can identify which real-world objects you are looking at. 1. True 2. False 3. I don’t know

[Q23]: The application can simulate your eye movement for your virtual avatar. 1. True 2. False 3. I don’t know

[Q24]: The application can authenticate your identity from the unique aspect of your eye (i.e., iris). 1. True 2. False 3. I don't know

[Q25]: The application can access user's eye calibration data (e.g., eye position) provided by the system. 1. True 2. False 3. I don't know

[Instruction] Answer the following true or false questions regarding the sensor privacy. This is not an evaluation of you; rather, we are attempting to evaluate the efficacy of these dialogs. [Q26]: The application requires your permission to access your eye tracking data. 1. True 2. False 3. I don't know

[Q27]: The application can access your eye tracking data when running in the background. 1. True 2. False 3. I don't know

[Q28]: The application can transfer your eye tracking data to an external device (e.g., a company server). 1. True 2. False 3. I don't know

[Q29]: The application can retain the unprocessed image of your eye within the application. 1. True 2. False 3. I don't know

[Q30]: The application only collects your final selection (instead of your eye movements) from the eye tracking data. 1. True 2. False 3. I don't know

[Instruction]Now that you have seen the permission settings for both the overall system and the app, we want to understand what information about both the system and the app can help you feel more comfortable using the technology in the future.

[Instruction] Please drag and drop the top three most important items from the list below that can influence your decision to use this technology in the future. (Don't worry about the ordering within the box)

[Item 1]: Knowing who will have access to this data. [Example includes: permission request; background access, control which app has access to your data].

[Item 2]: Knowing how will the data be stored. [Example includes: Delete after use, stores eye tracking data by default; provide options to delete your data.]

[Item 3]:Knowing how will the data be transmitted.[Example includes: keep your data only on device; transfer your data to an external device]

[Item 4]: Knowing what type of data will be collected. [Example includes: eye movement data (how long you look); eye gaze data (where you look); final selection (where you indicate); unique aspect of your eye (iris).]

[Item 5]: Knowing what is the purpose of collecting this data. [Example includes: indicate selection; generate virtual avatar; identity authentication]

A.1.2 Survey for Hand-tracking on Oculus

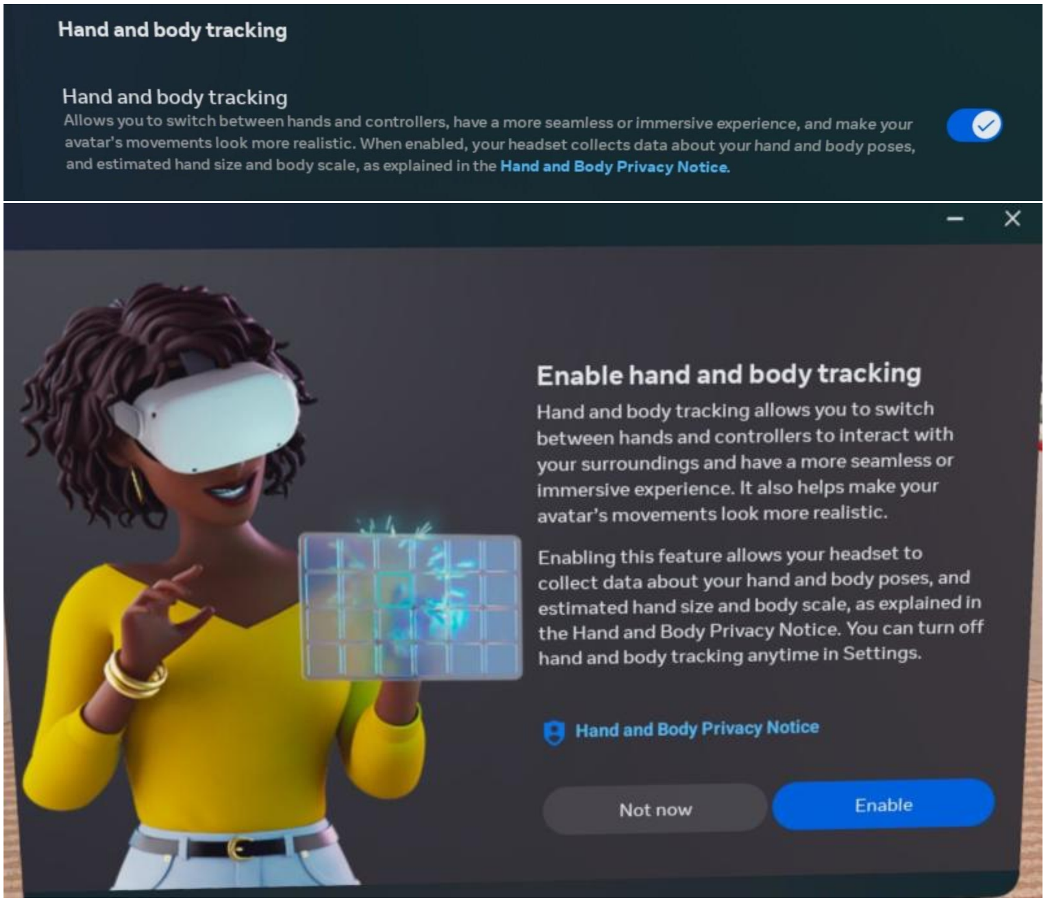
[Instruction] Suppose you want to use an AR/MR application with a hand-tracking feature. Below is what you see in the process of granting permission for hand tracking. We would like to ask you about your comfort levels and how informed you feel during this permission-granting flow. You will first navigate the system-level permission settings for hand tracking. You can enable hand-tracking permission for the system in this dialog from device permission in the system setting. After you toggle the button, the following dialog appears (Figure A.5):

[Hand-tracking tutorial] After you enable the hand-tracking feature, the system will present tutorials on how to interact with the virtual content using your hand (Figure A.6):

[Q31]: I feel informed about the utility of this permission. (5pt Likert scale from "Strong disagree" to "Strong agree")

[Q32]: I feel informed about the associated privacy risk of this permission. (5pt Likert scale from "Strong disagree" to "Strong agree")

[Q33]: I feel confident that this AR/MR system will securely store my hand-tracking data.



Hand and body tracking

Hand and body tracking

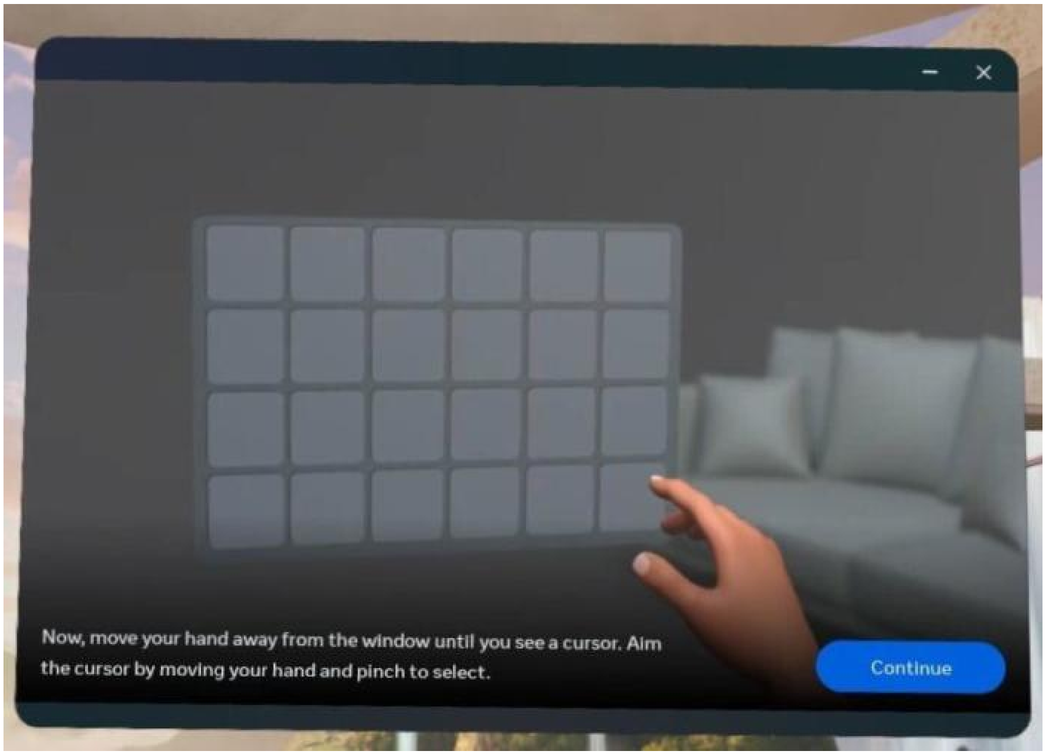
Allows you to switch between hands and controllers, have a more seamless or immersive experience, and make your avatar's movements look more realistic. When enabled, your headset collects data about your hand and body poses, and estimated hand size and body scale, as explained in the Hand and Body Privacy Notice.

Enable hand and body tracking

Hand and body tracking allows you to switch between hands and controllers to interact with your surroundings and have a more seamless or immersive experience. It also helps make your avatar's movements look more realistic.

Enabling this feature allows your headset to collect data about your hand and body poses, and estimated hand size and body scale, as explained in the Hand and Body Privacy Notice. You can turn off hand and body tracking anytime in Settings.

Figure A.5: System-level hand-tracking permission dialog (Oculus)



Now, move your hand away from the window until you see a cursor. Aim the cursor by moving your hand and pinch to select.

Figure A.6: System-level hand-tracking tutorial dialog (Oculus)

(5pt Likert scale from “Strong disagree” to “Strong agree”)

[Q34]: I know exactly what type data will be collected, how it will be used, and who will have access to it based on the information presented in the above permission screenshots. (5pt Likert scale from “Strong disagree” to “Strong agree”)

[Q35]: I feel comfortable using the device knowing the level of access it has to my hand-tracking data. (5pt Likert scale from “Strong disagree” to “Strong agree”)

[Instruction] Now, we are interested in the degree to which you understand what the system (i.e., the headset) can do with your data once you grant permission. Answer the following true or false questions regarding the **sensor capability**. This is not an evaluation of you; rather, we are attempting to evaluate the efficacy of these dialogs.

[Q36]: The system can understand your hand gesture to perform certain actions (e.g., select, scroll). 1. True 2. False 3. I don’t know

[Q37]: The system can identify which real-world objects you are holding. 1. True 2. False 3. I don’t know

[Q38]: The system can simulate your hand movement for your virtual avatar. 1. True 2. False 3. I don’t know

[Q39]: The system can authenticate your identity from the unique aspect of your hand (i.e., fingerprint). 1. True 2. False 3. I don’t know

[Q40]: The system can measure the hand size of new users. 1. True 2. False 3. I don’t know

[Instruction] Answer the following true or false questions regarding the **sensor privacy**. This is not an evaluation of you; rather, we are attempting to evaluate the efficacy of these dialogs.

[Q41]: The system requires your permission to access your hand tracking data. 1. True 2. False 3. I don't know

[Q42]: The system allows you to control which application has access to your hand tracking. 1. True 2. False 3. I don't know

[Q43]: The system can transfer your hand tracking data to an external device (e.g., a company server). 1. True 2. False 3. I don't know

[Q44]: The system can retain the image of your hand on the AR/MR headset. 1. True 2. False 3. I don't know

[Q45]: The system only collects your final selection (instead of your hand movements) from the hand tracking data. 1. True 2. False 3. I don't know

[Instruction] Now, you open an app on the headset, which doesn't need to request app-level permission for hand tracking since the app has automatic access to hand tracking data.

[Q46]: I feel informed about the utility of this permission. (5pt Likert scale from "Strong disagree" to "Strong agree")

[Q47]: I feel informed about the associated privacy risk of this permission. (5pt Likert scale from "Strong disagree" to "Strong agree")

[Q48]: I feel confident that this AR/MR application will securely store my hand-tracking data. (5pt Likert scale from "Strong disagree" to "Strong agree")

[Q49]: I know exactly what type data will be collected, how it will be used, and who will have access to it based on the information presented in the above permission screenshots. (5pt Likert scale from “Strong disagree” to “Strong agree”)

[Q50]: I feel comfortable using this AR/MR application knowing the level of access it has to my hand-tracking data. (5pt Likert scale from “Strong disagree” to “Strong agree”)

[Instruction] Now, we are interested in the degree to which you understand what the system (i.e., the headset) can do with your data once you grant permission. Answer the following true or false questions regarding the **sensor capability**. This is not an evaluation of you; rather, we are attempting to evaluate the efficacy of these dialogs.

[Q51]: The application can understand your hand gesture to perform certain actions (e.g., select, scroll). 1. True 2. False 3. I don’t know

[Q52]: The application can identify which real-world objects you are holding. 1. True 2. False 3. I don’t know

[Q53]: The application can simulate your hand movement for your virtual avatar. 1. True 2. False 3. I don’t know

[Q54]: The application can authenticate your identity from the unique aspect of your hand (i.e., fingerprint). 1. True 2. False 3. I don’t know

[Q55]: The application can measure the hand size of new users. 1. True 2. False 3. I don’t know

[Instruction] Answer the following true or false questions regarding the sensor privacy. This is not an evaluation of you; rather, we are attempting to evaluate the efficacy of these dialogs.
[Q56]: The application requires your permission to access your hand-tracking data. 1. True 2. False 3. I don’t know

[Q57]: The application can access your hand tracking data when running in the background 1. True 2. False 3. I don't know

[Q58]: The application can transfer your hand tracking data to an external device (e.g., a company server). 1. True 2. False 3. I don't know

[Q59]: The application can retain the image of your hand within the application. 1. True 2. False 3. I don't know

[Q60]: The application only collects your final selection (instead of your hand movements) from the hand tracking data. 1. True 2. False 3. I don't know

[Instruction]Now that you have seen the permission settings for both the overall system and the app, we want to understand what information about both the system and the app can help you feel more comfortable using the technology in the future.

[Instruction] Please drag and drop the top three most important items from the list below that can influence your decision to use this technology in the future. (Don't worry about the ordering within the box)

[Item 1]: Knowing who will have access to this data. [Example includes: permission request; background access, control which app has access to your data].
[Item 2]: Knowing how will the data be stored. [Example includes: Delete after use, stores hand tracking data by default; provide options to delete your data.]
[Item 3]:Knowing how will the data be transmitted.[Example includes: keep your data only on device; transfer your data to an external device]
[Item 4]: Knowing what type of data will be collected. [Example includes: hand movement data (how fast you move); hand gesture data (what guesture you perform); unique aspect of your hand (fingerprint).]

[Item 5]: Knowing what is the purpose of collecting this data. [Example includes: indicate selection; generate virtual avatar; identify authentication]

A.1.3 Survey for Eye-tracking on HoloLens

[Instruction] Suppose you want to use an AR/MR headset with an eye-tracking feature. Below is what you see in the process of granting permission for eye-tracking. We would like to ask you about your comfort levels and how informed you feel during this permission-granting flow. You will first navigate the system-level permission settings for eye tracking. You can enable eye-tracking permission, pause eye-tracking, and control eye calibration data for the system in this dialog from the system setting. (Figure A.7):

[Instruction] After you enable the eye-tracking feature, you will be asked to perform a calibration process. After the calibration process, the system provides an alternative sign-in process using the eye-tracking feature. This feature is optional (Figure A.8):

Questions are identical to Q1-Q15 in Appendix A.1.1

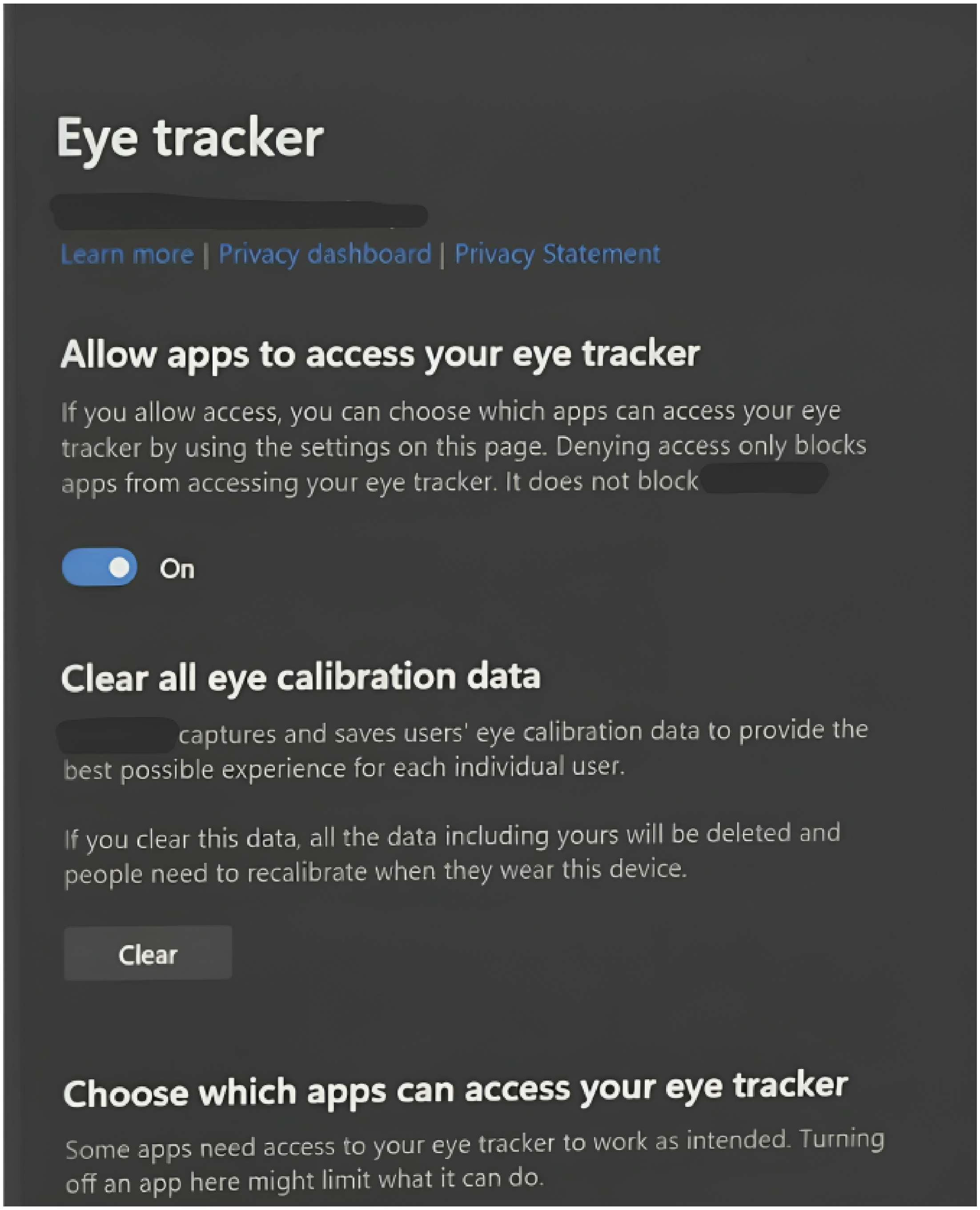
[Instruction] Now, you open an app on the headset, which has its own app-level permission settings for eye tracking. The following app dialog appears after you open the application for the first time (Figure A.9):

Questions are identical to Q16-Q30 in Appendix A.1.1

A.1.4 Survey for Hand-tracking on HoloLens

[Instruction] Suppose you want to use an AR/MR application with a hand-tracking feature. Below is what you see in the process of granting permission for hand tracking. We would like to ask you about your comfort levels and how informed you feel during this permission-granting flow. You will first navigate the system-level permission settings for hand tracking. The hand tracking permission for this system is enabled by default. You are informed about the hand-tracking for the system through this visualization. (Figure A.10):

[Instruction] Hand tracking does not require calibration from the user. Currently, the system



Eye tracker

[redacted]

Allow apps to access your eye tracker

If you allow access, you can choose which apps can access your eye tracker by using the settings on this page. Denying access only blocks apps from accessing your eye tracker. It does not block [redacted]

Clear all eye calibration data

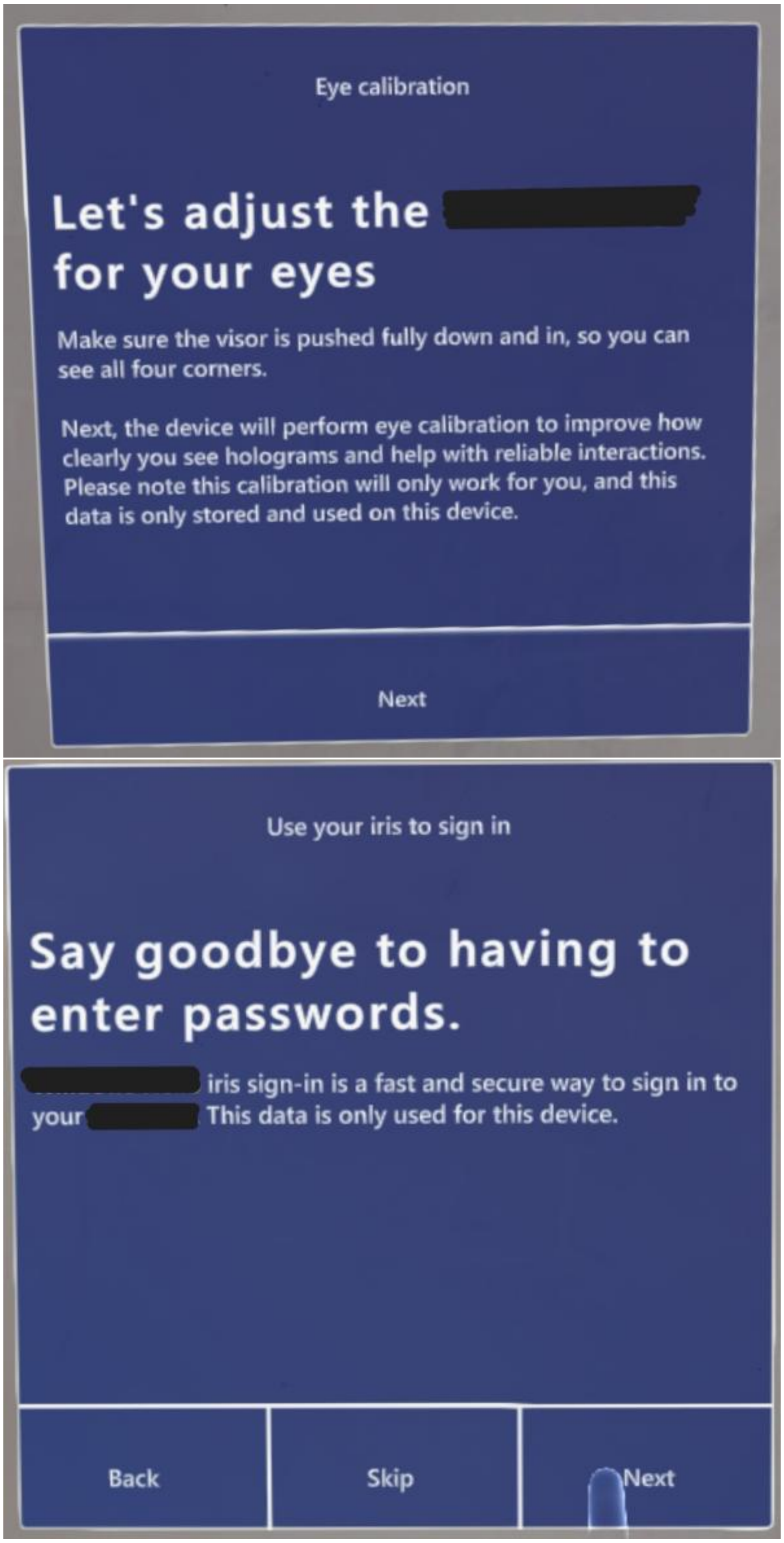
[redacted] captures and saves users' eye calibration data to provide the best possible experience for each individual user.

If you clear this data, all the data including yours will be deleted and people need to recalibrate when they wear this device.

Choose which apps can access your eye tracker

Some apps need access to your eye tracker to work as intended. Turning off an app here might limit what it can do.

Figure A.7: System-level eye-tracking dialog and app permission control (HoloLens)



Eye calibration

Let's adjust the [redacted] for your eyes

Make sure the visor is pushed fully down and in, so you can see all four corners.

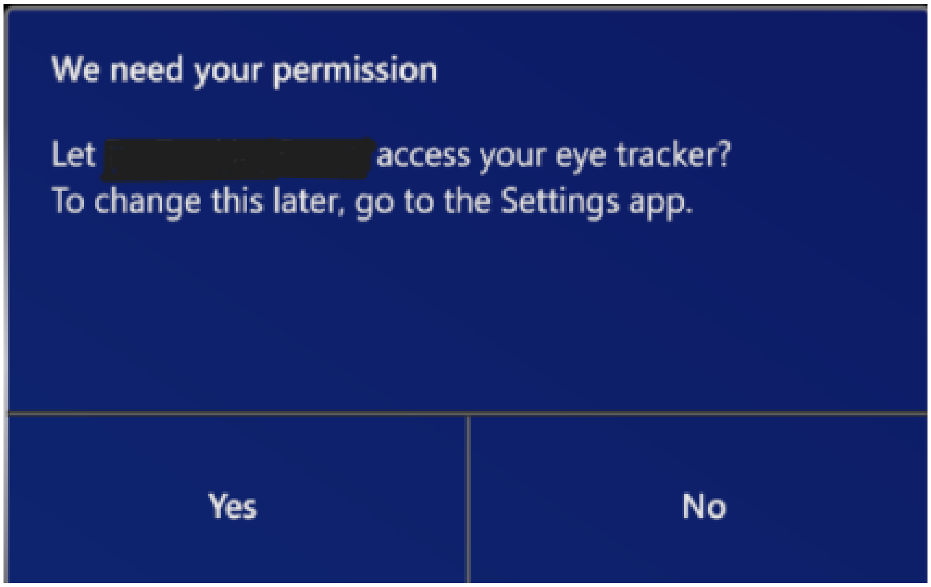
Next, the device will perform eye calibration to improve how clearly you see holograms and help with reliable interactions. Please note this calibration will only work for you, and this data is only stored and used on this device.

Use your iris to sign in

Say goodbye to having to enter passwords.

[redacted] iris sign-in is a fast and secure way to sign in to your [redacted] This data is only used for this device.

Figure A.8: System-level eye-tracking calibration and iris sign-in dialog (HoloLens)



We need your permission

Let [redacted] access your eye tracker?
To change this later, go to the Settings app.

Figure A.9: App-level eye-tracking permission dialog (HoloLens)

does not offer a way to control which applications can access your hand-tracking data in the system setting.

Questions are identical to Q31-Q45 in Appendix A.1.2

[Instruction] Now, when you open an app on the headset, it does not need to request app-level permission for hand tracking. The hand-tracking permission in this system is enabled by default, so the application automatically has access to the hand-tracking data. You can control the permission for hand-tracking background access for the applications in the system settings (Figure A.11):

Questions are identical to Q46-Q60 in Appendix A.1.2

A.1.5 Survey for Eye-tracking on Vision Pro

[Instruction] Suppose you want to use an AR/MR application with an eye tracking feature. Below is what you see in the process of granting the permission for eye tracking. We would like to ask you about your comfort level and how informed you feel during this

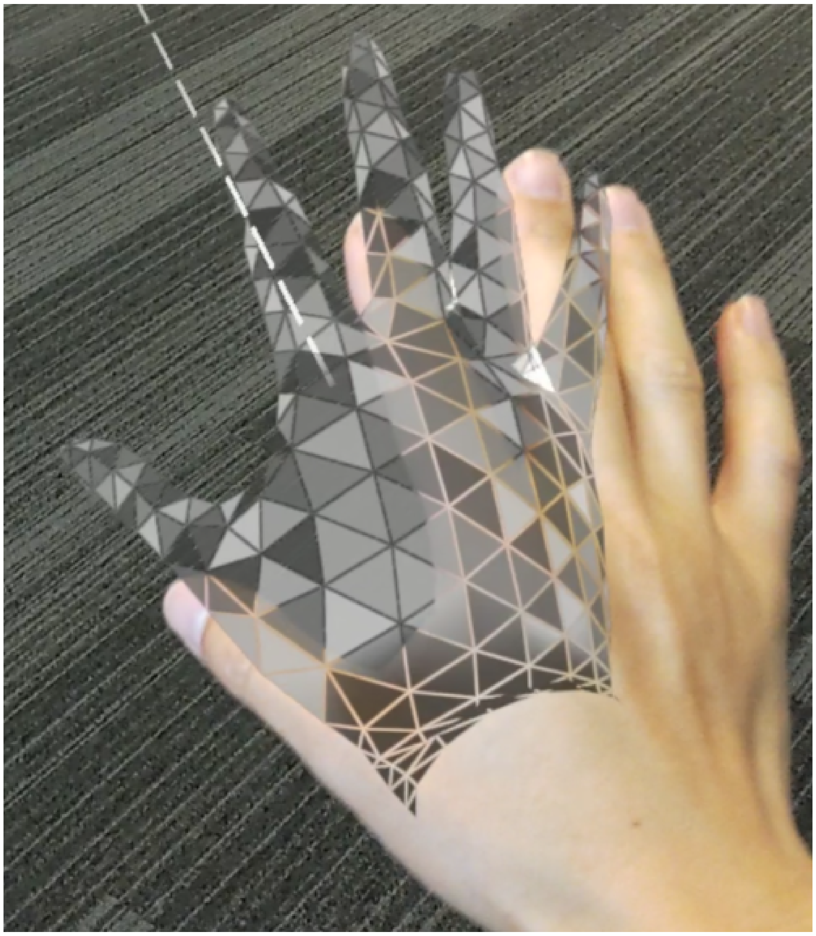
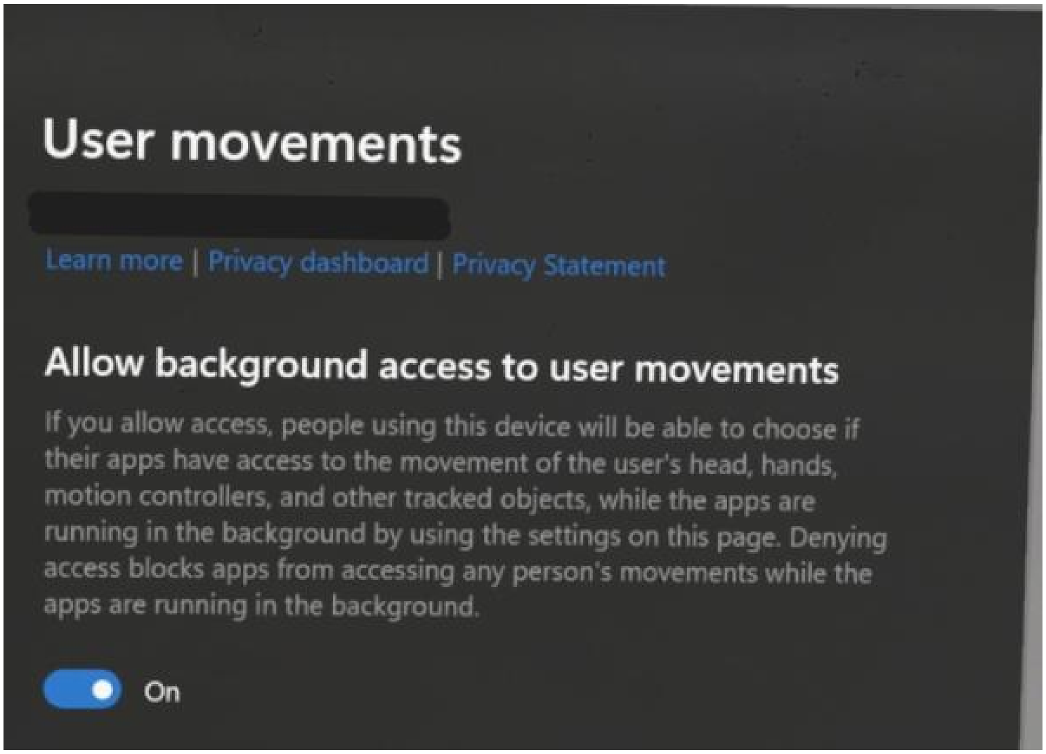


Figure A.10: Hand-tracking visualization (HoloLens)



User movements

[redacted]

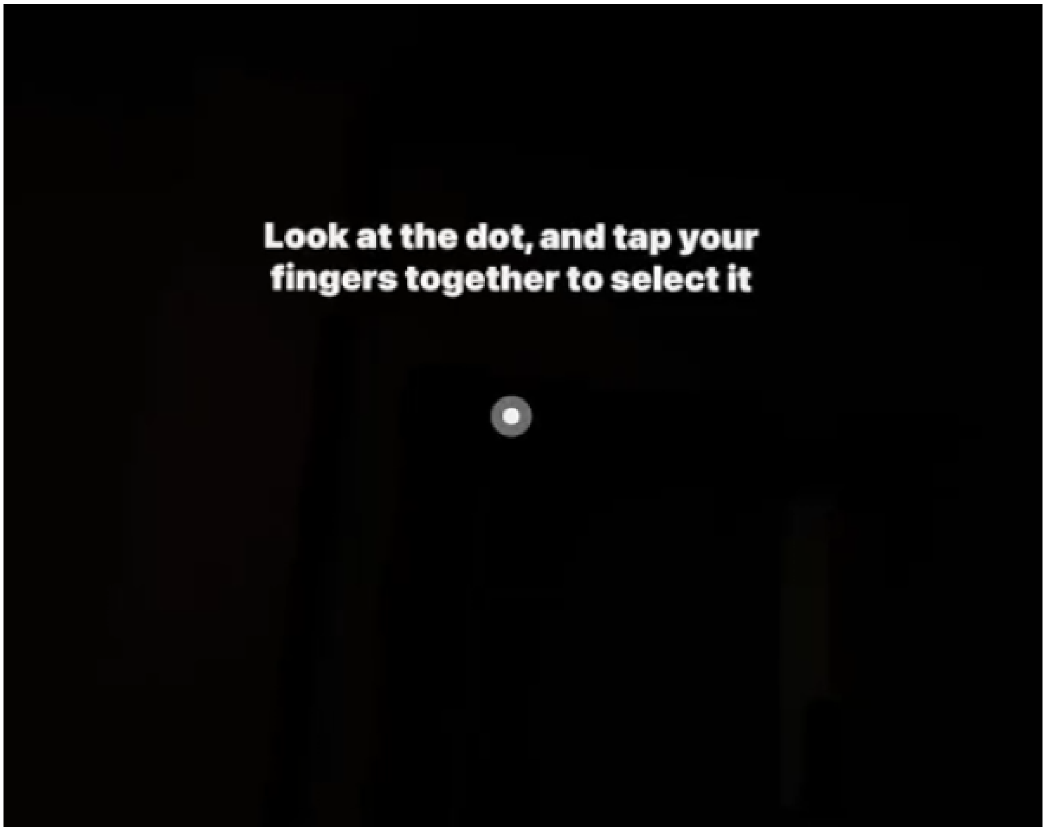
Learn more | Privacy dashboard | Privacy Statement

Allow background access to user movements

If you allow access, people using this device will be able to choose if their apps have access to the movement of the user's head, hands, motion controllers, and other tracked objects, while the apps are running in the background by using the settings on this page. Denying access blocks apps from accessing any person's movements while the apps are running in the background.

● On

Figure A.11: Background access permission for hand-tracking (HoloLens)



Look at the dot, and tap your fingers together to select it

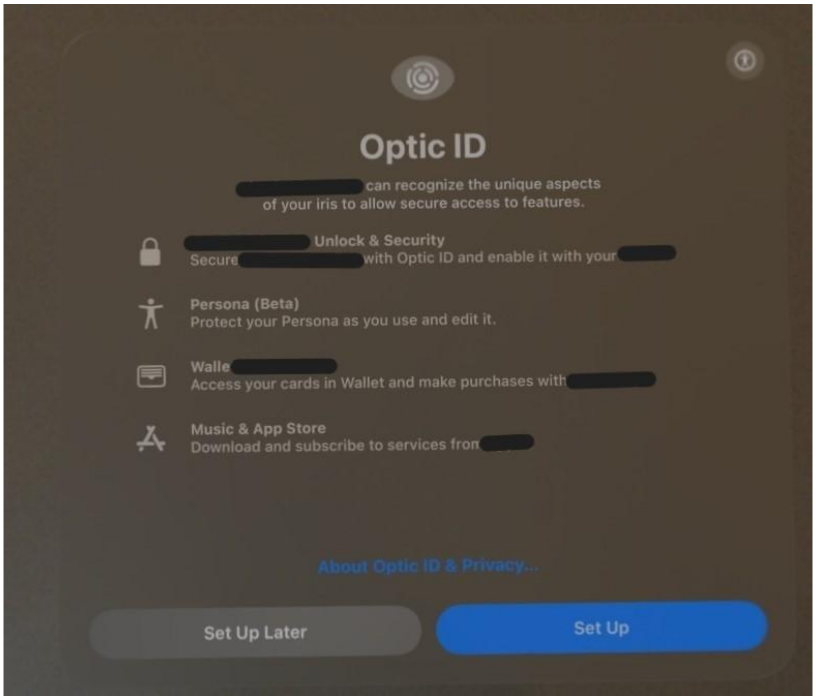
Figure A.12: Eye-tracking calibration (Vision Pro)

permission-granting flow. You will first navigate the system-level permission settings for eye tracking. The eye tracking permission for the system is enabled by default. You are informed about the eye-tracking calibration for the system in this dialog. (Figure A.12):

[Instruction] After the calibration process, the system provides an alternative sign-in process using the eye-tracking feature. This feature is optional(Figure A.13):

Questions are identical to Q1-Q15 in Appendix A.1.1

[Instruction] Now, you open an app on the headset, which doesn't need to request app-level permission for eye tracking since device doesn't share eye-tracking data with applications.



Optic ID

[redacted] can recognize the unique aspects of your iris to allow secure access to features.

[redacted] Unlock & Security
Secure [redacted] with Optic ID and enable it with your [redacted]

Persona (Beta)
Protect your Persona as you use and edit it.

Walle [redacted]
Access your cards in Wallet and make purchases with [redacted]

Music & App Store
Download and subscribe to services fron [redacted]

About Optic ID & Privacy...

Figure A.13: Eye-tracking Optid ID (Vision Pro)

Questions are identical to Q16-Q30 in Appendix A.1.1

A.1.6 Survey for Hand-tracking on Vision Pro

[Instruction] Suppose you want to use an AR/MR application with a hand-tracking feature. Below is what you see in the process of granting permission for hand tracking. We would like to ask you about your comfort levels and how informed you feel during this permission-granting flow. You will first navigate the system-level permission settings for hand tracking. The hand tracking permission for this system is enabled by default. You are informed about the hand-tracking for the system through this visualization. (Figure A.14):

[Instruction] You can control which application has access to your hand-tracking data in the system setting (Figure A.15):

Questions are identical to Q31-Q45 in Appendix A.1.2

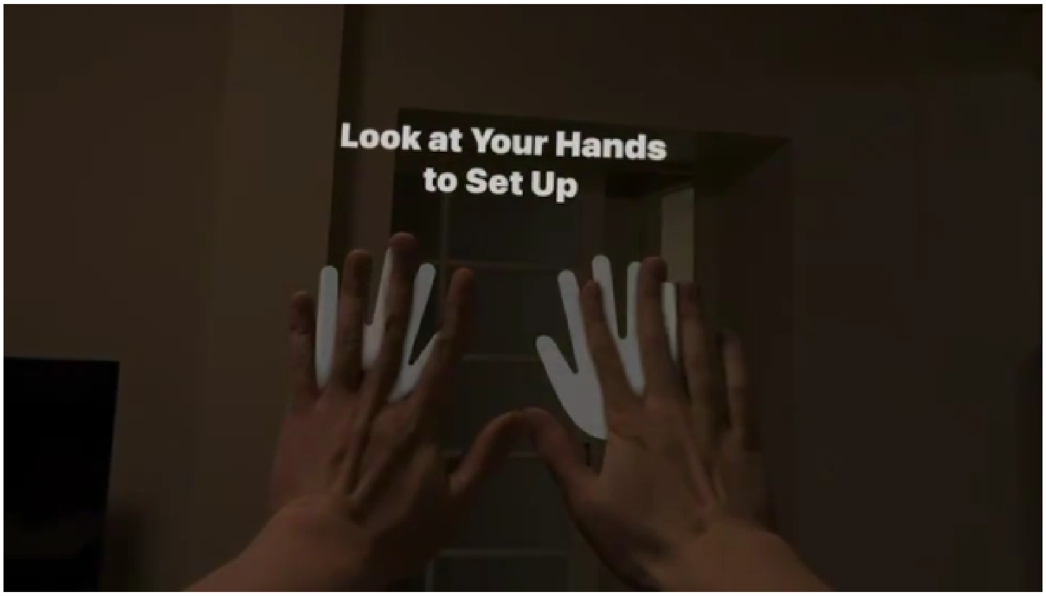
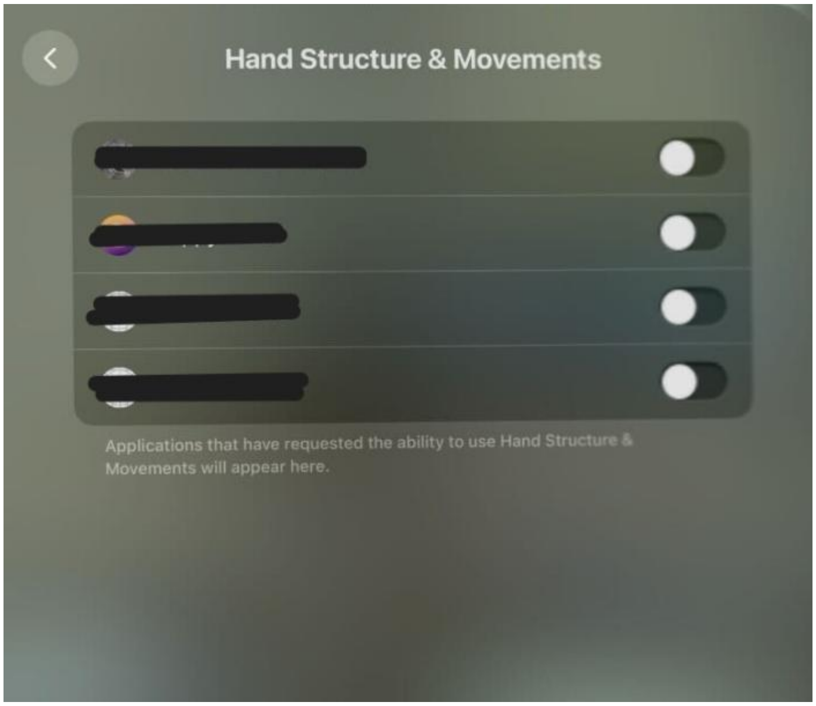


Figure A.14: Hand-tracking calibration (Vision Pro)


Hand Structure & Movements

[redacted] ○

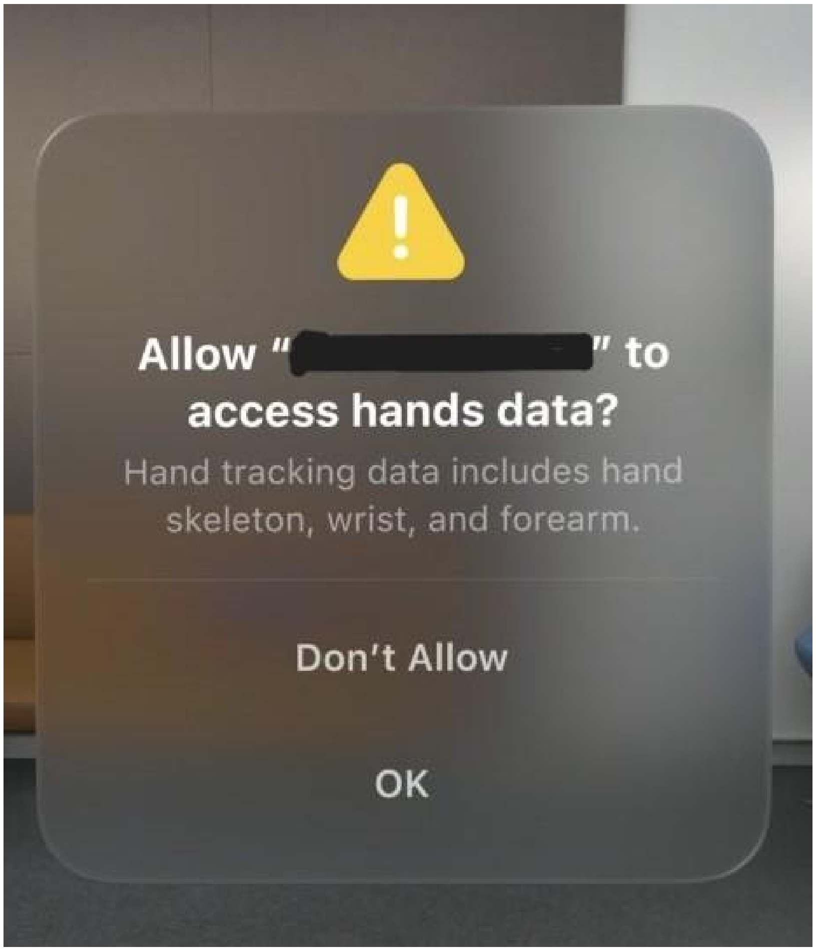
[redacted] ○

[redacted] ○

[redacted] ○

Applications that have requested the ability to use Hand Structure & Movements will appear here.

Figure A.15: Hand-tracking app permission control (Vision Pro)



Allow "[redacted]" to access hands data?

Hand tracking data includes hand skeleton, wrist, and forearm.

Figure A.16: App-level hand-tracking permission dialog (Vision Pro)

[Instruction] Now, you open an app on the headset, which has its own app-level permission settings for hand tracking (Figure A.16):

Questions are identical to Q46-Q60 in Appendix A.1.2

A.2 Comprehension Questions Answer Key

As part of our analysis of participant comprehension, we determine our own best assessment of the correct answer. We did this based on our own understanding of the APIs, documentation, and privacy policies. We document our answers and justifications for eye-tracking on the system level in Table A.1, eye-tracking on the application level in Table A.2, hand-tracking on the system level in Table A.3, and hand-tracking on the application level in Table A.4. Quotes from privacy policies or documentation are in italics in the tables.

Table A.1: Justification for system-level eye-tracking comprehension questions

Comprehension Question	**HoloLens**	**HoloLens Justification**	**Oculus**	**Oculus Justification**	**Vision Pro**	**Vision Pro Justification**
The system requires your permission to access your eye-tracking data.	False	Denying access only blocks apps from accessing your eye tracking. It does not block HoloLens (Figire A.7)	True	Run-time system level permission model (Figure A.2)	False	Eye-tracking is enabled by default for the system.
The system allows you to control which application has access to your eye tracking.	True	Built-in function (Figure A.7)	True	Built-in function (Figure A.3)	False	Eye input is not shared with Apple, third-party apps, or websites. [83]
The system can transfer your eye-tracking data to an external device (e.g., a company server).	False	Microsoft doesn't store any biometric or other identifiable information [80].	True	We collect and retain certain data about your interactions with eye tracking [63]	False	Eye input is not shared with Apple, third-party apps, or websites. [83]

The system can retain the unprocessed image of your eye on the AR/MR headset.	False	We store calibration information locally on device correlated with bit codes from the Iris pattern [76]	False	The raw image data is deleted from your headset after the abstracted gaze data is generated. [63]	False	Optic ID data — including mathematical representations of your iris — is encrypted and protected by the Secure Enclave [83]
The system only collects your final selection (instead of your eye movements) from the eye tracking data.	False	Abstracted eye-tracking data is available to the system [70]	False	Abstracted eye-tracking data is available to the system [57]	True	Data minimization for eye-tracking data [83]
The system can understand where your eyes look to indicate which virtual object to select.	True	Built-in function (EyesPose.Gaze [70]).	True	Built-in function (OVREyeGaze [57]).	True	Built-in function [83].
The system can identify which real-world objects you are looking at.	True	Access to passthrough camera data is available [53].	False	Identifying real-world objects requires integrating passthrough camera data, which the eye-tracking API does not offer [50].	False	Identifying real-world objects requires integrating passthrough camera data, which the eye-tracking API does not offer [83].

The system can simulate your eye movement for your virtual avatar.	True	Abstracted eye-tracking can simulate eye movement (EyesPose.Gaze [70]).	True	Abstracted eye-tracking can simulate eye movement (OVREyeGaze [57]).	True	Built-in function (Persona [83]).
The system can authenticate your identity from the unique aspect of your eye (i.e., iris).	True	Store calibration information locally on device correlated with bit codes from the Iris pattern [76]	False	Iris-scanning function is not supported	True	Optic ID data is encrypted, never leaves your device, and is accessible only to the Secure Enclave processor. [83]
The system can adjust eye calibration for new users.	True	Built-in function [76].	True	Built-in function [57]).	True	Built-in function [83].

Table A.2: Justification for application level eye-tracking comprehension questions

Comprehension Question	**HoloLens**	**HoloLens Justification**	**Oculus**	**Oculus Justification**	**Vision Pro**	**Vision Pro Justification**
The application requires your permission to access your eye-tracking data.	True	Run-time app level permission model (Figure A.9)	True	Run-time app level permission model (Figure A.4)	False	Eye input is not shared with Apple, third-party apps, or websites [83].
The application can access your eye-tracking data when running in the background.	False	Background access for eye-tracking is not supported.	False	Background access for eye-tracking is not supported.	False	Eye input is not shared with Apple, third-party apps, or websites. [83]
The application can transfer your eye-tracking data to an external device (e.g., a company server).	True	System provide no control over how third-party used user's eye-tracking data .	True	Oculus does not control how a third-party app uses, stores, or shares your abstracted gaze data [63]	False	Eye input is not shared with Apple, third-party apps, or websites [83].
The application can retain the unprocessed image of your eye on the AR/MR headset.	False	Only abstracted eye-tracking data is available to the application [70]	False	Only abstracted eye-tracking data is available to the application [63]	False	Eye input is not shared with Apple, third-party apps, or websites [83].

The application only collects your final selection (instead of your eye movements) from the eye-tracking data.	False	Application has access to the abstracted eye-tracking data [70]	False	Application has access to the abstracted eye-tracking data [57]	True	Only when you select the button, by both looking at it and tapping your fingers together, does where you are looking get communicated to the app. [83].
The application can understand where your eyes look to indicate which virtual object to select.	True	Built-in function (EyesPose.Gaze [70]).	True	Built-in function (OVREyeGaze [57]).	False	Eye input is not shared with Apple, third-party apps, or websites [83].
The application can identify which real-world objects you are looking at.	True	Access to passthrough camera data is available [53].	False	Identifying real-world objects requires integrating passthrough camera data, which the eye-tracking API does not offer [50].	False	Eye input is not shared with Apple, third-party apps, or websites [83].
The application can simulate your eye movement for your virtual avatar.	True	Abstracted eye-tracking can simulate eye movement (EyesPose.Gaze [70]).	True	Abstracted eye-tracking can simulate eye movement (OVREyeGaze [57]).	False	Eye input is not shared with Apple, third-party apps, or websites [83].

The application can authenticate your identity from the unique aspect of your eye (i.e., iris).	False	All calibration data is stored securely on the device locally and only available to the system [76]	False	Iris-scanning function is not supported	False	Optic ID data is encrypted, never leaves your device, and is accessible only to the Secure Enclave processor. [56]
The application can access user's eye calibration data (e.g., eye position) provided by the system.	False	All calibration data is stored securely on the device locally and only available to the system. [76]	False	The eye-tracking API may only request eye-tracker calibration instead of directly accessing the data [87].	False	Data used to calibrate your Apple Vision Pro to your eyes is protected on-device [83].

Table A.3: Justification for app-level hand-tracking comprehension questions

Comprehension Question	**HoloLens**	**HoloLens Justification**	**Oculus**	**Oculus Justification**	**Vision Pro**	**Vision Pro Justification**
The system requires your permission to access your hand-tracking data.	False	Hand-tracking is enabled by default for the system.	True	Run-time system level permission model (Figure A.5)	False	Hand-tracking is enabled by default for the system.
The system allows you to control which application has access to your hand tracking.	False	System automatically grants applications access to the hand-tracking API	False	System automatically grants applications access to the hand-tracking API	True	Run-time application level permission model (Figure A.15)
The system can transfer your hand-tracking data to an external device (e.g., a company server).	False	HoloLens also detects hand gestures intended for system interactions (such as menu navigation, pan/zoom, and scroll). This data is processed on your HoloLens device and is not stored. [86].	True	Meta processes and shares the hand-tracking data with the Oculus server, where it is retained for 90 days [64]	False	Apps do not need access to your hands set up information in order to help you interact with content [83].

The system can retain the unprocessed image of your hand on the AR/MR headset.	False	HoloLens also detects hand gestures intended for system interactions (such as menu navigation, pan/zoom, and scroll). This data is processed on your HoloLens device and is not stored. [86].	False	All of this analysis is done on your device in real-time as you move, and the images and estimated points are deleted in real time after processing. We do not collect or store this data on Meta servers [64].	False	Apple Vision Pro measures and stores information on-device about the size and shape of your hands and finger joints to make it easier for you to interact with content [83].
The system only collects your final selection (instead of your hand movements) from the hand tracking data.	False	Abstracted hand-tracking data is available to the system [52]	False	Abstracted hand-tracking data is available to the system [59]	False	Abstracted hand-tracking data is available to the system [68]
The system can understand your hand gestures to perform certain actions (e.g., select, scroll).	True	Built-in function [51]).	True	Built-in function [65].	True	Built-in function [66].

The system can identify which real-world objects you are holding.	True	Access to passthrough camera data is available [53].	False	Identifying real-world objects requires integrating passthrough camera data, which the hand-tracking API does not offer [59].	False	Identifying real-world objects requires integrating passthrough camera data, which the hand-tracking API does not offer [83].
The system can simulate your hand movement for your virtual avatar.	True	Built-in function [52]).	True	Built-in function [59].	True	Built-in function (Persona [83]).
The system can authenticate your identity from the unique aspect of your hand (i.e., fingerprint).	False	Fingerprint authentication function is not supported	False	Fingerprint authentication function is not supported	False	Fingerprint authentication function is not supported
The system can measure the hand size of new users.	True	Built-in function [52]).	True	Built-in function [59].	True	Built-in function [68].

Table A.4: Justification for app-level hand-tracking comprehension questions

Comprehension Question	**HoloLens**	**HoloLens Justification**	**Oculus**	**Oculus Justification**	**Vision Pro**	**Vision Pro Justification**
The application requires your permission to access your hand-tracking data.	False	System automatically grants applications access to the hand-tracking API	False	System automatically grants applications access to the hand-tracking API	True	Run-time app level permission model (Figure A.16)
The application can access your hand-tracking data when running in the background.	True	Built-in function (Figure A.11)	False	Background access for hand-tracking is not supported.	False	Background access for hand-tracking is not supported.
The application can transfer your hand-tracking data to an external device (e.g., a company server).	True	System provides no control over how third-party used user's eye-tracking data	True	...we do not control how a third party app uses, stores, or shares your abstracted hand and body data. [64]	True	It's [developer] responsibility to protect any data your app collects, and to use it in responsible and privacy-preserving ways [55]
The application can retain the unprocessed image of your hand on the AR/MR headset.	True	Access to passthrough camera data is available [53].	False	Only abstracted hand-tracking data is available to the application [59].	False	Only abstracted hand-tracking data is available to the application [68].

The application only collects your final selection (instead of your hand movements) from the hand-tracking data.	False	Abstracted hand-tracking data is available to the application [52]	False	Abstracted hand-tracking data is available to the application [59]	False	Abstracted hand-tracking data is available to the application [68]
The application can understand your hand gestures to perform certain actions (e.g., select, scroll).	True	Built-in function [51]).	True	Built-in function [65].	True	Built-in function [66].
The application can identify which real-world objects you are holding.	True	Access to passthrough camera data is available [53].	False	Identifying real-world objects requires integrating passthrough camera data, which the hand-tracking API does not offer [59].	False	Identifying real-world objects requires integrating passthrough camera data, which the hand-tracking API does not offer [83].
The application can simulate your hand movement for your virtual avatar.	True	Built-in function [52]).	True	Built-in function [59].	True	Built-in function [68].

The application can authenticate your identity from the unique aspect of your hand (i.e., fingerprint).	False	Fingerprint authentication function is not supported	False	Fingerprint authentication function is not supported	False	Fingerprint authentication function is not supported
The application can analyze the hand size of new users.	True	Built-in function [52]).	True	Built-in function [59].	True	Built-in function [68].

Table A.5: Participants' comprehension. The underlined percentages correspond to the correct answer. The red color highlights cases where the most common answer was incorrect. The green color highlights cases where the most common answer was correct. The Hol-Sys column corresponds to the Hololens system version of the question, Hol-App to HoloLens application, Oc-Sys to Oculus system, Oc-App to Oculus application, Vis-Sys to Vision Pro system, Vis-App to Vision Pro application.

Sensor	Category	Permission Comprehension Question	Options	Hol-Sys	Hol-App	Oc-Sys	Oc-App	Vis-Sys	Vis-App
Eye	Privacy	The system (application) requires your permission to access your eye-tracking data.	True	92.5%	91.6%	96.6%	95.5%	87.1%	67.1%
			False	2.8%	4.7%	1.1%	2.2%	9.4%	27.1%
			I Don't Know	4.7%	3.7%	2.3%	2.3%	3.5%	5.9%
		The system allows you to control which application has access to your eye-tracking data.	True	87.9%	N/A	87.5%	N/A	55.3%	N/A
			False	3.7%	N/A	3.4%	N/A	31.8%	N/A
			I Don't Know	8.4%	N/A	9.1%	N/A	13.0%	N/A
		The application can access your eye tracking data when running in the background.	True	N/A	49.5%	N/ A	25.0%	N/A	29.4%
			False	N/A	4.7%	N/A	20.5%	N/A	22.4%
			I Don't Know	N/A	45.8%	N/A	54.5%	N/A	48.2%
		The system (application) can transfer your eye-tracking data to an external device (e.g., a company server).	True	24.3%	21.5%	23.9%	51.1%	23.9%	24.7%
			False	26.2%	17.8%	30.7%	9.1	15.3%	32.9%
			I Don't Know	49.5%	60.7%	39.8%	30.7%	56.5%	42.4%
		The system (application) can retain the unprocessed image of your eye.	True	47.7%	43.9%	35.2%	26.1%	43.5%	47.1%
			False	11.2%	7.5%	14.8%	15.9%	11.8%	16.5%
			I Don't Know	41.1%	48.6%	50.0%	58.0%	44.7%	36.5%
		The system (application) only collects your final selection (instead of your eye movements) from the eye tracking data.	True	19.6%	21.5%	20.5%	23.9%	18.8%	23.5%
			False	26.2%	22.4%	25.0%	22.7%	18.8%	25.9%
			I Don't Know	54.2%	56.1%	54.5%	53.4%	62.4%	50.6%

	Utility	The system (application) can understand where your eyes look to indicate which virtual object to select.	True	91.6%	72.9%	93.2%	93.2%	95.3%	81.2%
			False	0.9%	6.5%	3.4%	3.4%	1.2%	8.2%
			I Don't Know	7.5%	20.6%	3.4%	3.4%	3.5%	10.6%
		The system (application) can identify which real-world objects you are looking at.	True	41.1%	34.6%	39.8%	44.3%	43.5%	40.0%
			False	19.6%	25.2%	34.1%	34.1%	24.7%	29.4%
			I Don't Know	39.3%	40.2%	26.1%	21.6%	31.8%	30.6%
		The system (application) can simulate your eye movement for your virtual avatar.	True	68.2%	59.8%	93.2%	95.5%	69.4%	67.1%
			False	8.4%	13.1%	3.4%	2.3%	5.9%	8.2%
			I Don't Know	23.4%	33.3%	3.4%	2.3%	24.7%	24.7%
		The system (application) can authenticate your identity from the unique aspect of your eye (i.e., iris).	True	84.1%	61.7%	22.7%	25.0%	85.9%	72.9%
			False	5.6%	9.3%	35.2%	34.1%	4.7%	15.3%
			I Don't Know	10.3%	29.0%	42.0%	40.9%	9.4%	11.8%
		The system can adjust eye calibration for new users. The application can access users' eye calibration data	True	79.4%	75.7%	86.4%	84.1%	76.5%	65.9%
			False	7.5%	4.7%	1.1%	4.5%	2.4%	14.1%
			I Don't Know	13.1%	19.6%	12.5%	11.4%	21.2%	20.0%
	Privacy	The system (application) requires your permission to access your hand-tracking data.	True	57.9%	71.0%	89.8%	55.7%	82.4%	96.5%
			False	32.7%	21.5%	3.4%	34.1%	12.9%	1.2%
			I Don't Know	9.3%	7.5%	6.8%	10.2%	4.7%	2.4%
		The system allows you to control which application has access to your hand-tracking data.	True	39.3%	N/A	40.9%	N/A	83.5%	N/A
			False	45.8%	N/A	38.6%	N/A	5.9%	N/A
			I Don't Know	15.0%	N/A	20.5%	N/A	10.6%	N/A
		The application can access your hand-tracking data when running in the background.	True	N/A	87.9%	N/ A	55.7%	N/A	37.6%
			False	N/A	3.7%	N/A	9.1%	N/A	14.1%
			I Don't Know	N/A	8.4%	N/A	35.2%	N/A	48.2%
		The system (application) can transfer your hand-tracking data to an external device (e.g., a company server).	True	21.5%	27.1%	35.2%	34.1%	23.5%	22.4%
			False	20.6%	15.9%	8.0%	11.4%	12.9%	12.9%

Hand

			I Don't Know	57.9%	57.0%	56.8%	54.5%	63.5%	64.7%
		The system (application) can retain the unprocessed image of your hand	True	46.7%	48.6%	48.9%	55.7%	49.4%	61.2%
			False	12.1%	5.6%	9.1%	5.7%	10.6%	7.1%
			I Don't Know	41.1%	45.8%	42.0%	38.6%	40.0%	31.8%
		The system (application) only collects your final selection (instead of your hand movements) from the hand-tracking data.	True	12.1%	20.6%	19.3%	19.3%	24.7%	22.4%
			False	27.1%	21.5%	27.3%	28.4%	9.4%	14.1%
			I Don't Know	60.7%	57.9%	53.4%	52.3%	65.9%	63.5%
	Utility	The system (application) can understand your hand gesture to perform certain actions (e.g., select, scroll).	True	92.5%	94.4%	100.0%	96.6%	95.3%	89.4%
			False	0.9%	0.9%	0.0%	1.1%	0.0%	1.2%
			I Don't Know	7.5%	4.7%	0.0%	2.3%	4.7%	9.4%
		The system (application) can identify which real-world objects you are holding.	True	42.1%	43.9%	28.4%	30.7%	40.0%	32.9%
			False	18.7%	17.8%	28.4%	28.4%	22.4%	24.7%
			I Don't Know	39.3%	38.3%	43.2%	40.9%	37.6%	42.4%
		The system (application) can simulate your hand movement for your virtual avatar.	True	77.6%	74.8%	94.3%	93.2%	80.0%	72.9%
			False	3.7%	4.7%	1.1%	2.3%	2.4%	3.5%
			I Don't Know	18.7%	20.6%	4.5%	4.5%	17.6%	23.5%
		The system (application) can authenticate your identity from the unique aspect of your hand (i.e., fingerprint).	True	24.3%	36.4%	22.7%	31.8%	41.2%	42.4%
			False	31.8%	21.5%	34.1%	28.4%	21.2%	24.7%
			I Don't Know	43.9%	42.1%	43.2%	39.8%	37.6%	32.9%
		The system (application) can measure the hand size of new users.	True	42.1%	40.2%	59.1%	62.5%	64.7%	64.7%
			False	15.0%	12.1%	13.6%	14.8%	7.1%	4.7%
			I Don't Know	43.0%	47.7%	27.3%	22.7%	28.2%	30.6%

Appendix B

Additional material for: Exploring User Reactions and Mental Models Towards Perceptual Manipulation Attacks in Mixed Reality

B.1 Recruitment & Screening Survey

Our primary recruiting messages were short, announcing the study and sharing a link to our recruiting survey, which provided significantly more information (see below). The recruiting messages were sent to members of our institution over Slack, mailing lists, or other private messages.

The recruiting message: "Hello everyone, I am looking for students who might be interested in participating in a user study wearing a Mixed Reality headset. The goal is to compare your performance on certain tasks with or without the MR headset. We will follow necessary COVID precautions with open windows in the user study room. The study is around 60 minutes and we will pay you $30 in Amazon gift card for your valuable time. Please let me know if you have any questions."

The full screening survey: "Thank you for taking the survey. We are a group of researchers from the University of Washington, Paul G. Allen School, and we are hoping to evaluate the impact of wearing a mixed reality headset while conducting a primary task. We appreciate your interest in our experiment and would like to conduct a quick survey beforehand. This study will take place (with COVID-19 precautions in place) on the University of Washington campus. We will reach out to you to schedule the experiment separately. This study has been reviewed by the University of Washington Human Subjects Review Board (IRB)."

1. How many times have you used a AR/MR/VR headset (HTC Vive, Oculus Rift, Windows Mixed Reality, etc.)?
   (i) I never tried it.
   (ii) I tried it a few times.
   (iii) I am a regular user.
   (iv) I use it everyday.
2. How do you feel when doing tasks in AR/MR/VR? [Open-ended]
3. Do you experience nausea when using AR/MR/VR headsets?
   (i) N/A or I'm not sure.
   (ii) No.
   (iii) Yes.
4. Do you feel eye strain when using AR/MR/VR headsets?
   (i) N/A or I'm not sure.
   (ii) No.
   (iii) Yes.
5. Do you feel dizziness when using AR/MR/VR headsets?
   (i) N/A or I'm not sure.
   (ii) No.
   (iii) Yes.
6. The headset we are using for our experiment is not compatible with some types of glasses frames. If you participate in the experiment, will you be able participate without glasses?
   (i) I don't need glasses/contacts.
   (ii) I will wear contacts to the experiment.
   (iii) I will wear glasses and if they are incompatible, I will participate without them.
   (iv) I'm not sure.

B.2 Interview Script

Notes: As is standard with semi-structured interviews, not all interviews followed exactly this script, as researchers may have followed up on participants' responses or otherwise reordered, omitted, or adapted questions according to the context in the moment. We began each study by following COVID-19 safety procedures (e.g., sanitizing equipment).

B.2.1 Warm Up Phase

Thank you for participating in our research. Before we begin the study, we'd like to give you a chance to review and sign this consent form. This study has been approved by UW Human Subject Research review board. You may experience mild discomfort from using the mixed reality device or some level of motion sickness or vertigo. We will make sure that your MR headset is adjusted correctly to minimize these risks. You will also be asked to stay seated during the task, minimizing the risk of motion sickness or bumping into any real-world objects. You may choose to end the experiment at any time, without loss of promised compensation.

With your permission, we'd like to video record the study. You can still participate in the study even if you'd prefer not to be audio or screen recorded, and you can ask us to delete the recording at any time later.

This study will have three parts: We prepare a demo app in Mixed Reality to get you used to the environment, and will ask you some follow up questions. Then we will ask you to conduct three different tasks both with and without the MR headset.

At last, we will have a comprehensive discussion at the end of our experiment to learn about your experience.

1. What do you think about MR?
2. Tell us a bit about your prior MR exposure, including devices or apps that you have used or observed others using, as well as in literature or film that you have seen.
3. How do you feel about completing tasks in MR?
4. If you don't feel comfortable completing tasks in MR, what concerns do you have?

B.2.2 Experiment Phase

In this part of the study, we'd like you to try some games with and without the MR headset. We are not comparing your performance with others, and we focus on evaluating this technology approach. But we still hope that you try your best.

As you are completing the tasks in MR, feel free to vocalize any reactions. If you experience any buggy situation, please feel free to vocalize them as well — we won't interrupt your task to answer them, but we'd be happy to discuss them later on. Again, if you experience any severe dizziness or discomfort, feel free to end the experiment at any time.

B.2.3 Post-Task Interview Phase

1. How do you like the MR experience?
2. What stood out to you the most?
3. [One researcher selected one or more of the participant's experimental results to describe to them.] Is there anything that you think impacted your performance in these experiments?
4. If you were affected by the content, could you go over that moment and elaborate on it?
5. If you noticed the misleading content and successfully performed the task, could you go over that moment and elaborate on it?
6. What would you attribute the misleading content to?
7. [If participants talked about bugs and attacks] At what points do you feel the content is buggy vs the content is actually an attack?
8. What mitigating strategy did you use during the attacks?

B.2.4 Debrief

Now we would love to debrief with you our research purpose. The goal of our research is to evaluate whether it is possible to design mixed reality applications that mislead participants given today's technology, and measure its efficacy based on your performance. Our assumption is that, under a time or attention limited condition, people may rely on their instinct or intuition to make decisions. If the virtual generated objects blending in our physical world are similar enough to real ones, they have the ability to trigger our intuition to either make false judgement, or impact our performance.

B.2.5 Post Debrief Questions

- Reflect on their performance when PMA occurred.
- How has this experiment changed your trust towards AR/MR/VR?
- How will this type of technology affect our daily life in ten years?
- Do you have any concerns about adapting this technology in your daily life?

B.3 Qualitative Codebook

The full codebook, with themes and subthemes, from qualitatively analyzing user reflection on PMA. Codes were not mutually exclusive.

Attribution of attacks

- The attack was a bug from the device or a glitch from the application.
- The attack was a part of the real world.
- The researcher deliberately programmed the attack for some reason.
- Identify the purpose of the attack and this study.

Self-reported impact of attacks

- Thought they were not impacted by the attacks.
- Thought they were impacted by the attacks because they didn't know how to proceed.
- Thought they were impacted by the attacks because they were distracted by the attacks.
- Thought they were impacted by the attacks because they believed the attack content was a part of the real world.
- Thought they were impacted by the attacks because they believed the attacks were in a different modality (audio).
- Didn't notice the attack.

Developed defensive strategies

- Focus more on the task.
- Concentrate on non-affected areas.

- Mentally filter out the attack content.
- Learn from past attacks and ignore them in later tasks.
- Try to swipe the attack content away.

User reflection on effectiveness of defensive strategies

- Thought the strategies were useful.
- Thought the strategies made them more cautious and thus react slower.
- Thought the strategies were not sufficient, and thus user was still affected by attacks.
- Thought the strategies they developed for one attack backfired against another attack.