UCLA
UCLA Electronic Theses and Dissertations

Title
Toward aTrustworthy Metaverse

Permalink
https://escholarship.org/uc/item/74k2b86h

ISBN
9798247966180

Author
Cai, Kunlin

Publication Date
2026-06-03

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA

Los Angeles

Toward a Trustworthy Metaverse

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Electrical and Computer Engineering

by

Kunlin Cai

2026

© Copyright by

Kunlin Cai

2026

ABSTRACT OF THE DISSERTATION

Toward a Trustworthy Metaverse

by

Kunlin Cai

Doctor of Philosophy in Electrical and Computer Engineering

University of California, Los Angeles, 2026

Professor Yuan Tian, Co-Chair

Professor Xiang Chen, Co-Chair

The Metaverse is widely understood as a persistent, shared, immersive virtual space that integrates physical and digital realities. As the Internet evolves toward this paradigm, the convergence of spatial computing and machine learning is reshaping human-computer interaction. The resulting ecosystem rests on two complementary capabilities: a spatial interaction layer that captures continuous, embodied user kinematics, and a cognitive intelligence layer that synthesizes this data through predictive and generative models. Together they enable immersive experiences but also introduce system-level risks. Without careful design, the Metaverse may evolve into a pervasive sensing and inference infrastructure with expanded attack surfaces, opaque data practices, and entrenched centralization across both the spatial and cognitive layers.

This dissertation reexamines the security and privacy foundations of Metaverse applications around this dual-pillar architecture. Rather than relying on reactive, point-wise defenses, it advocates for a proactive approach that prioritizes trustworthiness at the architectural level.

For the spatial interaction layer, we bridge vulnerability discovery with structural reform. We empirically uncover a novel class of side-channel threats showing that seemingly benign VR kinematics expose sensitive user intentions. To understand why such vulnerabilities persist, we conduct an in-depth interview study with 23 professional XR developers and surface ecosystem-level root causes. Finally, instead of treating decentralization as a binary switch between keeping the central server and removing it entirely, we redistribute each role the central operator plays—identity,

shared state, behavior auditing—to system participants one role at a time. We ground this design in a 22-participant interview study of user-stated needs, and a 20-participant within-subjects double-blind comparison against a centralized baseline returns no evidence of a perceptual gap on the constructs participants themselves consider load-bearing for a usable collaborative experience.

For the cognitive intelligence layer, we establish rigorous privacy boundaries using model-centric attacks. We show that predictive spatial-temporal models leak precise user trajectories, and that fine-tuned zero-shot Text-to-Speech models expose biometric voice features at both speaker-level identity and record-level verbatim granularities, with attack performance grounded in an information-theoretic characterization of the optimal query.

Together, these contributions move the Metaverse toward an architecture that is secure, accountable, and user-sovereign by design.

The dissertation of Kunlin Cai is approved.

Mohammad Rostami

Habiba Farrukh

Nader Sehatbakhsh

Yuan Tian, Committee Co-Chair

Xiang Chen, Committee Co-Chair

University of California, Los Angeles

2026

“What is now proved was once only imagined.”

—William Blake

Table of Contents

List of Acronyms . . . . . **xviii**

1 Introduction **1**

1.1 Contributions 6

2 Background **8**

2.1 The Metaverse: A Dual-Pillar Ecosystem 8

2.2 The Spatial Interaction Pillar 10

2.2.1 Mechanics and Transformative Applications 10

2.2.2 Security and Privacy Landscape in Immersive Interactions 11

2.2.3 Usable Security and the Immersive Development Ecosystem 12

2.2.4 Infrastructure Design and Decentralized Architectures 13

2.3 The Cognitive Intelligence Pillar 15

2.3.1 Predictive Services and Semantic Reasoning 15

2.3.2 Generative Intelligence and Biometric Synthesis 17

2.3.3 Privacy Auditing of Machine Learning Services 17

3 Identify Threats: Vulnerabilities in Immersive Interactions **21**

3.1 Background on Multi-user VR Applications 23

3.2 Threat Model 25

3.3 Methods 27

3.3.1 Step 1: Packet Extraction 28

3.3.2 Step 2: Field Extraction 29

3.3.3 Step 3: Semantics Extraction 30

3.3.4 Step 4: Click Position Extraction . . . 31
3.3.5 Reverse Engineering Challenges . . . 33
3.4 Data Collection . . . 37
3.5 Evaluation . . . 40
3.5.1 Our Attack is Robust Under Traffic Loss . . . 42
3.5.2 Keyboard Layout Affects Attack Accuracy . . . 43
3.5.3 Our Attack Remains Robust Under Varying Typing Speed . . . 44
3.5.4 Our Attack Can Be Stealthy Under Practical Scenarios . . . 45
3.5.5 Our Attack Generalizes Across Applications . . . 46
3.5.6 Per-Participant and Per-Hand Variation . . . 46
3.6 Keylogging is Possible Even with Partial Reverse Engineered Packets . . . 48
3.7 Discussion . . . 54
3.8 Chapter Summary . . . 57

4 Diagnosing Causes: A Developer-centered Study of XR S&P . . . 59
4.1 Methods . . . 62
4.1.1 Recruitment . . . 62
4.1.2 Study Design . . . 63
4.1.3 Qualitative Analysis . . . 65
4.2 Developers' Perceptions on Privacy in XR . . . 69
4.2.1 Developers have limited awareness of sensitive data in XR . . . 70
4.2.2 What makes data overall more sensitive in XR . . . 70
4.2.3 What makes certain data less sensitive . . . 72
4.2.4 What makes leakage channels more realistic in XR . . . 73
4.2.5 What makes certain leakage channels less realistic in XR . . . 74

4.2.6 Analysis: Potential misconceptions on privacy in XR 75
4.3 Developers' Perceptions on Security in XR 75
4.3.1 Developers exhibited limited awareness of XR attacks 76
4.3.2 What makes attacks overall more important in XR 77
4.3.3 What makes certain attacks less important in XR 79
4.3.4 What makes attacks overall more practical in XR 79
4.3.5 What makes certain attacks less practical in XR 80
4.3.6 Potential misconceptions on security in XR 81
4.4 Developers' perceptions on current mitigation practices and supports on XR S&P . 82
4.4.1 Limitations of current practices 82
4.4.2 Absence of supports from the XR community 84
4.4.3 Analysis: Misconceptions on the supports for S&P in XR 87
4.5 Discussion 88
4.5.1 Awareness gaps due to the rapidly evolving nature of XR interaction technologies 88
4.5.2 Diffusion of responsibility due to the user-experience oriented nature of XR threats 89
4.6 Chapter Summary 90

5 Architectural Reconstruction: A Decentralized Metaverse Framework 92
5.1 Background and Related Work 95
5.1.1 Decentralized Metaverse 95
5.1.2 Decentralized Applications 96
5.1.3 Human-Subjects Studies of Decentralized Technologies 97
5.2 Methods 98
5.2.1 User-Study Method 99

5.2.2 Decentralized Framework Design and Validation . . . 102
5.3 User-Study Results . . . 115
5.3.1 Threats that drive the demand for decentralization . . . 116
5.3.2 Ownership properties decentralization is asked to provide . . . 118
5.3.3 How users reconstitute trust without a central operator . . . 120
5.3.4 When users will switch from centralized to decentralized . . . 123
5.3.5 Folk models users bring to decentralization . . . 125
5.4 Validation Results . . . 128
5.4.1 Sample . . . 129
5.4.2 Subjective ratings: no evidence of a perceptual gap . . . 129
5.4.3 Architectural-invisibility check: identification did not exceed chance (N = 10 dyads) . . . 131
5.4.4 Debrief: semantic-correctness rollbacks went unnoticed . . . 131
5.4.5 Update round-trip latency: indistinguishable on the mean . . . 131
5.4.6 Synthesis . . . 133
5.5 Discussion . . . 133
5.5.1 Rethinking Decentralization in the Metaverse . . . 135
5.5.2 Folk Models of Decentralized Trust . . . 136
5.5.3 Scope, Scale, and Architectural Visibility . . . 138
5.5.4 Grounding System Design in User-Stated Needs . . . 139
5.5.5 Limitations and Future Work . . . 140
5.6 Chapter Summary . . . 142

6 Auditing Predictive Models: Privacy Leakage in Spatial-Temporal Intelligence . . . 144

6.1 Point-of-Interest Recommendation . . . 147

6.2 Threat Models . . . 148
6.3 Attack Suite . . . 149
6.3.1 Data Extraction Attacks . . . 149
6.3.2 Membership Inference Attacks . . . 151
6.3.3 Practical Implications of the Attack Suite . . . 153
6.3.4 Attack Algorithms . . . 154
6.4 Experiments . . . 156
6.4.1 Experimental Setup . . . 156
6.4.2 Implementation Details . . . 159
6.4.3 Experimental Results and Analysis . . . 161
6.4.4 Vulnerability Factor Analysis: Methodology Details . . . 169
6.4.5 The Impact of Training and Attack Parameters . . . 170
6.5 Defense . . . 180
6.5.1 Defense Techniques . . . 180
6.5.2 Takeaway Messages from the Defense . . . 182
6.5.3 Defense Implementation and Evaluation Details . . . 182
6.6 Discussion . . . 188
6.6.1 Broader Impacts . . . 188
6.6.2 Related Work . . . 189
6.6.3 Related Work on Defenses . . . 189
6.7 Chapter Summary . . . 190

7 Auditing Generative Models: Privacy Boundaries in Biometric Content Generation 192

7.1 Background . . . 194
7.1.1 Text-to-Speech (TTS) Models . . . 195

7.1.2 Membership Inference Attacks . . . 196
7.1.3 Threat Model . . . 197
7.2 Methodology . . . 198
7.2.1 Challenges of Naïve Adaptation . . . 198
7.2.2 Query Generation (G_Q) . . . 199
7.2.3 Representation Engineering (ϕ) . . . 203
7.3 Experiment . . . 208
7.3.1 Experiment Setup . . . 208
7.3.2 Attack Performance . . . 210
7.3.3 Ablation and Analysis . . . 213
7.3.4 Data Characteristics and Vulnerability . . . 222
7.4 Related Work . . . 227
7.5 Discussion . . . 228
7.5.1 Potential Defenses and Their Limitations . . . 228
7.6 Chapter Summary . . . 230

8 Conclusion and Future Work . . . 232
8.1 Summary of Contributions . . . 233
8.1.1 The Spatial Interaction Pillar . . . 233
8.1.2 The Cognitive Intelligence Pillar . . . 234
8.1.3 The Broader Argument: From Reactive Patching to Structural Trust . . . 235
8.2 Future Research Directions . . . 235
8.2.1 Privacy-Preserving ML for Spatial-Temporal Data . . . 235
8.2.2 Expanding ML Robustness: From Privacy to Adversarial Security . . . 236
8.2.3 Trustworthy World Models for Spatial Computing . . . 236

8.2.4 Generative Model Auditing at Scale 237
8.2.5 Human-Centric Security at the Intersection of HCI and ML 237
8.2.6 Toward Principled Trust Protocols 237
8.3 Final Remarks 238

References **239**

List of Figures

2.1 Metaverse Dual-Pillar Architecture 9

2.2 Example of user head motion mapped in 6DOF 11

2.3 Transformative applications of spatial analytics and mobility data processing 16

3.1 Motion data flow in multi-user VR applications. 24

3.2 Example of a participant typing in Rec Room using a virtual keyboard. 25

3.3 Our approach consists of four extraction steps to extract keystrokes from network traffic. 28

3.4 Cursor-keyboard intersection for keystroke logging 33

3.5 Mapping hand transforms in network packets to cursor transforms 35

3.6 Geometric Test 1: validating guessed cursor orientation 36

3.7 Our attack is robust against random packet drop 43

3.8 Far keys are harder to infer than near keys 44

3.9 Our attack is unaffected by typing speed 45

3.10 Attack accuracy varies slightly across participants due to typing habits 47

3.11 The attack accuracy is not affected by the hand the victim uses to type. 48

3.12 Machine Learning Attack Overview 49

4.1 Curated XR attack examples shown to interview participants 60

5.1 Ownly instance overview 106

5.2 Subjective Likert ratings, decentralized vs centralized 130

5.3 Per-update round-trip latency, decentralized vs centralized 132

6.1 Privacy attack suite against POI recommendation models 145

6.2 Data-extraction attack performance across three victim models and two datasets . . 162

6.3 Attack performance of LocMIA and TrajMIA across three victim models and two datasets . . . 163
6.4 How location-level aggregate statistics are related to LocMIA (4sq) . . . 166
6.5 How user-level aggregate statistics are related to TrajMIA (4sq) . . . 167
6.6 How trajectory-level aggregate statistics are related to TrajMIA (4sq) . . . 168
6.7 Optimal query timestamp improves LocExtract and TrajExtract (4sq) . . . 168
6.8 Query-timestamp and query-location counts affect LocMIA performance . . . 169
6.9 How user-level aggregate statistics are related to TrajExtract . . . 170
6.10 How user-level aggregate statistics are related to TrajMIA (Gowalla) . . . 171
6.11 How location-level aggregate statistics are related to LocMIA (Gowalla) . . . 172
6.12 How trajectory-level aggregate statistics are related to TrajMIA (Gowalla) . . . 172
6.13 The impact of the model generalization on the performance of four privacy attacks. 174
6.14 LocExtract is effective within a small query budget . . . 175
6.15 TrajExtract is effective with a small beam size . . . 176
6.16 Optimal query timestamp improves LocExtract and TrajExtract (Gowalla) . . . 176
6.17 Comparing soft voting with hard voting for logits aggregation in LocExtract . . . 177
6.18 The location sequences of shorter (sub-)trajectories are more vulnerable to TrajExtract. . . . 177
6.19 LocMIA performance improves with more shadow models . . . 178
6.20 TrajMIA performance improves with more shadow models . . . 179
6.21 The longer trajectories are less vulnerable to TrajMIA. . . . 180
6.22 Defense performance on protecting all corresponding sensitive information for each attack. . . . 184
6.23 Defense performance on protecting the targeted subset of sensitive information for each attack. . . . 185

6.24 Geo-Ind defense outperforms JFT for LocExtract on common-location ground truths 188

7.1 DTW-based temporal alignment for record-level MIA 207

7.2 Attack performance of speaker-level and record-level MIA across three TTS models and two datasets 210

7.3 Record-level MIA maintains low-FPR performance against seen-speaker non-members 212

7.4 Exact match queries consistently outperform weaker strategies for speaker-level MIA 213

7.5 Exact match queries substantially outperform mismatched audio for record-level MIA. 213

7.6 SV features best expose speaker-level leakage, while WavLM best exposes record-level leakage. 215

7.7 Temporal concatenation outperforms embedding averaging for speaker-level MIA . 216

7.8 Multi-level features improve record-level MIA. 217

7.9 DTW substantially outperforms alternative alignment methods for record-level MIA 218

7.10 Speaker-level leakage is driven by the flow-matching acoustic decoder 219

7.11 Speaker-level MIA reaches strong performance with few attacker utterances 220

7.12 Effect of query budget on speaker- and record-level MIA 220

7.13 Male speakers are more vulnerable to speaker-level MIA than female speakers . . . 222

List of Tables

3.1 Keylogging attack accuracy from 20 participants across three different typing tasks. 41

3.2 Keylogging accuracy across three additional VR applications 46

3.3 Comparison of ML models for keystroke inference from partially reversed raw bytes 51

3.4 Effect of training-data size on decoding accuracy 52

3.5 Secrets typed with the right hand are slightly more vulnerable to our machine-learning models. 53

3.6 Attack accuracy for different typing tasks with partially reconstructed data. 53

4.1 Participant overview – 23 professional XR developers in the security and privacy interview study 63

4.2 Data leakage channels in XR collected from the literature and prior discussions . . 64

4.3 Sensitive data categories in XR summarized from the literature 66

4.4 XR security and privacy attacks summarized from the literature 67

4.5 Mitigation tools and best practices for XR security and privacy. 68

5.1 Participant overview – 22 end-users in the decentralized-metaverse interview study 100

6.1 Performance of POI-recommendation victim models 154

6.2 Dataset statistics for the two POI-recommendation benchmarks 160

6.3 Sensitive-information exposure per attack 182

7.1 Threat model assumptions across auditing granularities 198

7.2 Effect of multi-query statistics on record-level MIA 221

7.3 Record-level MIA performance across silence ratio bins; lower silence ratio increases record-level risk. 223

7.4 Record-level MIA performance across phoneme density bins; higher phoneme density can increase risk. 224

7.5 Record-level MIA performance across high-frequency energy ratio bins; richer high-frequency content consistently increases vulnerability. 225

7.6 Record-level MIA performance across record-duration bins; longer records show higher vulnerability. 225

7.7 Record-level MIA performance across speech energy bins; higher energy recordings are more vulnerable. 226

7.8 Attack performance under early stopping defense for speaker-level and record-level MIA. 228

7.9 Attack performance under perturbation-based defenses. 229

7.10 Attack performance under DP-SGD defense for speaker-level and record-level MIA at two privacy budgets. 230

List of Acronyms

AR	Augmented Reality
ASR	Attack Success Rate
AUC	Area Under the Curve
CRDT	Conflict-free Replicated Data Type
DNN	Deep Neural Network
DP	Differential Privacy
DP-SGD	Differentially Private Stochastic Gradient Descent
DTW	Dynamic Time Warping
ETD	Electronic Thesis and Dissertation
FPR	False Positive Rate
GDPR	General Data Protection Regulation
HMD	Head-Mounted Display
IRB	Institutional Review Board
LBS	Location-Based Service
LDP	Local Differential Privacy
LiRA	Likelihood Ratio Attack
LLM	Large Language Model
LSTM	Long Short-Term Memory
MIA	Membership Inference Attack
ML	Machine Learning
NDN	Named Data Networking
POI	Point-of-Interest
RE	Reverse Engineering

RNN	Recurrent Neural Network
S&P	Security and Privacy
TPR	True Positive Rate
TTS	Text-to-Speech
VR	Virtual Reality
XR	Extended Reality

Acknowledgments

I am deeply grateful to my advisor, Professor Yuan Tian, for her guidance and support throughout my five-year doctoral journey. I am especially thankful for her trust in allowing me to pursue research in two distinct areas—Trustworthy XR and Machine Learning—and for instilling in me the rigor and intellectual discipline reflected throughout this dissertation.

I thank the rest of my doctoral committee—Professors Xiang "Anthony" Chen, Nader Sehatbakhsh, and Habiba Farrukh, together with Dr. Mohammad Rostami—for their thoughtful feedback, careful review of this work, and insightful questions that strengthened the dissertation in ways I could not have achieved alone.

I especially want to thank Jianfeng Chi and Fnu Suya for their mentorship, particularly when I was just beginning my PhD journey and exploring unfamiliar directions. I also thank my labmates Jinghuai Zhang, Ying Li, Tamjid Rahat, Tu Le, Faysal Hossain Shezan, Kaiyuan Zhang, Zihang Xiang, Peiran Wang, Holly Zhang, Will Shank, Sean Tang, and Nasir Hussain for collaborating with me and going through this process together. I am also grateful to my collaborators Jerry Su, Zhiqing Hong, Guang Wang, Rongchao Xu, Tianyuan Yu, McKenna McCall, Ilya Grishchenko, Rui Song, Tianshi Li, Benjamin Seligman, Cathy Lee, Jiaqi Ma, Giovanni Vigna, Christopher Kruegel, Lixia Zhang, Limin Jia, and Lujo Bauer for the chance to work alongside them; their ideas, expertise, and generosity made this research far richer than I could have managed on my own.

I gratefully acknowledge the Amazon Fellowship (2025), which funded the generative-model security and privacy work in Chapter 7, and the administrative support of the UCLA Department of Electrical and Computer Engineering staff.

Finally, I want to thank my parents, Man Cai and Yanchun Chen, for the sacrifices I only understood once I was far from home, and for supporting me unconditionally; my cats, Mandu and Oolong, for keeping me company along the way; and my friends, for making the long stretches between milestones a lot more livable. And most of all, my wife, Xiaohan Chen—for your patience, for holding everything else together while I chased this to the finish, for standing by me through every failure and celebrating every success with me, and for reminding me that work isn't everything.

Vita

2016–2021	B.A. and M.S. in Computer Science, Boston University
2021–2022	Ph.D. studies in Computer Science, University of Virginia
2022–2026	Ph.D. studies in Electrical and Computer Engineering, University of California, Los Angeles
2025	Amazon Fellow

CHAPTER 1

Introduction

The Metaverse represents a persistent, shared spatial environment that integrates physical and digital realities [Mys22, LBZ21]. Driven by immersive technologies such as extended reality (VR/AR) and advanced machine learning, it functions as a complex computational platform where users interact, socialize, and create in real time [LBZ21]. Compared with the 2D internet that came before, the Metaverse adds three properties that operate at the same time: embodied immersive interaction, in which VR/AR hardware streams continuous, high-dimensional kinematic, gaze, voice, and spatial telemetry; persistent multi-user shared state, in which avatars broadcast embodied behavior across co-present peers for cross-peer synchronization; and tightly-coupled algorithmic inference, in which predictive and generative ML pipelines consume the resulting streams in real time. These three properties are the conditions under which immersive social presence becomes possible, and together they make the Metaverse substantially more complex than the 2D internet and substantially more demanding to secure.

This shift has attracted substantial investment. Projections indicate that by 2030, the Metaverse will generate over $5 trillion in economic value and reach a global user base of up to 5 billion [McK22, Cit22]. As spatial computing matures, this growth is driven by the convergence of immersive interfaces and intelligent backend systems, fundamentally changing how humans interact with computers and with each other. The same convergence also opens two new surfaces of privacy risk at a new scale: the rich, fine-grained data continuously collected from users (e.g., embodied telemetry, gaze, voice, and behavioral signals), and the predictive and generative ML models trained on it. Recent surveys characterize this concentration as a defining trust challenge for the field [WSZ23a, GNS23], and the work in this dissertation addresses it directly across both the interaction and intelligence pillars.

While prior work defines the Metaverse at a conceptual level, its realization consistently relies on two fundamental capabilities: real-time, embodied interaction and intelligent, adaptive system behavior. Building on this observation, we conceptualize the Metaverse as a **Dual-Pillar Architecture**, consisting of the spatial interaction layer and the cognitive intelligence layer. This abstraction aligns with prior work, which commonly distinguishes between immersive infrastructures (e.g., XR, rendering, networking) and AI-driven services (e.g., personalization, prediction, and generative modeling) [LBZ21, DBG13, GR24].

- **The Spatial Interaction Pillar**: This pillar constitutes the perceptual and embodiment substrate of the Metaverse. Driven by VR/AR technologies and 3D game engines, it continuously captures ultra-high-frequency, multi-modal user kinematics, including motion, gaze, and gesture signals. By transforming these streams into synchronized avatar behaviors and environmental feedback, it establishes the core properties of presence, embodiment, and physical immersion [CTK23, CZL25].
- **The Cognitive Intelligence Pillar**: Complementing perception with reasoning, this pillar functions as the semantic and generative core of the Metaverse. Built upon advanced machine learning and generative AI models, it processes and abstracts the rich multimodal signals produced by the interaction layer to enable adaptive, predictive, and personalized experiences. This includes capabilities such as user intent inference, spatial-temporal prediction, and biometric or content generation, which collectively endow the Metaverse with intelligence and autonomy [BQS25].

As this dual-pillar architecture erodes traditional security boundaries, it catalyzes a threat landscape that is substantially more demanding than the 2D-web setting along three axes [WSZ22, GNS23]: (i) adversaries observe *continuous*, multi-modal embodied streams (kinematics, gaze, voice, spatial geometry) rather than discrete click-or-keystroke events; (ii) the multi-user, shared-state operation of the Metaverse means these streams are broadcast across co-present peers for avatar synchronization, turning a signal that could remain local into one that is network-visible; and (iii) the same streams are immediately ingested by predictive and generative models whose memorization compounds the leakage. Each contribution of this dissertation targets a class of threat that is

significantly worsened when these three operating conditions reinforce one another in a deployed Metaverse system rather than appearing in isolation.

Within the *spatial interaction layer*, the continuous capture of fine-grained interaction data is essential for delivering immersive experiences, yet the rapid deployment of this technology has far outpaced our understanding of its security implications. The Metaverse setting makes this gap particularly acute. High-dimensional VR/AR telemetry that is already sensitive at the device level is broadcast across co-present peers for avatar synchronization, turning it into a remote, network-visible attack surface that prior 2D-web threat models do not anticipate. These are precisely the conditions under which the kinematic side channels documented in Chapter 3 become exploitable. While recent research has begun to identify specific threats, current efforts remain confined to isolated discovery and reactive defenses, leaving the same classes of vulnerabilities to persist across platforms. This persistence points to the need for a systematic investigation into the root causes embedded within the development ecosystem itself. More broadly, a trustworthy Metaverse requires more than patching individual flaws. Confronted with the compounding risks of centralized data collection over persistent shared spaces and the structural potential for surveillance, the ecosystem requires a shift in system design, one that natively distributes data control, safeguards user sovereignty, and lets developers build applications that are secure by default.

At the *cognitive intelligence layer*, the widespread adoption of deep neural networks introduces a separate class of privacy risks. Traditional leakage vectors such as re-identification and linkage attacks have been extensively studied, but they are insufficient to capture the vulnerabilities inherent to modern machine learning. Predictive and generative models deployed in the Metaverse tend to memorize the high-dimensional behavioral and physiological data they continuously ingest, and the Metaverse setting raises the stakes in two specific ways. First, the data is *sequential and semantic*: continuous spatial-temporal trajectories with behavioral context, rather than isolated GPS pings. Second, the generated outputs are themselves *identity-bearing*: synthesized avatar voices function as the user's vocal identity in shared social spaces. Bridging this gap requires a corresponding shift in threat assessment. Rather than focusing on data transmission alone, we have to evaluate the models themselves as potential leakage surfaces, using model-centric auditing tools, in particular Membership Inference Attacks (MIA) and data extraction [SSS17a, CHN23, SCE23]. Rigorously

quantifying how centralized models expose sensitive user information is a prerequisite for any meaningful privacy guarantee at the intelligence layer.

To secure this dual-pillar architecture, this thesis proposes a comprehensive security and privacy framework for the Metaverse. Rather than relying on reactive patches, we address the root causes of centralized control and model opacity through architectural and algorithmic innovations. To build a trustworthy ecosystem where immersive interactions and AI safely coexist, the core contributions of this dissertation are organized into two synergistic research lines:

- **Research Line 1: System & Interaction (The Spatial Interaction Ecosystem).** Dedicated to the interactive layer, this line aims to cultivate a secure application development ecosystem through a rigorous, three-stage approach:
  - (1) Vulnerability Discovery (Chapter 3): We investigate the attack surfaces introduced by immersive interaction in the multi-user Metaverse. Because the Metaverse broadcasts embodied behavior across co-present peers for avatar synchronization, the ultra-high-frequency VR kinematic streams (e.g., head and hand tracking) that are core to the Spatial Interaction Pillar become accessible as a remote side channel. We demonstrate that these streams can be weaponized to infer sensitive user actions, such as typed keystrokes, by any peer co-present in the same virtual space, without local access to the victim's device or any installed malware.
  - (2) Root-Cause Dissection (Chapter 4): The recurring interaction-layer vulnerabilities across the XR ecosystem suggest a systemic pattern that reactive patching cannot break. To understand this persistence, we shift focus from the applications to the people who build them. We conduct 90-minute in-depth interviews with 23 XR developers, presenting each with curated immersive-attack scenarios (e.g., VR keylogging, AR perception) with bottom-up open coding, covering developers' threat perceptions (RQ1) and views on mitigation practices and community support (RQ2). Three ecosystem-level flaws surface—systemic practice gaps, cognitive and awareness biases, and infrastructural deficits in dominant game engines (e.g., Unity, Unreal) and security documentation—synthesizing into two structural root causes: pervasive awareness gaps that prevent

developers from recognizing spatially-specific threats, and a diffusion of responsibility reinforced by opaque engine architectures and fragmented community documentation. This analysis reframes interactive XR vulnerabilities from individual developer error to ecosystem-level dysfunction and motivates Chapter 5's architectural reconstruction.

  - (3) Architectural Reconstruction (Chapter 5): We reframe decentralization not as a binary architectural switch but as a function-by-function redistribution of distinct operator roles, each tied to a separate centralized failure mode our 22-participant interview study surfaces. Grounded in user-stated needs, including the Digital Doorbell permission-gated access intuition (ten of 22 participants, unprompted) and the Accountability Paradox between privacy and evidence, we build a serverless framework around three mechanisms: workspace-scoped cryptographic identity over Named Data Networking with a Standardized State Abstraction Layer (G_1), a CRDT-backed semantic-correctness pipeline (G_2), and peer-driven quorum-based behavior auditing (G_3). A within-subjects double-blind validation of 20 participants in 10 dyads against a centralized baseline returns four converging outcomes: no significant Likert difference on the three load-bearing constructs, dyad-level identification at chance, semantic-correctness rollbacks unnoticed in every decentralized session, and round-trip latency within 2 ms of the baseline—placing the framework inside the small-group, high-stakes deployment surface participants drew.

- **Research Line 2: Data & Model Privacy (The Cognitive Intelligence Boundary).** Focused on the analytical core of the Metaverse, this line aims to rigorously demarcate the privacy boundaries of centralized machine learning services by evaluating their propensity for data memorization:

  - (4) Auditing Spatial-Temporal Intelligence (Chapter 6): We systematically analyze the privacy vulnerabilities of predictive models, focusing on the kind of data the Metaverse actually produces: continuous, semantically annotated spatial-temporal trajectories rather than isolated GPS pings. By developing advanced adversarial frameworks, we investigate the extent to which Next Point-of-Interest (POI) recommendation systems,

the predictive backbone of Metaverse location-based services, memorize and expose users' sensitive trajectories and behavioral patterns at a fine spatial-temporal granularity.

- (5) Auditing Generative Intelligence (Chapter 7): Extending our evaluation to biometric content, we examine generative models in their Metaverse-specific role: synthesizing the avatar voices that carry a user's vocal identity in shared social spaces. Specifically, we design Membership Inference Attacks (MIA) against state-of-the-art Text-to-Speech (TTS) models to audit how much of those identity-bearing voice features, and the linguistic content the models were fine-tuned on, are retained inside the deployed system.

1.1 Contributions

Building upon the proposed research lines, this dissertation shifts the security paradigm from reactive patching to proactive, architecture-level trustworthiness. The core technical and conceptual contributions are summarized as follows:

- **Discovery of Novel Spatial Attack Surfaces:** Empirically uncovers a new class of side-channel threats, demonstrating that multi-modal VR kinematics can be exploited to infer sensitive user intentions with a top-1 keystroke recovery accuracy of **97.62%** on Rec Room.
- **Systematization of Ecosystem-Level Root Causes:** Conducts a pioneering, in-depth empirical study involving 23 XR developers, revealing how developers' cognitive biases and opaque game engine models perpetuate widespread interactive vulnerabilities across the broader XR ecosystem.
- **Architectural Shift to Decentralized Governance:** Proposes a decentralized protocol for immersive interactions, grounded mechanism by mechanism in user-stated needs surfaced by a 22-participant interview study (introducing the Digital Doorbell access primitive and the Accountability Paradox that audit-without-surveillance design must navigate). A 20-participant within-subjects double-blind validation against a centralized baseline returns paired Likert tests with $p > 0.10$ on all three load-bearing constructs (Perceived Latency, Synchronization Consistency, Collaborative Fluency), dyad-level architectural-invisibility

identification at chance, semantic-correctness rollbacks unnoticed in every decentralized session, and per-update round-trip latency within **2 ms** of the centralized baseline.

- **Empirical Privacy Auditing for Predictive Models:** Formulates a comprehensive auditing methodology through a suite of four tailored privacy attacks. Our empirical results demonstrate significant data leakage: models expose sensitive locations in 32.5% of random queries despite a training presence of only 18.7%, while extraction attacks achieve success rates that outperform random baselines (e.g., 10^{-8}%) by multiple orders of magnitude.

- **Demarcating Biometric Boundaries for Generative AI:** Leverages model-centric privacy attacks to quantify the privacy risks within TTS models. Through systematic MIA across diverse architectures, we find that speaker-level attacks reach AUC up to 1.000 in the strongest configurations and remain above 0.80 throughout, while record-level attacks consistently achieve AUC between 0.80 and 0.90. These results indicate that modern acoustic models memorize fine-grained training data well beyond what task performance requires.

Beyond these specific technical contributions, this dissertation advances a broader argument. The Metaverse builds directly on VR/AR, machine learning, and networked communication, and the trust failures it produces inherit much of their character from each of these underlying technologies. What makes the Metaverse distinctive, and substantially more complex than the 2D internet that preceded it, is that these technologies now operate simultaneously, persistently, and over multi-user shared state, and that operating setting significantly raises the security and privacy implications of each. Securing the Metaverse therefore cannot be achieved by chasing individual vulnerabilities with ad-hoc patches, nor by importing security models designed for any single layer in isolation. Real resilience requires understanding the systemic root causes that let threats persist across the coupling between the two pillars, and rebuilding both the architectural power dynamics and the algorithmic accountability frameworks from the ground up. By targeting the structural drivers of data centralization and model opacity at the level of this coupling, this research provides a practical blueprint for future Metaverse applications, one where, as physical and digital realities continue to merge, the resulting ecosystem is built on structural transparency, empirical accountability, and meaningful user sovereignty.

CHAPTER 2

Background

This chapter provides the technical and conceptual foundation for the security and privacy (S&P) challenges studied throughout this dissertation. Following the Dual-Pillar Architecture introduced in Chapter 1, we first examine the Spatial Interaction Pillar, covering the mechanics of immersive sensing, the resulting security and privacy threats, the human factors of the XR development ecosystem, and the case for decentralized infrastructure. We then turn to the Cognitive Intelligence Pillar, surveying predictive and generative Metaverse services and the privacy auditing frameworks needed to evaluate their risks. At each stage, we identify the critical research gaps that motivate the contributions in the chapters that follow.

2.1 The Metaverse: A Dual-Pillar Ecosystem

The Metaverse represents a fundamental paradigm shift from the traditional, two-dimensional internet to a persistent, three-dimensional spatial web. Rather than a single standalone application, it is comprehensively defined as a continuous, multi-user computational environment that seamlessly integrates physical reality with digital virtuality [LBZ21, Mys22]. In this paradigm, users transcend the role of external observers; they become *embodied* participants, interacting, socializing, and collaborating through digital avatars within real-time synchronized virtual spaces. Driven by the convergence of immersive Extended Reality (XR), high-frequency networking, and advanced artificial intelligence, the Metaverse transforms isolated digital simulations into a ubiquitous socio-economic platform characterized by unbounded interoperability, continuous state persistence, and tightly coupled algorithmic services [DBG13, WSZ22].

To systematically analyze the multifaceted S&P challenges inherent to this complex ecosystem,

we deconstruct its infrastructure through the **Dual-Pillar Architecture**. This conceptual framework bridges the critical operational gap between local perceptual interfaces and remote algorithmic processing pipelines, highlighting how the illusion of a seamless spatial experience fundamentally relies on a constant, bidirectional flow of highly sensitive data. Compared with the 2D internet that came before, three properties of this flow stand out as Metaverse-distinctive and as factors that significantly worsen its security and privacy implications. First, the captured signals are *embodied and high-dimensional*, including kinematics, gaze, voice, and spatial mesh produced by VR/AR hardware rather than discrete keystrokes or clicks. Second, the state is *persistent and multi-user-visible*, since avatars in a shared room broadcast embodied behavior continuously for cross-peer synchronization. Third, the same signals are *immediately consumed by predictive and generative ML pipelines* that retain memorizable traces of the individuals who produced them. Each of these properties has antecedents in prior technologies such as VR/AR research, online games, and machine learning; what makes the Metaverse setting harder to secure than any single antecedent is their simultaneous, real-time co-occurrence over multi-user shared state. Each chapter of this dissertation targets this co-occurrence in turn.

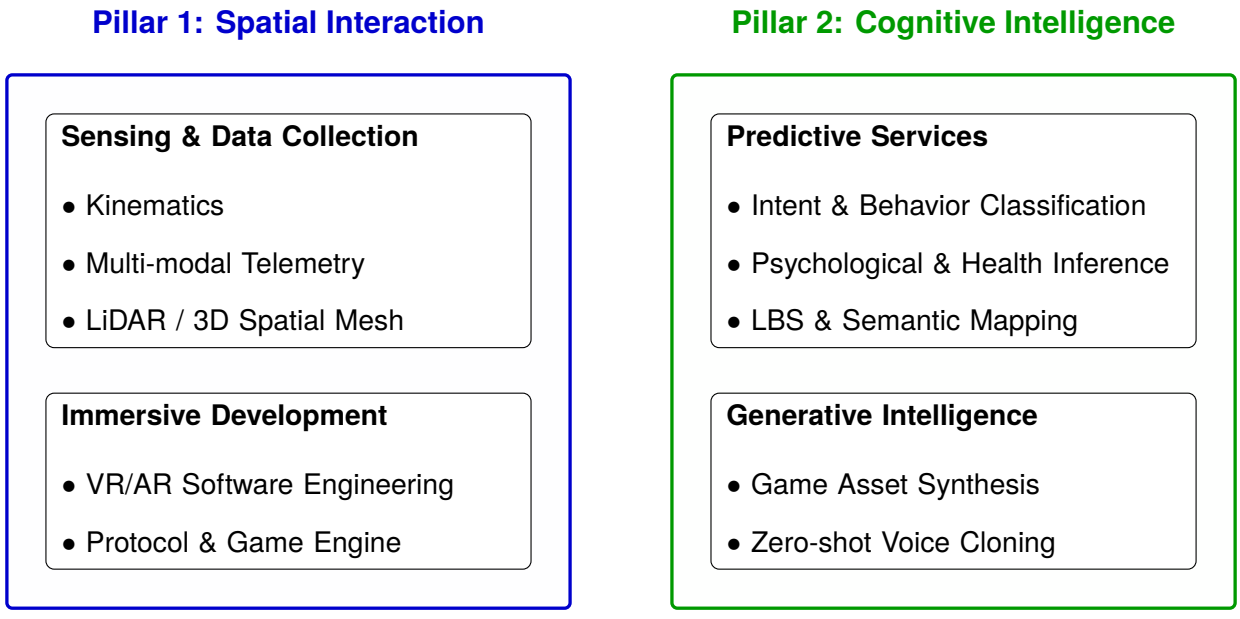


Figure 2.1: Metaverse Dual-Pillar Architecture. Pillar 1 manages embodied interaction and sensing, while Pillar 2 processes these data streams for predictive and generative intelligence.

2.2 The Spatial Interaction Pillar

To systematically evaluate the security and privacy landscape of the Metaverse, it is essential to first deconstruct its perceptual layer: the Spatial Interaction Pillar. This section explores the fundamental tension between the technical mechanics required to achieve seamless immersion and the systemic vulnerabilities that these mechanics inevitably introduce. We begin by detailing the continuous, multi-modal telemetry necessary for embodied interaction and its transformative utility. Building upon this technical context, we then examine how this pervasive data harvesting catalyzes severe privacy threats, investigate the underlying root causes of these vulnerabilities within the developer ecosystem, and finally outline the architectural imperative for decentralized infrastructures to restore user data sovereignty.

2.2.1 Mechanics and Transformative Applications

Functioning as the perceptual foundation of the dual-pillar architecture, the Spatial Interaction Pillar acts as the bidirectional bridge between human physiology, the physical environment, and the digital ecosystem. Unlike traditional 2D interfaces—which rely on discrete, low-bandwidth inputs such as keystrokes or mouse clicks (historically characterized as the WIMP paradigm, after Windows, Icons, Menus, Pointers)—spatial computing represents a fundamental shift toward continuous, embodied human-computer interaction [LBZ21, DBG13, MTU95].

In this paradigm, the user's physical body and real-world environment fundamentally become the interface itself [WSZ22]. Fueled by a spatial computing and AR/VR market projected to exceed $420 billion in value and engage over 3.7 billion users globally by 2030 [Gra23, Sta25b], modern immersive hardware (e.g., Apple Vision Pro, Meta Quest Pro) necessitates pervasive multi-modal telemetry. This aggressive data harvesting encompasses inward-facing infrared cameras for gaze and pupillary tracking, micro-facial expressions, electroencephalogram (EEG) signals, and environmental Light Detection and Ranging (LiDAR) scanners that generate high-fidelity 3D meshes of private physical spaces [App24, Met22, GST21, NGS22].

To understand the sheer granularity of this telemetry, consider basic motion capture. The fundamental baseline for spatial interaction relies on Six Degrees of Freedom (6DOF) kinematics.

In a 6DOF system, headsets and controllers continuously track both the translational (x, y, z) and rotational (pitch, yaw, roll) movements of the user's head and hands at extremely high frequencies (e.g., exceeding 90Hz) [Uni23]. This dense kinematic mapping is an absolute architectural requirement to render responsive virtual avatars and maintain the cognitive illusion of physical presence.

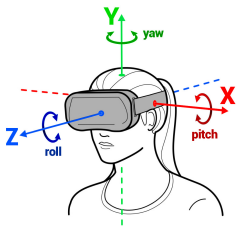


Figure 2.2: Example of user head motion mapped in 6DOF [dsk], representing the baseline kinetic telemetry essential for spatial interaction.

The convergence of this advanced, multi-modal sensing serves as the critical enabler for transformative spatial applications. In the enterprise sector, platforms like NVIDIA Omniverse leverage this high-precision tracking to construct industrial digital twins for collaborative simulation [New22]. Simultaneously, consumer social platforms (e.g., VRChat) and healthcare applications (e.g., remote robotic surgery) depend heavily on the continuous transmission of kinematics and facial tracking to preserve human empathy, non-verbal communication, and real-time physiological monitoring across geographic boundaries [FZM22, Mys22].

2.2.2 Security and Privacy Landscape in Immersive Interactions

Immersive interaction encompasses emerging spatial technologies that deliver deeply engaging digital environments. Recent research highlights that the unique characteristics of these immersive experiences make them particularly vulnerable to S&P threats [GNS23]. Specifically, immersive applications continuously collect an extensive range of fine-grained human data (e.g., multi-modal motion, voice, and spatial behavior) which inherently contains significantly more granular and sensitive biometric information than traditional 2D applications.

Consequently, breaches within immersive environments pose severe and novel risks to user identity [NGS22, NGM23, NRO23], the physical surrounding environment [FMN23, GST21], and

private physical actions such as typing [LNF24]. These VR/AR-rooted risks become substantially more acute under the Metaverse's multi-user, shared-state regime. The broadcasting of baseline kinematics (e.g., 6DOF data) is required precisely because remote peers must render the user's avatar in real time, which lifts what would otherwise be a local, device-bounded signal into a network-visible attack surface. The continuous, multi-user network broadcasting of these streams creates an implicit, high-resolution side-channel that bypasses traditional application permission models. Once exposed, the streams are biometrically identifying at population scale (e.g., users uniquely identifiable from motion alone [NGM23]), turning a synchronization requirement into a persistent re-identification surface. Furthermore, immersive experiences introduce entirely new attack surfaces that must be rigorously considered during systemic and interactive design. Researchers have identified domain-specific vulnerabilities stemming from overlooked design choices and specialized functionalities [YSH23], virtual social dynamics [BEE19a, FZM22, JF20, SFL23, ZXW23], content security [CBL24, SZS24, TAM23], physiological user perception [CTK23, CBY19, TBM22], systemic side-channels [AGA21, ZSC23], and insecure framework implementations [TLC22, GDL24].

Moreover, the current landscape severely lacks comprehensive standards, policies, and guidelines [Cyb21, RWP23] tailored to the development of immersive interactions. While existing policies provide generic suggestions for avoiding basic security issues (e.g., identity theft, intellectual property) and standard privacy leakage (e.g., location, financial information), they remain fundamentally insufficient. They fail to cover threats unique to spatial computing [Gia24] and the novel types of sensitive continuous telemetry involved [GNS23], resulting in a critical disconnect between application development and threat mitigation.

2.2.3 Usable Security and the Immersive Development Ecosystem

While technical threat vectors elucidate how vulnerabilities manifest, understanding their persistence requires examining the development ecosystem through the lens of **Usable Security** [AFM16]. The "usable security for developers" paradigm posits that developers are human users of security APIs, framework documentation, and permission models. When these tools are opaque or misaligned with

developers' mental models, they become primary catalysts for vulnerable implementations [VFP20, AC18]. Consequently, extensive research has characterized developers' attitudes toward privacy constraints [LLD21, TFV21] and the cognitive friction introduced by security policies [HDG24, LRA22] to foster more adoptable defenses.

However, translating this paradigm to immersive spatial computing reveals a critical research void. The unique architectural threats of immersive interactions—specifically continuous biometric harvesting and physical side-channels—introduce design challenges absent in traditional 2D software [HRR24, DFH23]. Furthermore, immersive applications predominantly rely on legacy 3D engines (e.g., Unity, Unreal) whose data-handling pipelines were optimized for local entertainment rather than privacy-preserving telemetry. This *structural mismatch* creates significant usability friction for developers attempting to implement robust security within spatial environments.

Current literature fails to address this ecosystem-level friction, as S&P research in spatial computing remains heavily skewed toward end-user perspectives [KWC24, SAY25, ASD24] or theoretical expert assessments [ASM22]. Existing developer-centered studies in this domain are limited in scope: early work (e.g., Adams et al. [ABB18]) predates contemporary biometric threats, while recent inquiries focus narrowly on social dynamics such as harassment [ASD24]. These efforts omit systemic vulnerabilities like hidden data leakage and the cognitive burden imposed by opaque engine permissions.

A significant gap remains in empirically assessing how immersive developers perceive architectural threats and navigate the complexities of privacy-conscious spatial design. Without investigating these ecosystem-level usability challenges, the Metaverse remains susceptible to a cycle of reactive patching instead of evolving toward developer-friendly, secure-by-design architectures.

2.2.4 Infrastructure Design and Decentralized Architectures

The Metaverse operates as an interconnected, persistent network of immersive environments where users interact in real time [Wei22, DBG13]. Because legacy game engines inherently default to client-server networking topologies, developers are structurally funneled into building centralized applications. However, relying on this centralized infrastructure to host these environments

introduces severe systemic vulnerabilities. Beyond single points of failure, centralization mandates the surrender of continuous biometric telemetry to platform operators (as discussed in Section 2.2.2). This concentration of power strips users of data sovereignty and enables large-scale surveillance, necessitating a structural shift toward decentralized architectures [WSZ23a, HGW23].

Existing decentralized Metaverse research predominantly focuses on asynchronous, blockchain-based infrastructures for asset ownership and digital provenance [HGW23, LYY23]. However, the application-level requirements for *real-time, embodied interaction* remain severely underexplored. Unlike traditional ledgers, spatial applications require the continuous, concurrent mutation of hierarchical data structures (e.g., Scene Graphs). Consequently, a true spatial decentralized application (dApp) must natively integrate lightweight localized identity for accountability [GMA23] and *semantic state synchronization*. While conventional Conflict-Free Replicated Data Types (CRDTs) ensure syntactic data convergence in distributed systems [PBS18, LLK22], they fail to capture the physical and spatial constraints of virtual environments (e.g., ensuring two peers cannot simultaneously manipulate a shared rigid object in conflicting directions). To preserve chronological causality and prevent physical exploits (e.g., unauthorized teleportation), semantic consistency and peer-driven moderation must be treated as primary architectural concerns alongside message routing [SCT23, SLF23]. Adjacent CSCW work on decentralized social platforms further reinforces this need to move beyond a binary "with-server vs. without-server" framing: members of decentralized communities reason about specific stewardship functions of an instance operator rather than the architecture as a whole [HNS25], motivating a *function-by-function* redistribution of operator roles (identity, shared state, behavior auditing) rather than a single architectural switch.

As these underlying decentralized protocols mature, understanding their human factors becomes critical. Current human-computer interaction (HCI) research on decentralization predominantly evaluates asynchronous systems, documenting the usability friction of cryptographic wallets [VWM21, YSD24] or the governance of federated microblogging platforms [LSL22, ZZW24]. While highlighting users' intrinsic desire for digital sovereignty [LNE23], these studies fail to capture the nuances of synchronous, embodied interaction. Methodologically, prior HCI work also documents how lay folk models of security, privacy, and platform behavior recur in predictable, architecturally targetable patterns [Was10, KDF15, MDD24], suggesting that the way

users misread decentralized systems is itself a design contract decentralized metaverse frameworks have to articulate against. A critical research gap remains in evaluating the usability and human experience of decentralized spatial protocols when multiple users actively collaborate and manipulate shared virtual environments. Specifically, the open question is whether such a framework can stay collaboratively indistinguishable from a centralized baseline on the constructs that users themselves consider load-bearing for a usable collaborative experience.

2.3 The Cognitive Intelligence Pillar

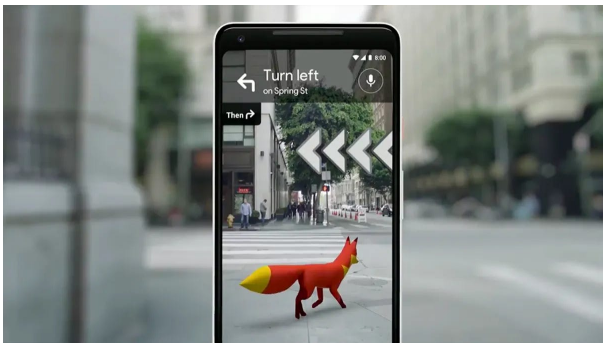
While the spatial interaction layer captures raw multi-modal telemetry, the **Cognitive Intelligence Pillar** functions as the semantic and generative core of the Metaverse [SAD24, ZJC22]. This section examines the transition from data acquisition to data comprehension, focusing on how machine learning models transform high-dimensional telemetry into adaptive, predictive, and personalized experiences. We first outline the broad landscape of predictive services and generative intelligence, highlighting their transformative utility. Building on this technical foundation, we then investigate the inherent privacy risks of these black-box models—specifically their propensity to memorize sensitive training data—and evaluate the necessity of empirical auditing frameworks to quantify the resulting leakage at scale.

2.3.1 Predictive Services and Semantic Reasoning

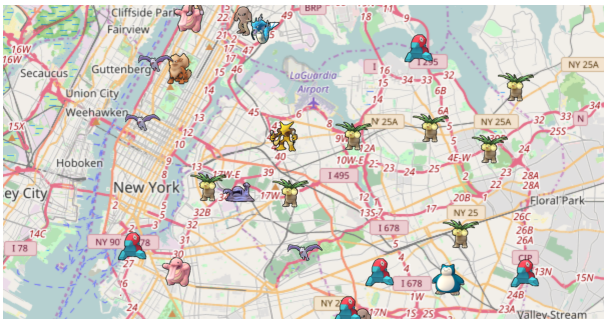
Within the Metaverse, the Cognitive Intelligence Pillar relies extensively on predictive machine learning models to classify user behavior, infer intent, and anticipate future states. Rather than simply reacting to explicit discrete commands, these predictive services ingest the continuous, high-dimensional telemetry captured by the spatial interaction layer, such as kinematics, gaze vectors, and environmental interactions, to construct semantic profiles of user activity. This capability encompasses a broad spectrum of algorithmic tasks, including real-time action classification, continuous biometric authentication, and behavioral preference modeling [WSZ23a, MHJ20]. The kind of data feeding these models is itself Metaverse-distinctive. Unlike traditional analytics on aggregated 2D logs, the Pillar consumes *sequential, semantically rich, behavior-annotated* streams,

where each record is not an isolated observation but a component of a coherent embodied trajectory tied to a persistent avatar identity. This sequence-level coherence is what gives the models their predictive utility, and as Chapter 6 demonstrates, it is also what makes them disproportionately prone to memorizing individual users.

Among these diverse applications, spatial analytics and the processing of *mobility data* serve as a prominent and highly sensitive example. In immersive environments, mobility modeling extends beyond basic 2D positioning to parse fine-grained spatio-temporal trajectories across blended physical-digital landscapes [BSL23]. By feeding this continuous data into predictive spatial models, developers offer highly optimized Location-Based Services (LBS). For instance, frameworks like the Google ARCore Geospatial API leverage environmental telemetry to anchor digital navigation overlays to precise real-world coordinates at scale [BQS25, Goo24], while interest-driven applications like *PokeMap* [Pok24] utilize historical spatial trajectories to infer underlying user preferences, proactively predicting intent to suggest customized routes or optimal "next-step" destinations tailored to each user.



(a) ARCore Geospatial API [Goo24]



(b) LBS Inference in *PokeMap* [Pok24]

Figure 2.3: Transformative applications of spatial analytics and mobility data processing. Left: Contextual navigation via real-world coordinate anchoring; Right: Predictive intent inference based on historical trajectories.

2.3.2 Generative Intelligence and Biometric Synthesis

Complementing predictive reasoning, the intelligence pillar heavily utilizes generative artificial intelligence to dynamically synthesize high-fidelity, multimodal content. Because rendering complex virtual environments and maintaining continuous immersion requires immense computational resources and creative bandwidth, developers increasingly rely on deep neural networks [CLL23]. This macro-level generative capability spans a wide array of Metaverse applications, including the automated synthesis of 3D environmental assets and textures [PJB22, LGT23], algorithmic motion blending for avatar kinematics [TRG22, CJL23], and Large Language Model (LLM) driven narrative generation for virtual ecosystems [POC23, BQS25].

One particularly intimate example within this broad generative core is *biometric synthesis* and acoustic modeling—specifically advanced Text-to-Speech (TTS) systems. These models are widely deployed to enhance accessibility, provide real-time cross-lingual translation, and populate virtual environments with interactive audio. Driven by recent architectural breakthroughs in flow-matching and diffusion models (e.g., F5-TTS [CNM25], Voicebox [LVS23]), these generative acoustic models deliver vital utility in immersive scenarios. For example, utilizing zero-shot voice cloning [WCW23, Ope24], platforms can synthesize text-based inputs into a user's authentic physiological voice, allowing individuals to maintain their unique vocal identity in social hubs without continuous microphone streaming. Similarly, state-of-the-art TTS engines generate context-aware, emotionally resonant speech for Non-Player Characters (NPCs) in real-time, enabling seamless and dynamic human-agent interaction [GTZ24, HHL24].

Together, these predictive and generative models provide the indispensable utility required to sustain a personalized Metaverse. However, achieving this unprecedented convenience demands the continuous ingestion of deeply sensitive user data—from granular physical trajectories to physiological voice patterns—creating profound model-level privacy vulnerabilities.

2.3.3 Privacy Auditing of Machine Learning Services

To systematically quantify the severe memorization risks inherent in the Cognitive Intelligence Pillar, researchers employ empirical privacy auditing frameworks. Rather than relying on theoretical privacy

bounds, these frameworks proactively attack the model to measure how much sensitive training data leaks through its observable outputs. The most prominent vectors for assessing this vulnerability are Membership Inference Attacks and Training Data Extraction attacks [SSS17b, CLE19].

Membership Inference Attacks (MIA). The primary objective of an MIA is to determine whether a specific data record was included in a target model's training dataset [SSS17b, SZH18, CCN22, GAA23]. Threat models for these attacks are fundamentally categorized by the level of access granted to the adversary. In a white-box setting, the attacker possesses full transparency into the target model's architecture, internal weights, and gradients. This allows them to exploit explicit internal metrics—such as intermediate layer activations or gradient norms with respect to the target data—yielding highly accurate signals of model memorization [NSH18].

However, in practical Metaverse deployment scenarios, proprietary AI models are typically concealed behind commercial API endpoints. Consequently, adversaries and external auditors must operate under a black-box threat model. Under this restricted setting, the attacker cannot observe the model's internal parameters or gradients, and must instead rely entirely on the observable outputs (e.g., predicted spatial coordinates or generated audio) to perform inference.

Recent works evaluating conditional generative and predictive models often formulate these black-box MIAs as a score-based decision process [PW23]. Rather than probing the model with explicit internal queries, the adversary crafts a targeted attack prompt and evaluates the model's corresponding output. By analyzing metrics such as reconstruction error, output similarity, or structural representation gaps between the model's generation and the target sample, the attacker derives a continuous membership score. This score quantifies the likelihood of memorization: a model will typically exhibit disparate behavior—such as unexpectedly high confidence or unusually precise reconstruction—when processing data it was exposed to during training compared to unseen data. The final membership decision is then made by evaluating this score against an empirically defined threshold, making the attack's effectiveness entirely dependent on designing a robust scoring metric that captures this behavioral discrepancy.

Training Data Extraction Attacks. While MIA focuses on binary presence, data extraction attacks aim to explicitly reconstruct verbatim training samples directly from the model's learned

weights [CLE19, CHN23]. Unlike model inversion attacks [FJR15a], which generally reconstruct representative class features, extraction attacks exploit model overfitting to recover the exact, idiosyncratic data points—such as sensitive user coordinates or unique vocal patterns—that the model has memorized. This profound vulnerability is fundamentally rooted in the phenomenon of *unintended memorization*. Theoretical and empirical studies demonstrate that over-parameterized deep neural networks—particularly advanced language and generative models—often internalize highly specific, low-frequency training examples rather than exclusively generalizing underlying statistical distributions [Fel20, CTW21]. Furthermore, recent empirical scaling laws reveal that as model capacity and parameter counts increase, their propensity to memorize and explicitly regurgitate exact training sequences drastically exacerbates [CIJ22]. In the context of the Metaverse's intelligence layer, this implies that highly personalized spatio-temporal trajectories or physiological voice profiles are not simply used to learn general heuristics; they are actively embedded within the network weights, rendering them highly susceptible to precise adversarial extraction [SSG23].

The Auditing Gap in the Cognitive Intelligence Pillar. Existing privacy auditing methodologies, optimized primarily for image classifiers and discrete Large Language Models, are structurally insufficient for the advanced predictive and generative models of the Metaverse. This introduces two critical domain-specific gaps:

- *Mobility and Spatial Analytics:* Previous privacy research on mobility data primarily targeted system-level vulnerabilities via side-channel attacks on aggregated statistics [SH12, Kru07, VHB14]. However, the paradigm shift toward utilizing Deep Neural Networks (DNNs) for Location-Based Services introduces a fundamentally new attack surface. Existing MIA frameworks fail to account for the continuous spatio-temporal dynamics of this data. Currently, the community lacks an empirical framework to systematically investigate how centralized DNNs inherently memorize and leak highly sequential user-location trajectories.

- *Biometric and Acoustic Generation:* Applying MIA to generative acoustic models, such as zero-shot TTS, introduces three intertwined hurdles. First, an inherent input mismatch complicates the auditing process: modern zero-shot TTS is dual-conditioned on a text prompt and a reference audio, and standard attack queries pair the target text with a different

reference snippet to prompt voice cloning. This forces the model into an exploratory zero-shot adaptation state whose high-variance prosodic drift masks the subtle signals of training-time memorization, making it difficult to determine whether an accurate generation reflects verbatim memorization or generalization from the reference prompt. Second, raw acoustic features (Mel-spectrograms, MFCCs) lack the resolution to disentangle memorization traces from surface phonetic and prosodic content, requiring richer representation spaces that isolate speaker identity and sequence-specific memorization. Third, the continuous waveforms TTS generates carry no explicit confidence scores, and naive frame-level comparison is dominated by benign temporal misalignment, rendering point-to-point distance metrics hypersensitive to micro-prosodic variation that bears no relation to membership.

Bridging these dual gaps necessitates domain-specific MIA and extraction frameworks tailored to spatial trajectories and physiological acoustic features, establishing the technical focus for the remainder of this dissertation.

Together, the gaps identified across the spatial interaction and cognitive intelligence layers form the research agenda of this dissertation. Chapter 3 begins by examining the most immediate layer of risk: the interaction data itself, and how its network-level transmission creates exploitable side channels.

CHAPTER 3

Identify Threats: Vulnerabilities in Immersive Interactions

Building upon the "Dual-Pillar Architecture" introduced in Chapter 1, this chapter focuses on the **Spatial Interaction Pillar**. As Virtual Reality (VR) becomes increasingly ubiquitous—projected to reach over 171 million global users in 2024 [Abb24]—the shift toward multi-user multi-user VR applications has fundamentally expanded the attack surface of spatial computing. Delivering highly immersive, collaborative environments necessitates the continuous collection and transmission of ultra-high-frequency kinematic data to synchronize virtual avatars across remote clients. While this architectural necessity bridges the gap between physical and digital realities, it simultaneously introduces severe, understudied privacy risks inherent to spatial interaction.

Rather than attempting an exhaustive catalog of all potential interactive flaws, this chapter leverages a highly representative vulnerability as a methodological anchor: **keylogging via spatial kinematics**. While recent research has demonstrated the feasibility of VR keylogging, existing methods—such as VR-Spy [AGA21], HoloLogger [LHY22], TyPose [SZA23], and rendering-based attacks [ZSC23]—rely on a stringent assumption: the adversary must install local malware, tap into local WiFi channel state information, or deploy dedicated surveillance implants on the victim's device.

This chapter removes that constraint by introducing a novel, **remote side-channel attack**. We demonstrate that an adversary requires only a presence in the same virtual room as the victim to execute a highly accurate keylogging attack. By extracting standard avatar rendering information directly from network packets received by an attacker-controlled game client, we can infer sensitive user-typed secrets, including passwords, financial credentials, and private communications.

Adapting keylogging to a remote context presents significant technical hurdles. Because commercial multi-user VR applications operate as black boxes—often shielded by anti-cheating

engines and Digital Rights Management (DRM) components—understanding how motion data is formatted and transmitted is highly complex. Furthermore, network-transmitted motion data is typically compressed and under-sampled to ensure efficient routing, resulting in lower-fidelity telemetry compared to raw, local sensor data.

To overcome these challenges, we establish a systematic, four-step blueprint for vulnerability discovery. Our methodology extracts fine-grained kinematic data from standard network traffic and maps these continuous, low-fidelity motion trajectories to precise keystroke entries.

To rigorously evaluate this discovery, we conducted comprehensive empirical evaluations and user studies. Testing our methodology on Rec Room [rec23]—a leading multi-user VR applications with over 15 million users [Lyo]—our attack successfully reconstructed typed secrets with a top-1 accuracy of **97.62%**. Crucially, the attack maintains a comparable efficacy (**97.53%**) in complex, real-world conditions, including crowded multi-user environments and Non-Line-of-Sight (NLOS) scenarios where the attacker cannot visually observe the victim. We further validated the generalizability of this threat by successfully replicating the attack across three additional widely adopted VR applications (including Sing Together: VR Karaoke [sin24]), achieving top-1 accuracies of **98.24%**, **98.27%**, and **99.07%**.

Moving beyond threat discovery, this chapter proposes robust, system-level defense mechanisms to mitigate this fundamental data leakage. These countermeasures have since been adopted and implemented by major industry stakeholders, including Rec Room and SteamVR [ste24a], underscoring both the critical severity of this vulnerability and the urgent need to secure the spatial interaction ecosystem.

By dissecting this remote keylogging vulnerability, this chapter demonstrates a systematic approach to uncovering and mitigating interactive threats in spatial computing. The core contributions are summarized as follows:

- **Pioneering Remote Keylogging in Social VR:** We demonstrate the first feasible remote keylogging attack in multi-user VR environments. By eliminating the need for local malware or physical proximity, this approach highlights a highly practical and stealthy threat vector affecting all multi-user VR applications users.

- **Novel Kinematic Extraction Methodology:** We introduce an innovative technique to recover typing-related motion directly from a remote application client's network packets, overcoming challenges related to data compression, under-sampling, and black-box application design. Additionally, we provide new analytical tools for VR motion data processing, including precise motion input control and cursor/keyboard measurement.

- **Machine Learning Attack Alternative:** We introduce an alternative strategy effective on partially reverse-engineered network packets. By applying machine learning techniques directly to raw byte extraction, we prove that highly accurate remote keylogging is achievable with minimal manual reverse engineering.

- **Comprehensive Empirical Evaluation:** We conducted extensive user studies to assess attack efficacy, practicality, and generalizability. Analyzing tens of thousands of keystrokes across multiple applications and under NLOS conditions, we provide concrete insights into the vulnerabilities of real-world typing behaviors and gauge user awareness of these emerging threats.

- **Industry-Adopted Countermeasures:** We propose practical defense mechanisms to prevent spatial motion data leakage over the Internet. Demonstrating the real-world impact of this research, major platforms including Rec Room and SteamVR have acknowledged the vulnerability, awarded a bug bounty, and actively implemented our proposed security patches.

3.1 Background on Multi-user VR Applications

As the name suggests, a multi-user VR application provides interaction opportunities for users, allowing them to communicate with other players across the Internet. Typically, users can communicate by both typing and speaking. The typing functions are usually achieved using a virtual keyboard, either in-application or through overlay applications such as Steam Chat or Messenger. These newly emerging services allow users to interact in realistic digital worlds, and they distinguish themselves from traditional social networking services by capturing and translating real-world user motions into their virtual landscapes [WJF22].

Motion Update in Multi-user VR Applications. Motion, which has six degrees of freedom (6DOF) as shown in Figure 2.2, is normally described by position (x, y, z) and rotation (roll, pitch, yaw). In this chapter, we define TRANSFORM to be the composition of position and rotation.

Popular game engines describe motion with their own data structures that encapsulate TRANSFORM. For example, the Unity Engine, the most popular game engine for multi-user VR applications [Boo], has a standard data structure to represent TRANSFORM using a Quaternion (for rotation) and a Vector3 (for position) [Uni23]. In Unity-based applications, this standard data structure is transmitted over the network to synchronize user motions.

For multi-user VR applications, the server continuously receives motion data updates from clients and broadcasts them to all clients [Pho], as seen in Figure 3.1. Typically, the synchronization process employs UDP packets with established libraries such as Photon PUN/Fusion [Pho], Unity Netcode [net23], and Mirror [mir] to ensure stable and accurate motion transmission.

However, the motion data may not be fully identical after the transmission, as sometimes, applications (e.g., Rec Room) may use lossy compression on the motion data update, trading off a slight loss of fidelity of motion data in exchange for a lower burden on the transmission process.

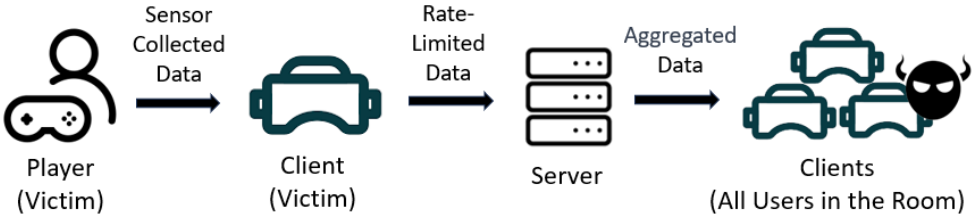


Figure 3.1: Motion data flow in multi-user VR applications.

Moreover, the transmitted motion data is also under-sampled from the original motion data. In current multi-user applications, the server usually sends updates to the clients at a rate of 20-30 packets per second. This rate takes into account the network bandwidth limitation on both the server and client sides, ensuring efficient communication without overloading either end [net]. But applications usually run at 60-120 frames per second (FPS) [Vir22], so they have adapted mechanisms like interpolation [Gama] to compensate for performance loss.

VR Typing Mechanism. In VR, there are multiple options for text input. The most commonly

deployed methods include using voice or typing on a virtual keyboard [DBW19]. In this chapter, our primary focus is on the typing method that utilizes a virtual keyboard. Typing on a virtual keyboard involves two major steps: (1) moving the cursor, which can be a controller cursor or a virtual hand, to the target key on the virtual keyboard, and (2) selecting the key by performing a click, either by pressing a button on the controller or by poking at the key with a hand gesture. Since both of these steps involve the movement of the user's hand, multi-user VR applications map the corresponding motion to the avatar. This ensures the avatar feels realistic and accurately represents the details of the user's hand motion.



Figure 3.2: Example of a participant typing in Rec Room using a virtual keyboard.

For the click mechanism, we focus on clicks represented by button presses on a controller, which is a widely adopted method for both interaction between users and typing on virtual keyboards. Considering the importance of click functionality, Unity offers standardized APIs, such as OVR [ovr] and Input system [Tec], to help applications detect these clicks. These APIs use the trigger value (i.e., how deep the trigger button is pressed down, from 0.0 to 1.0) to detect clicks. Typically, the API registers a click when the trigger value is higher than a specific threshold. Since the trigger button is typically held for more than one frame per click, to avoid repeated firing, the first frame where the trigger is pressed is recognized as the click moment [inp].

3.2 Threat Model

Adversary Objective. Our attack aims to extract user-typed secrets inside multi-user VR applications. By extracting these secrets, an attacker may gain access to the following types of sensitive information:

(1) *Credit Card Information:* Modern VR applications often incorporate payment gateways to

facilitate transactions, requiring users to input sensitive financial details, such as credit card numbers. Such information represents a profitable target for adversaries aiming for unauthorized financial access. Conventionally, credit card numbers are sequences of digits.

(2) *User Authentication Data:* Within the context of multi-user VR applications, authentication mechanisms, such as password inputs, are employed when users attempt to access their accounts, private virtual rooms, or private virtual assets. Adversaries can actively seek these credentials to gain unauthorized access to user accounts. These credentials typically consist of alphanumeric characters, and may sometimes include special characters.

(3) *Private Conversation:* Multi-user VR applications incorporate social elements that include both professional and social activities. Users often engage in private chats within these applications to communicate with business partners or friends, which could include the exchange of sensitive information, such as business-related data or personal matters. The private conversation is usually composed of strings of alphabetical characters.

Given that recent applications offer functionalities such as in-app purchases of virtual items and private user chats, the entry of private information, as explained above, has become common in multi-user VR applications.

Adversary Knowledge. For the scope of the attacks discussed in this chapter, we operate under the assumption that the adversary operates a remote client (an unmodified client located on the adversary's side that can receive updates about other players sent by the server) of a multi-user VR application and does not have access to the victim's devices or local environment. The target multi-user VR application should have both typing capabilities and the functionality to synchronize user motion across a network infrastructure. To the best of our knowledge, these features are ubiquitously supported across current multi-user VR applications.

Also, we assume that an adversary is a legitimate user (by downloading the application and registering an account in the application) who can enter a virtual room with other users. This presence makes the other users in the room potential victims (i.e., the attacker does not necessarily follow and target a specific user). This assumption is reasonable because, in multi-user VR applications like Rec Room, most rooms are public to facilitate the applications' purposes of socializing and meeting

strangers.

Being in the same virtual room allows the adversary's application client to receive motion updates from all other users in the room, including their typing-related motion data, even if their avatars are not visually seen by the adversary in the application. Moreover, the attacker can differentiate and group motion updates from different victim users. This is because current network protocols for multi-user VR applications require a unique user identifier to associate user avatars with their network updates. Therefore, by grouping motion updates using this user identifier, the attacker can analyze motion updates from different user entities, and perform the attack independently on each of them, thereby stealing keystrokes from all users in the room. Note that this user identifier only allows the attacker to group the inferred keystrokes and associate these keystrokes to any network updates linked to the user identifier (e.g., in-game username); further linking such information back to each user's identity in the real world is out of scope.

Additionally, given that the adversaries control their own clients, we assume they have the ability to access the binary files of the application client and capture the network traffic that their client sends and receives. Furthermore, we assume that the adversary can prepare for the attack by studying the application behavior. For example, the attacker can create multiple accounts, perform typing-related experiments on their own clients using these accounts, and observe the visual outputs of the application (e.g., how the keyboard looks).

3.3 Methods

The basic insight behind our approach is that multi-user VR applications transmit a user's VR motion data over the network and make it available to all other users who are in the same virtual room. This includes the motions that occur when a user is typing. Thus, typing-related motions are received by everyone in the same room. Consequently, anyone can analyze the network traffic and reconstruct the user's typing motion to infer the user's keystrokes.

As mentioned previously, it is challenging to extract motion data from packets and perform accurate keylogging attacks using low-fidelity motion data. We solve these challenges with a four-step approach, which is outlined in Figure 3.3. The first step takes the network packets and

filters for those including motion data. It is followed by a step that parses the packets and identifies the data fields within each packet. The third step recovers the semantics of these fields and extracts the motion data. In the last step, the extracted motion data is mapped to key positions, which provides us with the prediction of the user's keystrokes. Our approach differs from existing keylogging approaches as it only requires network packets collected from an attacker's own application client as inputs.

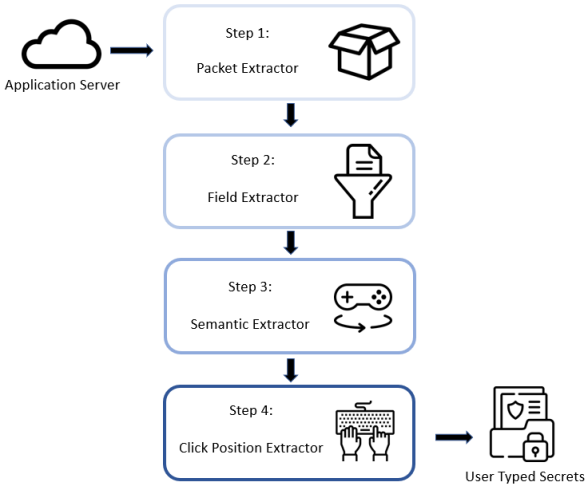


Figure 3.3: Our approach consists of four extraction steps to extract keystrokes from network traffic.

In this section, we present how we implemented each of these steps for a specific target. In particular, we choose Rec Room, one of the most popular multi-user VR applications that has more than 15 million users [Lyo], as an example to demonstrate our attack.

3.3.1 Step 1: Packet Extraction

The first step of the approach is to capture the raw network traffic and filter out the packets irrelevant to motion data. To recover motion data, we use Wireshark [wir23] and locate the incoming network traffic of our Rec Room client by its application port. Note that a single VR application is typically communicating with different servers (IP addresses) that handle different channels (such as voice data, messages, motion updates, etc.), but each server uses a fixed IP address while we stay in the room. In order to find the server that sends traffic specifically related to motion data, we group the network traffic from the VR application by source IP address, and take the packets from the IP

address with the highest transmission frequency. These packets will most likely contain the motion data sent from Rec Room's server, as it updates motion data with clients at a high frequency (a fixed rate of 15 packets per second). Therefore, by performing this filtering step, we can exclude the majority of network traffic irrelevant to motion data, and limit the scope of analysis for the next step.

For other multi-user VR applications, this strategy can be similarly applied, as the application needs to update users' motion data at a high frequency to ensure smooth motion, which makes it easy to isolate the motion update traffic. In addition, we can also use other common characteristics of motion updates to further eliminate irrelevant packets (e.g., motion updates are usually sent as UDP packets).

3.3.2 Step 2: Field Extraction

Once we obtain the packets that contain the user's motion data, we need to parse the packets and extract data fields serialized within the packets, so that we can extract the semantics from the fields later. In Rec Room, we implemented this step by first using a parser for Photon Engine [Pho], a networking library that Rec Room uses and whose network protocol parser can be found in an open-sourced project [Alt]. With this parser, we are able to parse the packet into data fields, and extract the types (e.g., integer, float, etc.) and values of the data fields. However, Photon allows developers to define and transmit their own custom objects (i.e., objects of custom data type, which may include multiple fields defined by the developers), and Rec Room uses this feature. Without knowing the fields serialized inside of these objects, we only see them as blobs of raw bytes. To parse these objects, we decompile Rec Room and find how it defines and deserializes the relevant objects, which allows us to build a parser that breaks down these objects into individual fields.

For other multi-user VR applications, we can replicate this implementation with mostly automated steps, and some reversing effort. It is likely that we can reuse the network protocol parser. Many applications use well-established networking libraries (e.g., Photon Engine and Unity Netcode [net23]) instead of implementing networking functionalities from scratch. For example, among the **34** multi-user VR applications that we investigated, **21** of them use Photon Engine, including Rec Room and VRChat [VRC23], two of the most popular multi-user VR applications

on the market. Therefore, we can spend a one-time effort to find or implement the protocol parser and reuse it on other multi-user VR applications, and only spend reversing effort to recover the custom objects if the applications define them (as Rec Room does). However, if it is a rare case that the application does not use any networking library, we will need to reverse how the application deserializes the entire packet, which requires more manual effort.

One possible extra obstacle in this step is that the network traffic may be encrypted. While it is rare that VR applications encrypt the packets that include motion data (only **1** out of the **34** multi-user VR applications that we investigated encrypts their packets that include motion data), this obstacle can still be overcome. The key exchange process and decryption happen on the attacker's own client. Therefore, the attacker can decrypt the network traffic with a user CA certificate to conduct a Man-in-the-Middle attack on their own client, and previous work [TLC22] has successfully performed this step and decrypted network traffic for 140 VR applications.

3.3.3 Step 3: Semantics Extraction

After parsing all the fields, it remains unknown which fields correspond to the typing-related information, and hence, we need to associate the fields with their semantics. Specifically, we aim to find the fields with the following semantics, which are needed for the next step:

(1) *Body motion data*, which includes TRANSFORM (i.e., the position and rotation) of the left hand, right hand, and head. They are used to track a user's typing motion. Such data is widely used in multi-user VR applications to update a user's body position.

(2) *Click data*, which includes the trigger values of the left controller and right controller. They are used to determine when a user is performing a click. This data is typically included in multi-user VR applications to update users' hand gestures.

(3) *Keyboard-opening event data*, which is signaled through the menu-opening event, is used to determine the keyboard position and when the user may start typing. This event data is specific to certain applications, but it is optional, as previous work [GSW23] demonstrates that the keyboard location can also be accurately approximated using the bounds of hand motion data. Also, the starting point of user typing can be inferred using click patterns as discussed in [WSZ23b].

(4) *User identifier*, which associates the motion update with the user (avatar) who that update belongs to. This field allows us to extract a separate motion data stream for each user when there are multiple users in the room.

To identify fields (1) to (3), we perform an experiment in which we (a) programmatically provide controlled motion data by running simulated VR hardware inputs, (b) compute the changes in packet fields, and (c) manually observe how changes to the inputs correspond to changes in the packet fields. This experiment is explained in detail in Section 3.3.5.1. Identifying field (4) is trivial since each motion update is annotated with a user identifier (a built-in field of the Photon protocol).

This step can also be replicated in other multi-user VR applications. Since most multi-user VR applications use Unity, which uses a standard data structure to represent motion data, we observed similar associations between the motion data inputs and the packet fields across applications. Furthermore, this experiment can be performed on any other multi-user VR application, since it only relies on manipulating the VR hardware inputs and observing packet data. Moreover, even if the mapping is not obvious (e.g., if an application applies obfuscation in the packets), we can resort to reversing the semantics from the application's binary by tracing how the packet fields are being used, which would require some more manual effort.

3.3.4 Step 4: Click Position Extraction

Once we extract the motion data, the last step is to infer keystrokes by (1) finding the TRANSFORM (i.e., the position and rotation) of all keys, (2) detecting the timing of the clicks, (3) calculating the cursor's TRANSFORM (the cursor refers to a point that projects a line to select keys) at click time, and (4) finding the intersection of the cursor's projection and keys. Essentially, these steps are reversing and emulating how multi-user VR applications compute keystrokes.

Firstly, we need to calculate the keys' TRANSFORM. To do this, we first need to know (a) how the keyboard is positioned relative to a user (see **Keyboard Measurement** in Section 3.3.5.2 for how this is measured), and (b) how the keys are positioned on the keyboard (see **Key Measurement** in Section 3.3.5.2), both of which only need to be measured once for an application in the attack preparation stage, as they are fixed. Then, we can find the packet that contains the *keyboard-opening*

event, and use the *body motion data* in the packet to find the user's TRANSFORM at the time of keyboard opening. With this information, we can use the pre-measured values in (a) to calculate the absolute position of the keyboard, and use the pre-measured values in (b) to calculate the positions of each key.

Secondly, we need to determine the timing of the clicks. That is, we find in which packet each click happens, matching one packet to a click, so that we can later analyze the motion data in each of these packets to understand where the cursor is at the time of the click. This can be done by analyzing the *click data* and performing click detection (see Section 3.1 for details about how clicks are performed and detected in VR).

Thirdly, we need to calculate the cursor's TRANSFORM. To do this, we first need to know how the cursor is positioned relative to the hand, which is a fixed relation (e.g., the hand position is at the palm of the hand, whereas the cursor is at the tip of the index finger). Similar to the pre-measured values of the keyboard, this also needs to be measured only once for an application (see **Cursor Measurement** in Section 3.3.5.2). Then, for each packet matched to a click, we find the hand's TRANSFORM from the *body motion data* and use this pre-measured value to obtain the cursor's TRANSFORM.

In the fourth phase, knowing the cursor's and the keys' TRANSFORM during a click, we can detect the actual key that was clicked. Specifically, as illustrated in Figure 3.4, we project a line (the blue arrow lines in the figure) following the cursor, and when this line intersects with any of the keys on the keyboard, we log this information (the blue dots on the keys in the figure). Intuitively, one can visualize a line through the fingertip of the avatar's hand and see where this line points to and intersects the virtual keyboard. Note that this process is the same as how applications would calculate user keystrokes on local clients, and we are essentially "replaying" the keystroke entries extracted from the network packages in this step.

In other multi-user VR applications, the pre-measured values of the cursor and keyboard may be different from Rec Room, and we need to repeat this one-time measurement process. Note that this process can be performed on any other multi-user VR applications, as it operates by manipulating the VR hardware data and observing visual outputs.

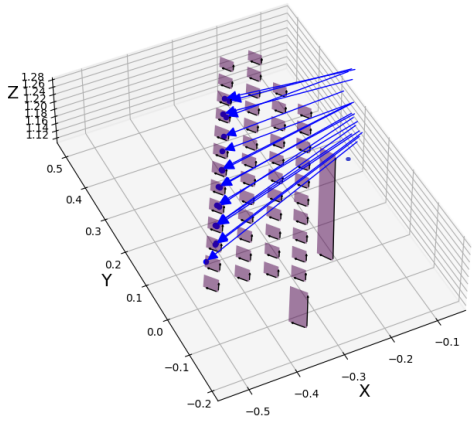


Figure 3.4: By visualizing the extension of the cursor with blue arrow lines, we calculate its intersection with the keyboard, as indicated by the blue dots on the keys, to log the key selections.

3.3.5 Reverse Engineering Challenges

In our approach, there is information that needs to be extracted once per application in the attack preparation stage: the semantics of key fields in network packets (Step 3) and the pre-measured values of the cursor and keyboard (Step 4). One of the most effective ways to obtain such information is to debug an application by accessing the application's memory while manipulating the inputs and observing the outputs at runtime. However, due to protections from anti-cheat engines [EPI23, fac23, bat23], attaching a debugger to multi-user VR applications is difficult. In this section, we introduce two methods to overcome these challenges without the need to bypass the anti-cheat engines. They employ only the readily available channels: VR hardware input, network packet output, and visual output (as mentioned in our threat model in Section 3.2). These methods enable us to (1) understand the associations between hardware inputs and packet fields by providing controlled inputs and (2) precisely measure the cursor and keyboard TRANSFORM (i.e., position and rotation) using geometric manipulations of the hardware inputs and visual validation.

3.3.5.1 Precise Input Control

In Step 3, we aim to recover the semantics of packet fields. To do that, we want to provide precisely controlled inputs to the application and observe how changes are reflected in the fields of observed packets. VR device inputs are highly dimensional (e.g., a controller has 6DOF inputs to represent

its position and rotation), and we need to isolate and control each input dimension to understand the effect of their changes. Moreover, such inputs cannot be easily controlled manually because of the sensitive sensors built into the VR hardware. For example, if we want to determine the effect of only changing the x coordinate of the left controller position, we need to (1) fix the right controller and head-mounted display (i.e., the VR headset) and avoid even the slightest movements, (2) move the left controller along the x-axis without any movement in the y-axis and z-axis, and (3) prevent any rotation in any directions. Using the actual controller, performing such a task "by hand" is essentially impossible.

To solve this challenge, we leverage Nvidia's VR Capture and Replay (VCR) tool [Nvi23], which allows developers to record tracking data from VR devices and replay them. Specifically, we utilize the "replay" component of VCR to serve as a VR hardware simulator and use it to run programmatic inputs. With this feature, we can (1) locate the fields affected by each input by isolating changes in the input, and (2) uncover conversions from inputs to fields.

First, we associate inputs and fields by observing their correspondence. For example, if we want to locate the x coordinate of the left hand, we can (1) fix all other input components in our replay script and only change the x coordinate, then (2) only fix the x coordinate and change every other input component. If there is a field in the network packets that changes if and only if x changes, it is highly likely that this field corresponds to x.

Second, after we locate an input component's corresponding field, we can reason how they are converted by observing how the field value changes with the input. For example, we can vary the x coordinate from -1.0 to 1.0 and see how the field for x changes accordingly.

3.3.5.2 Precise Measurement

In Step 4, we can parse the TRANSFORM (i.e., position and rotation) of hands and head from packets, but the cursor and the keyboard information are not directly (explicitly) included in the motion data, even though they have certain fixed relations with the motion data (hands or head). We could attempt to leverage the visual outputs to observe these hidden attributes. However, this presents a challenge due to the difficulty in obtaining precise direct observations. For example, just by looking

at the visual outputs, it is hard to gauge the distance between two virtual objects, and an error of just 10 cm can shift a key prediction by three keys (a key has a side length of about 3cm) and render the attack ineffective.

We present a method to solve this challenge and measure the "hidden attributes" with visually verifiable tests. Specifically, we design geometric tests, which move virtual objects in specific ways to visualize geometric properties, so that we can verify our measurements visually. We perform the measurements in two steps: (1) Cursor Measurement: we measure the relations between cursor and hand, and uncover how we can convert the hand's TRANSFORM that is present in packets to the cursor's TRANSFORM. (2) Key Measurement: we use the cursor as a reference point to measure the keys.

Cursor Measurement. Cursor measurement is the problem of finding the fixed *offset* (i.e., a constant transformation in position and rotation) between the hand's TRANSFORM in packets (referred to as TRANSFORM_{hand}) to the cursor's TRANSFORM (referred to as $\text{TRANSFORM}_{cursor}$). From inputs, we can only manipulate the controller's TRANSFORM (referred to as $\text{TRANSFORM}_{controller}$), which has a fixed *offset* to both the hand and the cursor. These three TRANSFORM values can be all different (in position and rotation), as seen in Figure 3.5. At a high level, we can solve this problem by first finding the *offset* between the hand and the controller, then finding the *offset* between the controller and cursor, and combining these two *offsets* to get the *offset* between the hand (which appears in the packet data) and the cursor. The detailed steps are as follows:

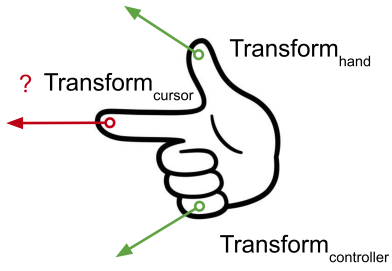


Figure 3.5: We need to uncover conversion from hand's TRANSFORM in packets to cursor's TRANSFORM by manipulating controller's TRANSFORM.

(1) Find the *offset* between TRANSFORM_{hand} and $\text{TRANSFORM}_{controller}$. Since we can read out TRANSFORM_{hand} from packets, it is straightforward to find this *offset* by inputting one sample value of

TRANSFORM$_{controller}$ and observing the corresponding TRANSFORM$_{hand}$, then calculate the constant *offset* in position and rotation between them.

(2) Find the offset between TRANSFORM$_{cursor}$ and TRANSFORM$_{controller}$. This can be broken into three steps: (a) We start with a guess for one part (i.e., position or rotation) of this *offset*. For example, if we want to measure the rotational offset, we can start with the guess that the controller and cursor have the same orientation (offset is zero). (b) Next, we perform a spatial binary search of the real offset. That is, we plug in the guessed offset as a part of controller inputs and run geometric tests to visually inform us whether our guess is off from the real offset, and in which direction the real offset is. For instance, we can perform the geometric test as seen in Figure 3.6 and move the hand along the initial guessed direction. If the offset is non-zero, we are not moving the cursor along its pointed direction, and the reticle (cursor's projection on a screen) will move. (c) Then, we iteratively adjust our guess until the geometric test shows that the guessed offset aligns with the real offset. Continuing from the example in (b), we can keep adjusting the guess based on which direction the reticle deviates, until the reticle does not move, which is easy to verify visually. At this point, we know that the hand is moving along the cursor's orientation, and thus our guess is correct.

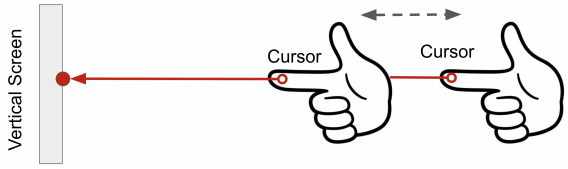


Figure 3.6: Geometric Test 1: test if the guessed cursor orientation is correct by moving the hand along the guess.

(3) Calculate our target *offset* by combining the two obtained *offsets*. The *offset* in (1) takes us from TRANSFORM$_{hand}$ to TRANSFORM$_{controller}$, and the *offset* in (2) then takes us from TRANSFORM$_{controller}$ to TRANSFORM$_{cursor}$, so combining them takes us from TRANSFORM$_{hand}$ to TRANSFORM$_{cursor}$, which is our goal.

Key Measurement. Key measurement is the problem of finding the positions of the four corners of a key at a fixed player location. The four corners of a key determine the key's location, as they define a bounded plane region as the key's region. For this problem, we need to know the cursor's TRANSFORM, and use it as a reference. To measure a corner of a key, we (1) position the cursor at an

arbitrary position and find a way to point the cursor at the corner (again through a spatial binary search procedure), then (2) find the distance between the cursor and the key corner using a spatial binary search with another geometric test, and lastly (3) calculate the key's position by adding a vector that is along the cursor's pointing direction and of the length equal to the distance to the cursor's position.

Keyboard Measurement. Keyboard Measurement is the problem of finding each key's position when the player is in an arbitrary position and orientation. This problem is straightforward to solve once we are able to perform **Key Measurement**. We can use the originally measured keys as a baseline, and repeat measuring all keys' positions at multiple player positions and rotations. This allows us to calculate how the changes in player positions and rotations transform into changes in the keys' positions. Therefore, for a new player's position and rotation, we simply calculate the changes in position and rotation from the baseline, and apply this transformation.

3.4 Data Collection

We performed data collection experiments to assess the effectiveness of our attack in real-world scenarios. Specifically, we conducted a user study to collect real-world typing activity data from Rec Room, one of the most popular multi-user VR applications, as the main experiment. We have also conducted further experiments to demonstrate that our attack works well in three additional applications of various genres and in realistic scenarios.

Data Collection Setup. During the user study, we installed and ran Rec Room in VR mode through SteamVR on a Windows PC, with an Oculus Quest 2 connected via Quest Link (we will refer to this as the *Victim PC* later). We installed and ran Rec Room in non-VR mode on another Windows PC to simulate the attacker behavior (we refer to this as the *Attacker PC*). The two PCs then join one private virtual room, as mentioned in our threat model. After we begin capturing network traffic with Wireshark on the *Attacker PC*, the participant, wearing the Quest 2 headset, starts entering the prepared text into Rec Room's chat functionality using the virtual keyboard. Lastly, to serve as a reference during the data labeling process, on the *Victim PC*, we use VCR to record the VR tracking data during each session.

User Study Recruitment. We conducted a user study to collect VR typing data from volunteers using the *Victim PC*. This study has been approved by our university's Institutional Review Board (IRB). Through email advertisements via our institution's email list, we recruited 20 participants of varying ages, heights, genders, and experience with VR from our institution's campus. Prior to the experiment, participants were provided with instructions on how to operate the Quest 2 headset and controllers. Additionally, they were guided on the typing functionalities and practiced typing within Rec Room. Before the start of the experiment, the participants were also informed that their typing-related information would be recorded and that they could stop the study at any time. To minimize bias and more closely mirror real-world typing scenarios, participants were not initially informed of the study's exact purpose. Instead, they were simply told that the goal of the study was to examine VR typing behavior. After the participants completed all the typing experiments, we debriefed them about the real purposes of the study and asked them to fill out a survey to understand their perceptions of our attack. The study lasted approximately 60 minutes per participant, and each participant who took part in our study was compensated with a $20 gift card.

User Study Typing Trials. For each experiment, a participant started by opening the Rec Room chats to perform typing trials. In each trial, the participant was presented with a prompt and was asked to type all the characters in the prompt into the chat. In total, each participant completed 65 trials, of which 30 trials were with number prompts, 20 were with password prompts, and 15 were with sentence prompts. To simplify the setup, all prompts were shown to the participants in the same chat in which they performed the trials, so they could see the prompt while typing. We had a researcher continuously observing the participants' inputs and asking the participants to retype a prompt if they mistyped it (e.g., they typed "124" when the prompt was "123"). By doing this correcting step, the prompts could conveniently serve as the ground truth without modifications during the evaluation of the attack accuracy later.

As previously mentioned, the participants engaged in trials with three types of prompts, each representing a common typing behavior:

(1) *Numbers*: The prompts are random numbers with lengths of 3 (e.g., "823"), 9 (e.g., "804458083"), and 12 (e.g., "595397360820") digits, with 10 prompts per length. This simulates entering credit card information.

(2) *Passwords*: The prompts are passwords with characters, numbers, and punctuation marks. To emulate real-world password typing activity, we generated 10-17 characters long combinations of English words with numbers and punctuation marks, using the Memorable Password Generator [mem23].

(3) *Sentences*: The prompts are English sentences that consist of 3 words (e.g., "Clouds are drifting"), 6 words (e.g., "The sunset today looked absolutely stunning"), and 9 words (e.g., "My cat just did the funniest thing this morning"), with 5 prompts for each length. These sentences are generated by ChatGPT [Ope22]. This experiment is to simulate the scenario of typing in private chats.

After the study concluded, we asked the participants if they wished to exclude their data and address any questions they might had. At the time of writing, we have not received any requests for excluding data.

Data Labeling. During the study, we utilized a timing application to mark the beginning and end of each trial. This ensured only data from the trials was considered for labeling. Then, we check the recording of tracking data during the trials to verify that the number of clicks corresponds to the expected number of clicks from the prompts. For example, if the prompt asks the user to type "123," then we should expect three clicks in the recording of tracking data. If the number of clicks matches, we then label each click using the characters in the prompts along with its timestamp in the tracking data recording, so that we can later compare the clicks detected in the network traffic with the ground truth clicks. However, if the number of clicks does not match, it signals that the prompt was mistyped, which may happen when the prompts are long, and the researchers did not spot the errors in what the participants typed during the study. The data for such trials is discarded and not considered for evaluation. However, these cases are rare. In total, we dropped a combined 638 clicks out of 22,730 clicks (29 out of 1,300 trials) collected from the 20 participants in our study. This leaves 22,092 keystrokes (from 1,271 trials) for the evaluation.

Other Practical Scenarios. We also conducted an experiment to demonstrate the practicality of our attack in scenarios where (1) there are multiple users in a virtual room, and (2) the attacker does not see the victim users. We created the same experiment setup as the **User Study Typing Trials**

experiment but with two modifications: (1) we placed another four different victim users in the room instead of one; (2) the attacker faces a wall, thus they are unable to see any other users. We then conducted a user study with the same procedure as the main experiment but with two participants typing concurrently (the other two users in the room were dummy users). Since the attack worked exactly the same in this setting, we ran this experiment only once. In total, we collected 2,431 clicks from the participants.

Other Multi-user VR Applications. To show that our attack generalizes beyond Rec Room, we tested it on three additional popular applications from diverse genres: Galaxity [gal24], Sing Together: VR Karaoke [sin24], and oVRshot [ovr24]. Specifically, we conducted an end-to-end attack on each of these applications following the same attack procedure (Section 3.3) and setup of the experiment on Rec Room (the **User Study Typing Trials** experiment). Among these applications, we targeted keystrokes on the full keyboards of two (Galaxity and Sing Together: VR Karaoke), whereas in oVRshot, attention was directed to the numeric keyboard, given that it is the only type available. During the typing experiment, which was conducted with three participants for each application, we collected a total of 3,423 clicks for Galaxity, 3,473 clicks for Sing Together: VR Karaoke, and 760 clicks for oVRshot (numbers trials only). Note that for passwords and sentences trials, we selected random words that can have slightly different lengths, which leads to a slight variance in total keystrokes for different applications.

3.5 Evaluation

In this section, we evaluate our attack and analyze various factors that may affect the attack accuracy. We have analyzed whether the attack accuracy is affected by packet drop rates, key position on the keyboard, and typing speed.

To evaluate the attack accuracy, we use top-k accuracy, which assesses how many of the user-typed keys are correctly predicted within our attack's top-k predicted keys, sorted by distance from the position of the cursor's projection on the keyboard. In other words, the top-k accuracy is calculated as (number of successfully inferred keys) / (number of total keys). If the click is not identified by the keylogging attack, we automatically mark the prediction as wrong.

	Top 1	Top 3	Top 5
Numbers	96.78%	97.98%	98.23%
Passwords	97.42%	97.88%	98.17%
Sentences	98.16%	98.41%	98.52%
Total	97.62%	98.15%	98.34%

Table 3.1: Keylogging attack accuracy from 20 participants across three different typing tasks.

Table 3.1 presents the overall accuracy of our attack based on 22,092 keystrokes collected from all 20 participants in the user study. The evaluation setup is consistent with the character-level evaluation in prior research on VR keylogging attacks [LHY22, GSW23, WSZ23b, SZA23].

Across all three types of prompts, our keylogging attack consistently exhibited a very high accuracy, inferring 97.62% (21,567 out of 22,092 keystrokes) of all keystrokes correctly with top-1 prediction, 98.15% (21,683 out of 22,092 keystrokes) with top-3 prediction, and 98.34% (21,726 out of 22,092 keystrokes) with top-5 prediction. Against a 2.13% random guess baseline for one-key inference, our attack's effectiveness increases over 45 times.

Although our attack is nearly perfect in accuracy, there are a few incorrect predictions. This can happen for one of the following two reasons: (1) Loss of motion data precision due to lossy compression of the data: To ensure stable network performance, multi-user VR applications such as Rec Room compress motion data items before transmitting them over the Internet. When this data is decompressed by the adversary client, it will be slightly off compared to the ground truth motion. This slight imprecision in motions and cursor positions might cause inaccurate predictions. (2) Stale hand motion update: From our experiment, we observed that some hand motions are not updated (propagated) immediately. This might happened because of the lag on the victim client or the interpolation mechanism applied on the previous motion update. Consequently, when this happens, the hand's TRANSFORM associated with a click tends to be closer to the previous click

rather than the intended target click and leads to incorrect keystroke predictions by our attack.

It should be noted that the loss of motion data precision due to the compressed data is not significant. Additionally, the cases of stale hand motion occur with a low probability. This explains why only 2.38% of the keystrokes in our experiment are predicted incorrectly.

These findings from our attack highlight the significant risk of privacy breaches due to motions detected in the network packets, emphasizing the urgent need for protection against such vulnerabilities.

3.5.1 Our Attack is Robust Under Traffic Loss

Since the attack is conducted remotely, it is crucial for us to study the packet loss during the transport of motion data and how it affects the accuracy of our attack. According to a recent article by Obkio [Lam23], apps with a packet loss rate of more than 10% are considered unusable [Lam23]. To stress-test our attack, we conduct an analysis by dropping random packets at an even higher rate of up to 20% from the participants' network captures. The corresponding attack accuracy is illustrated in Figure 3.7. From the result, we observe that the attack accuracy after 20% of random packet drops rate still remains very high (94.97% top-1 accuracy), showing that our attack is robust even under significant packet loss.

In total, we lost 2.65% of the keystroke recovery accuracy due to the following two reasons: (1) Loss of brief keystroke data due to packet loss: In instances where the click duration is extremely short, the motion data corresponding to this key might only be present in a few packets. If there is packet loss, these packets might be dropped. As these packets are never received, they cannot be used for key recovery; (2) Packets after a click motion are considered as the actual click: By definition, a click occurs the moment the victim presses the key beyond the click threshold, as mentioned in Section 3.1. However, a packet loss might occur at the moment of a click. As a result, the initial packet representing the click might be missing. In this case, our attack identifies a subsequent packet during the click period as the click packet (i.e., the attack thinks this later packet is the first frame of a click). However, the subsequent packet might show a minor difference in the hand's TRANSFORM from the original click packet, resulting in a variation in the click motions and

causing a wrong keystroke identification.

It should also be noted that we observed only an additional 0.42% of clicks undetected when a 20% packet loss is introduced. Also, the situation where selecting later packets results in a decrease in attack accuracy only arises in specific cases. It happens only when the click motion changes rapidly and the initial few packets are lost. This is why our attack accuracy is only slightly affected by the significant packet drop rate.

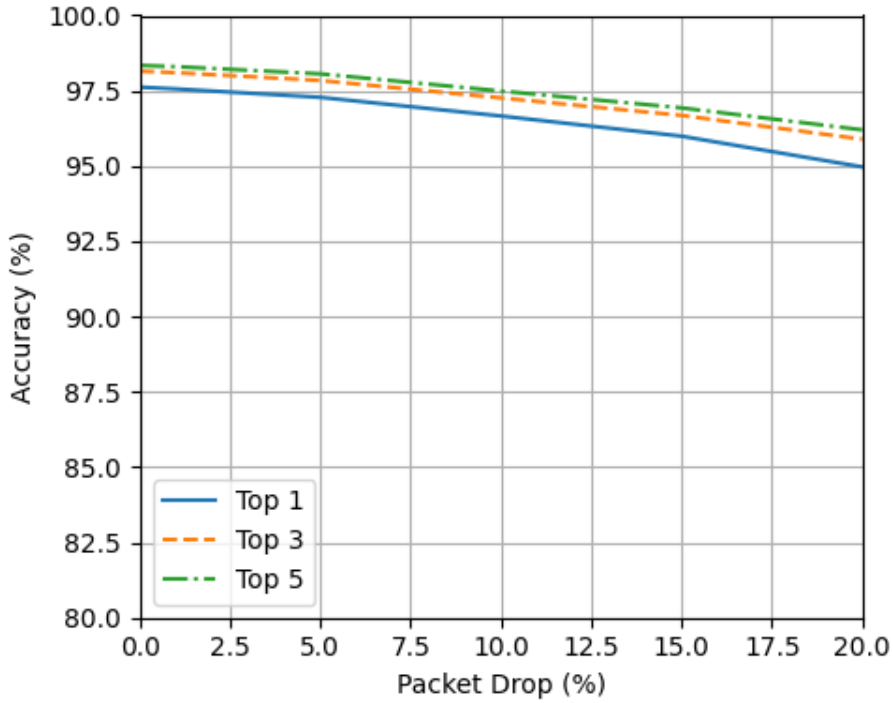


Figure 3.7: Our attack is robust against random packet drop, achieving a top-1 accuracy of 94.97% even when 20% of the packets are dropped.

3.5.2 Keyboard Layout Affects Attack Accuracy

Given that VR users utilize their hands to control the cursor to select keys on the virtual keyboard, we hypothesize that keys positioned farther away from the user may be more challenging to predict accurately as these keys are relatively smaller in the user's field of view. Our intuition is based on the fact that the cursor, when pointing at a distant key, adopts a more tilted angle. Consequently, this might increase the likelihood of an incorrect prediction by the attack as the inference to the

relatively smaller keys is more sensitive to deviations introduced by the transmission process.

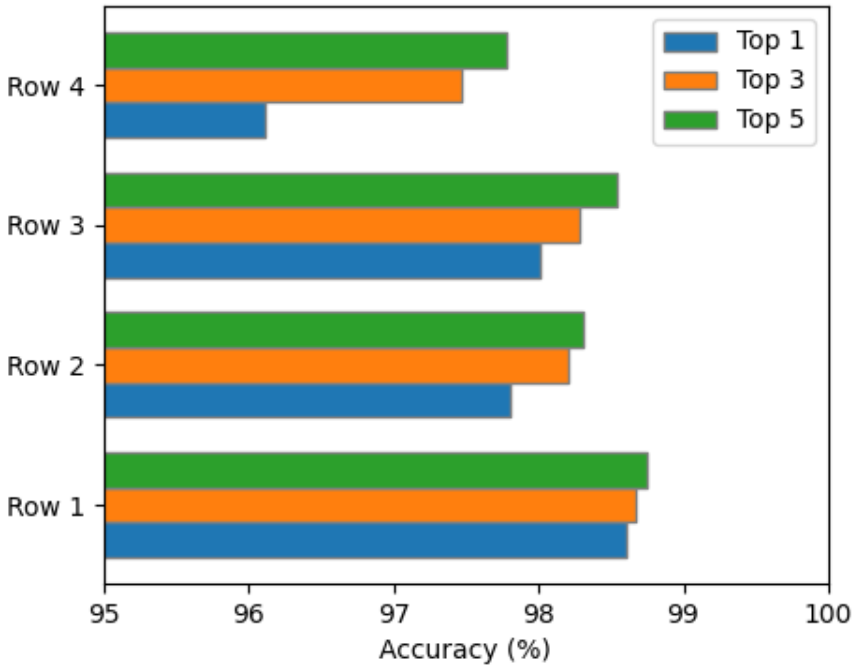


Figure 3.8: The keys positioned further away from the user are relatively more difficult to infer compared to the closer keys.

To test this hypothesis, we categorized the keys on the keyboard into four rows based on their layout. Row 1 starts with keys 'z,' 'x,' and 'c,' Row 2 starts with keys 'a,' 's,' and 'd,' Row 3 starts with keys 'q,' 'w,' and 'e,' and Row 4 starts with keys '1,' '2,' and '3.' The accuracy rates for each row are displayed in Figure 3.8. Although the attack accuracy remains high (with a top-1 accuracy of >96%) across all rows, there is a noticeable trend: the accuracy diminishes as keys are positioned further away from the user (for instance, top-1 accuracy drops from 98.6% in Row 1 to 96.12% in Row 4).

3.5.3 Our Attack Remains Robust Under Varying Typing Speed

We also studied how the typing speed impacts the accuracy of our attack. Intuitively, predicting faster clicks might be more challenging since the cursor moves more quickly. Even a slight lag could significantly affect the position of the cursor's projection on the keyboard. We present accuracy data

grouped by click duration percentiles (e.g., the 0-20 percentile group comprises the fastest 20% of clicks). Contrary to our initial expectations, we did not observe a significant effect of typing speed on the attack accuracy. This finding could be attributed to the fact that even the fastest click (with a duration between 0.255 and 0.721 seconds for the 0-20 percentile group) is captured in multiple motion updates, given that motion updates occur frequently (15-20 updates per second, or 0.05 to 0.067 seconds between updates).

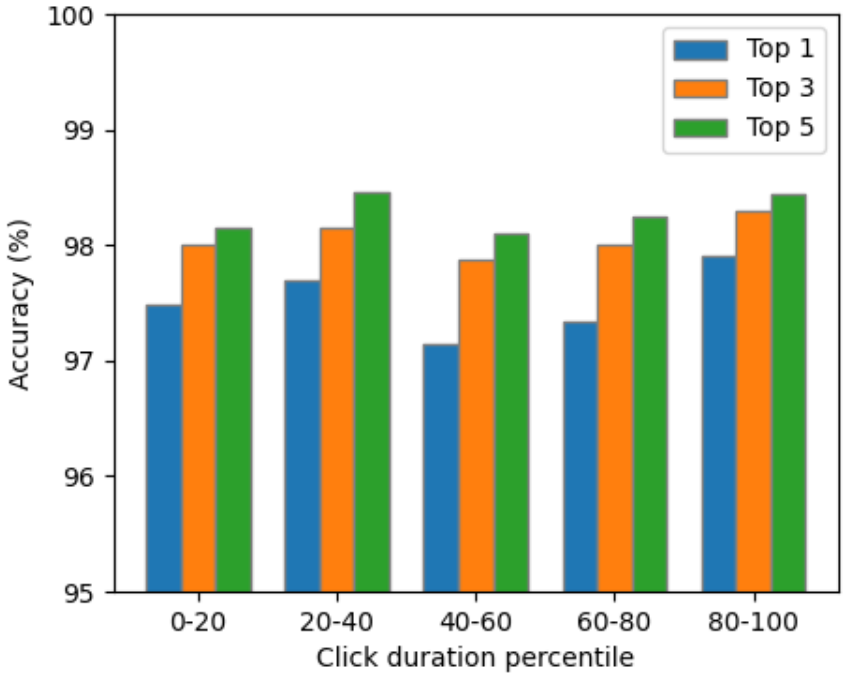


Figure 3.9: Our attack is unaffected by typing speed, with similar accuracy across different click duration percentiles.

3.5.4 Our Attack Can Be Stealthy Under Practical Scenarios

For the experiment setting in which (1) there are multiple users typing concurrently in the same room and (2) the attacker is facing the wall, the attack has comparable performance to the **User Study Typing Trials** experiment. Our attack correctly inferred 97.53% (2,371 out of 2,431) of all keystrokes with top-1 prediction, 99.51% (2,419 out of 2,431) with top-3 prediction, and 99.59% (2,421 out of 2,431) with top-5 prediction. This result shows that (1) our attack is not affected by

the number of participants in the room, and (2) even when the attacker cannot see the victim, the motion data is still received and can be used to recover the keystrokes. Thus, we conclude that our attack is applicable in a multi-user environment, and the attack is stealthy since the attacker can hide anywhere in the room while performing the attack.

3.5.5 Our Attack Generalizes Across Applications

	Top 1	Top 3	Top 5
Galaxity	98.25%	99.71%	99.73%
Sing Together: VR Karaoke	98.27%	99.97%	99.97%
oVRshot	99.07%	99.61%	99.61%

Table 3.2: Keylogging attack accuracy from three participants typing in the selected three additional applications.

By performing our proposed attack on Galaxity [gal24], Sing Together: VR Karaoke [sin24], and oVRshot [ovr24], we have successfully recovered the motion data from the collected traffic and infer the key participants typed.

The performance of our attack is as detailed in Table 3.2. The slightly varied attack accuracy across applications may be attributed to differences in keyboard layouts and participant typing habits. Nevertheless, all attack results demonstrated performance comparable to our main experiment on Rec Room, underscoring the generalizability of our attack across different applications.

3.5.6 Per-Participant and Per-Hand Variation

We further break down the attack's accuracy by participant and by hand to characterize how robust the attack is to individual variation.

Attack Accuracy Across Participants. Figure 3.10 shows individual differences in the attack's accuracy. While accuracy remains high across all participants (top-1 minimum 94.8%), there are individual differences (top-1 median 98.13%, maximum 99.47%). Several factors may contribute, including different typing habits and positioning since we did not constrain the participants' typing process. During the study, we observed that some participants leaned back and typed characters from a long distance (some even typed with awkward poses, positioning their hands above their shoulders), whereas others typed right in front of the keyboard. Therefore, it may be harder to predict keystrokes from those who were far from the keyboard, similar to how farther keys were harder to predict (see Section 3.5.2). Despite these differences, accuracy remains high across all participants, so the attack is robust to individual variation.

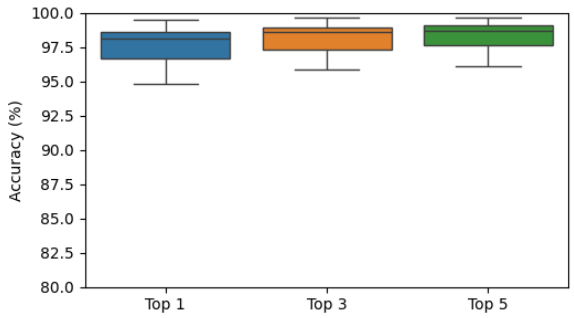


Figure 3.10: The accuracy of the attack varies slightly among different participants due to their varying typing habits.

Difference Across Hands. Figure 3.11 shows the attack's accuracy for both hands, which do not differ noticeably (top-1 accuracy: left 98.1%, right 97.51%). This is expected, as our attack uses hand motions to precisely calculate the keystroke, and the motions likely do not fundamentally differ between the left and right hands (e.g., they are positioned roughly the same distance from the keyboard).

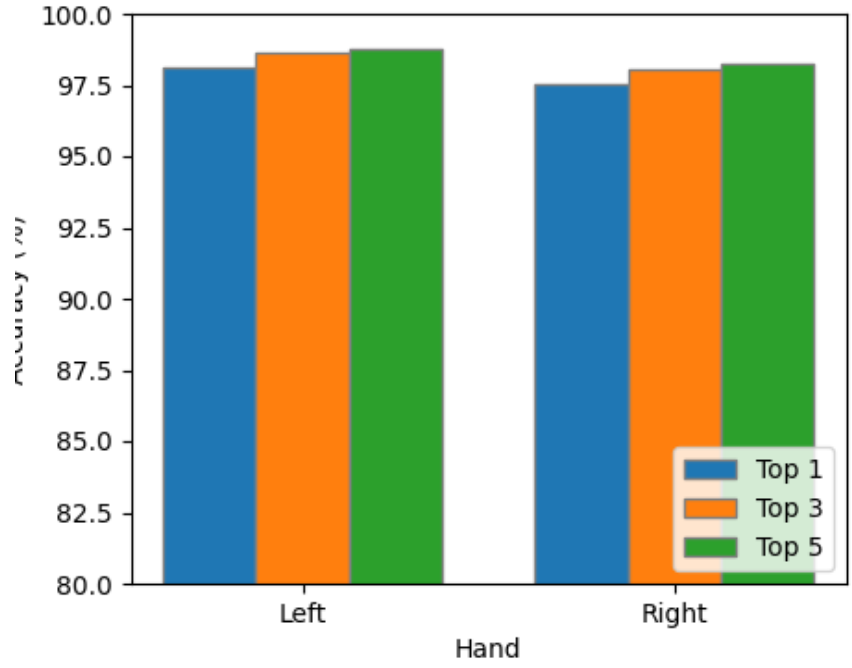


Figure 3.11: The attack accuracy is not affected by the hand the victim uses to type.

3.6 Keylogging is Possible Even with Partial Reverse Engineered Packets

In the previous sections, we demonstrated the effectiveness of performing a keylogging attack by fully reconstructing the typing process. However, the involved steps require some manual effort and can be time-consuming. If an adversary aims to execute our remote keylogging attack rapidly, with minimal manual reversing effort, the keylogging attack can still be performed with the help of machine learning. This approach operates under an additional assumption compared to the threat model introduced in Section 3.2. In particular, we allow the adversary to obtain some motion data from a victim's typing that is paired with the actual text being typed, serving as labeled (ground truth) data. This assumption is commonly accepted and used in previous VR keylogging work [SZA23, AGA21, ZSC23]. In the case of multi-user VR applications, adversaries can collect the necessary data by chatting with the victim (via the application chat) and capturing any messages with the corresponding motion data. Since the victim can perform multiple tasks without closing the menus, and the keyboard position is fixed once they open the menu, this collection step allows the adversary to use the labeled data to predict other keystrokes on the same keyboard location.

Attack Setup. To show that our attack works with limited reversing, we use the example of Rec Room to further demonstrate this attack. Figure 3.12 outlines the process.

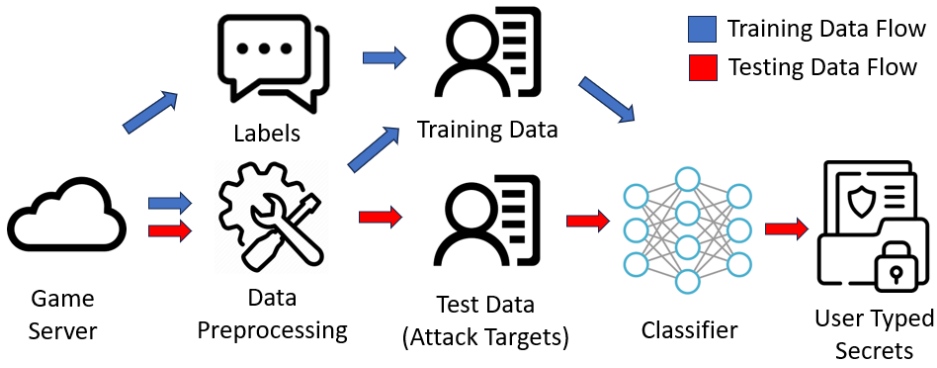


Figure 3.12: Machine Learning Attack Overview: The blue arrows illustrate the model training process, while the red arrows describe how an adversary might utilize the trained model to infer user keystrokes.

To build the training dataset for the machine learning model, we first collect raw packets and their corresponding labels (the typed text) using the aforementioned phishing approach. To preprocess the raw packets, we follow the steps from Section 3.3 that do not require significant manual effort and can be consistently applied across applications:

(1) Filter for the packets that contain motion data, as explained in Section 3.3.1.

(2) Use the protocol parser for Photon, as referenced in Section 3.3.2, to parse the packets, given that Rec Room employs the Photon library for motion updates. This step parses the packets into different fields of readable formats (e.g., integers, floats), except for the custom object fields, which are left as raw bytes. This step is easy because the Photon protocol parser is a readily available tool that can be applied to all applications that use the Photon libraries.

(3) Run the **Precise Input Control** experiment (see Section 3.3.5.1) to identify the fields that correspond to the click data (which are stored in floats), and the custom object fields that contain the motion data. This step is easy because the experiment can be run automatically, and we only need to manually observe the associations between the input data and the fields.

(4) Determine the timing of clicks by performing click detection, which is done in the same way as in Section 3.3.4. We can use this information to find the packets that correspond to clicks

and select the custom object field that contains the hand motion in these click packets (e.g., select the custom object field that contains the right-hand motion if the participant uses the right hand to click). Recall that click identification is straightforward via standard Unity APIs (see Section 3.1).

Compared to the attack described in Section 3.3, the resulting data will no longer have the following information: (1) The parsed motion data in a readable format. This requires reversing the parsing of custom objects, and this process needs to be redone for each application, as mentioned in Section 3.3.2; (2) The TRANSFORM of the cursor and key. This is because we do not have motion data in a "meaningful" format, so we cannot perform calculations on top of it (e.g., measure cursor and keyboard offsets).

After these data preprocessing steps, for each click, the resulting data provides only a custom object field in raw byte format that contains the motion data associated with typing. The resulting data is paired with the corresponding labels to create a dataset to train a machine learning classifier. Subsequently, we gather more traffic as the victim types. These attack targets (traffic with keystrokes) can be processed in the same manner as the training data and fed into the trained machine learning classifiers to infer corresponding keystrokes.

Attack Evaluation and Results. For evaluation purposes, we utilize the traffic traces collected from the user study and divide them into an 8:1:1 ratio for training, testing, and validation. Then, we follow the attack steps above to carry out the attack and evaluate the attack result using the top-k accuracy of the testing data.

To identify the best machine learning model for the task, we compare the attack results from various models, including SVM [CV95], GBM [KMF17], MLP [Ros58], and CNN [LBB98], to the Random Guess baseline. These models are trained with a learning rate of 10^{-4} for 500 epochs, from which we chose the checkpoint with the highest validation accuracy as the best attack model.

From the results in Table 3.3, we observe that all ML models extract the victim-typed keystrokes significantly better than the random guess baseline. Additionally, we observe that the CNN model performs the best, with a 68.07% top-1 accuracy, 85.96% top-3 accuracy, and 90.28% top-5 accuracy in predicting keystrokes. This result highlights that machine learning models like CNN can effectively learn about mapping the bits in packets to corresponding motions and click positions,

so the model can predict keystrokes with reasonable accuracy.

Our CNN model structure consists of three convolutional layers. Each layer has a one-dimensional kernel size of three, with 32, 64, and 128 neurons in each layer. Subsequent to these convolutional layers, the architecture includes two fully-connected layers and a softmax layer. The output space of this CNN model corresponds to the number of classes, equivalent to the distinct keys on the keyboard (47 in total).

The success of CNN might be attributed to its ability to learn from small datasets with generalizability [AZH21]. Also, the CNN model might be able to learn from the features with significant effects on the key presses (e.g., those bits directly related to the TRANSFORM of the typing hand) [AZH21], which can be effective in capturing relevant parts from the partially reversed data items that contain non-parsed custom fields used in this task.

	Top 1	Top 3	Top 5
Random Guess	2.13%	6.38%	10.64%
SVM	44.87%	64.47%	71.57%
LightGBM	46.49%	66.24%	71.61%
MLP	61.99%	79.81%	85.34%
CNN	68.07%	85.96%	90.28%

Table 3.3: Comparison of machine learning models on inferring the keystrokes with partially reversed data in raw bytes.

The accuracy is still lower compared to the full extraction keylogging attack we proposed earlier, which uses high-fidelity typing-related motion data. This is due to two reasons: (1) From the prior experiment, we find that a slight deviation in the motion data received by the adversary can cause an incorrect prediction since the motion data of typing nearby keys is similar due to the relatively small space of keys. As the model is trained on partially reversed data, the data might have less

learning signal to predict the key and might introduce errors. (2) As the model is trained on data items collected through chatting with a victim, the quantity of the data is also limited. This can potentially lead to the overfitting of the model and cause a deviation when inferring the motion data that the model has never seen, leading to a decrease in prediction accuracy. However, even with the drop in accuracy, the attack can still very effectively infer the victim's keystroke.

We further analyze the attack on partially reverse-engineered packets along three axes: training-data size, hand, and typing task.

More Training Data Allows Higher Attack Accuracy. From the results in Table 3.4, training the machine-learning classifier with more data lets it predict keystrokes with higher accuracy. This implies that if an adversary can collect more labeled typing motion from the victim, the attack results improve further.

Train Data	Top 1	Top 3	Top 5
20 Percent	30.16%	50.47%	63.01%
40 Percent	45.30%	67.81%	76.38%
60 Percent	57.14%	79.29%	86.54%
80 Percent	68.07%	85.96%	90.28%

Table 3.4: More training data enables the model to better learn how to decode the data.

Secrets Typed with the Right Hand are More Vulnerable. Table 3.5 shows that the data typed with the right hand is slightly more vulnerable to our attack. We attribute this to the fact that most users predominantly use their right hand for typing: more than 70 percent of the clicks in our dataset originate from the right hand. As a result, the model is trained on a larger volume of right-hand data, improving generalization on right-hand test samples.

	Top 1	Top 3	Top 5
Left(Total)	66.73%	83.34%	88.91%
Right(Total)	72.31%	86.44%	90.70%

Table 3.5: Secrets typed with the right hand are slightly more vulnerable to our machine-learning models.

Performance of the Attack on Different Tasks. From the results in Table 3.6, a user's typing of just numbers can be inferred with a top-1 accuracy of 83.63%. Sentences can be inferred with an accuracy of 70.59%, while password typing yields a lower attack accuracy of 58.67%.

We attribute this ordering to class structure. Numbers are more vulnerable because they belong to fewer classes (only 10), so each number appears more frequently in the training set than the alphabetic characters. Sentences are slightly more vulnerable than the main result because the distribution of characters within a sentence is unevenly weighted; some characters appear more often than others, making them easier to attack. Password typing can include any key on the keyboard, and the rarely-seen keys make passwords harder to attack.

	Top 1	Top 3	Top 5
Numbers	83.63%	97.91%	100%
Passwords	58.67%	77.23%	86.61%
Sentences	70.59%	84.03%	92.44%

Table 3.6: Attack accuracy for different typing tasks with partially reconstructed data.

3.7 Discussion

User Awareness and Concerns. From the results of the post-study survey, none of the participants had ever imagined that their input in multi-user VR applications could be stolen by adversaries. However, after disclosure, 12 out of 20 participants expressed concerns regarding this type of attack (Score > 4). These findings suggest that VR users do recognize the potential harm from such attacks. The user study results highlight the importance of educating VR users about potential security and privacy threats, such as our keylogging attacks in multi-user VR applications. This education can serve as a warning, encouraging users to exercise caution and prevent privacy leakage.

Privacy policies of VR applications play a crucial role in educating users about their data privacy. However, to the best of our knowledge, major VR applications often fail to clearly explain sensitive data like motion data in their privacy policy. Furthermore, Trimananda et al.[TLC22] find that approximately 70 percent of PII data sent in traffic of VR applications are not properly disclosed. This highlights the need for better disclosure and compliance efforts from VR developers.

Impacts of Our Attack. We have reported our attack findings to all of the applications we evaluated, as well as SteamVR and Unity. Rec Room, Sing Together: VR Karaoke and SteamVR responded, acknowledging the vulnerability. Furthermore, Rec Room classified it as *P3 severity (Medium: Vulnerabilities that affect multiple users and require little or no user interaction to trigger)*. This underscores the fact that our attack is feasible and poses a privacy risk to numerous users. At the time of writing, we still have ongoing discussions with SteamVR and Unity.

It should also be noted that our reverse engineering approach is not limited to recovering typing-related motion; it can also recover other types of sensor data shared over the internet, such as voice data. Our tool can serve as a baseline for future studies aiming to further reveal the potential threats posed by the misuse of sensor data in VR.

Moreover, our attack poses a threat not only to users in applications which we have performed our attack. For multi-user VR applications, the function of correctly synchronizing avatar movements and displaying them to other users is foundational, serving as the cornerstone of real-time interaction within the application. To accurately render user avatars and their movements, it's important to note that even if motion data is encrypted when sent over the network, it will eventually be decrypted on

every client. Therefore, attackers can recover motion data (of other users) received by their own client, which allows them to fully track how a victim's avatar moves when selecting a key. Since our attack targets and exploits the transmission of motion data, which is fundamental to the design of multi-user VR applications, the threat posed by the attack is not limited to the applications we have evaluated. In addition to them, we have analyzed the functionality of **30** multi-user VR applications from various platforms, including Oculus and Steam. Of the applications we studied, **18** offer virtual keyboard typing functionality and transmit motion data online, making their users potential victims of our attack. Additionally, the wide support of overlay system messaging apps like Steam Chat or Messenger on VR devices makes all multi-user VR applications potentially susceptible to our attacks, since the applications may transmit motion data when these overlays are activated.

Our Attack is Applicable Regardless of Physical Setup. It is important to note that the transmitted motion data represents the movements of in-game objects (e.g., avatar hands), rather than the physical movements of devices (e.g., controllers). As such, the motion data remains consistent irrespective of the attacker's or victim's physical setup (e.g., VR device, WiFi router), with the setup only affecting the data's quality (e.g., refresh rate). In Section 3.5.1, we have shown that our attack is robust against degraded motion data quality.

Defenses. To the best of our knowledge, no current defense mechanism directly addresses our attack, and common data protection defenses struggle to mitigate our attack for the following reasons:

(1) Encrypting network traffic is not enough: as discussed by Trimananda et al.[TLC22], if an adversary controls the client application, they can intercept packets during the key exchange, thereby obtaining the encryption key. This enables the adversary to decrypt subsequent traffic to their client and access motion data, facilitating the described attack. Interestingly, among the applications we examined, only VRChat encrypts the user's motion data within the packets. This highlights a general oversight by multi-user VR applications with respect to the potential privacy breaches related to motion data.

(2) Differential privacy comes with utility trade-offs: by introducing noise to all motions, differential privacy can decrease attack accuracy and mitigate potential motion-related privacy breaches. However, as highlighted by Nair et al.[NGS22], incorporating differential privacy to

protect the motion data can lead to utility drops. That is, it can lead to altered and inaccurate avatar movements, and can potentially reduce the quality of the immersive experience in VR applications.

We propose a defense to mitigate this attack: full blockage of motion updates during typing activities. That is, a user's motion data should not be sent to remote clients when the user is typing, and perhaps an idle or random animation can be sent instead. While this solution may sound trivial, it requires efforts from developers, game engines, and VR systems alike.

For typing activities using an application's built-in typing functionalities, the burden of defense is now put on every developer to identify scenarios in which typing happens and fix them, which can be error-prone. Therefore, it may also be necessary for game engines to provide a standard API and defense mechanism for the typing functionality.

However, as previously mentioned, just protecting motion data for an application's built-in typing functionalities is insufficient because of the wide support of system overlays like Steam Overlay [ste23]. As the application is still active, motion data is still updated when the overlay is launched, and typing activities inside the overlay are also vulnerable to our attack. In this case, the application cannot know whether the users are typing and when to block motion updates. Consequently, a standardized system-level API for typing detection also needs to enable communication between the system and applications.

Adoption of Defense. As of August 2024, SteamVR and Rec Room have created patches to defend against our proposed attack, following our suggestions of the aforementioned defense. Specifically, in SteamVR Beta 2.7.1 [ste24b], SteamVR restricts VR applications from accessing hand motion data when the Steam keyboard is opened in Steam Overlay, so that the applications cannot inadvertently synchronize the typing motion data with remote users. Similarly, in Rec Room's "My Little Template" Edition update [rec24], Rec Room stops synchronizing a user's hand motion data with remote users when the user is typing in a sensitive field (e.g., password).

Limitation and Future Work. At this stage, our method predominantly targets a common input approach: using controllers to select inputs from virtual keyboards, which highlights vulnerabilities in motion data transmission. While alternative input methods exist, such as using one's hand to tap keys on a virtual keyboard, these mechanisms still rely on motion data for input. Therefore, the

foundational concept of our attack remains valid. Thus, our technique should also be effective in extracting typed secrets from these variations.

Also, our work primarily showcases that keylogging attacks can be performed with motion data extracted from network traffic. However, the use of motion data is not limited to updating remote avatars; it also plays a role in various stages of a VR system, such as rendering, haptic feedback, and video recording. We will leave this as future work to further explore other remote channels available for keylogging attacks.

As highlighted by Nair et al. [NRO23], motion data can lead to other forms of privacy breaches, such as user identification, anthropometric measurements (e.g., height and wingspan), and even demographic details (e.g., age and gender). It would be valuable to conduct a separate study in the future, exploring how to remotely exploit these vulnerabilities to deduce such information from VR players. Moreover, we primarily focused on utilizing motion data transmission as a side channel to perform a keylogging attack.

Ethics Statement. This project demonstrates a practical keylogging attack that utilizes an inherent issue of VR multiplayer applications. We have followed a responsible disclosure process to notify affected parties and to mitigate potential harm. Also, we believe that our work serves as an important warning to developers and we hope that it will inspire the design of better defenses and secure future applications.

3.8 Chapter Summary

This chapter demonstrated that the kinematic data broadcast by multi-user VR applications for avatar synchronization constitutes a practical and exploitable side channel. Without requiring local malware or physical proximity, an adversary sharing a virtual space with a victim can recover typed keystrokes with top-1 accuracy of 97.62% on Rec Room, and comparable accuracy across three additional VR platforms. The attack is robust to real-world conditions including crowded environments and non-line-of-sight scenarios. Following responsible disclosure, Rec Room and SteamVR both acknowledged the vulnerability, awarded a bug bounty, and deployed countermeasures.

This result establishes a foundational observation for the rest of this dissertation: the core data streams of immersive interaction are not just sensitive in the abstract—they are concretely exploitable by a remote adversary. Yet despite the systemic nature of this threat (every multi-user VR application uses kinematic synchronization), such vulnerabilities are routinely missed during development. Chapter 4 investigates why, turning from the what of VR vulnerabilities to the why—examining the cognitive biases and ecosystem-level gaps that allow these threats to persist across platforms and developers.

CHAPTER 4

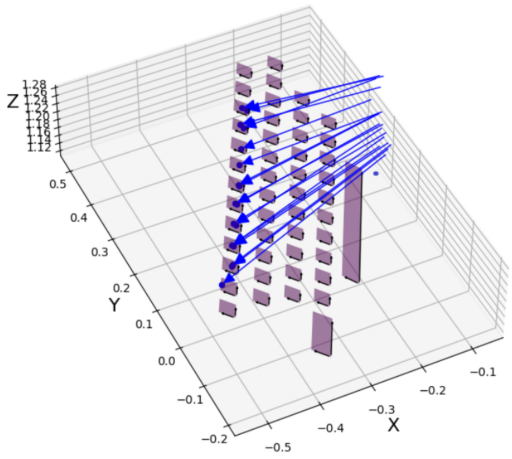
Diagnosing Causes: A Developer-centered Study of XR S&P

Chapter 3 demonstrated that a fundamental property of multi-user VR—kinematic synchronization—can be exploited as a side channel to recover sensitive user input. This vulnerability is not an implementation bug in a single application; it is rooted in how VR interaction data is collected and transmitted by design. And yet, across the industry, such threats are systematically missed.

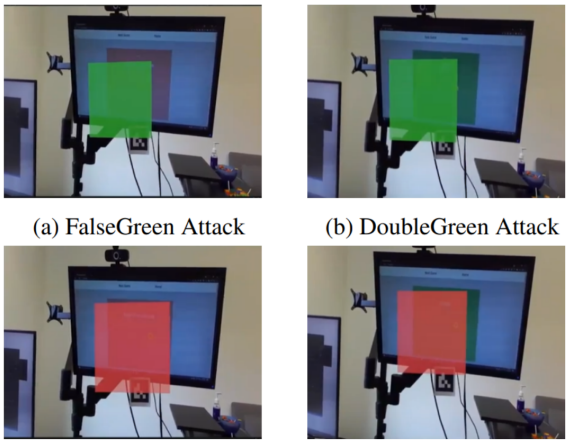
This pattern—where conceptually similar vulnerabilities appear repeatedly across different platforms—points to something deeper than individual developer oversights. The XR ecosystem has seen a growing catalog of side-channel threats, biometric leakage exploits, and interaction-layer privacy failures [AGA21, NGS22, ZSC23, LHY22], and ecosystem-scale measurement studies further document widespread privacy gaps across VR app stores [YLC25]. If each were simply a coding error, we would not expect such consistent recurrence. The more likely explanation is that systemic root causes—within how developers understand threats, how game engines expose (or conceal) data-handling logic, and how the broader XR community communicates security norms—are driving a cycle of insecurity that reactive patching cannot break.

The interaction designs and data collection channels in XR are primarily determined at the application level. Therefore, developers act as the primary architects of the spatial interaction ecosystem; their design choices directly shape the S&P posture of these applications [AFM16]. While current XR research primarily focuses on user-centric S&P concerns [KWC24, SAY25], there remains a critical lack of in-depth empirical investigation into the developers' actual practices, challenges, and cognitive blind spots. Existing studies often rely on developers' general subjective perceptions [ABB18], failing to uncover *unknown unknowns*—situations where developers cannot systematically identify or mitigate XR-specific threats introduced by immersive designs.

To bridge this gap, this chapter systematically investigates the human and ecological factors



(a) VR Keylogging Attack [SCB24]



(a) FalseGreen Attack (b) DoubleGreen Attack

(c) DoubleRed Attack (d) FalseRed Attack

(b) AR Perception Attack [CTK23]

Figure 4.1: Curated XR attack examples shown to the 23 professional XR developers during the interview study.

underlying XR S&P vulnerabilities. We conducted a comprehensive, developer-centered study involving 90-minute in-depth interviews with 23 professional XR developers. Specifically, we explore two core research questions to explain the widespread inadequacy of S&P measures in current applications:

RQ1: What are developers' perceptions of emerging S&P threats in XR?

RQ2: What are the developers' perceptions of current mitigation practices and the support from the XR community?

To investigate perspectives on threats unique to spatial computing, we curated and presented developers with a diverse set of attack scenarios sourced from recent literature (e.g., the attacks illustrated in Figure 4.1). Using a bottom-up, open-coding qualitative analysis, we examined developers' backgrounds, operational workflows, and threat models.

Our findings reveal that while developers recognize the importance of XR S&P, their efforts are severely hindered by fundamental ecosystem-level flaws: (1) systemic practice gaps, (2) profound cognitive and awareness biases, and (3) critical infrastructural deficits stemming from dominant game engines and the broader development community. For instance, we observed that many community solutions either reduce application utility (e.g., blocking raw camera feeds), rely on

outdated documentation (e.g., Meta's API docs), or require external S&P experts, making them highly impractical for average XR developers, especially those operating with limited budgets.

Ultimately, diagnosing these developer-centric challenges is an indispensable prerequisite for resolving interactive vulnerabilities within the spatial ecosystem. By synthesizing these socio-technical causes into two core problems—*awareness gaps* and *diffusion of responsibility*—we provide actionable solutions and outline structural directions to promote robust XR S&P. This analysis lays the essential groundwork for the architectural reconstruction proposed in Chapter 5.

To the best of our knowledge, we are the first to conduct a developer-centered study exploring developers' perceptions (e.g., awareness, misconceptions, attitudes, and capabilities) regarding diverse emerging threats in XR. The core contributions of this chapter are summarized as follows:

- **Empirical Investigation of XR Developers:** We conducted 90-minute in-depth interviews with 23 professional XR developers to identify contexts in which S&P threats are considered important or amplified during XR development (e.g., the amplification of sensitive data through immersive designs, the collection of rich sensor data, and unmoderated XR social experiences).
- **Identification of Ecosystem-Level Flaws:** We systematically uncovered the cognitive and infrastructural factors that make it difficult for XR developers to recognize potential S&P threats, highlighting the tension between S&P and the user-experience-driven nature of immersive environments.
- **Actionable S&P Roadmaps:** We identified critical gaps in developers' knowledge regarding existing mitigations and proposed stakeholder-aware improvement directions from both developers' and the broader community's perspectives. These findings highlight the urgent need for enhanced support in XR S&P strategies, technologies, and communication.

4.1 Methods

4.1.1 Recruitment

We recruited 23 professional XR developers with experience in developing and distributing XR applications. Participants were recruited by searching on LinkedIn using the keywords of *XR/VR/AR developers*. We assessed their qualifications by reviewing their LinkedIn profiles and personal CVs, and directly contacted those who met our criteria. Participants were required to have published at least one complete XR-specific project on a public distribution platform (e.g., Meta Quest) and to be currently employed as XR developers. Additionally, we confirmed their XR experience by asking about their past projects through an invitation email or message. Once their experience were verified, we proceeded by inviting them for a Zoom interview. Of the 412 qualified developers we contacted (from June 2024 to June 2025), 34 participants signed up, and 23 ultimately participated in the study. The decision to finalize the number at 23 was based on reaching saturation of themes, where no new themes emerged from additional interviews. All 23 participants completed the main study (approximately 90 minutes) and were compensated $70 USD.

Participant Demographics. Among our participants, eight identified as female, fourteen as male, and one preferred not to disclose their gender. The group was relatively young, with eight participants aged 18-25, eleven aged 25-35, three aged 35-45, and one aged 45-55. Despite randomly inviting all qualified participants we found online, most were young males, reflecting the gender and age makeup of the XR developer community, as observed in similar studies [KMR19, ABB18]. Table 4.1 shows the information on our diverse participant pool. Specifically, our participants have experience developing 15 types of XR applications across different domains. Moreover, our participants have published applications on 12 different platforms. Among all of our participants, only two participants have less than 1 year of experience, six have 1-2 years of experience, eight have 2-5 years of experience, six have 5-10 years of experience, and one has more than 10 years of experience. Notably, two participants have contributed to XR applications with over one million downloads.

Table 4.1: **Participant Overview** – Years of experience in XR development, number of XR applications developed, the categories and styles of developed applications, and the platforms where these applications were published and accessed.

ID	XR Development YOE	# of Apps	App Categories	App Style	Publish Channel
P1	0-1 Years	1	Single player	Education	Meta
P2	2-5 Years	2	Single player	Design, Action, Puzzle, Fitness	Meta
P3	1-2 Years	4	Single & Multi-player	Design, Education	itch.io
P4	2-5 Years	12	Single & Multi-player	Social, Action, Role-playing	Apple, Meta
P5	1-2 Years	4	Single & Multi-player	Design, Action, Role-playing, Puzzle	Steam, itch.io
P6	1-2 Years	3	Single player	Retail, Sport	Snap
P7	5-10 Years	2	Single & Multi-player	Social, Music	Steam, Meta
P8	5-10 Years	5	Single & Multi-player	Social, Event	WebXR
P9	5-10 Years	7	Single & Multi-player	Education, Healthcare, Action, Sport	Meta, Steam, Pico, WebXR
P10	>10 Years	15	Single & Multi-player	Social, Education, Retail, Design, Action, Sport	Meta, Apple
P11	2-5 Years	4	Single & Multi-player	Design, Action, Puzzle	Spark AR
P12	2-5 Years	5	Single player	Design, Action, Puzzle	Meta, Apple, Google Play
P13	5-10 Years	>40	Single & Multi-player	Social, Education, Healthcare, Retail	Meta, Directly to business
P14	1-2 Years	3	Single & Multi-player	Social, Design, Action	Meta
P15	2-5 Years	8	Single & Multi-player	Education, Healthcare, Retail, Action	Steam, itch.io
P16	2-5 Years	2	Single player	Education, Healthcare, Relax	Meta, Steam, Pico, itch.io
P17	0-1 Years	8	Single &Multi-player	Social, Museum, Other	itch.io
P18	5-10 Years	29	Single & Multi-player	Social, Education, Healthcare, Design, Puzzle	Steam, Apple, WebXR
P19	5-10 Years	2	Single player	Education, Design, Social	Meta, Apple
P20	1-2 Years	3	Single player	Retail, Design	Meta, Apple
P21	2-5 Years	6	Single player	Education, Social AR Filter, Other	Snap, YouTube, Instagram, Meta
P22	1-2 Years	3	Single & Multi-player	Education, Social	WebXR
P23	2-5 Years	20	Single player	Education, Design, Action	Meta

4.1.2 Study Design

After recruitment, the study included a pre-study survey and an interview phase. The pre-study survey collected participants' demographic information, development experience, and application

publishing channels. The survey was completed within three days prior to each scheduled interview.

During the main study session, we conducted semi-structured interviews with each participant to gather information on their development practices, perceptions of the threats we demonstrated, and opinions on the XR industry's security and privacy (S&P). We assured participants of anonymity and emphasized that our study aimed to address issues affecting the entire developer community, not to test individuals.

This study was approved by the Institutional Review Board. At the beginning of each study session, the interviewer briefed the participants on the study goals and procedures and then asked them to sign a consent form. The semi-structured interview contains the following four modules:

Background Questions. The interviewer asked participants about their XR development background, including training, involvement in S&P design, experiences with S&P issues, and perspectives on community practices. To elicit unique insights, the interviewer tailored prompts using participants' pre-study survey responses (e.g., application details). Both prepared and impromptu follow-up questions were used to explore topics such as their role in S&P-related decisions.

Threats in XR. The second part of the interview aimed to understand participants' perceptions of S&P-related threats in XR, including sensitive data, which we extended based on [GNS23] (Table 4.3), data leakage channels (Table 4.2), and potential attacks.

Table 4.2: Data leakage channels in XR collected from the literature and prior discussions [SCB24, ZSC23, Gia24].

No.	Leakage Channel	Example
1	App designed functions	Motion data for rendering avatar
2	Hardware side-channels	Rendering side-channels
3	Third-party services	Analysis APIs
4	Insecure operations	Unsafe storage, unencrypted data

As detailed in Table 4.4, we reviewed notable and representative attacks documented in existing

XR S&P literature from top venues, classifying them as XR-related (e.g., perception manipulation) or XR-amplified (e.g., social attacks). Building on the definitions proposed in [DAR23] and [Gia24], we organized these threats into categories for clarity and demonstration convenience. Initially, we categorized them based on primary attack purposes outlined in [DAR23], then refined the categorization by incorporating variations in attack surfaces related to XR design choices and use cases.

We then presented these threats (e.g., perception, social attacks in XR) to participants in two different orders (i.e., ascending and descending) to gather their feedback. Due to time constraints, the demonstration included all threat categories but did not exhaustively cover every XR S&P threat identified in the literature. However, as our aim was to explore developers' perceptions, we believe the selected examples were sufficient and broadly representative for XR threats. After the demonstration, the interviewer asked participants to reflect on the threats regarding their practicality and importance. The interviewer followed up with questions to assess their comprehension of the content and ask for potential mitigation.

Mitigation and Best Practices. The third part of the interview focused on understanding participants' awareness of existing mitigation strategies in XR and their perspectives on the responsibilities of various stakeholders (e.g., policymakers, users, platform providers) in the XR community. We compiled tools and practices from existing literature into a slide deck as provided in Table 4.5, asked developers about their awareness of these mitigations, and presented explanations and examples. We then asked participants to use the tools to address the threats mentioned previously and assess their understanding.

4.1.3 Qualitative Analysis

In alignment with our two research questions, we conducted a qualitative analysis of interview transcripts and screen recordings using a bottom-up open coding approach facilitated by MAXQDA. Following the best practices by Saldana [Sal21], our analysis involved two rounds of coding to ensure a thorough examination and identification of key themes.

Three researchers independently analyzed four interviews in the first round of coding to develop

Table 4.3: Sensitive data categories in XR summarized from the literature [GNS23, PS23].

No.	Data Type	Sensitive Information Examples
1	Motion-related data (e.g., motion tracking/gesture) [NRO23, NGM23, WSZ23b]	Identity, age, gender, physical state, biometric information, etc.
2	Voice data (e.g., microphone data) [LNF24, FMN23, CML25]	Speech content, gender, biometric, etc.
3	Camera data (e.g., surrounding environment) [GNS23, CBA24]	Environment, personal information, identity, other person identity, etc.
4	Device related data (e.g., network/HMD) [AGA21, GNS23, TLC22]	Wealth, network, IP address, etc.
5	Spatial data (e.g., location/play area) [GNS23]	Wealth, identity, address, etc.
6	Health related data (e.g., heart rate) [PS23, ZYM23, YMW25]	Cognition, emotion, health, etc.
7	Biometric (e.g., eye tracking/gait) [ND24, JDT23, WZS24]	Attention, identity, authentication related information, etc.
8	Observation (e.g., avatar/behavior) [SCB24, YSH23]	Age, gender, preference, health, emotion, etc.
9	Feedback (e.g., haptics) [GTM22, PS23]	Virtual behavior, users state, etc.

an initial codebook through repeated transcript readings and iterative discussions. Daily meetings were held to confirm, refine, and merge codes, resulting in a preliminary codebook comprising 124 codes. Building on this foundation, the researchers engaged in axial coding analysis. This phase involved merging similar codes and organizing them into overarching themes, providing a structured framework to address the research questions.

Following the initial coding process, the task of coding the remaining data was undertaken by three researchers using the agreed-upon codebook. Each data sample was then coded by a primary coder and verified by another researcher. Any new codes emerging from this coding process were discussed among all three researchers and incorporated into the codebook upon reaching consensus. Following the recommendations of McDonald et al. [MSF19], inter-rater reliability was not calculated for the coding process as the primary objective was to identify emergent themes rather than to seek coder agreement. This approach allowed the researchers to remain open to novel insights and patterns within the data.

Table 4.4: XR security and privacy attacks summarized from the literature.

No.	Attack Category	Attack Targets	Explanation
1	Shoulder Surfing Attacks	Spy XR user [GSW23], Spy on bystanders [DDK14], etc.	Recording without other people's consent
2	Software Side-channel Attacks	Infer keystroke [SCB24, YSH23], re-identify user [NGM23], etc.	Exploiting functionality data as side-channel
3	Input Attacks	Compromise input integrity [LKR16, Das24, Gia24, SW20], DoS [CAL23, El 24], etc.	Manipulating inputs to trigger dangerous operation
4	Social Attacks	Harass another user [BEE19a, FZM22, SFL23, ZXW23], social engineering [JF20], etc.	Utilizing virtual social experience as attack channels
5	Content Attacks	Overlay malicious content [CBL24, SZS24], impersonate others [TAM23], etc.	Introducing malicious virtual content in XR
6	Perception Attacks	Manipulate perception [CBY19, CTK23, TBM22], manipulate memory [BTM23] etc.	Manipulating perception and memory
7	Physiology Attacks	Introduce disorientation [CBY19], introduce content dizziness [FHF18], etc.	Introducing motion sickness and discomfort

The final codebook consists of 81 codes grouped into 18 themes.

Labeling Criteria. During our study, we analyzed the ratings of user responses to understand the relationship between developers' awareness and their threat perceptions. The primary labeling was performed by an author with expertise in XR S&P. To mitigate potential bias and ensure consistency, each response was independently coded by a second researcher, selected from a pool of three co-authors on the paper. The second coder then discussed the labeling decisions with the XR expert to reach an agreement. This approach reduced the risk of prejudice toward individual participants and improved the reliability of the labeling process. The interrater reliability scores are as follows: (1) awareness of attacks, we obtained a Cohen's Kappa of 0.812; (2) for awareness of mitigation, the Cohen's Kappa is 0.920; (3) and for mitigation quality, the quadratic weighted Kappa is 0.827. These scores indicate a high level of consistency among coders [McH12].

- Awareness of attacks: We classified participants as "Aware" of an attack if they mentioned it without our prompt, stated that they had considered such an attack before, or provided additional

Table 4.5: Mitigation tools and best practices for XR security and privacy.

No.	Approach	Examples	Applicable to
1	User notification & education	Privacy policy [GDL24, TLC22], in-app notification [ZSK18], etc.	Content attacks, perception attacks, data leakage, etc.
2	Static analysis	Ghidra [NSA24, GDL24], Roslyn [Mik24], etc.	Software side-channels, content attacks, data leakage, etc.
3	Runtime analysis	Moderation [SCT23], packet analysis [LTS23, TLC22], etc.	Social attacks, physiology attacks, data leakage, etc.
4	OS and game engine enforcement	Arya [LRK18], OS-level data control [Met24a], etc.	Content attacks, input attacks, perception attacks, etc.
5	Legislation and policymaking	Laws [Cal18, Eur16], best practices in XR [XRS23, Met24b], etc.	Shoulder-surfing attacks, social attacks, content attacks, etc.
6	Input and output processing	Input or output checking [RK21, LKR16], differential privacy [NRO23], etc.	Software side-channel attacks, shoulder-surfing attacks, etc.

examples of similar attacks.

- Awareness of mitigations: Participants were marked "Unaware" of a defense mechanism if they mentioned being unaware of it or misused it for solving demonstrated attacks.
- Quality of developer-proposed mitigation: The researcher rated the mitigation strategies proposed by developers on a *scale of three*, where *one* indicates strategies that do not make much sense, *two* indicates strategies that make some sense but are missing important details, and *three* indicates strategies that are well-thought-out or align with realistic solutions adopted in the industry.

Threat Model. Our threat model focuses on (1) external adversaries (e.g., network/system intruders, malicious XR users, bystanders, or third-party API providers capable of generating harmful content)

and (2) careless developers who may inadvertently introduce attack vectors or harmful content to XR applications. We do not consider malicious developers as we assume study participants do not have malicious intent.

During the presentation of each attack, we further discuss the specific attack channels and attacker capabilities. For example: (1) Many XR platforms and applications (e.g., VRChat) allow user-generated content (e.g., custom rooms or game objects), which—if not properly validated—can enable content, perception, or physiology attacks. (2) Malicious third-party API providers can introduce input, content, or software side-channel attacks when their services are integrated without proper vetting. (3) External adversaries may exploit hardware vulnerabilities to manipulate sensor inputs or engage in malicious activities within social XR applications, leading to shoulder-surfing, input-based, or social attacks.

4.2 Developers' Perceptions on Privacy in XR

XR technologies typically involve a wide range of user data, which introduces novel privacy threats. This section explores developers' perceptions of these threats, particularly those arising from the unique characteristics of XR user data. In our pre-study survey, developers rated privacy in XR as highly critical, with a median Likert score of 6/7; only three developers rated it 4 (moderately critical) or lower. With this in mind, we further investigate our research question on developers' perceptions of privacy in XR from three perspectives, following the demonstration described in Section 4.1.2:

- Developers' awareness of sensitive data in XR
- Developers' perceptions of sensitive data in XR on:
  - What makes data overall more sensitive?
  - What makes certain data perceived as less sensitive?
- Developers' perceptions of leakage channels in XR on:
  - What makes leakage channels overall more realistic?

– What makes leakage channels perceived as less realistic?

Additionally, we also analyze potential developer misconceptions at the end of this section. Understanding these perceptions is essential, as their awareness, misunderstanding, and capability directly influence their practices, application designs, and protection mechanisms in XR [AFM16, HDG24].

4.2.1 Developers have limited awareness of sensitive data in XR

When asked about the sensitive data types used or collected in their applications——prior to being introduced to the definitions of sensitive data types and sensitive information in XR (Table 4.3)——developers, on average, identified only *2.1* types of XR-sensitive data type (e.g., motion, biometric) and *0.5* types of non-XR data type (e.g., email, password) as being collected in their applications. This initial finding raises concerns about whether developers' applications indeed collect only this limited set of data types or if developers fail to recognize certain types of data as sensitive.

Developers come to realize that their applications collect more types of sensitive data than they had previously thought.. After being presented with XR-sensitive data definition, developers identified significantly more XR-sensitive data types, recognizing an average of *4.8* types collected in their applications and acknowledging protection needs. P3 reflected: *"It makes you realize that some data that seems pretty harmless is actually a lot more potent,"* highlighting potential risks from developers' lack of awareness.

4.2.2 What makes data overall more sensitive in XR

After being presented with various types of sensitive data (Table 4.3) and their implications, developers were asked to rate the sensitivity of each data type. From their ratings, we found a median Likert score of 5.0/7.0. We analyzed their reasoning and identified the key factors as follows.

Immersive interaction reduces the awareness of data leakage.. P4, P12 and P23 mentioned that the immersive interaction design in XR makes voice data more prone to leakage and increases its sensitivity: *"When you talk to people in VR, it brings back the same feeling that you're talking*

to people in real life, so often people don't have the expectation that they are being recorded." Additionally, P7 and P8 raised concerns about unintentional self-disclosure in XR applications due to non-intuitive interaction design. Informed by their prior experiences in XR, P8 noted that this could increase the sensitivity of voice and expressed an intention to incorporate clearer notification mechanisms in his future applications as mitigation: *"I did not realize that they could hear me. There was no point at which I was told or made aware of, or had visual feedback on the fact that I had an open microphone."*

Advanced tracking sensors produce sensitive data with greater details.. P7, P8, P13-16, and P19-22 highlighted that data collected in XR contain rich sensitive information, which allows adversaries to infer user information. P13, a healthcare XR developer, emphasized that XR technology amplifies the sensitivity of data leakage in healthcare applications: *"Because this [XR] device involves a lot of activity and emotion expressing capability. Using the controller like with the phone, you just use the vibration or some haptic features to identify things, but XR can track my hand motion, and in future it can comes with some sensor for your legs, or even if not, the sensors available in the excess can still detect your leg movement and the energy and the pace you move."* Similarly, P13 highlighted that XR camera data is more sensitive than data from other devices as it provides more detailed information and is physically attached to the users' direct view. Considering the sensitivity of the camera data, P8 mentioned that they would only collect this data for valid reasons and properly protect (e.g., via encryption) in development.

The value of XR data incentivizes extensive collection and surveillance.. P7, P13, P14, and P19-21 further noted that the economic value of the sensitive information inferred from the data types collected in XR can drive XR headsets, applications, and service providers to collect user data (e.g., camera and spatial data.), significantly increasing the privacy sensitivity of these data. P13 noted that scanning a room for guardian system with an XR headset can leak data to the platform:*"Where is my room? Where is my couch? Where is my table? Where is my TV? So this data definitely has a business value. Because, let's say most of the ADs now understand what we talk and what I'm interested in, and based on that, they will publish those ADs, right? ... at the same time, certain people are not comfortable with sharing that information."* These concerns align with findings in [TLC22, GDP25], which show that data is already being collected for profiling, analytics,

and personal advertisements in XR. More critically, P7, P8, P16, and P21 expressed concerns that headset providers or governments could exploit camera and biometric data for surveillance purposes since these data provide detailed information about XR users and their private space (e.g., home).

Opaque XR infrastructures impede effective mitigation.. P4, P9, and P13 highlighted that XR devices are increasingly being used for medical purposes, but the black-box XR systems present challenges related to data protection and compliance, thereby increasing the sensitivity of medical data. P9 mentioned the challenges he faced when shipping his applications with devices as a medical service, particularly in meeting compliance requirements from government agencies (e.g., VA) which require full visualization on collected data in devices [USnd]: *"The problems we have with the headsets are that it is bytedance, or it's Meta, HTC. There's no open source. But on the same day, you know, it's like we're following the API callback to mainland China, ... We're part of the VA, the US government doesn't want that."*

4.2.3 What makes certain data less sensitive

While developers generally perceive data as more sensitive in XR, they still rate some data types as less sensitive (score < 4). we analyzed their reasoning and identified the key factors:

The perceived benefits tend to overshadow the risks of privacy leakage.. P4, P7, and P17 described a trade-off, classifying functionally essential XR data types as less sensitive and prioritizing technical performance over privacy concerns. P4 stated: *"I think it's sensitive, but it's required for the application to work. It's hard to make a gesture-based game if you can't get the gesture."* Similarly, the importance of motion data for enabling immersive experiences contributes to P7's perception of it as less sensitive: *"If you want to exist in the same place as somebody else. You're gonna need to know what their hands are doing with their bodies, and people voluntarily add extra trackers to their bodies to have the avatar be more realistic."* This contradicts the finding in [NRO23], where the connection between motion data and crucial XR operations increases the potential to infer important information.

The insufficient knowledge of developers tends to obscure the consequences of privacy leakage.. P1, P2, P10, P13, P15, P16, P18, P20, and P21 stated that they do not consider certain data types

(e.g., motion, feedback) to be highly sensitive even after our demonstration, as they could not identify plausible risks of sensitive information leakage based on their experience. For example, P7 mentioned: *"I don't think it's super sensitive. I think most kinds of data that you're gonna get from that, it's like, Oh, this person whacked their controller into a wall. And so there's a wall there."* Moreover, P4, P10, P13, and P16 admitted that they were not aware and had not previously considered the potential consequences of using or collecting feedback data (e.g., haptics). Therefore, they perceive feedback data as less sensitive as P13 said: *"So this is a very niche and new thing, which we don't know, because the applicant is completely new, and people are still figuring out how to use. It's hard for me to say how much threat from my experience."*

4.2.4 What makes leakage channels more realistic in XR

After presenting potential sensitive data types and possible data leakage channels, we collected developer feedback on which channels are more likely in XR. We identified high realism ratings with a median Likert score of 6.0/7.0. Analysis of their reasoning identified the key factors as follows:

Prevalence of development misoperations due to technological immaturity.. P4, P6, P7, P9, P10, P12, P13, P16, and P18-22 reported, based on their own experiences or those of XR developers they worked with, developers tend to blindly rely on third-party packages (e.g., Photon [Gamb]) when developing XR applications. For example, P12 noted: *"It's a fact that people use APIs all the time without really knowing what's under it ... If you're using Unity to build anything, you're gonna use a bunch of packages."* P7, P9, P10, P12, P13, P15, and P16 also noted that insecure operations, as a data leakage channel, are likely to occur in XR. P13 believes the immaturity of XR technology and market makes this trend especially concerning: *"When it comes to the AR/VR set of things. It's still a maturing technology where a lot of freelancers, small agencies, people from all sizes of teams, and backgrounds are coming in. So, as I mentioned, no matter what the protocols we bring in, at the end of the day. It's a responsibility of the developing team."* These threats stem from the current landscape of the XR industry, which is largely composed of individual developers and small enterprises [XR425]. This context suggests that developers should be careful when using third-party packages to avoid potential data breach.

Ease of self-disclosure by users in an immersive environment. P1, P4, P5, P7, P9, P13, P14, P17 and P21 expressed concerns that beyond intentional information sharing, XR's immersive nature can more easily lead to the unintentional disclosure of private information (e.g., behavior patterns, voice). For example, P17 noted that users may accidentally leak private data and emphasized that developers should take responsibility for helping to protect them: *"I am a firm believer that users are stupid. So if you, if you have something, chances are they will screw it up. Doesn't mean it's intended. But there's always something right?"* P14 emphasized that such misuse is more likely to occur in XR due to its immersive interaction, which tends to encourage self-disclosure, aligning with the findings in [SMF22, MZF20]: *"In VRChat. I think a lot of times we are interacting with friends in a very natural way. I wouldn't pay extra attention to how I behave. So that data could be very suggestive of who I really am."*

4.2.5 What makes certain leakage channels less realistic in XR

While developers generally agree with the presented leakage channels as realistic, we identified cases where they give a lower score (score < 4). To better understand the reasoning behind these ratings, we classified developers' perception of leakage channels being less realistic due to:

The responsibility is not acknowledged in XR development.. P3, P6, P7, P9, P10, P13, P15, P16, and P21-23 mentioned that they were unfamiliar with hardware-side channels for data leakage. For example, P3 stated: *"I am not too knowledgeable about this one, so I'm giving a lower score."* Additionally, P16 believed that developers should not be responsible for it: *"I think that is possible. But I'm not really concerned about that. And also as a developer, I can't really do much about that."* Similarly, P7, P15 and P17 suggested that this issue should be addressed by XR headset providers and governments.

The belief that local XR applications are inherently secure.. When asked about the necessity of implementing protections (e.g., encryption or secure data storage) to prevent XR data leakage in their applications for the types of data demonstrated in the study, P1-3, P6, P7, P10-12, P16 and P17 believed that data handled by local XR applications (i.e., those without network functionality or limited to single-player use) would be inherently secure, and therefore would not require additional

security measures. For example, P11 commented: *"I think it's just fine because it's a local APP. So it only processes those data and uses it, but doesn't actually send it to anything."*

4.2.6 Analysis: Potential misconceptions on privacy in XR

Based on the developers' perceptions above, we analyze and summarize the following misunderstandings about XR privacy:

Misconceptions regarding data sensitivity.. From Section 4.2.3, we observed two common misconceptions: (1) the belief that the data with higher utility is less sensitive, and (2) the assumption that unfamiliar data is also less sensitive. However, according to the GDPR [Eur16] and widely accepted research guidelines [Off25b, Off25a, Sta25a], any data that can be used to reveal an individual's identity or personal information (e.g., financial or health data) should be classified as sensitive and protected using appropriate mechanisms (e.g., [NGS22]).

Misconception on the data leakage channels.. FromSection 4.2.5, we observed the following misconceptions: (1) Misconceptions of assuming that mitigating data-leakage channels (e.g., hardware side channels) is outside their scope. For example, developers may believe this responsibility lies with hardware teams or other stakeholders. However, prior work emphasizes that software developers should play an active role in preventing leakage, even when it originates in hardware. Developers are encouraged to participate in co-design efforts and implement software-level protections such as isolation [DCA20, YAI24, DCV23]. (2) The misconception that local XR applications are inherently secure. While local applications may offer more security than online apps, they remain vulnerable to leakages. For example, through unsafe Android API calls, side channels (e.g., rendering and motion), and insecure local storage [GDL24].

4.3 Developers' Perceptions on Security in XR

In addition to privacy threats, In this section, we explore developers' perceptions and understanding of threats happening in XR. Our pre-study survey reveals that developers generally perceive security in XR as a critical issue, with a median Likert score of 5/7. Among the 23 developers surveyed,

only four rated XR security <= 4 (Moderately Critical). Building on these findings, we further investigate our research question on developers' perceptions of security in XR from three aspects, following the demonstration in Section 4.1.2:

- Developers' awareness of security attacks in XR
- Developers' perceptions of attack importance regarding:
  - What makes attacks in XR overall more important?
  - What makes certain attacks perceived as less important?
- Developers' perceptions of attack practicability:
  - What makes attacks in XR overall more practical?
  - What makes certain attacks perceived as less practical?

Similar to the previous section, we analyze these responses and highlight potential misconceptions held by developers. We focus on these questions because developers' awareness, misconceptions, and capabilities directly influence the security and mitigation they adopt in XR applications [AFM16, HDG24].

4.3.1 Developers exhibited limited awareness of XR attacks

Without our demonstration of the categorized XR attacks, developers recalled an average of only *0.9/7.0* XR-specific attack types listed in Table 4.4. Furthermore, 6/23 developers reported being unaware of any XR-specific attacks. These findings suggest that, without reminders, developers may struggle to adequately consider potential XR-specific threats during the design and development of their applications. To better understand this gap, we further investigate whether the lack of awareness stems from developers never having encountered these threats or simply needing a prompt to recall them.

Developers had never heard of many attacks prior to our demonstration.. Based on developers' feedback and our analysis (as described in Section 4.1.3), developers had not heard of *3.7/7.0* attack

categories prior to our demonstration. Moreover, only 10/23 developers were introduced to more than half of our demonstrated attacks. Nevertheless, after the demonstration, developers expressed that they gained new insights into important attacks they did not consider but would be helpful for their future development. As P9 remarked regarding the software side-channel attacks used to infer keystrokes [SCB24], *"I've built keyboards in VR, and I will think about the possibility of these attacks now."*

Developers' attack awareness undermines their perceptions of attack importance and practicability.. Developers assigned lower scores to attacks that they did not know before. To investigate how awareness influences perceptions, we categorized ratings by attack awareness and applied Mann-Whitney U-tests to assess significant differences. Our analysis revealed a significant difference ($U = 4245.5$, $z = -3.458$, $p < 0.001$) between developers' ratings and awareness of attacks. Developers rated known attacks higher median importance score of 7.0/7.0 versus 6.0/7.0 for attacks they were unaware of before. Similar to attack importance, for attack perceived practicability, developers gave significantly higher ratings to the attacks they were aware of ($U = 5032.5$, $z = -6.125$, $p < .001$), with a median score of 7.0/7.0 compared to 5.0/7.0 for those they were unaware of before.

4.3.2 What makes attacks overall more important in XR

After being presented with various attack types (Table 4.4) and their implications, developers rated attack importance with a median score of 6.0/7.0. We analyze their reasoning and identify the key factors as follows:

Immersive experiences are considered highly personal.. P2, P8-10, P12, P14, and P16-22 noted that attacks like social attacks are particularly important because XR experiences are highly personal. P12 stated: *"It's a completely different feeling when people are in your personal space, you're using voice and you can hear them, you can talk to them, you can see their hands. It's very different than a social media app that you can ignore on a screen ... I think the damage can definitely be a lot more than other media, like I had a kid that came up to me and just sneezed in my face, and I felt so disgusted, I know, like it's virtual, and it was like: dude. Get away from me."* P12, P14, and P22

emphasized that social attacks, from these previous negative experiences, can be highly disruptive and should be mitigated in both their application and other multiplayer XR applications. As P22 noted: *"Especially social XR platforms like VRChat, at least consider it as part of the game's design."*

XR functionalities may negatively impact the large user population.. P6, P10, P19, P22 and P23 identified attacks as important due to potential large-scale sensitive information leakage from the community's perspective. As P6 stated about the shoulder-surfing attack using XR devices: *"This sounds like a very big issue. Because you know AR glasses are coming soon right? By a bunch of companies, or you know what, a lot of them have already been around. I think mass adoption is going to be coming within the next couple of years."* Similarly, without hesitation, P4, P6, P7, P9-11, P16, and P21 believed social attacks are the most important attacks in XR because they affect a large number of users. To avoid similar threats happening in their applications, developers stated that they would actively consider more of these attacks during implementation. For example, P20 mentioned their efforts to reduce the risk of social, physiological, and perceptual attacks:*"That's why we're always trying to consider that in our design. And we're gonna do a lot like testings internally before we bring it to test down users."* Moreover, P8, P15, P16 and P18 mentioned that developers are worried that negative experiences caused by these attacks could drive users away.

XR applications are beginning to carry out higher-risk activities.. P3, P4, P7, P8, P13, P14, P16, and P21-23 indicated that attacks like side-channel attacks could be used to infer sensitive user information during higher-risk activities in XR, such as password entry, private behaviors, conversations, or other personal details, making these attacks especially important to consider in XR development. P16 commented: *"Those can track a lot more physical data, and also your health data, and even the like. Your motions, signatures, or even capture, if you have certain injuries."* Similarly, P13 mentioned that this threat will become more severe in the near future as higher-risk operations, such as banking, are introduced in XR. Developers and the XR community should prioritize their focus on protecting data used in these higher-risk applications.

The immersive nature of XR makes mitigation more challenging.. P6, P16, P19, and P22 believe that attacks are more important in XR as they are harder to mitigate for developers, which aligns

with [Cas22]. P6 used social attacks as an example: *"You probably aren't able to do real-time voice detection."* Similarly, P16 commented: *"I think it is very important ... I also think it's difficult to moderate, most things like Rec Room, they're unmoderated experiences."*

4.3.3 What makes certain attacks less important in XR

While developers generally perceive attacks as important in XR, they consider some attacks less important (score < 4). We analyzed their reasoning and the key factors as follows:

Users are capable of performing mitigation.. As highlighted by P3, P7-9, and P14, physiology attacks can be perceived as less important since users can simply remove headsets or close their eyes for mitigation instead of implementing mitigation during the development process. For example, P8 mentioned:*" I think it's too easy to just take off the headset. It's not like I'm holding you hostage or anything."* Similarly, P7 mentioned that closing eyes can also prevent these attacks. However, this shifts the burden of mitigation onto users, which contradicts recommendations for preventing physiology attacks [CAL23, PRC22].

Certain attacks are perceived as applicable only within specific XR environments.. P4, P6, P7, P10, P16, P17, P20, and P21 mentioned that they consider certain attacks less important when they believe those attacks are only effective under certain circumstances. For example, P4 commented on physiology attacks: *"I think the damage is more or less minimal because it's impacting an individual user ... in most cases it's more so harmless and just a Disney effect."*

4.3.4 What makes attacks overall more practical in XR

Similar to the process of evaluating attack importance, we asked developers how practical the attacks were and identified a median Likert score of 6.0/7.0. We analyzed their reasoning and identified the key factors as follows:

XR provides stealthiness and flexibility through hardware and immersive technology.. P6, P8-10, P13, P14, P16, P19, P21 and P22 perceived attacks (e.g., shoulder-surfing attacks) as practical because of their stealthiness to perform attacks when comparing XR to other mobile devices. P13

mentioned: *"If you're using a mobile or a camera. People at least will get an awareness, right? Somebody is trying to steal something from me. Different cameras again, it's hard. But this one is like an easy option for a threat, and hard to identify, for the person who is being watched."* Moreover, P6, P8-10 and P16 mentioned that social attacks are more flexible in XR, making them more practical from a technical perspective. Based on these experiences, P16 and P19 emphasized that they want to contribute to mitigation for better user education and improved notification systems by developers and platforms.

Encouraging user-generated content in XR may enable adversaries.. P4, P9, P16 and P22 noted that since XR environments encourage users to create their own content (e.g., game rooms), the likelihood of physiology attacks increases. P4 noted: *"I think in VRChat, you can throw a flash bomb to introduce seizure...We rely on the users to like deal with this. Platform allows these kinds of behaviors."* P4 was also concerned with content attack introduced by user-created content: *"That's actually one of the things we tried to prevent when we're doing an event on Horizon, because the user can just jump on stage or show models or images that's not great."* Considering this issue, P4, P16, and P22 have mentioned the need in their applications for better access control and moderation tools to mitigate malicious user-generated content.

4.3.5 What makes certain attacks less practical in XR

While developers mostly agreed attacks were practical, some gave lower scores (< 4). We analyzed their reasoning and identified the key factors as follows:

Execution of attacks requires significant technical expertise.. P10, P12, P15, P17, and P23 mentioned that attacks, such as perception and input attacks, are technically difficult in XR, making them less practical. For example, P15 highlights perception attacks: *"You need a lot of technical know-how to be able to manipulate boundaries like that."* Similarly, P4, P7, P9, P15, and P16 noted that conducting input attacks requires technical expertise in XR or an understanding of users' states.

User-experience oriented attacks are perceived as less motivating.. P6, P8, P13, P14, P16, P17, P21, and P22 mentioned that attacks (e.g., perception attacks), seem less motivating for attackers, as they primarily target user experiences rather than direct benefits, making them impractical. As P6

commented: *"You can't really gain anything from this. Aside from harming people, basically like you can't earn money from it right?"*

4.3.6 Potential misconceptions on security in XR

Based on the developers' perceptions above, we analyze and summarize the following misconceptions about XR security:

Misconceptions on attack importance.. From Section 4.3.3, we identified the following misconceptions: (1) considered that attacks that can be mitigate by users as less important in development. While users can sometimes mitigate attacks (e.g., removing a headset during a physiology attack), developers should not shift the responsibility to users. Instead, they should implement robust defenses, especially when users may be unaware of potential threats. This view is supported by [WA18, GIS22]. (2) Misconception that the attacks only affect specific groups (e.g., physiology attacks on individuals with seizures) are less important. Developers should not overlook these risks. All users should be considered equally in security design, especially when user safety is at stake [Wan18].

Misconceptions on attack practicability.. From Section 4.3.5, we identified the following misconceptions: (1) Attacks with technical complexity are perceived as less practical. A more rigorous approach to assessing attack feasibility involves using a threat model that assumes attackers possess the necessary skills and have reasonable access to relevant information [Sho14]. (2) Developers may underestimate attacks due to a lack of appreciation for attacker motivations. Specifically, they may overlook user experience–oriented attacks, assuming they are less likely. However, as XR technology is increasingly deployed in critical domains such as the military [Ind25] and healthcare [Nur25], these types of attacks remain both feasible and attractive to adversaries. Moreover, all user deserves full protection, as emphasized in ethical development guidelines [GZK21].

4.4 Developers' perceptions on current mitigation practices and supports on XR S&P

After examining developers' perceptions of S&P threats in XR, we aim to understand whether they are capable of addressing these threats with appropriate mitigation strategies. Based on developers' responses to threats and our questions, we address our second research question by summarizing the findings that answer: (1) What are the limitations of the current mitigation practices for XR S&P threats? (2) What are the missing supports in current mitigation practices for XR S&P threats? Specifically, we present insights on these questions from the perspective of both the developers and the broader XR community (i.e., other stakeholders). This is important because we aim to identify the limitations of current mitigation practices and address existing problems. Moreover, we seek to understand how other stakeholders in community can better support developers in ensuring S&P.

4.4.1 Limitations of current practices

As a first step, we identified limitations in proposing mitigations, awareness of existing mitigation practices, and perceptions of those practices provided in XR communities.

Developers lack the ability to propose effective mitigation strategies.. After demonstrating each attack, we asked developers to propose a mitigation strategy and evaluated their responses using the criteria described in Section 4.1.3. Developers sometimes provided impractical or unclear mitigation strategies. Additionally, developers' awareness of attacks influenced their ability to propose effective mitigation strategies. We conducted a Mann-Whitney U-test to compare mitigation scores between the two groups. Developers provided significantly better mitigation for attacks they were aware of, rating familiar attacks higher ($U = 3956.5$, $z = -2.48$, $p < .01$).

Many developers are unaware of existing policies and standards that inform best practices for privacy in XR.. 12/23 developers were unaware of the GDPR [Eur16], and 14/23 were unaware of the CCPA [Cal18], both of which are relevant to S&P in XR applications. Moreover, none were aware of XR-specific standards such as Rosenberg et al. [RWP23].

Developers are unaware of existing mitigation practices for XR.. When presented with mitigation

tools for S&P in XR development, developers noted that they were unaware of some of these tools but believed these tools could be helpful in addressing S&P challenges in XR. For example, 14/23 developers reported unfamiliarity with program analysis tools (e.g., Ghidra [NSA24] and Roslyn [Mik24]) used for detecting vulnerabilities, and 14/23 were unfamiliar with existing system- or game-engine-level protection methods (e.g., Arya [LRK18]).

Existing mitigation sacrifices utility in exchange for S&P.. P3-6, P12-14, and P21 expressed concerns that existing privacy mitigation practices (e.g., restricting access to camera feeds) constrain developers' creativity and limit the potential of XR applications. P6 noted: *"They don't provide any access to the camera and information at all, It really limits what you can develop as a developer."*

Platforms need to improve the S&P checking process for application publication.. Developers highlighted that all current platforms publishing S&P policies need improvement. In our study, 14/23 participants had published applications on the Meta Quest platforms. Among them, four reported that their submissions did not undergo a detailed review of S&P, and five recommended that Meta enhance its review process by implementing more rigorous content checks and providing clearer feedback, guidelines, and documentation. Compared to Meta, fewer developers in our study (6/23) had published on the Apple platform. However, the satisfaction rate was slightly higher: only two of these six developers believed that Apple could benefit from stricter reviews or more developer guidance, while the remaining four considered Apple's policies to be adequate. Additionally, P11, P15, P19, and P22 observed that S&P requirements are largely absent on other distribution platforms (e.g., SideQuest, itch.io, and AR platforms).

Disclosure and feedback on S&P problems is not sufficient.. P1, P4-10, P13-16, P19, and P22 emphasized the importance of timely disclosure of S&P issues to ensure effective mitigation, but noted that such disclosure is largely lacking at present. To address this problem, developers called for the establishment of more testing and disclosure channels—both formal (e.g., beta testing, bug bounty programs) and informal (e.g., app reviews, user comments)—to help identify implementation and design flaws related to S&P issues. For example, P15 noted that collaboration and disclosure among developers was invaluable in his previous XR development projects at larger companies. *"Have developers that are all working on like VR or XR in general, when you publish it, and they all test*

it, and tell you what's good, what's not, and then you fix that... We need dedicated developer-focused forums, similar to Reddit, to gather feedback from experienced developers. This feedback can help you refine your application and enhance its privacy compliance." Moreover, P13 called for additional tests from users for S&P issues in their applications and help uncover previously unknown problems.*"I would like users to be open about the issues they are facing. However, I haven't gotten any feedback provided by my end users regarding the protection mechanism used in the application.*"

Regulations and best practices for XR are impractical or outdated.. P1-11, P13-16 and P21, P22 expressed frustration with existing guidelines and documentation in XR by describing them as consistently difficult to follow. They characterized these resources as overly lengthy, frequently updated, and occasionally inaccurate which is largely attributed to the current stage of the XR industry. P16 also noted that this issue stems from the suboptimal practices of industry-leading platforms: *"Meta is the biggest vendor out there, and they have so much old documentation mixed with new documentation. Oftentimes it is impossible to know what is current, and sometimes even the current stuff is not well documented."*

4.4.2 Absence of supports from the XR community

In addition to the limitations of existing solutions, developers' suggestions for mitigating threats and other suboptimal practices in XR (e.g., outdated documentations) revealed the lack of support within the XR industry. We categorized these needs into three groups demonstrated below:

4.4.2.1 Strategic Support

From the developers' response, we identify a need for strategic support (i.e., the provision of guidance or resources) as follows.

Better support for developers to address economic and utility concerns in S&P-aware XR development.. P2, P6-10, P12, P13, P15-19, and P21 highlighted that the tension between utility and security is fundamentally rooted in the economic constraints of XR development. P7 remarked:"*Security is definitely something we want to consider sooner rather than later, but it's essential to have more cost-effective solutions.*" Moreover, as previously discussed, developers called

for strategies different from existing ones (e.g., blocking the camera feed) that do not compromise utility. P2 and P11 also emphasized the need for strategic support to balance the economic needs of smaller companies with S&P in XR: *"[Companies/Policymakers] should reward developers that demonstrate greater responsibility toward the technology, to encourage them to build a compliant application."* P5, P7-9, P16, and P22 also emphasized the need for community to provide external S&P experts, which developers can not afford.

Better guidance for best S&P practices in XR.. After demonstrating the threats, P1-3, P5-13, P15, P16, and P22 emphasized the need for guidance and professional resources to address emerging attack surfaces in XR. For example, P5 and P16 highlighted the responsibility of industry leaders (e.g., Meta, Apple) to provide accessible resources such as checklists, templates, tutorials, and consulting services to support developers, particularly junior developers, in S&P practices such as privacy policy drafting. Moreover, developers found that general S&P policies (such as GDPR and CCPA) were often tedious, costly to implement. For example, P8 mentioned: *"The biggest problem I have with the GDPR is that it's so annoying that I would beg to have it go away."* Moreover, P20 mentioned that many requirements might not be suitable for XR: *"I don't made any experience that's telling the people ahead that we're tracking your motion data because that's kind of ruins the magic of the [XR] experience."* Based on this, P2, P4, P6, P8, P11, and P18-22 called for more XR-specific regulations to address the unique use cases (e.g., notification, immersive social experiences) in XR applications.

Better consideration for underrepresented groups.. All female developers (P1, P11, P18, and P19), raised concerns about insufficient support in XR for underrepresented groups. P11 pointed out that physiological attacks disproportionately affect women, since many XR testing focus primarily on male participants, and called on developers to address this imbalance. This mirrors our finding here that only female participants flagged these issues. P19 described having experienced social attacks: *"I got attacked a couple of times, especially when your avatar is a woman or a minor"* and urged stronger anti-discrimination measures and requirements in social XR experiences. Besides the worry of female developers, many developers (P6-13, P16, P17, P19, P21, and P23) emphasized that current XR platforms lack adequate protections for children, underscoring the urgent need for age-appropriate security safeguards in the XR community.

4.4.2.2 Technical Support

Developers also called for stronger technical support as follows:

Moderation tools. P2, P5, P8, P9, P11, P13, P14, P16, P17 and P19-22 requested moderation tools (e.g., vision-based) for content and user behavior management in XR applications. P13 specifically highlighted the need for integrated moderation tools to block malicious content.

I/O processing tools. P2, P6, P7, P8, P10, P11, P14, P16, and P19 emphasized the need for input/output processing tools. For instance, P6 suggested differential privacy-based automatic blurring as an output processing method for mitigating shoulder surfing attacks, while P8 highlighted that input checking, such as packet inspection can help prevent input attacks.

User education and notification systems. P1, P4, P6-10, P13, P16, P17, and P19-22 emphasized the need for improved user education and notification systems as a technique to mitigate content, social, shoulder-surfing, input, and physiology attacks. For example, P16 stressed the importance of implementing an interaction design to notify users of content attacks.

4.4.2.3 Communication Support

Developers also identified gaps in communication support as follows:

Communication for developers to understand XR S&P knowledges.. P1, P2, P4, P5, P7, P11-13, P16, and P18-22 stressed the importance of providing more threats knowledge with clear guidance on S&P in XR development. P2 highlighted: *"When I see, like XR related posts it's always about. Oh, what's the new tech like AI and stuff like that? But it's never about like security. So I think, having the community participate and educating everyone about it, and then giving suggestions."* Moreover, all developers hope researchers could share findings promptly and educate the broader community through accessible channels (e.g., online tutorials and developer summits). P1, P2, P8, P15, and P19 also mentioned the need for more communication channels dedicated to S&P. The absence of such channels may also contribute to the existing communication problems (e.g., disclosure on S&P issues).

Communication for technical transparency. (P2, P4, P5, P6, P7, P8, P9, P11, P12, P13, P14, P16

and P19). Developers expressed concerns about the lack of communication on technical transparency in the XR ecosystem, particularly regarding the black-box nature of third-party APIs (e.g., Avatar SDK, XR Tracker), which they feared could introduce S&P risks. Participants emphasized the need for clearer communication, reliability, and openness from third-party providers.

4.4.3 Analysis: Misconceptions on the supports for S&P in XR

Although most perceptions of unmet needs are valid, our analysis also identifies the following misconceptions.

Misconceptions on policies and standards.. Developers mistakenly believe that there are no XR-specific S&P policies or standards. While we agree that developing more XR-specific S&P policies is beneficial, as discussed in Section 4.4.1, there are already some initial efforts in this area. Addressing developers' lack of awareness of existing policies is also a crucial part of the problem [Eur16, Cal18, RWP23].

Misconceptions on technical support.. Developers also mistakenly believe that there are no usable tools for mitigating the proposed threats (e.g., content attacks). While we agree that additional tools are needed to address XR S&P threats, there have already been initial efforts and tools made available to developers [NGS22, NSA24, Mik24]. However, a lack of awareness among developers about these existing mitigation tools significantly limits their effectiveness.

Misconceptions on communication.. Besides, developers hold the misconception that there are no usable communication channels for XR S&P. While we agree that more communication channels and feedback mechanisms would be beneficial, there are existing platforms—such as forums [met25, nvi25, imm25] and conferences [awe25, iee25]—that already serve this purpose. Developers should proactively engage in S&P discussions through these channels, participate in bug bounty programs, and follow up with users to close the feedback loop.

4.5 Discussion

In this work, we investigate how developers perceive and reason about real-world XR S&P threats. By examining their perceptions of the sensitivity, realism, importance, and practicality of threats, we uncover gaps in development (see Section 4.2 and 4.3) and highlight key factors (see Section 4.4). Moreover, our comprehensive study reveals two common misconceptions: (1) awareness gaps regarding XR threats, and (2) diffusion of responsibility when discussing mitigation practices for XR threats. In this section, we further examine the consequences of these misconceptions, analyze their underlying causes, and propose corresponding solutions.

4.5.1 Awareness gaps due to the rapidly evolving nature of XR interaction technologies

In Sections 4.2.1, 4.3.1, and 4.4.1, we identified significant gaps in developers' awareness of emerging XR threats, existing mitigation tools, and support resources (e.g., reward programs [Met21], guidance [XRS23]). These awareness gaps lead to normalcy bias—developers tend to assume the threats as safe because they are unaware of potential risks (e.g., perception attacks are widely underestimated, especially in high-stakes contexts like military or medical XR) [CNA20, Mag21]. This underestimation reduces developers' motivation and capacity to implement mitigation, increasing the likelihood of suboptimal practices such as insecure interaction designs (e.g., poorly considered voice control in immersive contexts) and missing safeguards (e.g., unmoderated social XR). As noted earlier, developers identify these factors as amplifying XR threats.

We believe that this awareness gap exists due to the emerging nature of XR interaction technologies, where threats are closely tied to new interaction designs, especially the new modalities of data (e.g., motion, voice) and the heavy reliance on them to fully enable basic interactions in XR, rendering an inherent and difficult privacy-utility tradeoff. Knowledge about these emerging XR threats and mitigation support is difficult for developers to access because it evolves rapidly and is scattered across various sources. Unlike mature platforms such as mobile or PC, where established guidelines are readily available (e.g., [Goo25, SJT08]), XR S&P findings are typically shared through academic publications, industry reports, or informal developer-to-developer communication, requiring developers' active engagement.

Hence, we advocate for the development of standardized communication channels for XR developers [KBO21, XR 23], especially those integrated into tools that support the normal development workflow (e.g., IDE), as prior research in the mobile context suggests that these just-in-time reminders contextualized in the app implementation help raise developers' awareness and encourage the adoption of secure and privacy-preserving designs [LAH18]. We argue that it is important to empower developers to proactively follow emerging XR trends, engage in community discussions, and utilize existing channels (e.g., forums [met], feedback systems, and app reviews) to stay informed of new threats and mitigation supports.

4.5.2 Diffusion of responsibility due to the user-experience oriented nature of XR threats

From developers' perceptions on threats and mitigation (Section 4.2,4.3,4.4), we identified a trend of diffusion of responsibilities [DL68] (i.e., the tendency to assume that addressing XR S&P issues is someone else's responsibility). For example, developers suggested that users can mitigate physiology attacks by removing the headset, a problematic stance that disrupts the user experience and shifts both responsibility and risk onto users. Although users and platform providers also share responsibility, developers play a crucial role in collaboratively addressing these risks [NDT17, FHE12].

We believe this diffusion of responsibility is especially pronounced in XR because threats (e.g., physiology attacks that induce motion sickness) span both hardware and software domains and directly impact the users' body and mind [RKM14, Gia24]. Unlike traditional exploits (e.g., remote code execution) that map cleanly onto traditional vulnerability taxonomies, XR threats often look like UX, hardware, or platform issues, causing a misalignment in developers' mental models for threat identification and mitigation. Notably, although major platform providers develop built-in S&P protection, emphasizing considerations such as data minimization, transparency, control, and on-device data processing [App24], they are inevitably limited to relatively coarse-grained mitigations. For instance, although Apple Vision Pro confines camera and sensor inputs (e.g., eye-tracking) to on-device processing, an application-level gaze-driven UI still reveals where users are looking through button selections—effectively leaking the eye-tracking data [Smi24]. Failing to effectively engage developers to mitigate threats represents a missed opportunity to appropriately

balance S&P considerations with the nuanced demands of functionality and the complexities of social contexts.

We advocate for the development of a practical industry framework for XR that clearly assigns ownership of emerging threats to designated stakeholders and provides S&P checklists for each threat. This approach has proven effective in encouraging responsible development in other domains, such as the Web [Ope21], mobile [ASA21], and IoT [BTC18]. While there have been initial efforts in this space (e.g., XRSI's recommendations [XRS23]), our findings highlight a pressing need for: (1) more detailed practice guidelines that address real-world threat scenarios encountered in XR (e.g., social harassment [Alh24], motion as a side channel [SCB24]), similar to the OWASP Top Ten [Ope21] in web development; and (2) accompanying enforcement mechanisms to motivate secure development practices (e.g., incentives or penalties from policymakers for S&P, additional testing support from users, and auditing tool development from researchers and organizations). Such an approach is critical for fostering greater awareness among developers about their S&P responsibilities and for motivating proactive ownership of XR threats.

4.6 Chapter Summary

This chapter presented the first developer-centered empirical study of XR security and privacy, conducted through in-depth interviews with 23 professional XR developers. Our analysis reveals that the persistence of interaction-level vulnerabilities is not primarily a technical failure—it is a socio-technical one. Developers consistently underestimate the privacy sensitivity of continuous kinematic data, overestimate the protection offered by platform-level permissions, and lack actionable guidance from the XR community. Two structural root causes explain this pattern: pervasive *awareness gaps* that prevent developers from recognizing spatially-specific threats, and a *diffusion of responsibility* reinforced by opaque game engine architectures and fragmented security documentation.

These findings shift the frame from individual developer error to ecosystem-level dysfunction. The implication is clear: improving developer education alone is insufficient. If the tools, permission models, and defaults that developers rely on are structurally misaligned with privacy-preserving design, vulnerabilities will continue to emerge regardless of awareness. Chapter 5 responds to this

diagnosis directly, proposing an architectural reconstruction that addresses the root problem: a fully decentralized interaction protocol that removes the centralized data collection model entirely, building user sovereignty into the system structure rather than relying on developer vigilance.

CHAPTER 5

Architectural Reconstruction: A Decentralized Metaverse Framework

Having diagnosed the ecosystem-level root causes of interactive vulnerabilities in Chapter 4, it becomes evident that reactive technical patching is fundamentally insufficient: the ultimate barrier to a trustworthy Metaverse is rooted in user trust, in particular end-users' growing apprehension toward unchecked centralized power and pervasive surveillance. Traditional centralized platforms, often characterized as "walled gardens" [Opp22, LYY23], maintain absolute authority over the server, leading to opaque moderation and mandatory data surrender [WSZ23a, SCT23]. The design problem this chapter addresses, however, is not the binary one prior work tends to frame as "keep the server or remove it." Our 22-participant interview study (Section 5.2.1) shows that users reason about decentralized metaverse design at a finer grain, naming specific operator functions (recording, gating, sanctioning, vouching) and asking, function by function, whether each should be removed, moved onto user-held credentials, replaced by a peer process, or left in place. The result is a multi-adversary picture rather than a monolithic anti-centralization stance. Participants articulate four distinct centralized failure modes (Section 5.3.1): commercial commodification of behavioral data, state-coerced platform compliance, peer-weaponized moderation tooling, and, in immersive use, distrust of the centralized space provider who can manipulate the geometry of perceived space. Each names a different harm and requires a different decentralization move. We accordingly define decentralization throughout this chapter as the distribution of distinct operator roles (identity, shared state, behavior auditing) among system participants, rather than as a single architectural switch.

Bridging this paradigm shift into mainstream interactive use reveals steep technical hurdles. Modern game engines are built around a client-server paradigm and lack native protocols for true peer-to-peer (P2P) interaction [Gre18, YK13]. Conflict-free Replicated Data Types (CRDTs)

resolve syntactic data conflicts in P2P networks but are agnostic to application semantics, so without an authoritative server to enforce physics or resolve game logic, trustless collaborative environments degrade into chaos. Compounding this, our user study surfaces an Accountability Paradox (Section 5.3.3): nine of 22 participants want the privacy of an unmonitored space and the evidence trail of a surveilled one, a tension neither pure architectural pole resolves. To bridge syntactic network convergence and domain-specific truth, this chapter introduces a unified, serverless framework for interactive P2P metaverse applications, organized around three foundational mechanisms tied to specific user-stated needs (Section 5.2.2). **G_1 (User-controlled identity and decentralized state)** pairs a Standardized State Abstraction Layer, which maps the continuous 3D environment into an engine-agnostic, flattened JSON document schema, with workspace-scoped cryptographic identities under hierarchical naming over Named Data Networking. Together, these mechanisms operationalize the permission-gated access intuition we name the Digital Doorbell (articulated without prompt by ten of 22 participants; Section 5.3.2). **G_2 (Semantic consistency under concurrent interaction)** layers a peer-side correctness pipeline (optimistic local execution, continuous invariant verification, threshold-based consensus, and compensating state transitions; Section 5.2.2.2) to keep shared state meaningful under concurrent edits without synchronous stalls. **G_3 (Consensually auditable behavior without central surveillance)** runs a peer-driven auditing pipeline with quorum-based ostracization (Section 5.2.2.3) that treats moderation as the strict evaluation of domain contracts rather than heuristic anomaly detection. One participant (P14) independently proposed exactly this mechanism before being shown it, and the design explicitly refuses the stake-weighted-consensus collapse the same data exposed (Section 5.3.5).

To rigorously evaluate whether the framework delivers these properties without sacrificing the collaborative quality participants told us they would not give up (the Inconvenience Threshold of Section 5.3.4), we conducted a 20-participant within-subjects, double-blind validation in 10 dyads against a centralized baseline. Each dyad completed the same concurrent collaborative task (cooperative building plus synchronized chest-opening) under both conditions, with counterbalancing and blinding maintained for both participants and the immediate facilitator. Participants rated the system on three 7-point Likert constructs they themselves nominated as load-bearing (Perceived Latency, Synchronization Consistency, Collaborative Fluency), and we instrumented both clients

for per-update round-trip latency. Four outcomes converge (Section 5.4). First, paired tests on the three Likert constructs did not reach significance ($p > 0.10$); given the sample size and absence of a formal equivalence test, we read this conservatively as an absence of evidence for a perceptual gap rather than positive equivalence (Section 5.4.2). Second, dyad-level identification of the decentralized condition before any debrief did not exceed chance (4 of 10 correct, exact binomial $p \approx 0.75$). Third, the semantic-correctness rollbacks the framework triggered in every decentralized session went unnoticed by every participant in the structured debrief. Fourth, per-update round-trip latency lands within roughly 2 ms of the centralized baseline ($M_D = 101.8$ ms vs. $M_C = 99.9$ ms), inside the response-time band at which interactive systems are perceived as essentially instantaneous [Nie93]. Together, the validation places the framework inside the small-group, dense-collaboration, high-stakes deployment surface participants drew, without crossing the friction line they said would disqualify it.

In summary, this chapter makes the following core contributions:

- **An empirical map of user needs for a decentralized metaverse.** A 22-participant interview study (Section 5.3) names the multi-adversary picture motivating decentralization, the Digital Doorbell construct (ten of 22 spontaneously articulating the underlying permission-gated access intuition), the Accountability Paradox that audit-without-surveillance design must navigate, the intimate-scale, high-stakes deployment surface users themselves draw, and four predictable folk-model distortions (brand-as-security, anonymity-decentralization conflation, decentralized-as-unsecured, plutocratic-consensus default) that decentralized metaverse systems must articulate design contracts against.

- **A fully decentralized interactive metaverse framework.** We design and implement a serverless framework (Section 5.2.2) that answers those needs mechanism by mechanism: user-held workspace credentials over Named Data Networking with a Standardized State Abstraction Layer (G_1), a CRDT-backed semantic-correctness pipeline (G_2), and peer-driven quorum-based behavior auditing (G_3). To our knowledge, this is the first fully decentralized interactive collaborative metaverse system of this kind.

- **A within-subjects double-blind validation reporting absence of evidence for a perceptual**

gap. A 20-participant validation against a centralized baseline (Section 5.4) returns paired Likert tests with no significant difference on the three nominated constructs ($p > 0.10$ across all three), dyad-level architectural-invisibility identification at chance, semantic-correctness rollbacks that triggered in every decentralized session yet went unnoticed by every participant in the debrief, and per-update round-trip latency sitting within 2 ms of the centralized baseline.

5.1 Background and Related Work

We situate the work against three prior literatures: decentralized metaverse infrastructures and how they have addressed the application layer (Section 5.1.1), the dApp requirements an interactive metaverse framework has to satisfy (Section 5.1.2), and HCI human-subjects work on decentralized technologies that contextualises our user study (Section 5.1.3).

5.1.1 Decentralized Metaverse

The metaverse is typically described as an interconnected, persistent, and immersive network of virtual worlds in which users, represented by avatars, interact and exchange user-generated content [Wei22, DBG13, WSZ23a]. Building such worlds in a decentralized manner has become increasingly attractive as centralized platform control raises concerns beyond traditional system-level limitations: single points of failure, limited transparency, platform lock-in, and concentration of control over user data, behavioral traces, and virtual assets. The same concentration raises data-sovereignty issues, where users lack ownership and control over their digital identities and interactions, and creates room for large-scale surveillance within immersive environments [WSZ23a, HGW23]. As Chapter 3 demonstrated empirically, these concerns are not abstract: the kinematic side channels documented there are precisely the kind of behavioral traces a centralized operator concentrates by default, and the developer-side ecosystem analysis in Chapter 4 traced their persistence to the same operator-centric design defaults this chapter sets out to reconstruct.

Existing decentralized-metaverse work has focused primarily on blockchain-based infrastructures for asset ownership, economic transactions, and interoperability [HGW23, LYY23]. These

approaches provide important building blocks but emphasize system-level guarantees (integrity, provenance, trust) over the application-level requirements of interactive metaverse experiences, which remain comparatively underexplored as a unified protocol layer.

Viewed through this lens, a decentralized metaverse application is a decentralized application (dApp) whose real-time state, governed by hierarchical structures such as Scene Graphs, is concurrently mutated by peers as a shared, structured document [KWH19]. Decentralization is most valuable when user-owned identity, portable state, and cross-community trust have to persist beyond any single operator.

5.1.2 Decentralized Applications

A dApp is a system in which no single party controls the application's logic or state. In practice, many dApps remain only partially decentralized [CWE18]: concentration migrates to instance administrators or relay operators rather than disappearing. We group the requirements of a metaverse dApp into three components, which Section 5.2.2 elaborates as design goals:

- **Decentralized Identity.** Users have to hold self-sovereign identifiers without relying on centralized providers, with cryptographic binding between names, identifiers, and actions for verifiability [RSL20, GMA23, SLS22]. Interactive metaverses benefit from lightweight, locally scoped trust domains and hierarchical naming that support rapid verification and contextual accountability.

- **Decentralized Communication and Shared State.** Real-time dissemination of interaction and game state has to occur without centralized servers, with availability, resilience, and censorship resistance. Named Data Networking offers a name-based, data-centric substrate for peer-to-peer communication [JST09, ZAB14], and Conflict-free Replicated Data Types (CRDTs) give the formal basis on which distributed replicas converge without centralized coordination [SPB11, PBS18, LLK22, WQK22], instantiating the broader local-first philosophy that keeps data ownership at the edge [KWH19]. These mechanisms guarantee syntactic convergence; the spatial, temporal, and social semantics metaverse actions carry are not captured by eventual consistency alone, a limitation the peer-to-peer multiplayer literature has

long acknowledged [YK13].

- **Behavior Auditing without Central Authority.** Trustless peer-to-peer systems have to enforce domain logic, from social norms to physical anti-cheat, without re-introducing a central moderator whose privileged read access defeats decentralization's privacy properties. Decentralized moderation is largely unaddressed in protocol design [SCT23, SLF23]; how moderation infrastructure shapes platform accountability is a long-running concern in social computing [CG16].

Existing dApp families cover these requirements only partially. Federated platforms such as Mastodon distribute hosting but recentralize identity around instance administrators [LSL22, NKF23, BLL24]; protocols such as the AT Protocol used by Bluesky decouple identity from hosting yet still rely on centralized relays for high-bandwidth dissemination [KFG24]; blockchain-based dApps decentralize asset ownership but route real-time interactions through a small set of dominant API gateways. To bridge these gaps, our work introduces a decentralized protocol that pulls identity, semantic synchronization, and moderation into a single serverless substrate for interactive metaverse applications.

5.1.3 Human-Subjects Studies of Decentralized Technologies

A growing body of human-subjects research has looked at how people understand and use decentralized technologies, clustering around three themes. First, studies of cryptographic interfaces document the cognitive burden of managing wallets, keys, and trust assumptions, and show that usability remains a major barrier to adoption [VWM21, FWS21, YSD24, FWT22]. Second, research on federated social systems such as Mastodon has examined instance governance, moderation, and community boundaries [LSL22, NKF23, ZZW24, BLL24]; closest to the present work, Hwang et al. [HNS25] report that members of decentralized social communities reason about specific stewardship functions of an instance administrator rather than about the architecture as a whole, a function-by-function reading we return to in Section 5.5.1. Third, studies of digital sovereignty point to users' desire for greater control over data, identity, and platform governance as a key motivation for decentralized alternatives [LNE23]. Methodologically, our analysis of how users reason about

decentralized architectures draws on the HCI folk-model tradition that documents the predictable structure of lay mental models of security, privacy, and platform behavior [Was10, KDF15, MDD24].

The user-study questions and validation constructs sit on two further bodies of HCI work. The multi-adversary structure of user concerns reported in Section 5.3.1 draws on the networked-privacy and contextual-integrity tradition [Mb14, Nis04], while the responsiveness, synchronization, and collaborative-fluency constructs the validation measures (Section 5.2.2.4) draw on the CSCW tradition that takes shared-state friction as load-bearing for the experience of working together at a distance [OO00, Sch02].

Most empirical HCI work on decentralization focuses on asynchronous systems (wallets, microblogging, governance tools) rather than interactive collaborative environments. Even work on Web3 and metaverse platforms tends to emphasize identity, governance, or asset management rather than interaction quality [Opp22, WLS24, Per24]. The immersive setting has its own human-subjects literature on harassment patterns [BEE19b, LZK25], spatial-privacy concerns [HWN24], and spatial-deception attacks in which a central provider manipulates the user's perception of the virtual environment [CBY21, YLB24]; a recent twenty-year scoping review further finds that social-VR collaboration research operates predominantly at dyad and triad scale [SCG24]. The human experience of decentralized real-time interaction in such settings stays comparatively underexplored, which motivates the two-step study reported in Section 5.2.

5.2 Methods

Centralized metaverse platforms concentrate identity, state, and moderation in a single operator, but it is much less clear what users actually want from a decentralized alternative. Prior work has surveyed the technical building blocks of decentralized applications (Section 5.1); it has not systematically asked which user-reported needs those blocks would in fact have to address. We therefore organize this investigation as a two-step exploration: first ask the users, then design against what they say.

Step one (RQ1) is empirical. We conduct a semi-structured qualitative interview study (Section 5.2.1) to surface, in users' own terms, *when, why, and what concrete needs users have for*

a decentralized metaverse alternative. The findings, reported in Section 5.3, tell us which design problems are worth solving and which framings users will misread the framework against if we do not address them. **Step two (RQ2) is constructive.** Informed by step one's findings, we design a decentralized framework (Section 5.2.2) that addresses a small number of foundational challenges those findings put on the table, and we empirically validate that the framework does not sacrifice the collaborative quality users said they would not give up (Section 5.2.2.4; results in Section 5.4). We do not attempt a complete decentralized metaverse product; the contribution of step two is an exploration of how a protocol can credibly respond to user-stated needs, scoped deliberately to the foundational design problems step one establishes as load-bearing.

5.2.1 User-Study Method

5.2.1.1 Participants and Recruitment

We recruited 22 participants through campus-wide mailing lists, departmental newsletters, and social media. Participation was open to all interested members of the campus community; we did not apply screening criteria beyond informed consent.

We conducted the interviews one-on-one rather than in groups. Several of the six themes the interview probes (privacy-driven self-censorship, distrust of corporate moderation, attitudes toward state-level surveillance; see Section 5.2.1.2) touch on disclosures that participants are unlikely to volunteer in front of peers, and the individual format minimizes the social-desirability pressures that focus groups and paired interviews introduce on exactly these topics [Mor13, Kru94]. The two interviewers worked from a shared script (Section 5.2.1.2) to keep coverage consistent across participants while allowing semi-structured follow-ups on participant-specific experiences.

Our final sample comprised 15 participants self-identifying as male, 6 as female, and 1 as gender queer, drawn from majors spanning Computer Science, Computer Engineering, Electrical and Computer Engineering, Computer Science and Engineering, Mathematics, Applied Mathematics, Biology, Bioengineering, and Computational Biology (Table 5.1). All procedures, including data collection and audio recording, were conducted under our institution's IRB-approved protocol. Each participant received $10 (USD) for an in-lab session of approximately 30 minutes. At the end of

Table 5.1: **Participant Overview** – Demographics and gaming experience for the 22 interview-study participants. "Familiarity to Minecraft" is binarised from a five-point self-report (1 = Moderately/Very/Extremely familiar; 0 = Slightly/Not familiar at all).

ID	Familiarity to Minecraft	Multiplayer Games Played	Age	Gender	Major
P1	1	9	20	Male	CS or Eng
P2	1	7	21	Male	CS or Eng
P3	1	45	19	Female	CS or Eng
P4	0	2	19	Female	Non-CS or Eng
P5	1	3	25	Male	CS or Eng
P6	1	1	27	Male	CS or Eng
P7	1	10	20	Female	Non-CS or Eng
P8	1	5	21	Female	Non-CS or Eng
P9	1	15	25	Male	CS or Eng
P10	0	6	27	Female	CS or Eng
P11	1	3	24	Male	CS or Eng
P12	0	3	22	Female	CS or Eng
P13	0	4	26	Male	CS or Eng
P14	1	8	21	Male	CS or Eng
P15	1	5	22	Male	CS or Eng
P16	1	5	21	Male	CS or Eng
P17	1	11	22	Male	CS or Eng
P18	1	10	21	Male	CS or Eng
P19	1	12	21	Male	Non-CS or Eng
P20	1	52	19	Gender queer	Non-CS or Eng
P21	1	20	20	Male	Non-CS or Eng
P22	1	60	20	Male	Non-CS or Eng

the session, participants were invited to opt in to a second, separately-scheduled session in which they would experience the framework as part of a paired collaborative task; that session and its enrollment are reported in Section 5.2.2.4.

5.2.1.2 Study Design

The interview addresses RQ1 by probing six themes that prior work on decentralization has flagged in isolation but has not jointly examined in interactive metaverse settings (Section 5.1): (i) agency, surveillance, and asymmetric threat models; (ii) data sovereignty and spatial autonomy; (iii) the intimacy boundary and scalability of trust; (iv) the accountability paradox in trustless spaces; (v) the epistemic shift from institutional trust to cryptographic verification; and (vi) the Inconvenience Threshold, the friction level above which participants reported they would abandon decentralization regardless of its principled appeal. Each theme corresponds to one anchor question in the shared interview script and to one or more semi-structured follow-ups that the interviewer used to probe participant-specific examples.

Two interviewers worked from a shared script, asking the anchor questions in a consistent order across participants and following up on participant-volunteered specifics. We chose a semi-structured rather than a fully scripted format because several themes (notably accountability and the epistemic shift) require participants to reason through unfamiliar scenarios in real time, and over-scripting the follow-ups would have foreclosed the comparative reasoning the RQ depends on. Each interview lasted approximately 30 minutes (see Section 5.2.1.1) and was audio-recorded with the participant's consent for downstream coding (Section 5.2.1.3).

The Inconvenience Threshold theme ((vi)) is the only theme operationalized through an in-session artifact rather than purely verbal reasoning. Participants were shown a short hands-on demonstration of our decentralized onboarding flow (the Ownly identity bootstrap) and asked to compare its perceived complexity with the email-and-password login they routinely use. The artifact lets participants ground their answer in concrete friction rather than abstract speculation about decentralization complexity; we treat the resulting verbal comparisons as qualitative data and report them in Section 5.3.4 alongside the other five themes.

5.2.1.3 Qualitative Analysis

We analyzed the interview transcripts and the verbal comparisons collected during the Ownly bootstrap demo (Section 5.2.1.2) through inductive coding, with the goal of producing the macro-

theme framework that Section 5.3 reports. Because the demo responses elicited the same kind of comparative reasoning as the verbal interview portions, we integrated them into the same codebook rather than maintaining a separate sub-codebook for demo-only data. Our coding strategy was bottom-up and open-ended, following Saldaña's iterative approach [Sal21] and supported by MAXQDA. We split the work into two stages: codebook development and full-corpus coding.

In the first stage, two researchers independently coded a subset of four interviews, generating descriptive codes through recursive reading and inductive reasoning. Daily debriefing sessions compared the independent labels, resolved discrepancies, and consolidated overlapping concepts, yielding a preliminary codebook of 82 open codes. We then performed axial coding to identify thematic relations among the descriptive codes, clustering them first into a small set of mid-level themes and then into five overarching macro-themes that frame Section 5.3 (four substantive macro-themes plus a fifth, methodological macro-theme on folk-model distortions).

In the second stage, the same two researchers coded the remaining interviews under a primary-coder and secondary-validator model: each transcript was coded by one researcher and independently verified by the other to maintain interpretive consistency. Emergent codes were reviewed by both researchers and incorporated into the master codebook only after unanimous agreement.

Following McDonald *et al.* [MSF19], we did not compute formal inter-rater reliability statistics. Because our themes are exploratory rather than confirmatory and our analysis privileges the breadth of rich, participant-grounded patterns over a single coder-agreement scalar, McDonald *et al.*'s argument that IRR is the wrong metric for emerging socio-technical practices applies directly to our setting.

The final analytical framework comprises 27 sub-findings organized into the five macro-themes reported in Section 5.3.

5.2.2 Decentralized Framework Design and Validation

Design Goals. Section 5.3 surfaces a long list of properties users want from a decentralized metaverse. To turn that list into a tractable design agenda, we asked of each finding two questions: what centralized failure mode does it identify, and is decentralization an architectural answer to that

failure, or only a constraint on the resulting system's deployment? Three failure modes recurred across the user-study findings as ones that any decentralized alternative would have to address, in the sense that leaving any of them unaddressed undermines the rest: the centralized operator's privileged read access and unilateral control over user-held state (cutting across Sections 5.3.1 and 5.3.2), the inability of straightforward replication schemes to keep concurrent action meaningful (foregrounded by the small-group framing of Section 5.3.4), and the absence of any audit trail that does not also reintroduce surveillance (Sections 5.3.3 and 5.3.3). The remaining findings tell us when and where users would adopt the system rather than what mechanism it needs. We treat the three recurring failure modes as the foundational design problems for this chapter, name them G_1 through G_3, and use Sections 5.2.2.1–5.2.2.3 to explore mechanisms that look promising against each. The exploration is exactly that; we do not claim a complete decentralized metaverse product, and the longer list of user-stated needs is real and important but lies outside this chapter's scope.

G_1 (User-controlled identity and decentralized state). Users want decentralization to remove the centralized operator from two places at once: from the path of their interaction data (driven by commercial commodification, state-coerced platform compliance, and, in XR, distrust of the centralized space provider itself; Section 5.3.1), and from the policy that gates access to the space and creations they consider theirs (the asset-loss and Digital Doorbell findings of Section 5.3.2, on which ten of 22 participants converged without prompt). Section 5.2.2.1 explores a substrate that combines a state-abstraction layer over peer-to-peer transport with a cryptographic-identity layer scoped to workspace-local trust domains.

G_2 (Semantic consistency under concurrent interaction). Users see decentralization as viable only for small-group, dense-collaboration settings (Section 5.3.4); even there, the standard state-replication guarantees that keep peer copies syntactically aligned do not by themselves guarantee that the shared state remains meaningful under the application's domain rules (for instance, that no two players can simultaneously claim the same resource). Section 5.2.2.2 explores a peer-side semantic-correctness pipeline that combines optimistic local execution, continuous invariant verification, threshold-based consensus, and compensating state transitions.

G_3 (Consensually auditable behavior without central surveillance). Users articulate the Accountability Paradox: they want the centralized arrangement's evidence trail without its surveillance read access (Section 5.3.3), and one participant independently proposes a peer-side resolution from first principles (P14 in Section 5.3.3). Section 5.2.2.3 explores peer-driven cryptographic auditing with quorum-based ostracization over an append-only behavior history that any honest peer can replay.

The remaining user-study findings calibrate the framework rather than dictating new mechanisms here. The intimacy-boundary preference (Section 5.3.4) scopes the framework to small workspaces, not massively-multiplayer environments. The Inconvenience Threshold (Section 5.3.4) constrains the friction budget; Section 5.2.2.4 tests it empirically against a centralized baseline on the three constructs participants told us they would not sacrifice: Perceived Latency, Synchronization Consistency, and Collaborative Fluency. The folk-model distortions of Section 5.3.5 name the architectural collapses against which the Discussion (Section 5.5) explicitly defends each design choice; they do not dictate new framework mechanisms.

5.2.2.1 Decentralized Framework Substrate

The substrate of our framework addresses goal G_1 (user-controlled identity and decentralized state) by reframing the metaverse's real-time state as a collaborative document that peers co-author over a serverless network. The substrate has two components: a Standardized State Abstraction Layer that maps the 3D environment into an addressable schema, and an Ownly-based identity and communication substrate that carries authenticated document mutations between peers and scopes trust through workspace-local naming. We describe each in turn.

Standardized State Abstraction Layer. Modern game engines manage virtual worlds via a Scene Graph [Gre18], a nested tree conceptually analogous to a Document Object Model. By serializing the environment into an addressable, JSON-like object tree, we reframe real-time peer-to-peer synchronization as a collaborative document editing problem [KB17]. Raw Scene Graphs, however, are bloated with rendering data and tightly coupled to a specific engine, making them ill-suited for direct P2P exchange. Our Standardized State Abstraction Layer maps the continuous 3D

environment into a discrete, engine-agnostic document schema via two mechanisms:

- **Selective Synchronization.** Following the network-relevance principle of [YK13], we isolate semantic state from local rendering. Only entities tagged with network-aware `Components` (e.g., `Interactable`) are extracted, so the network carries behavioral intents rather than visual noise.

- **Flattened Document Schema.** Directly serializing nested Scene Graphs produces deep JSON trees prone to performance overhead and topological conflicts during concurrent edits [KB17]. We assign each network-active entity a unique identifier (UUID) as a top-level key and flatten its semantic attributes into direct key-value pairs, turning complex graph updates into atomic document mutations.

By collapsing the per-engine graph into a uniform document schema, the Standardized State Abstraction Layer is what makes serverless convergence tractable: the network sees only flat, addressable mutations, no engine internals.

Ownly-based Identity and Communication Substrate. With the 3D environment abstracted into concurrent document patches, the framework requires a decentralized substrate that carries those patches between peers without an authoritative arbiter and that scopes identity to a workspace-local trust domain. We adopt Ownly [YMP24], a user-controlled serverless collaborative editor[1], as that substrate. Peers communicate directly via intermediate packet forwarders rather than authoritative servers. To maintain consistency without a central arbiter, the framework uses Conflict-free Replicated Data Types (CRDTs): as long as honest peers eventually receive the same set of edits, their local replicas converge under concurrent modification and arbitrary delay. Edits are exchanged by (i) encapsulating local document mutations as named, secured data, (ii) joining a Sync group to broadcast updates and serve incoming Interests, and (iii) fetching and validating new data from the network. Each edit is tracked by a name composed of the workspace name, the user name, and a per-user sequence number, as illustrated in Figure 5.1.

[1] `https://ownly.work`

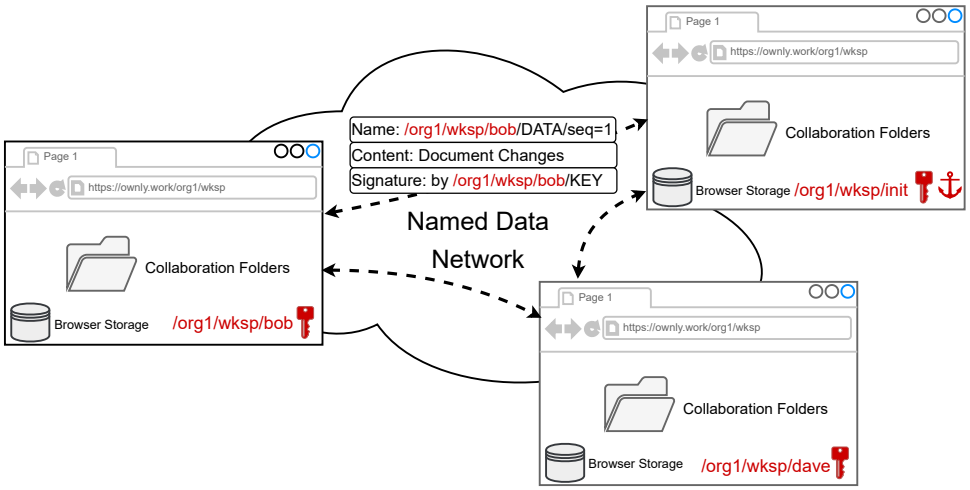


Figure 5.1: In the Ownly instance `/org1/wksp`, users and the initiator collaboratively edit the world document. Document mutations are structured as named secured data and exchanged via a Named Data Network.

To support user-controlled identity (the second half of G_1), the substrate models each virtual world as a localized workspace that doubles as a self-contained trust domain [YMP24]. The workspace creator serves as the domain controller and invites collaborators out-of-band (e.g., by email) to exchange access credentials. Hierarchical naming then resolves decentralized identity: if Alice (controlling the namespace "`/org1`") initiates a workspace "`/org1/wksp`" and out-of-band invites Bob ("`/bob`"), Bob is issued a locally scoped cryptographic identity "`/org1/wksp/bob`" that he uses to sign his state updates. All authenticated members hold equal read and write permissions over the shared world document.

From Substrate to Application Guarantees. The substrate guarantees only syntactic convergence over the world document; it is agnostic to whether a converged state respects the metaverse's domain logic and is unable, by itself, to identify or isolate compromised peers. Sections 5.2.2.2 and 5.2.2.3 describe the two application-layer guarantees we layer on top: enforcing semantic correctness under benign concurrency (G_2) and decentralized behavior auditing under adversarial conditions (G_3).

5.2.2.2 Enforcing Semantic Correctness

The substrate guarantees only syntactic convergence of the world document; it does not by itself ensure that a converged state remains valid under the application's domain logic. This subsection addresses goal G_2 (semantic consistency under concurrent interaction) by introducing a pipeline that detects and repairs invariant violations without sacrificing the low-latency interactivity that the validation in Section 5.2.2.4 tests against a centralized baseline. We motivate the problem with a micro-case study of concurrent artifact manipulation, then describe an asynchronous validation pipeline that combines optimistic local execution, continuous semantic verification, threshold-based consensus, and compensating state transitions.

Case Study: Semantic Divergence and Resource Invariant Violations

Context: Consider a highly concurrent environment in which actors interact with shared digital artifacts. Unlocking a shared resource receptacle requires the consumption of a discrete token; a fundamental resource conservation invariant dictates that tokens consumed must equal successful unlocks.

Structural Resolution: If Actor A and Actor B claim the resource simultaneously, the underlying CRDT deterministically orders the events and allocates the contents exclusively to Actor A.

Semantic Breakdown: To preserve low-latency collaboration, both clients optimistically execute the action locally and deduct a token before network synchronization.

Outcome: The network converges on a syntactically sound but semantically invalid state: one resource was dispensed, yet two tokens were permanently consumed. This is a decentralized resource-duplication anomaly.

Resolving these anomalies without stalling client execution for synchronous coordination is the central design problem. Our pipeline (Algorithm 1) combines:

- **Optimistic Local Execution and Hypothetical States.** Clients apply inputs immediately to a local state view. When a client at logical state S_k executes operation op, it transitions to a *hypothetical state* $S'_k = f(S_k, op)$ before network convergence. This zero-latency paradigm mirrors client-side prediction and optimistic UI updates validated in interactive web and mobile applications, and is necessary to keep collaborative tasks fluid.

- **Continuous Semantic Verification.** Each client functions as a background validator. Following the principle of *Invariant Confluence* (I-confluence) [BFF14], that synchronous coordination can be safely bypassed if application invariants are continuously preserved, the client evaluates the converged state $S_{\text{converged}}$ against the domain invariants $\mathcal{I}_{\text{domain}}$ upon high-stakes transitions. For example, it validates the resource invariant

$$\mathcal{I}_{\text{resource}} : \sum(\text{TokensConsumed}) \equiv \sum(\text{SuccessfulUnlocks}) \tag{5.1}$$

  and flags an anomaly only when the assertion fails.

- **Threshold-Based Consensus.** To prevent malicious actors from weaponizing state corrections, rollbacks cannot be unilateral. Drawing on quorum-based consensus and peer-auditing protocols, a validator that detects a violation broadcasts a rollback request; peers independently re-evaluate the invariant and broadcast approval votes. Corrective action proceeds only once a quorum is reached.

- **Compensating State Transitions.** Once the quorum τ_{quorum} is satisfied, we avoid arbitrary state rewinding, which would violate the append-only nature of CRDTs. Inspired by the *Saga pattern* [GS87], application state is versioned by logical clock vectors (`[user:sequence]`), and the network injects a formally defined compensating transaction (e.g., refunding the erroneously consumed token to Actor B). The logical clock continues forward while the *content* is reconciled to semantic validity.

Algorithm 1 Decentralized Semantic Verification: Detection and Rollback Request

Require: S_{local} (Local state), $\mathcal{I}_{\text{domain}}$ (Set of domain invariants)

procedure OnStateUpdate(Δ_{net})
  $S_{\text{converged}} \leftarrow$ CRDT_Merge($S_{\text{local}}, \Delta_{\text{net}}$)
  $anomalyDetected \leftarrow$ False ▷ Step 1: Continuous Evaluation of Semantic Invariants
  for each $I \in \mathcal{I}_{\text{domain}}$ **do**
    if not I.assert($S_{\text{converged}}$) **then**
      $anomalyDetected \leftarrow$ True
      $\vec{v}_{\text{err}} \leftarrow S_{\text{converged}}$.vector
      break
    end if
  end for ▷ Broadcast Compensating Rollback Request
  if $anomalyDetected$ **then**
    $req \leftarrow$ CreateRollbackReq($\vec{v}_{\text{err}}, S_{\text{converged}}$)
    Broadcast(req)
  end if
end procedure
procedure OnReceiveRollbackRequest($\vec{v}_{\text{err}}, S_{\text{err}}$) ▷ Step 2: Independent Peer Verification
  if not VerifyInvariants($S_{\text{err}}, \mathcal{I}_{\text{domain}}$) **then**
    Broadcast(Vote($\vec{v}_{\text{err}}$, APPROVE))
  end if
end procedure

Algorithm 2 Decentralized Semantic Verification: Threshold Consensus and Compensation

Require: $Votes$ (Vote registry), τ_{quorum} (Consensus threshold), $S_{\text{converged}}$ (Current converged state)

procedure OnReceiveVote($\vec{v}_{\text{err}}, peerID$)
    $Votes[\vec{v}_{\text{err}}]$.add($peerID$) ▷ Prevent replay ▷ Step 3: Threshold Consensus and Compensating Transition
    if $|Votes[\vec{v}_{\text{err}}]| \geq \tau_{\text{quorum}}$ **then**
        $S_{\text{prior}} \leftarrow$ GetPriorValidState($S_{\text{converged}}, \vec{v}_{\text{err}}$)
        $\Delta_{\text{comp}} \leftarrow$ GenerateCompensation(S_{prior})
        $\vec{v}_{\text{new}} \leftarrow$ IncrementClock($S_{\text{converged}}$.vector)
        ApplyAndBroadcast($\Delta_{\text{comp}}, \vec{v}_{\text{new}}$)
        Clear($Votes[\vec{v}_{\text{err}}]$)
    end if
end procedure

The pipeline preserves semantic correctness without blocking the main rendering or interaction threads: speculative CRDT mutations, anomaly alerts, and approval votes are all published asynchronously to the workspace Sync group introduced in Section 5.2.2.1. Section 5.2.2.3 extends this validator pattern from benign concurrency to adversarial behavior.

5.2.2.3 Decentralized Behavior Auditing and Moderation

While Section 5.2.2.2 addresses semantic anomalies that arise from benign concurrency, an operational metaverse must also withstand deliberate adversarial behavior. Our participants explicitly raised this question in the Accountability Paradox and the broader epistemic shift toward cryptographic rather than institutional verification (G_3). This subsection introduces a peer-driven auditing pipeline that extends our application of domain invariants $\mathcal{I}_{\text{domain}}$ to function as enforceable contracts. We motivate the problem with a micro-case study of spatial manipulation, then describe how continuous background logging, deterministic constraint evaluation, peer-driven accusations, and quorum-based ostracization together identify and isolate malicious actors without a central moderator.

Case Study: Spatial Discontinuity and the Violation of Embodied Constraints

Context: Consider a persistent environment governed by embodied constraints. A fundamental kinematic invariant $\mathcal{I}_{\text{kinematic}}$ dictates that the distance traversed (Δd) over elapsed time (Δt) cannot exceed the domain's maximum theoretical velocity $V_{\max}$.

Exploit (Structural Validity): A malicious actor manipulates their local client to instantly update coordinates from Point A to a distant Point B and broadcasts the transition. The underlying P2P protocol accepts the coordinate update as structurally well-formed.

Semantic Breakdown (Invariant Violation): Although syntactically valid, the transition fails the $\mathcal{I}_{\text{kinematic}}$ assertion, as $(\Delta d/\Delta t) > V_{\max}$.

Outcome: Without an application-layer auditor, the system validates a physically impossible trajectory and grants the adversary an unearned advantage.

Replacing the centralized moderator's policing function without introducing a privileged peer requires conceptualizing moderation not as heuristic anomaly detection but as the strict evaluation of domain contracts. Our auditing pipeline (Algorithm 3) comprises:

- **Continuous Background Logging.** Each client operates as a continuous background model checker, maintaining an ephemeral sliding-window log of state transitions $S_{t-k} \rightarrow \cdots \rightarrow S_t$ received from peers.

- **Deterministic Constraint Evaluation.** To isolate application semantics from network propagation delay, verification relies exclusively on client-declared event timestamps T_{event}. The evaluator computes the delta between consecutive state payloads (e.g., $\Delta d/\Delta t \leq V_{\max}$) and verifies domain constraints. Anchoring audits in the internal temporal logic of events natively prevents false positives during network partitions.

- **Peer-Driven Accusations.** If a local model checker detects an assertion failure, it broadcasts an *Accusation* packet whose payload is the logical clock vector `[user:sequence]` corresponding to the illicit transition. The accusation acts as a pointer to the evidence rather than a self-contained proof.

- **Quorum-Based Ostracization.** To prevent malicious actors from weaponizing the moderation

system via false accusations, peers receiving an accusation must independently corroborate it. They retrieve the cited events from their own logs and deterministically replay the transition; a violation strike is registered only when the peer independently reproduces the assertion failure. Once a node observes validated accusations from a predefined quorum (the `EVICTION_QUORUM` parameter, set to three independent peers in our prototype), the community ostracizes the offending actor and drops subsequent packets originating from that name.

Limitations and Open-Source Extensibility. We acknowledge the intractable nature of exhaustively codifying all possible adversarial behaviors. The set of invariants $\mathcal{I}_{\text{domain}}$ in our prototype demonstrates the framework's viability rather than providing a comprehensive anti-cheat solution. Because the auditing protocol enforces externally specified invariants rather than coded heuristics, an open-source extension surface is available: communities can add domain-specific invariants by appending to $\mathcal{I}_{\text{domain}}$ without modifying the substrate, and the same quorum-based ostracization mechanism applies. We treat the construction of a richer invariant library as an explicit boundary of this work and a direction for community contribution.

5.2.2.4 Validation Method

Participants. The validation drew its participants from the Section 5.2.1.1 sample. Of the 20 participants who accepted the opt-in invitation extended at the close of their individual interview, all 20 enrolled in the validation and were paired at random into 10 dyads. We chose the dyad as the smallest configuration that meaningfully exercises the framework's concurrent state synchronization, since the validation rests on whether two simultaneously active clients can coordinate over a shared world document.

The validation session lasted approximately 30 minutes and took place in an in-lab setting. Each participant received $15 (USD) for the validation session; combined with the $10 received for the Section 5.2.1.1 session, total compensation per participant across both sessions was $25. Procedures, including a post-session debrief, were conducted under our institution's IRB-approved protocol.

Algorithm 3 Consensus-Driven Invariant Checking and Peer Ostracization

procedure AUDITPEERTRANSITIONS($targetPeer, recentLog, \mathcal{I}_{\text{domain}}$)
  $S_{prev}, S_{curr} \leftarrow$ GetLatestTransition($targetPeer, recentLog$)
  $violationDetected \leftarrow$ False ▷ Step 1: Deterministic Constraint Evaluation
  for each $invariant \in \mathcal{I}_{\text{domain}}$ **do**
    if not $invariant$.assert(S_{prev}, S_{curr}) **then**
      $violationDetected \leftarrow$ True
      break
    end if
  end for ▷ Step 2: Initiate Peer-Driven Accusation
  if $violationDetected$ **then**
    $evidence \leftarrow \{S_{prev}.vector, S_{curr}.vector\}$
    Broadcast(Accusation($targetPeer, evidence$))
  end if
end procedure

procedure ONRECEIVEACCUSATION($targetPeer, evidence, \mathcal{I}_{\text{domain}}$)
  $S_{prev}, S_{curr} \leftarrow$ RetrieveLocalState($evidence$) ▷ Step 3: Independent Corroboration
  if VerifyInvariants($S_{prev}, S_{curr}, \mathcal{I}_{\text{domain}}$) == False **then**
    $AccusationTally[targetPeer] \leftarrow AccusationTally[targetPeer] + 1$ ▷ Step 4: Community Ostracization upon Quorum
    if $AccusationTally[targetPeer] \geq$ EVICTION_QUORUM **then**
      DisconnectAndBan($targetPeer$)
    end if
  end if
end procedure

Design. The validation employs a within-subjects, double-blind design to assess whether participants detect a perceptual difference between the decentralized framework and a centralized counterpart when they experience the two side by side. Each dyad completes the same collaborative task

under both conditions: **Condition A (Centralized Control)** and **Condition B (Decentralized Treatment)**. The active condition is concealed from both the participants and the immediate session facilitator, and counterbalancing (A→B for half the dyads, B→A for the other half) mitigates order and learning effects.

Collaborative Task. To stress concurrent state synchronization and conflict resolution, each dyad performs a two-stage mission. In the first stage, the dyad jointly constructs a virtual house, which generates a continuous stream of small, asynchronous state mutations (placing, removing, and re-coloring blocks). In the second stage, the dyad performs a high-concurrency synchronization task: the dyad locates five treasure chests distributed in the environment and is explicitly instructed to coordinate over voice and trigger the opening action on all five chests at the same moment. The second stage is designed to maximize near-simultaneous state transitions and to expose any divergence between the two conditions' handling of concurrent edits to the world document.

Construct Isolation. Before each session the facilitator instructs the dyad to evaluate the underlying network protocol, in particular its responsiveness, stability, and synchronization quality, rather than the game's aesthetics or task mechanics. After each condition, each participant rates the system on three 7-point Likert constructs: Perceived Latency (responsiveness of inputs and smoothness of avatar motion), Synchronization Consistency (alignment between the participant's view of the world and the partner's view, including absence of teleporting, vanishing, or jumping objects), and Collaborative Fluency (the system's tendency to support rather than interrupt joint action). Each construct is operationalized by three items. The facilitator also records instances of "verbal confusion" (e.g., *"I placed a block here, why is it gone?"*) as a behavioral indicator of perceived synchronization failure.

Architectural Invisibility Test. At the close of the second condition, with both participants still blinded to the condition assignment, each participant is asked which of the two systems they believe was the decentralized one. We frame this as an "architectural invisibility test": if the perceived quality of the two conditions is statistically indistinguishable (Likert ratings paired across conditions) and participants' identification accuracy is at chance, we conclude that no perceptual gap is detected in this sample. We treat as the null hypothesis that participants perceive no statistically significant

difference in Perceived Latency, Synchronization Consistency, or Collaborative Fluency between the two conditions, and that identification accuracy does not differ from 50%.

Debrief. After the identification question, the facilitator conducts a brief structured debrief in which the dyad is invited to comment on any specific moment of confusion, divergence, or smoothness they noticed during either condition. The structured debrief preserves the blinding: it does not disclose which condition was decentralized, nor does it disclose that the study compared a decentralized framework against a centralized baseline, in order to keep the participants' qualitative reflection naive to the comparison. The debrief is audio-recorded and integrated into the Section 5.2.1.3 codebook only for codes that bear on perceived quality between the two conditions; analyses of validation-specific themes are reported in Section 5.4.

5.3 User-Study Results

The qualitative interview study described in Section 5.2.1 was designed to surface, in participants' own terms, what they want from a decentralized alternative to today's centralized metaverse platforms. This section reports what those terms turned out to be, answering RQ1: *when, why, and what concrete needs do users have for a decentralized metaverse alternative?* Across 22 participants, our analysis clustered their reasoning around five questions they returned to repeatedly: (i) which centralized failures actually push users toward decentralization (Section 5.3.1); (ii) which ownership properties they ask decentralization to provide (Section 5.3.2); (iii) how trust is reconstituted once no central operator is available to provide it (Section 5.3.3); (iv) under what conditions they would actually switch from a centralized default to a decentralized alternative (Section 5.3.4); and (v) where lay intuitions about decentralization systematically break down, and what that tells us about how to explain decentralization to its prospective users (Section 5.3.5). Section 5.3.5 closes by naming the small subset of findings that load-bear the framework's design in Section 5.2.2.

Each sub-finding opens with the list of participants who engaged with it (P*n*); verbatim quotations are italicized and attributed to the specific participant.

5.3.1 Threats that drive the demand for decentralization

We begin with the question participants treated as primary in their own answers: what is wrong with the centralized arrangement that decentralization would have to fix? In their framing, decentralization is a response to specific failures: who watches, who can suppress, and what is at stake when that watching is misused. Five distinct adversary models surfaced, and the model a participant held determined which property of decentralization they considered indispensable. We catalogue the five, then close by naming which of them decentralization actually answers and which it does not.

Users want decentralization to escape the centralized operator's monetization of their behavior. P3, P4, P5, P6, P11, P12, P13, P14, P19 raised the data-as-business-input chain that the centralized operator participates in as something they wanted to escape. P19 distinguished raw leakage from algorithmic targeting: *"if you like secretly access some of my extra data and choose to like, I mean, only seeing like some of the content, that's pretty bad"* (P19). Several already practised channel-switching as a coping behavior [HM16], treating the centralized arrangement as a coerced trade they would prefer to exit.

Users want decentralization to escape state-coerced platform compliance. P5, P6, P11, P13, P21, P22 reached for governmental adversaries grounded in lived geopolitical experience, locating the harm not in the state directly but in the centralized platform's obligation to comply with state requests. P13 recalled a concrete platform sanction triggered by content the participant considered innocuous: *"I have actually got banged from social media because I just post some tax questions, and I was like, what's wrong with that? How I cannot post about tax questions? It's just text!"* (P13). We read the vividness of the recall as diagnostic: opaque centralized moderation reads as state-adjacent to these participants even when no state actor is directly involved, which is the condition decentralization aims to dissolve.

Users want decentralization to escape peer-weaponized centralized moderation. P3, P4, P7, P8, P11, P13, P15 named a different adversary: not the centralized operator, but other users wielding the operator's moderation pipeline against them. The most diagnostic instance came from P15, who

described a chilling effect on reporting misbehaviour rather than on doing it: *"I'd rather just stay quiet and just let them continue the trouble"* rather than risk being chased through reports (P15). The desired escape is therefore not from moderation per se but from moderation as a centralized lever: in a decentralized arrangement, the lever does not exist for peers to seize.

A subset of users wants decentralization because they distrust the centralized space provider in XR, not only the data collector. A smaller cluster (P1, P15, P16) named threats text platforms cannot host: the centralized space provider can manipulate the geometry of the user's spatial perception. *"there's like attacks like men in the room. . . you're tricking. . . you're like modifying the dimension of like the geometry of the virtual space so that you cause the player to like fall down or something"* (P1). The threat reorients the centralization critique from who reads the data to who renders the world, which is a meaningfully larger surface; the framework's peer-side invariant checking (Section 5.2.2.3) is calibrated to it.

A subset of users does not want decentralization because they do not believe any networked arrangement is sufficient. P3, P4, P11, P17, P19, P20, P21, P22 articulated the threats clearly but reported no behavior change [HM16]: *"Yeah, but I guess hasn't translated to my actions yet"* (P17). Several reserved their most sensitive communication for non-digital channels, treating both centralized and decentralized systems as insufficiently private. This stance does not adjudicate the centralized-vs-decentralized question; it brackets it.

Synthesis. Across the five adversary models, decentralization is an apt answer for three (commercial commodification, state-coerced compliance, XR space-provider distrust): each names a harm whose locus sits at the centralized operator itself, and removing the operator removes the harm. For peer-weaponized moderation, decentralization is a partial answer: it removes the lever peers seize, but it does not by itself answer the underlying social harm peers cause, which is why Section 5.3.3 treats the resulting accountability paradox as a first-class design problem. The privacy-cynicism stance brackets rather than answers the comparison, and we read it as the upper-bound calibration on what any decentralized framework can claim: the right framing for ours is substantially better

than centralized, not equivalent to no network. One finding cuts against the standard HCI privacy narrative: in our data, the most concrete reports of day-to-day behavioral self-censorship trace back to peer-weaponized moderation rather than to corporate surveillance, which gives us reason to question any centralization defence that leans on the moderation pipeline as the chief user protection.

5.3.2 Ownership properties decentralization is asked to provide

If Section 5.3.1 catalogued the centralized failures users want decentralization to escape, this section catalogues the positive properties they want decentralization to put in their place. Participants did not ask decentralization for ownership as a single thing. They asked for four distinct properties the centralized arrangement was perceived to deny: continuity (the platform cannot take it away), authorship-honoured permanence (the platform cannot retroactively invalidate it), permissioned access (the platform cannot let strangers in without my consent), and key-based exclusivity (only my key, not their policy, governs access). Each property names a centralized failure mode and a desired decentralized counterpart.

Users want decentralization to remove the centralized operator's unilateral power to delete their creations. P5, P7, P11, P13, P15, P21 cited past experiences of losing in-game property to platform decisions, recalled with strong affect. P7: *"I'd rather have it stored myself. Cuz what if they lose it?. . . When they took away my dog. Yeah. . . I was mad"* (P7). The implicit comparison is local custody against centralized custody; participants did not engage with the storage-redundancy comparison, on which the framework's CRDT-replicated world document (Section 5.2.2.1) actually wins on both axes (we develop this in Section 5.5).

Users want decentralization to disable the centralized operator's power to retroactively invalidate already-built artifacts. P5, P11, P13, P15 identified a subtler form of the same demand: terms-of-service changes that invalidate artifacts the user has already laboured over. P13: *"the company's policy. . . they can change very, like, instantly. . . what if one day I spend tons of work, tons of my sweat on that creation, and suddenly I'm banned for using that?"* (P13). The defensive response inside the centralized arrangement is self-backup as veto; the desired decentralized

arrangement makes the veto structural rather than improvised.

Users want decentralization to gate access on permission they grant, not on policy the operator sets: the Digital Doorbell. P3, P5, P6, P7, P9, P11, P15, P17, P19, P21 articulated a permission-gated access intuition for what they considered a truly private virtual space: even the legitimate host has to request entry. P7 was the one participant to frame this intuition through a door metaphor, which we adopt as the name for the underlying requirement: *"Even though they own it, I would still want to have control on who could enter my space or who could edit what I built. . . make sure they notify the user. Like kind of like ringing the door"* (P7). The other nine participants articulated the same permission-based gating without invoking the door image directly; we call the resulting requirement the Digital Doorbell after P7's framing. P15 spontaneously surfaced the abuse vector any naive "anyone can claim any space" decentralized policy implies (the user who circles the whole world and declares it theirs); the participant's instinct that gating must be credentialed rather than declared is exactly the affordance the framework's hierarchical naming provides (Section 5.2.2.1).

Users want decentralization to make access governed by their key, not by the operator's contract. P3, P5, P9, P13, P17, P19, P21 articulated the qualitative difference between ownership tied to a private key and ownership tied to a TOS clause [LNE23]. P5: *"without my personal key no one else can access [that] still can be considered as private"* (P5). One participant (P13) pushed this maximalism to its limit by treating only self-hosted infrastructure as "private," a stance internally consistent but one that excludes every realistic deployment, centralized or decentralized.

Users want decentralization selectively: communicative artifacts should be local, but structural artifacts can stay centralized. P3, P4, P5, P11, P15, P19, P20 explicitly distinguished what they wanted decentralized from what they were content to leave on a commercial server. In our coding, communicative artifacts (chat, voice, intent) merited local custody, while structural artifacts (placed blocks, built environments) did not; we read the pattern as a request for architecturally selective decentralization that matches the artifact's sensitivity, rather than wholesale decentralization. This is the lay version of the Selective Synchronization mechanism (Section 5.2.2.1).

Synthesis. Four of the five ownership properties bind directly to specific affordances the framework provides: continuity and retroactive-invalidation defence bind to CRDT-replicated state (G_1); the Digital Doorbell binds to workspace-scoped credentialed access (G_1); key-based exclusivity binds to the cryptographic identity layer (G_1). The fifth, selective sovereignty, binds to Selective Synchronization rather than to identity. The convergence is unusually strong: ten of 22 participants articulated the permission-gated access intuition we name Digital Doorbell (after P7's door metaphor) without prompt, which we read as evidence that the underlying intuition is intuitive rather than imposed, and which licenses the chapter to adopt it as a load-bearing design metaphor. Two cautions calibrate the claim. P7's affective reaction to asset loss underspecifies the storage-redundancy comparison, on which multi-peer CRDT replication is in fact more loss-resilient than either centralized custody or single-laptop backup; the Discussion makes that non-obvious counter-claim explicit. P13's privacy maximalism (self-hosted infrastructure as the only "truly private" configuration) is internally consistent but excludes any networked deployment, and we treat it as the same upper-bound calibration adopted at the close of Section 5.3.1.

5.3.3 How users reconstitute trust without a central operator

Sections 5.3.1 and 5.3.2 described what users want decentralization to remove (the centralized operator's surveillance role) and what they want it to put in its place (continuity, exclusivity, gated access). The natural next question is how the resulting arrangement is supposed to hold together: with no central operator, who records, who sanctions, who decides, and who vouches? This is the densest cluster in our corpus because it is where the comparison is hardest. The centralized operator does a great deal of trust work participants want kept, even after they remove the operator: recording who did what, sanctioning bad actors, making collective decisions, vouching for software, and serving as a fallback when interpersonal trust runs out. Participants converged on a paradox they could name but not resolve, and then independently proposed several mechanisms for resolving it, some of which closely mirror the framework's design.

The Accountability Paradox: users want decentralization for privacy but want the centralized arrangement's evidence trail at the same time. P1, P3, P9, P11, P13, P15, P17, P19, P21

wanted both the privacy of an unmonitored space and the accountability of a surveilled one. P3: *"if like... grooming and stuff [is going on], it's very important that there are logs and like evidence in place so... malicious users can be removed"* (P3). P1 inferred too quickly that decentralization implied the absence of any guidelines, and P15 cited 4chan-style failures as evidence that decentralization would amplify trolling. Both inferences confuse anonymity with decentralization (Section 5.3.5); both nonetheless name the real demand, which is that decentralization preserve an evidence trail while removing the centralized operator's privileged read access to it.

Users independently propose an append-only behavior log as the resolution of the paradox. P1, P7, P15 named the construct the framework provides without prompt. P7 articulated it most cleanly: *"there's history or a record of what people are doing at least you have proof of that and everyone has access to it"*, and later: *"you can erase the evidence, but at least it's on there"* (P7). This is the cleanest match in the corpus between a user intuition and a specific construct in the framework: the immutable, peer-replicated log of state mutations that the world document provides (Section 5.2.2.1). The participants are not endorsing the design after seeing it; they are reaching for the same construct as the right resolution before seeing it.

Users propose peer-driven cryptographic auditing as the substitute for the centralized operator's enforcement role. P14 reached for the framework's auditing mechanism (Algorithm 3) without exposure to it: *"if they cheat, they can only do it on their computer, so when they are communicating with another computer, there will be some, like, verification of the local data. So... you can actually let everybody follow the rules"* (P14). P19 and P20 separately recognised and endorsed the same construct once it was demonstrated in the post-task debrief. One spontaneous articulation is not statistical evidence of convergent design intuition, but it is a useful existence proof: the centralized enforcer is not the only conceivable enforcer, and the alternative reads as natural rather than alien once articulated.

Users reach for consensus vocabulary, but import the cryptocurrency-vernacular failure mode. P1, P5, P11, P15 reached for cryptocurrency-style consensus, and P1 articulated the failure of stake-weighted consensus mid-sentence: *"if the person who's powering the server is like more than 51% then he gets to say that oh yeah we should allow bloodshed in this meta. . . Not necessarily right but given the constraints that's probably the best way"* (P1). The framework adopts the form of the construct (quorum-based decision) but explicitly refuses the content: `EVICTION_QUORUM` (Section 5.2.2.3) counts independent peers rather than weighting by stake. The takeaway for the comparison: users are willing to accept consensus-based decision-making as a replacement for the centralized operator, but they will import the wrong consensus shape if the design does not name its alternative.

A counter-current argues for social norms instead of encoded rules. We coded a counter-current in P3, P11, P13, P17, P19, P21, P22's responses: decentralized governance should emerge as social norms rather than be encoded in protocol [NKF23, BLL24]. In this reading, the centralized operator's enforcement role should not be replaced at all; instead, the decentralized arrangement should rely on the social fabric the operator usually obscures. The framework's stance, articulated in Section 5.2.2.3, is hybrid: enforceable invariants are quorum-checked, and social norms remain the appropriate vehicle for context-specific community rules.

Users reject the listen-server hybrid as a centralized-vs-decentralized middle ground. We coded nine of the 22 participants (P3, P5, P9, P11, P13, P15, P17, P19, P21) as rejecting the single-host-peer hybrid the interview protocol offered as a middle ground. Three independent objections recurred in their responses: the host gets asymmetric power to cheat, the host is asymmetrically exposed to attack, and the social problem of selecting and compensating the host adds fresh friction. P21 named the one condition under which the hybrid would be acceptable: pre-existing real-world trust in the host. That is the out-of-band credential exchange the framework already provides (Section 5.2.2.1). We read the pattern as a preference for either a structurally absent operator role (fully decentralized) or one filled by a known and trusted human, rather than a designated peer.

Trust does not vanish under decentralization; two participants reconstitute it along three articulated axes. P1 and P5 (only two of the 22 participants in this cluster) articulated three potential replacements for brand-based trust: cryptographic verification of software identity, third-party audit institutions, and pre-existing real-world relationships. P1: *"you have some kind of PGP keys. . . trusted developers. . . and you can verify that the software is signed by them"* (P1). With only two informants, we treat the three-axis structure as suggestive rather than convergent; the framework nonetheless provides the first reconstitution natively, supports the second via the open-source extension surface (Section 5.2.2.3), and treats the third as the implicit premise of out-of-band credential exchange.

Synthesis. This cluster carries both the strongest convergence between lay reasoning and framework design and the most consequential lay misreadings, and the two readings cut in opposite directions. The convergence (P7's spontaneous append-only-log proposal, P14's spontaneous peer-audit proposal, the near-unanimous rejection of the listen-server hybrid) is the strongest single signal in our data that the framework's design reads naturally as an answer to the question rather than as a researcher-imposed artifact. The misreadings (anonymity-decentralization conflation, plutocratic-consensus reach) are the strongest signal that the chapter's prose must explicitly refuse each architectural collapse, since users will reach for these defaults if the framework does not name its alternative. The Accountability Paradox is not resolvable from within either pure architecture, which is why the framework's hybrid stance, enforceable invariants for hard rules and social norms for soft community context, maps onto both halves of the user demand at once.

5.3.4 When users will switch from centralized to decentralized

Sections 5.3.1–5.3.3 described what users want a decentralized alternative to be and how they think it should work. This section asks the consequential follow-up: when, in practice, would they actually choose it? Participants did not present decentralization as a global replacement for centralization. They drew specific boundaries between contexts where they would switch and contexts where they would not, and those boundaries are themselves design requirements for the deployment surface.

Decentralization wins for small, intimate, known-peer settings; centralization wins at scale. P1, P3, P9, P11, P15, P21 bounded decentralization's use case to small-group, dense-collaboration settings, treating it as the digital living room rather than the public square. P3: *"pretty much two person game"* (P3). The framework's workspace abstraction (Section 5.2.2.1) is deliberately scoped to this regime, not to massively multiplayer environments.

Decentralization wins for competitive play because it removes the centralized host's ping advantage. A less-expected use case came from P15, P17, P18, P19's responses, one prior decentralized-systems work has not foregrounded: in their reasoning, peer-to-peer removes the centralized host's geographic ping advantage, making P2P fairer for competitive play. We note this as a candidate adoption driver that cuts against the conventional intuition that competitive play requires server authority.

Decentralization wins only above a stakes threshold; for casual use, centralization remains preferred. P3, P5, P9, P11, P13 will tolerate decentralization's friction only for high-stakes, narrow use cases. P3 named the boundary explicitly: *"if I was doing work. . . professionally or. . . classified or anything, I could see this having any use"* (P3). P5 quantified the tolerance as a bounded performance sacrifice: small acceptable, large not. This bounded tolerance directly motivates the no-degradation hypothesis Section 5.2.2.4 tests and Section 5.4 reports: decentralization can only win above the stakes threshold if it does not simultaneously lose on perceived collaborative quality.

Centralization wins by default through habituation, regardless of architectural merit. P1, P3, P5, P7, P11, P13, P15, P21 default to centralized not out of preference but out of familiarity. P7: *"Yes, cuz that's what we're used to"* (P7). P1 reflected survivorship bias [Was10]: only stable centralized systems have been observed, so centralization is inferred to cause stability. This is the one adoption barrier the framework cannot architect away; it can only keep its friction bounded enough that the high-stakes use cases participants do enumerate become routine.

Decentralized onboarding sits at the upper edge of the tolerable friction range. P3, P5, P9, P11, P13, P14, P15, P16 rated the demo onboarding flow against a familiar 2FA flow. P5: *"[I]t's relatively the same. . . people may get may feel a little weird since it's not the conventional login method"* (P5); P15: *"It seems quite complex. . . Six or seven. Complicated maybe eight"* (P15). The friction source is unfamiliarity rather than intrinsic complexity, but the friction is real and sits at the upper end of what participants self-report as tolerable for high-stakes contexts.

Synthesis. The deployment surface where decentralization wins is sharply bounded by participants themselves: small-group, dense-collaboration, high-stakes use, with a friction budget at the upper edge of what they self-report tolerating. Inside that envelope, decentralization wins on continuity, ownership, evidence-without-surveillance, and (perhaps surprisingly) competitive fairness; outside it, centralization continues to win, either because the scale is too large for dense P2P state, or because the use is too casual to repay the onboarding cost, or because habituation overrides architectural reasoning. The framework's task is not to dethrone centralization but to make the small high-stakes envelope reliably win-able, which is what Section 5.2.2.4 validates and Section 5.4 reports. The competitive-fairness finding is the one genuinely surprising adoption driver and is foregrounded in the Discussion as a use case the prior literature has not adequately considered.

5.3.5 Folk models users bring to decentralization

If Section 5.3.4 mapped where users' reasoning is most defensible (the small high-stakes envelope they themselves bounded), this section maps where the same reasoning systematically fails. The failures are not noise. We treat them as a meta-finding: lay folk models of the comparison fail in systematic ways, in predictable directions, and each failure is architecturally targetable [Was10, KDF15]. Four architectural misconceptions and one methodological one recurred.

Brand-as-security distorts the comparison by treating corporate reputation as a security primitive. P1, P7, P15, P17 reached for brand reputation as a security signal. P1: *"I would trust a central corporation more than like some guy in Nebraska"* (P1). Brand reputation is a legibility signal, not a security signal: a well-known company can hold data badly, and an open-source signed

build can be inspected by anyone. The framework's hierarchical naming with cryptographic identity (Section 5.2.2.1) substitutes legibility-of-code for legibility-of-brand. The chapter has to name this substitution explicitly; otherwise the design choice reads as arbitrary against the brand-as-security default.

The anonymity-decentralization conflation distorts the comparison by importing centralized-platform moderation pathologies. In our coding, P1, P15, P16 talked as if decentralization implied anonymity, and as if a decentralized system would inherit the moderation pathologies of anonymous platforms. The framework's identity layer is not anonymous; it is pseudonymous with provenance: cryptographic identities are bound to out-of-band-exchanged real-world contacts, and every state mutation is signed. We read the misconception as predicting that users will reject the framework on grounds that do not actually apply to it, which is the strongest argument for naming the alternative explicitly.

Decentralized-as-unsecured distorts the comparison by treating "server removed" as "security removed". We coded P1, P7, P15, P19, P20 as expressing variations on the folk model: removing the server means everyone can access the data, or the network as a whole is harder to defend than a centralized stronghold. In the cases the framework targets the opposite holds: out-of-band-credentialed peer-to-peer transport plus CRDT-confined state restricts visibility to credentialed peers, while centralized aggregation creates a single point of compromise. The framework corrects the misconception in practice; the chapter has to correct it in prose.

Plutocratic consensus distorts the comparison by importing stake-weighted voting from cryptocurrency discourse. P1, P5, P11 named consensus from the cryptocurrency vernacular and imported the stake-weighted form by default (the proof-of-work quote in Section 5.3.3). The participant noticed the failure mode (*"Not necessarily right"*) and could not escape it. The framework's choice of one-peer-one-vote with a configurable `EVICTION_QUORUM` (Section 5.2.2.3) is a direct response, but the design choice is only legible if the chapter explains why peer-weighted voting was selected.

Self-report distortion: participants do not reliably introspect on whether they self-censor under centralized observation. We coded apparent self-contradictions on behavior-change-under-surveillance in P9, P11, and P17's responses, with P11's pair of statements occurring within the same interview; P17 articulated the attitude-behavior gap directly. The chilling effect is by definition partially unconscious [Mb14], so we read self-reported behavior change as unreliable evidence and triangulate instead against what participants felt comfortable saying once the unmonitored-P2P framing was articulated (e.g., P13 escalating from polite generalities to specific political examples mid-interview).

Synthesis. The diagnostic value of these distortions is not that participants "got the technology wrong" but that the distortions are systematic across participants of differing technical sophistication, and that each is architecturally targetable. Read individually, each distortion is a design contract the chapter has to honour: brand-as-security obliges us to make cryptographic signing legible to non-experts; the anonymity-decentralization conflation obliges us to articulate pseudonymous-with-provenance identity as the framework's actual identity model; decentralized-as-unsecured obliges us to show that credentialed CRDT confines state more tightly than centralized aggregation, not less; plutocratic consensus obliges us to justify peer-weighted voting against the cryptocurrency-vernacular default. The self-report distortion is methodological rather than architectural and warns us not to lean on self-reported behavior change as primary evidence of the framework's value; we triangulate against the content participants felt comfortable expressing once the unmonitored framing was established. Read together, the distortions are the part of the corpus with the most direct design consequence: they tell us which design choices the chapter has to explicitly name and defend, because users will otherwise import the standard architectural collapses by default and judge the framework against the wrong baseline.

Synthesis: what the findings require of the framework

Three findings translate directly into design requirements Section 5.2.2 answers. First, the convergent intuition that an immutable, peer-replicated record resolves the Accountability Paradox (P7 in Section 5.3.3) maps onto the world document (Section 5.2.2.1). Second, the permission-gated

access intuition from ten participants that we name Digital Doorbell (Section 5.3.2) maps onto workspace-scoped credentialed access. Third, the framework's auditing pipeline (Section 5.2.2.3) has one spontaneous lay articulation (P14 in Section 5.3.3), with two further post-hoc endorsements; we treat this as a design implication validated by recognition rather than a hard requirement.

Two findings impose bounded operating envelopes: the intimacy-boundary preference (Section 5.3.4) restricts the framework to small-group workspaces, and the Inconvenience Threshold constrains the friction budget Section 5.2.2.4 tests empirically. The folk-model distortions of Section 5.3.5 are diagnostic rather than dictative; they oblige Section 5.5 to articulate the design choices that anticipate each misreading.

The throughline of RQ1 is the following. Users do not want decentralization in the abstract. They want specific properties they perceive the centralized arrangement as denying them (continuity, key-based exclusivity, audit-without-surveillance, escape from peer-weaponized moderation); they want those properties only inside a sharply bounded use envelope (small-group, high-stakes, with friction below their tolerance ceiling); and they will judge any decentralized alternative against folk-model distortions the design has to explicitly refuse. The framework in Section 5.2.2 is the response to that triple constraint. We turn next, in Section 5.4, to whether the framework is empirically indistinguishable from a centralized baseline on the constructs participants told us they would not sacrifice.

5.4 Validation Results

Section 5.2.2.4 described the within-subjects, double-blind validation that asks whether participants perceive our decentralized framework as indistinguishable from a centralized baseline on the three constructs they themselves told us they would not sacrifice (Section 5.3.4). This section reports the validation's four outcomes: participants' subjective Likert ratings (Section 5.4.2), dyads' identification guesses before any debrief (Section 5.4.3), the post-session debrief observation that semantic-correctness rollbacks went unnoticed (Section 5.4.4), and per-update round-trip latency captured from the system logs of both conditions (Section 5.4.5). Under the small-group, dense-collaboration regime where Section 5.3.4 found that decentralization is asked to win, none

of the four measurements reveals a degradation participants explicitly said would disqualify the framework. Section 5.4.6 reads them together.

5.4.1 Sample

The validation cohort and procedure are described in Section 5.2.2.4. Of the 22 participants from Section 5.2.1.1, 20 opted in to the validation session and were paired into 10 dyads; the two non-returners are both from the same interview-session dyad and are excluded from all analyses below. The high opt-in rate is consistent with the recruitment design, in which participants were told at the close of the first session that the second was a paid follow-up; we therefore treat the validation cohort as a convenience subsample of the Section 5.2.1.1 cohort rather than as an independently recruited one. Because the validation cohort is drawn from the same participants who nominated the load-bearing constructs in Section 5.3, we read the Likert outcomes below as construct-valid for the population that named them rather than as evidence of generalisation to a naive population; Section 5.5.5 returns to this scope explicitly. The dyad scale matches where adjacent social-VR laboratory work concentrates (79% dyads in a recent twenty-year scoping review [SCG24]), and Section 5.5.5 returns to the workspaces-of-five-to-ten question this leaves open. The system logs were captured continuously during each session and aggregated post-hoc for the latency analysis in Section 5.4.5.

5.4.2 Subjective ratings: no evidence of a perceptual gap

Figure 5.2 reports the mean rating in each condition on the three 7-point Likert constructs defined in Section 5.2.2.4: Perceived Latency, Synchronization Consistency, and Collaborative Fluency.

Two patterns are visible. First, ratings under both conditions sit at the upper end of the scale across all three constructs (between 5.65 and 6.35 on the 7-point range), which suggests participants experienced both arrangements as collaboratively functional rather than as uniformly poor or uniformly excellent. Second, the differences between the two conditions are small in absolute terms and not statistically significant under paired tests. Decentralized was rated slightly lower on Perceived Latency (mean 5.85 vs. 6.00; paired $t(19) = -0.45$, $p = 0.66$), slightly higher on Synchronization

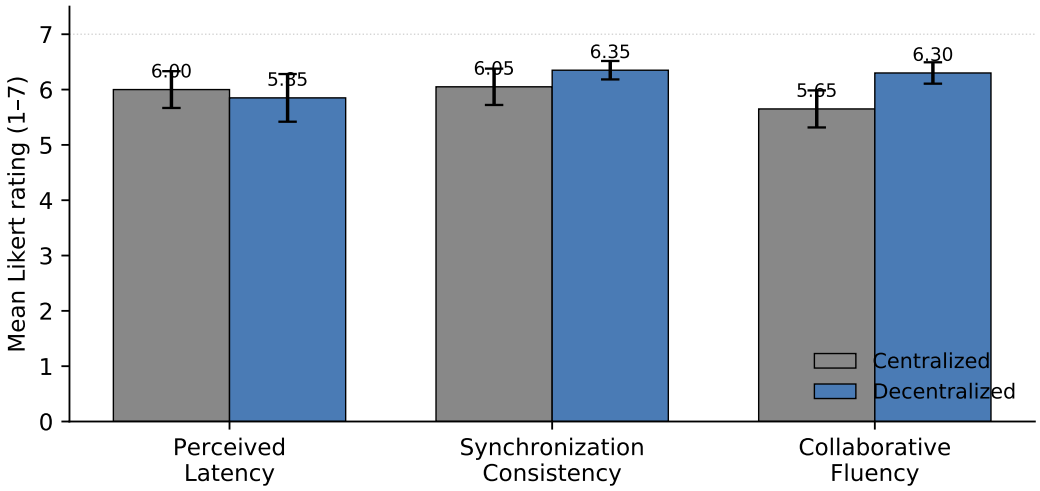


Figure 5.2: Mean Likert ratings (1–7, higher is better) on the three constructs under the centralized baseline and the decentralized framework. Error bars are ±1 standard error of the mean. None of the three pairwise differences is statistically significant ($N = 20$ paired participants). Per-construct paired effect size and 95% CI of the mean difference (D − C): Perceived Latency $d_z = -0.10$, $[-0.85, 0.55]$; Synchronization Consistency $d_z = 0.17$, $[-0.51, 1.11]$; Collaborative Fluency $d_z = 0.37$, $[-0.17, 1.47]$.

Consistency (6.35 vs. 6.05; $t(19) = 0.78$, $p = 0.45$), and slightly higher on Collaborative Fluency (6.30 vs. 5.65; $t(19) = 1.66$, $p = 0.11$). Non-parametric Wilcoxon signed-rank tests reach the same conclusion in all three cases ($p > 0.10$).

With $N = 20$ paired participants the validation is conservatively powered, so we frame the outcome as the absence of evidence for a perceptual gap, not as positive evidence of equivalence. We did not apply a multiple-comparisons correction; under Bonferroni at $\alpha = .05$ the threshold would tighten to .017, which all three reported p-values still exceed. The direction of the differences does not, however, consistently favour centralization on any construct, which is the result the Inconvenience Threshold framing of Section 5.3.4 required us to deliver: participants told us they would tolerate decentralization's friction only if it did not degrade the collaborative experience, and on the three constructs they themselves nominated as load-bearing, no degradation is detectable in our sample.

5.4.3 Architectural-invisibility check: identification did not exceed chance ($N = 10$ dyads)

The architectural invisibility test described in Section 5.2.2.4 asks each participant, before any debrief and while still blinded to the condition assignment, which of the two systems they believed was the decentralized one. Of the 10 dyads, 4 correctly identified the decentralized condition and 6 did not; an exact two-tailed binomial test does not reject the chance-level null ($p \approx 0.75$). Where dyads volunteered remarks alongside their guess, the modal pattern matched the near-chance outcome: participants reported that the two versions felt indistinguishable and that they were essentially guessing.

5.4.4 Debrief: semantic-correctness rollbacks went unnoticed

A third subjective outcome comes from the post-session debrief. By design, the decentralized condition triggered semantic-correctness rollbacks (Section 5.2.2.2, Algorithm 1) whenever a concurrent edit produced an invariant violation; the chest-opening stage of the collaborative task (Section 5.2.2.4) was deliberately engineered to provoke at least one such rollback per session, and the system logs confirm that rollbacks were in fact triggered in every decentralized session. During the structured debrief, no participant spontaneously reported noticing a rollback, an inconsistency between their own and their partner's view of the world, or any moment where the world state appeared to revert or jump. When the facilitator probed for it at the close of the debrief, no participant could identify a specific instance from memory either. We take this as positive but conservative evidence that the compensating-transition mechanism layered on top of optimistic local execution preserved perceptual continuity for the dyads it was acting on in this sample: the rollbacks happened, the framework's invariants held, and the experience did not surface to participants as a recognisable artefact.

5.4.5 Update round-trip latency: indistinguishable on the mean

To check that the subjective outcomes are grounded in the system's actual behavior rather than in a perceptual ceiling effect, we instrumented both client applications during each session to log per-update round-trip latency: the wall-clock interval at client A between issuing a local update and

observing the resulting state confirmation propagated back from client B (the A → B → A path). Both clients ran on the same in-lab network with an inter-client ping of approximately 5 ms in both conditions, so the measured value reflects the system's application-level overhead (CRDT merge, signing, validation, and the per-frame Unity loop) on top of the round-trip network cost, not the network ping itself. We aggregate the raw per-update samples across all sessions in each condition; centralized: $N = 869$ updates across six trials; decentralized: $N = 460$ updates across six sessions. Figure 5.3 reports the mean latency in each condition.

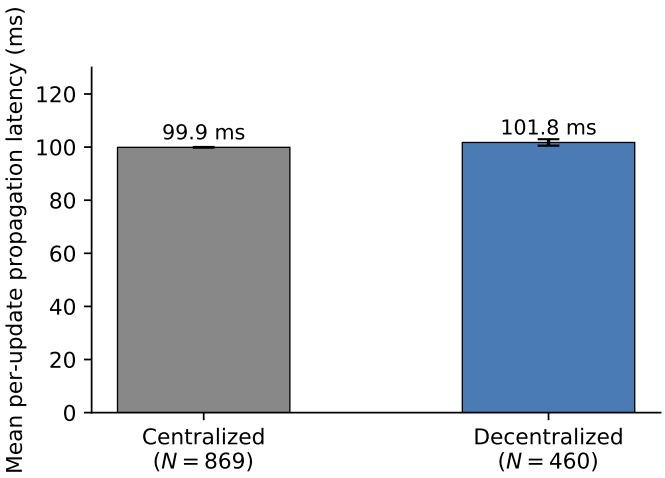


Figure 5.3: Mean per-update round-trip latency in milliseconds (A → B → A path) with ±1 standard error of the mean. Both clients run on the same in-lab network (≈ 5 ms inter-client ping). Centralized: $N = 869$ updates aggregated across six trials; decentralized: $N = 460$ updates aggregated across six sessions.

The two conditions are essentially co-located. The centralized baseline produced a mean round-trip latency of 99.9 ms ($N = 869$ updates; median = 99 ms), and the decentralized framework produced a mean of 101.8 ms ($N = 460$ updates; median = 97 ms). A Welch's t-test on the per-update means does not reach significance ($t = -1.51$, $p = 0.13$), with a negligible effect size (Cohen's $d = -0.12$). Decentralized is marginally faster at the median (97 ms vs. 99 ms), although we treat this gap as too small to interpret. The headline reading is that the decentralized framework delivers per-update round-trip latency that lands within roughly 2 ms of the centralized baseline. The ∼100 ms we observe sits at the end-to-end response-time limit at which interactive systems are

perceived as essentially instantaneous to the user [Nie93], consistent with the Likert finding that participants did not report a perceived-latency gap.

5.4.6 Synthesis

The validation's four outcomes converge on a single, conservatively framed conclusion. Subjectively, participants did not distinguish the decentralized framework from a centralized baseline on the three constructs they themselves nominated, and they did not surface the semantic-correctness rollbacks the framework triggered in every decentralized session. Across the 10 dyads providing identification guesses before any debrief, accuracy did not differ from chance (4 of 10 correct, exact binomial $p \approx 0.75$). Objectively, the system delivers per-update round-trip propagation at a mean latency within 2 ms of the centralized baseline, and a median that is in fact marginally lower. Together with the qualitative findings of Section 5.3, the validation places the framework inside the deployment surface participants drew for us in Section 5.3.4: it sits at the upper end of the friction range they self-reported as tolerable (the onboarding ratings of Section 5.3.4) without crossing into a per-update or per-construct degradation they said would disqualify it. We did not report task-completion metrics: the Inconvenience Threshold framing of Section 5.3.4 pitched the validation question as whether the decentralized framework felt collaboratively indistinguishable, not whether it was task-equivalent on rate-of-work, and the chest-opening task was engineered to provoke rollbacks rather than to be raced. Whether the result holds under larger or more diverse groups, and under heavier rollback frequencies than the chest-opening task elicited here, are open questions we return to in Section 5.5.

5.5 Discussion

Centralized metaverse platforms concentrate identity, state, and moderation in a single operator. Our participants did not reject that concentration in the abstract; in Section 5.3 they named specific consequences they wanted out of it (commercial commodification of behavioral data, the threat of arbitrary asset removal, the bind between privacy and accountability) and specific properties they wanted in its place. This chapter is one attempt to build a decentralized framework against

that list rather than against a generic "no-server" specification. Our scope throughout is metaverse-application decentralization specifically, by which we mean interactive shared 3D state under concurrent multi-user use; the interview study in Section 5.2.1, the framework in Section 5.2.2, and the validation in Section 5.4 are all scoped to this setting, and we make no general claim about decentralization as a property of internet applications at large. To our knowledge, no fully decentralized interactive collaborative metaverse system of this kind exists today: the collaborative virtual environments deployed in this setting remain server-mediated, and the most recent CSCW work on decentralized social platforms [HNS25, ZZW24] addresses asynchronous text federation rather than interactive shared state. The work in Section 5.2.2 is therefore best read as an exploration of whether such a system can deliver what our participants said they needed, and the validation in Section 5.4 is a first signal that it can at the small-group, dyadic scale our participants drew.

With that framing in mind, this section develops four implications, organised at four successive levels of analysis: the architectural concept, the user, the deployment regime, and the research methodology. At the level of the concept (Section 5.5.1), we reconsider what decentralization should mean for metaverse design and argue against reading it as a single architectural switch from "with server" to "without server"; the more productive reading is a function-by-function redistribution of distinct operator roles. At the level of the user (Section 5.5.2), we turn from the system to the people who use it and argue that the folk models users bring to decentralized metaverse systems recur predictably enough to be treated as design contracts the system has to articulate against by name. At the level of deployment (Section 5.5.3), we argue that the scope at which a decentralized metaverse system operates and the visibility of its internal coordination together decide whether its architectural advantages survive contact with users; for our framework, this resolves into a small-group regime in which the machinery can stay hidden. At the level of methodology (Section 5.5.4), we read this chapter as one instance of a research template for metaverse decentralization in which user-stated needs elicit design choices that are then examined on the constructs participants themselves named. Section 5.5.5 closes by naming what this chapter cannot answer and what could plausibly follow it.

5.5.1 Rethinking Decentralization in the Metaverse

The conventional framing of decentralization, including in much of the metaverse literature, treats it as a binary architectural choice: keep the application, remove the central operator. Our participants did not reason at that grain. They named specific things the centralized metaverse operator does for everyone (records who did what; gates who enters the space; sanctions bad actors) and asked, function by function, whether they wanted that function removed, moved onto user-held credentials, replaced by a peer process, or left in place. This function-layer reasoning has surfaced in adjacent settings as well; Hwang et al. [HNS25] report that members of decentralized social communities reason about specific stewardship functions of an instance administrator rather than about the architecture as a whole. The Accountability Paradox of Section 5.3.3 is the cleanest example in our data: participants did not ask for the operator's evidence-keeping function to disappear; they asked for it to be stripped of its privileged read access. The Digital Doorbell finding of Section 5.3.2 is another: ten of 22 participants articulated the same permission-gated access intuition (which we name Digital Doorbell after one participant's door metaphor), not for "no operator" but for an access-gating function that runs on permission they grant rather than on platform policy they did not write.

The framework's three foundational mechanisms in Section 5.2.2 are a concrete pass at this redistribution, one function at a time. The workspace abstraction with user-held credentials (G_1) takes the access-gating function off the operator and puts it on the people who own the workspace. The world document with semantic-correctness invariants (G_2) takes the function of deciding who wins a conflict off a server and puts it on a small peer-quorum process. The peer-driven auditing with quorum-based ostracization (G_3) takes the function of policing misbehaviour off the operator's privileged view and puts it on the same small peer-quorum process. What looks at the architecture layer like one choice ("no central server") becomes, at the function layer, three separate moves, each tied to a different user-stated concern.

The same function-layer reading also helps explain why a popular middle-ground in metaverse deployments, the local-host arrangement in which one client acts as the server for the others, does not really answer what our participants asked for. Section 5.3.3 reported that nine of 22 participants

explicitly rejected this option for three reasons that came up independently: the designated host gets the same kind of unfair advantage the centralized operator had (they can cheat more easily); the same host is the most exposed to attack; and the social question of who takes on the host role, and who pays them for it, becomes a fresh source of friction. The first two concerns are documented in security accounts of deployed multiplayer architectures, where listen-server arrangements give the host both unfair gameplay advantages and asymmetric exposure to attack on the host machine. In the function-by-function frame above, a local-host arrangement does not actually move any of the operator functions onto user-held credentials; it just shifts the same concentration to a peer who is not the user. Our participants seemed to read this as a different version of the problem they were trying to leave behind, rather than as a solution to it, which is one reason we use a genuine peer-to-peer substrate rather than a designated-host shortcut. The one condition under which Section 5.3.3 found the hybrid acceptable, pre-existing real-world trust in the host, is precisely the kind of trust the workspace's out-of-band invitation mechanism (Section 5.2.2.1) is built to express, so the framework keeps that property and discards the asymmetry.

The implication for metaverse-decentralization research is methodological rather than prescriptive. Our framework, we suspect, is legible to readers because it specifies, function by function, which operator role is moved and which user-stated need motivates the move. Papers on decentralized metaverse systems that report only the architectural choice (a CRDT, a peer-to-peer transport) leave it harder to tell which operator functions have actually moved and which have quietly migrated to a different concentration point (a federation administrator [HNS25], a community moderator, an instance owner). It seems plausible, though we make the observation tentatively, that work in this area would benefit from reporting a function-by-function account of what their architecture redistributes alongside the architectural description.

5.5.2 Folk Models of Decentralized Trust

Users do not approach a decentralized metaverse system as a blank slate; they bring folk models of trust, security, and authority shaped by the centralized platforms they have used for a decade or more. Section 5.3.5 reported four architectural misreadings that recurred across participants of

varying technical sophistication: brand-as-security, the anonymity-decentralization conflation, the decentralized-as-unsecured intuition, and the cryptocurrency-style plutocratic-consensus default. The standard HCI response to misreadings like these is to treat them as user-education problems and design onboarding flows that close the gap. We are inclined to think this misses the structural fact about them in the metaverse setting. The misreadings recur across populations and technical sophistication levels [Was10, KDF15, MDD24], which leads us to treat them less as local confusions to be taught around and more as design contracts a metaverse framework has to articulate against by name.

Each contract is a short sentence the system has to say, anchored to something concrete the framework already does. Brand-as-security can be addressed without "trusted developer chains" or other vocabulary the chapter does not actually develop: metaverse workspaces are populated by people the user invited (Section 5.2.2.1, picking up the Digital Doorbell of Section 5.3.2), and membership is something a user can see directly, which is a different shape of trust signal than a corporate logo. The anonymity-decentralization conflation is addressed by the fact that the framework's identities come from out-of-band contact between people who already know each other rather than from anonymous discovery on an open network; the system is decentralized but not anonymous. The decentralized-as-unsecured intuition is addressed by workspace scoping: state changes flow inside the Sync group only, so a non-member cannot see what is going on, which leaves the framework's state visibility narrower than a typical centralized aggregation rather than wider. The plutocratic-consensus default is addressed by the eviction protocol counting independent peers rather than weighting by compute or stake (Section 5.2.2.3), so well-resourced adversaries do not get a louder vote simply by paying for it.

The implication for the metaverse subfield follows from how predictable the misreadings turn out to be: decentralized metaverse systems will be read against the wrong baseline if their prose does not say which architectural collapses they are not. The four refusals above are short to write down and easy to leave out. We suspect that articulating these contracts as a routine matter, in the same spirit as usable-security papers articulate threat models, would help future systems be evaluated against the baseline they actually offer rather than the one a reader brings.

5.5.3 Scope, Scale, and Architectural Visibility

A decentralized metaverse system's design implications are bounded by the scale at which it is deployed and the visibility of the coordination work it does. Two findings from our exploration combine, along exactly these two dimensions, into a single deployment principle. Section 5.3.4 reported that participants drew the decentralization envelope narrowly: small, intimate, known-peer settings, bounded by the social trust an out-of-band invitation can carry. Section 5.4.4 reported that the framework's compensating-transition mechanism (Section 5.2.2.2) triggered semantic-correctness rollbacks in every decentralized session and that no participant surfaced a single one. The two findings are not independent: they describe the same regime from two sides.

The participants' side of this regime is a sizable rather than marginal slice of deployed metaverse activity. Cooperative crafting and building among a handful of known players (the kind of activity our user study most directly elicited, also characteristic of small-realm hosting), collaborative virtual-world creation among already-acquainted creators, family-scale shared multiplayer experiences, and small-team remote collaboration in shared 3D workspaces all sit inside the envelope our participants drew. The adjacent research literature in the more bandwidth-intensive modality of social VR concentrates at the same scale: a recent twenty-year scoping review reports that 79% of laboratory studies in that adjacent space operate with dyads and 16% with triads [SCG24]. Spatial-privacy concerns raised in immersive systems [HWN24] and embodied-attack concerns documented in social-VR research [BEE19b, CBY21, LZK25] sharpen rather than diminish the case for a small-group decentralized arrangement: the workspace as a permissioned room, evidence kept without an operator reading it, and intimate-scale state consistency are particularly decisive where centralized space-provider concentration is most uncomfortable. We think this should be foregrounded as the shape of the contribution rather than apologised for as a limitation.

The system's side of the same regime is what lets the framework's internals stay hidden. The observation in Section 5.4.4 that participants did not notice the rollbacks is, we think, the operational form of a more general principle: if the user-visible experience has to compete with a centralized baseline, the architectural mechanics that produce it benefit from staying out of sight. Systems that surface those mechanics directly (sync states, consensus rounds, internal warnings) tend to pay

a friction cost that is not load-bearing for the user. We do not claim that scale and architectural invisibility cannot be combined, but they are hard to combine: at larger group sizes the per-update variance, the credential-exchange overhead, and the audit-quorum latency all become harder to keep out of the user's awareness. The small-group regime is the one in which the principle has room to live, and we think future decentralized-metaverse work, ours included, would benefit from bounding its claims to that regime rather than promising more.

5.5.4 Grounding System Design in User-Stated Needs

The methodological move this chapter most wants to make explicit is the alignment between empirical findings, design mechanisms, and validation outcomes. This chapter offers one demonstration that a decentralized metaverse framework can be built mechanism by mechanism from user-stated needs and examined, at the dyad scale, on the constructs participants nominated as load-bearing: the framework's design choices in Section 5.2.2 are grounded mechanism by mechanism in specific findings from Section 5.3, and Section 5.4 reports that paired tests on those same constructs did not reach statistical significance, an outcome we read under the absence-of-evidence framing of Section 5.4.2. Hwang et al.'s recent CSCW work on decentralized social platforms [HNS25] reads as a sibling effort in adjacent territory, where user-stated trust frictions motivate a set of design moves for the federated text setting.

Three alignments are worth following concretely, each connecting a user-study finding to a framework mechanism and, where the validation can speak to it, to a validation outcome.

End-to-end alignment. The intimacy preference and concurrent-collaboration requirement surfaced in Section 5.3.4 motivated the compensating-transition layer of Section 5.2.2.2; Section 5.4.4 reports that the rollbacks this layer produced, in every decentralized session, did not register as artefacts to the participants experiencing them. This is the needs → design → validation chain in its tightest form: a stated user need, a mechanism that responds to it, and a validation outcome consistent with the mechanism not surfacing to the user as a recognisable artefact in this sample.

Design alignment with supporting validation. The Digital Doorbell of Section 5.3.2 motivated the workspace and out-of-band-credential design of Section 5.2.2.1; Section 5.4.2 reports that participants' Collaborative Fluency ratings did not differ significantly between the two conditions (decentralized $M = 6.30$ vs. centralized $M = 5.65$, $p = 0.11$), consistent with the access-gating work the substrate was doing not interrupting the participants' sense that the collaboration was working.

Design refusal of a user-named failure mode. The Accountability Paradox of Section 5.3.3 and the plutocratic-consensus folk model of Section 5.3.5 jointly motivated Section 5.2.2.3; one participant (P14) reached for the same peer-audit construct unprompted, before being shown ours, and two further participants (P19, P20) endorsed it once it was demonstrated in the post-task debrief. The framework's audit pipeline operationalises that user intuition while refusing the cryptocurrency-vernacular collapse the same data exposed. We take this as a useful existence proof for the construct, alongside an explicit design refusal of the failure mode the same data named.

What we want to highlight from these three alignments is the shape of the exploration that produced them: a small set of foundational mechanisms grounded in user-stated needs from a qualitative metaverse study, examined against a centralized metaverse baseline on the constructs participants themselves nominated, with the design choices traceable through every step of the argument. This shape is what makes the specific framework we built useful to extend, replace, or argue against, rather than only an artifact to be inherited. The framework in Section 5.2.2 is one instance; we hope follow-on metaverse-decentralization work treats it as a starting point for replacing, extending, or arguing against its specific choices, while keeping the needs-design-validation alignment as the common ground on which the conversation proceeds.

5.5.5 Limitations and Future Work

The work has limitations the reader should hold against its claims. We are not claiming the framework is the only possible answer to the user-study findings, and we are not claiming the answers we report are final. The user-study sample (22 participants in Section 5.2.1.1, validation

cohort of 20 in Section 5.4.1) was recruited as a convenience sample from a single research-active university; the findings are scoped to that population. With $N = 20$ paired participants and no formal equivalence test (e.g., TOST), the validation's negative result is best read as an absence of evidence for a perceptual gap rather than positive equivalence; under higher power the Collaborative Fluency comparison ($M_D = 6.30$ vs. $M_C = 5.65$, $p = 0.11$, direction favouring decentralized) could yet shift, and the three pairwise tests were not corrected for multiple comparisons. The architectural invisibility test pre-registered in Section 5.2.2.4 is reported in Section 5.4.3, but at 10 dyads the dyad-level binomial test is itself conservatively powered, and a single dyad's identification flip would meaningfully shift the point estimate. The collaborative task surfaced rollbacks heavily by design; lighter usage patterns may surface artefacts this task masked. The framework was evaluated at the dyad scale, the smallest metaverse group at which its mechanisms can be exercised; the `EVICTION_QUORUM` = 3 path in Section 5.2.2.3 was therefore structurally unreachable in this validation, and whether the framework's properties hold at workspace sizes of five to ten peers is an empirical question this chapter does not answer.

Four lines of future work follow naturally.

Adversarial exposure of the audit pipeline. Section 5.2.2.3 responds to the Accountability Paradox and the plutocratic-consensus folk model, but those responses were not put under adversarial pressure in our validation. Plant-and-detect studies, in which a confederate participant performs constraint-violating behavior while genuine participants run the framework, would let us check whether the peer-audit construct holds up against the kind of misuse multiplayer and social-metaverse platforms have documented at scale [BEE19b, LZK25].

Domain-invariant extension. Section 5.2.2.3 left the domain-invariant set deliberately small and the open-source extension surface explicit; whether community contribution can populate that surface at the pace adversarial behavior evolves is an open empirical question rather than an engineering exercise.

Cross-workspace identity portability. The workspace-local credentials the framework currently uses do not yet let a user carry a metaverse identity across workspaces in a way that preserves both privacy and continuity; this is a precondition for the hybrid arrangement Section 5.3.4 gestured at, in which small high-stakes decentralized hubs sit alongside, rather than replace, the familiar centralized metaverse infrastructure people already use casually.

Metaverse-application instantiation. The small-group regime mapped out in Section 5.5.3 corresponds directly to the deployed use cases listed there. The more spatially sensitive variant of the same regime, immersive virtual environments where the centralized space-provider model has been a documented source of spatial-deception attacks [CBY21, YLB24, HWN24], is a parallel candidate setting where the framework could be specialised. Each of these is a place a future research effort could test a different cell of the design contracts we listed in Section 5.5.2.

What we hope readers take from this exploration is less the specific framework we built and more the move it makes. Decentralized metaverse systems, we suspect, are best specified by which operator functions they redistribute, defended by which folk-model collapses they refuse to inherit, and bounded by the small-group regime that lets their machinery stay hidden. The framework in Section 5.2.2 is one such redistribute-refuse-bound move; we hope its specifics invite both extension and disagreement.

5.6 Chapter Summary

This chapter addressed the structural root cause of the vulnerabilities identified in Chapters 3 and 4: the centralized data collection and moderation model that is standard in Metaverse applications. Rather than treating decentralization as a binary architectural switch from "with server" to "without server," we framed it as a function-by-function redistribution of distinct operator roles (recording, gating, sanctioning, vouching), each tied to a specific user-stated concern surfaced by our 22-participant interview study. The resulting framework operationalizes three foundational mechanisms: a Standardized State Abstraction Layer with workspace-scoped cryptographic identity over Named Data Networking (G_1), a peer-side semantic-correctness pipeline with compensating state transitions

(G_2), and a peer-driven quorum-based auditing pipeline (G_3). Together they show that collaborative, immersive multi-user experiences can function without routing user data through a central authority. A within-subjects, double-blind validation against a centralized baseline returned absence of evidence for a perceptual gap on the three constructs participants themselves nominated as load-bearing, identification at chance, semantic-correctness rollbacks that triggered in every decentralized session yet went unnoticed by every participant, and per-update round-trip latency within 2 ms of the centralized baseline. Collectively, these results place the framework inside the small-group, dense-collaboration regime participants drew as decentralization's high-value deployment surface.

This concludes the first research line of this dissertation, which addressed the Spatial Interaction Pillar through three stages: threat discovery (Chapter 3), root-cause diagnosis (Chapter 4), and architectural reconstruction (this chapter). The next research line shifts attention to the second foundational component of the Metaverse: the Cognitive Intelligence Pillar. While the preceding chapters examined how the act of interaction leaks information, the following chapters ask a different question: how much sensitive information do the ML models that process this data retain and expose? Chapter 6 begins this investigation by auditing the privacy behavior of predictive spatial-temporal models used in location-based services.

CHAPTER 6

Auditing Predictive Models: Privacy Leakage in Spatial-Temporal Intelligence

Having addressed the systemic vulnerabilities and structural power dynamics of the **Spatial Interaction Pillar** in the preceding chapters, this dissertation now shifts its focus to the second foundational component introduced in Chapter 1: the **Cognitive Intelligence Pillar**. While the interaction layer governs user kinematics and perception, the analytical core of the Metaverse relies on sophisticated, centralized machine learning (ML) models. In this chapter, we explore the privacy boundaries of the predictive spatial-temporal models that drive personalized Metaverse experiences.

Specifically, immersive XR applications increasingly rely on Location-Based Services (LBS) to seamlessly anchor digital content to physical coordinates and predict future user behaviors. These services encompass functionalities ranging from traffic management [BYL20, LMH22] and store location selection [LLL17] to Point-of-Interest (POI) recommendations [SQC20, YLZ22]. Among these, POI recommendation has gained immense traction[1], being integrated into ubiquitous platforms like Yelp and Google Maps to suggest relevant future destinations based on personal preferences [IMD20].

As these LBS ingest vast quantities of high-dimensional human mobility data to train and optimize ML models, they unlock unprecedented user convenience but simultaneously introduce severe privacy risks. Mobility data intrinsically contains highly sensitive identity markers, such as specific home addresses, work locations, social relationships, and trip purposes [BE09]. While a significant body of research exists regarding mobility data privacy [GL05, Kru07, ABC13, STP13], these studies primarily focus on protecting data during the transmission and release processes.

[1]From 2017 to 2023, more than 111 papers were published on POI recommendation built upon mobility data collected by location service providers [WSZ23a].

Previous efforts have demonstrated how mobility data can be linked via side channels—such as social networks [HSS13, HHB18], open-source datasets [GKP14, PB23], and network packets [JWH07, VHB14]—leading to data aggregation defenses [GL05, MR09, BCP14].

However, these traditional defense mechanisms overlook a critical, modern threat vector: the privacy risks inherent to the mobility-based ML models themselves. Specifically, adversaries can extract sensitive attributes directly from the ML models trained on this data, which are inherently susceptible to memorization [SSS17b, CCN22]. There remains a critical lack of empirical privacy risk evaluation regarding how POI recommendation models implicitly leak the data they digest.

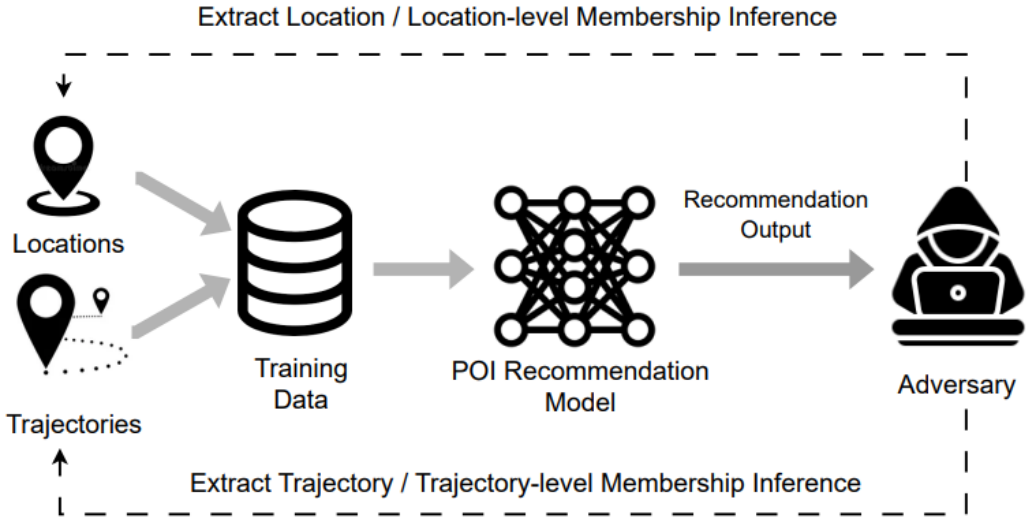


Figure 6.1: Privacy attack suite against POI recommendation models. We demonstrate that an adversary can extract or infer membership information of locations or trajectories in the training dataset.

Evaluating privacy risks in POI recommendation models is highly challenging, as existing attack and defense mechanisms are rendered ineffective by the unique features of mobility data. Previous model-centric privacy attacks have predominantly targeted image and text classifiers [SSS17b, FJR15b, CLE19], where individual data points possess high semantic uniqueness. In contrast, mobility data (e.g., discrete geographic coordinates) lacks semantic uniqueness without broader context. Furthermore, mobility data is multimodal, combining spatial and temporal dimensions to describe continuous behavioral patterns over time. Existing privacy attacks fail to reconstruct this meaningful context, resulting in failure when applied to spatial-temporal models. Consequently, the

efficacy of traditional defense mechanisms [ACG16, SCZ22, SCL22]—designed for image or text tasks—remains largely unknown when applied to POI recommendation.

To rigorously quantify the privacy leakage inherent to these centralized algorithmic services, we designed a comprehensive privacy attack suite specifically tailored for POI recommendation systems. As illustrated in Figure 6.1, our suite utilizes both data extraction and Membership Inference Attacks (MIA) to assess vulnerabilities at both the **location** and **trajectory** levels. By operating under varying adversary knowledge assumptions, our attacks successfully extract different typologies of sensitive information, providing a holistic privacy risk assessment.

Our empirical evaluation, conducted on three representative POI models across two real-world mobility datasets, demonstrates that state-of-the-art recommendation models implicitly memorize and expose sensitive user trajectories. Furthermore, we provide an in-depth analysis to characterize which specific types of mobility data are most susceptible to leakage. Notably, we discover an "outlier effect" within POI privacy attacks, where training examples associated with specific user behaviors or rare trajectories are disproportionately vulnerable. Finally, we evaluate potential defensive countermeasures, highlight remaining structural challenges, and outline future directions for privacy-preserving spatial-temporal intelligence. To facilitate ongoing research and reproducibility, our complete attack suite has been open-sourced at https://github.com/KunlinChoi/POIPrivacy.

By bridging the gap between advanced model-centric privacy attacks and spatial-temporal intelligence, this chapter systematically audits the cognitive intelligence boundaries of predictive Metaverse services. The core contributions are summarized as follows:

- **Tailored Privacy Attack Suite for Mobility Data:** We introduce a novel privacy attack suite that integrates the unique multimodal characteristics of mobility data (i.e., spatial-temporal context) into the attack design. To the best of our knowledge, this work is the first to comprehensively evaluate privacy risks in neural-network-based POI recommendation models via inference attacks at both the location and trajectory levels.
- **Empirical Quantification of Model Memorization:** Through extensive experiments on state-of-the-art POI models and real-world datasets, we conclusively demonstrate that spatial-temporal

intelligence models are highly vulnerable to both data extraction and membership inference attacks.

- **Identification of Vulnerability Factors:** We provide an in-depth empirical analysis detailing the unique factors in mobility data that drive vulnerability. Specifically, we expose the "outlier effect," demonstrating how atypical locations and user trajectories exacerbate privacy leakage.
- **Evaluation of Countermeasures:** We test existing defense mechanisms and prove that they fail to effectively thwart our attacks without inducing unacceptable utility loss, thereby identifying critical challenges and outlining future directions for developing fundamentally privacy-preserving POI recommendation models.

6.1 Point-of-Interest Recommendation

POI recommendation has recently gained much attention due to its importance in many business applications [IMD20], such as user experience personalization and resource optimization. Initially, researchers focused on feature engineering and algorithms such as Markov chain [ZCL14, CLY14], matrix factorization algorithms [LZX14, CYL13], and Bayesian personalized ranking [HLL17, ZZY16] for POI recommendation. However, more recent studies have shifted their attention towards employing neural networks like RNN [LWW16, YFR20], LSTM [KW18, SQC20], and self-attention models [LLL21, LWG20]. Neural networks can better learn from spatial-temporal correlation in mobility data (e.g., check-ins) to predict users' future locations and thus outperform other POI recommendation algorithms by a large margin. Meanwhile, this could introduce potential privacy leakage. Thus, we aim to design an attack suite to measure the privacy risks of neural-network-based POI recommendations systematically.

We first provide the basics of POI recommendations and notations used throughout this chapter. Let $\mathcal{U}$ be the user space, $\mathcal{L}$ be the location space, and $\mathcal{T}$ be the timestamp space. A POI recommendation model takes the observed trajectory of a user as input and predicts the next POI that will be visited, which is formulated as $f_\theta : \mathcal{U} \times \mathcal{L}^n \times \mathcal{T}^n \rightarrow \mathbb{R}^{|\mathcal{L}|}$. Here, the length of the input trajectory is n. We denote a user by its user ID $u \in \mathcal{U}$ for simplicity. For an input

trajectory with n check-ins, we denote its trajectory sequence as $x_T^{0:n-1} = \{(l_0, t_0), \ldots, (l_{n-1}, t_{n-1})\}$, where $l_i \in \mathcal{L}$ and $t_i \in \mathcal{T}$ indicate the POI location and corresponding time interval of i-th check-in. Also, the location sequence of this trajectory is denoted as $x_L^{0:n-1} = \{l_0, \ldots, l_{n-1}\}$. The POI recommendation model predicts the next location l_n (also denoted as y by convention) by outputting the logits of all the POIs. Then, the user can select the POI with the highest logit as its prediction $\hat{y}$, where $\hat{y} = \arg\max f_\theta(u, x_T^{0:n-1})$. Given the training set D_{tr} sampled from an underlying distribution $\mathcal{D}$, the model weights are optimized to minimize the prediction loss on the overall training data, i.e., $\min_\theta \frac{1}{|D_{\text{tr}}|} \sum_{(u, x_T^{0:n-1}, y) \in D_{\text{tr}}} \ell(f_\theta(u, x_T^{0:n-1}), y)$, where ℓ is the cross-entropy loss, i.e., $\ell(f_\theta(u, x_T^{0:n-1}), y) = -\log(f_\theta(u, x_T^{0:n-1}))_y$. The goal of the training process is to maximize the performance of the model on the unseen test dataset $D_{te} \in \mathcal{D}$, which is drawn from the same distribution as the training data. During inference, this prediction $\hat{y}$ is then compared to the next real location label l_n to compute the prediction accuracy. The performance evaluation of POI recommendation models typically employs metrics such as top-k accuracy (e.g., $k = 1, 5, 10$).

6.2 Threat Models

Adversary Objectives. To understand the potential privacy leakage of training data in POI recommendation models, we design the following four attacks from the two most common privacy attack families: membership inference attack [SSS17a] and data extraction attacks [CTW21], based on the characteristics of the mobility data for POI recommendation, namely *common location extraction* (LocExtract), *training trajectory extraction* (TrajExtract), *location-level membership inference attack* (LocMIA), and *trajectory-level membership inference attack* (TrajMIA). These four attacks aim to extract or infer different sensitive information about a user in the POI recommendation model training data.

LocExtract focuses on extracting a user's most frequently visited location; TrajExtract extracts a user's location sequence with a certain length given a starting location; LocMIA infers whether a user has been to a location and used for training; TrajMIA infers where a trajectory sequence has been used for training.

Adversary Knowledge. For all attacks, we assume the attacker has access to the query interface of

the victim model. Specifically, the attacker can query the victim model with the target user and obtain the corresponding output logits. This assumption is realistic in two scenarios: (1) A malicious third-party entity is granted access to the POI model query API hosted by the model owner (e.g., location service providers like Foursquare or Yelp) for specific businesses such as personalized advertisement. This scenario is well-recognized by [Shu22, Mik21, XLX21]. (2) The retention period of the training data expires. Still, the model owner keeps the model and an adversary (e.g., a malicious insider of location service providers) can extract or infer the sensitive information using our attack suite, even if the training data have been deleted. In this scenario, the model owner may violate privacy regulations such as GDPR [Eur16].

Depending on different attack objectives, the adversary also possesses different auxiliary knowledge. In particular, for TrajExtract, we assume the attacker can query the victim model with a starting location l_0 that the target user visited. This assumption is reasonable because an attacker can use real-world observation [VFB11, SH12], LocExtract, and LocMIA as cornerstones. As for LocMIA and TrajMIA, we assume the attacker has access to a shadow dataset following the standard settings of membership inference attacks [SSS17b, CCN22].

6.3 Attack Suite

Our privacy attack suite includes the two most prevalent types of privacy attacks on machine learning models: data extraction and membership inference attacks. These are used to evaluate the privacy vulnerabilities of POI recommendation models at both location and trajectory levels. The subsequent sections detail the technical approaches and design of attacks, taking into account the unique aspects of mobility data.

6.3.1 Data Extraction Attacks

Our data extraction attacks are rooted in the idea that victim models display varying levels of memorization in different subsets of training data. By manipulating the spatial-temporal information in the queries, the attacker can extract users' locations or trajectories that these victim models

predominantly memorize.

LocExtract Common location extraction attack (LocExtract) aims to extract a user's most frequently visited location in the victim model training, i.e.,

$$\text{LocExtract}(f_\theta, u) \rightarrow \hat{l}_{top1}, \ldots, \hat{l}_{topk}.$$

The attack takes the victim model f_θ and the target user u as the inputs and generates k predictions $\hat{l}_{top1}, \ldots, \hat{l}_{topk}$ to extract the most frequently visited location of user u. The attack is motivated by our key observation: querying POI recommendation models with a random location reveals that these models tend to "over-learn" a user's most frequently visited locations, making these locations more likely to appear in the model output. For example, we randomly choose 10 users and query the victim model using 100 randomly selected locations. Of these queries, 32.5% yield the most frequent location for the target user. Yet, these most common locations are present in only 18.7% of these users' datasets.

In LocExtract, we first generate a set of different random inputs for a specific user and use them to make iterative queries to the victim model. Each query returns the prediction logits with a length of $|\mathcal{L}|$ outputted by the victim model. The larger the logit value, the more confident the model is in predicting the corresponding location as the next POI. Therefore, by iterating queries to the model given a target user and aggregating the logit values of all queries, the most visited location is more likely to have a large logit value after aggregation. Here, we use a soft voting mechanism, i.e., averaging the logit values of all queries, as the aggregation function (see also Sec. 6.4.5 for the comparison with different aggregation functions). With the resulting mean logits, we output the top-k locations with k largest logit values as the attack results. Algorithm 4 gives the outline of LocExtract. Though the attack is straightforward, it is effective and can be a stepping stone for TrajExtract in our attack suite.

TrajExtract Our training trajectory extraction attack (TrajExtract) aims to extract the location sequence $x_L^{0:n-1} = \{l_0, \ldots, l_{n-1}\}$ in a training trajectory of user u with a length of n from the victim model f_θ. Formally,

$$\text{TrajExtract}(f_\theta, u, l_0, n) \rightarrow \hat{x}_{L_0}^{0:n-1}, \ldots, \hat{x}_{L_\beta}^{0:n-1},$$

where $\hat{x}_{L_0}^{0:n-1}, \ldots, \hat{x}_{L_\beta}^{0:n-1}$ indicate the top-β extracted location sequences by the attack.

The key idea of the training trajectory extraction attack is to identify the location sequence with the lowest log perplexity, as models tend to demonstrate lower log perplexity when they see trained data. We denote log perplexity as:

$$\mathrm{PPL}_{f_\theta}(u, x_T^{0:n-1}) = -\log \Pr_{f_\theta}(u, x_T^{0:n-1}) = -\sum_{i=0}^{n-1} \log \Pr_{f_\theta}(u, x_T^{0:i-1}),$$

where $\Pr_{f_\theta}(\cdot)$ is the likelihood of observing $x_T^{0:n-1}$ with user u under the victim model f_θ. In order to get the lowest log perplexity of location sequences with a length of n, we have to enumerate all possible location sequences. However, in the context of POI recommendation, there are $O(|\mathcal{L}|^{n-1})$ possible location sequences for a given user. $|\mathcal{L}|$ equals the number of unique POIs within the mobility dataset and can include thousands of options. Thus, the cost of calculating the log perplexity of all location sequences can be very high. To this end, we use beam search to extract the location sequences with both time and space complexity $O(|\mathcal{L}| \times n \times \beta)$, where β is the beam size. In particular, to extract a trajectory of length n, we iteratively query the victim model using a set of candidate trajectories with a size of β and update the candidate trajectories until the extraction finishes. As highlighted in the prior work [FLD18], when using beam search to determine the final outcome of a sequential neural network, there is a risk of generating non-diverse outputs and resembling the training data sequence. However, in our scenario, this property can be leveraged as an advantage in TrajExtract, as our primary objective revolves around extracting the training location sequence with higher confidence. As a final remark, both LocExtract and TrajExtract need a query timestamp to query the victim model, and we will show the effects of the timestamp in our experiments. Algorithm 5 in Section 6.3.4 gives the detailed steps of TrajExtract.

6.3.2 Membership Inference Attacks

Membership inference attack (MIA) aims to determine whether a target data sample is used in the model training. We extend the notion to infer whether certain sensitive information (e.g., user-location pair (u, l) and trajectory sequence (u, x_T)) of the user's data is involved in the training of the victim model f_θ. Since POI recommendation models use multi-modal sequential data as

inputs and adversaries lack sufficient information to construct a complete input, we propose attack designs to manipulate the spatial-temporal information in queries to enhance effectiveness of attacks. The membership inference attack can be formulated as follow:

$$\text{MIA}(f_\theta, X_{target}, D_s) \rightarrow \{\text{member}, \text{nonmember}\},$$

where X_{target} represents the target sensitive information ($X_{target} = (u, l)$ in LocMIA and $X_{target} = (u, x_T)$ in TrajMIA), and D_s is the shadow dataset owned by the adversary.

To effectively infer the membership of a given X_{target}, we adapt the state-of-the-art membership inference attack – likelihood ratio attack (LiRA) [CCN22] to the context of POI recommendation. The key insight of LiRA is that the model parameters trained with X_{target} differ from those trained without it, and the effect of the model parameter on a data sample can be well approximated using a loss value. By conducting a hypothesis test on the distributions of the loss values, we can identify if the victim model is trained with the X_{target} or not. LiRA consists of four steps: (1) train multiple shadow models, (2) query the shadow models trained with X_{target} and without X_{target} to obtain two distributions, (3) query the victim model X_{target} to obtain the output logits, and (4) conduct a Λ hypothesis test to infer the membership of the X_{target} based on the two distributions and the query results. We present the full LiRA procedure in Section 6.3.4.

LocMIA In this attack, the adversary aims to determine whether a given user u has visited a location l in the training data. However, it is not feasible to directly apply LiRA to LocMIA as the victim model takes the trajectory sequences as inputs, but the adversary only has a target location without the needed sequential context. In particular, LocMIA needs the auxiliary inputs to calculate the membership confidence score since this process cannot be completed only using $X_{target} = (u, l)$. This attack is a stark contrast to MIA for image/text classification tasks where the X_{target} itself is sufficient to compute the membership confidence score.

To this end, we design a spatial-temporal model query algorithm (Algorithm 6 in Section 6.3.4) to tailor LiRA to LocMIA and optimize membership confidence score calculation. The idea behind the algorithm is that if a particular user has been to a certain POI location, the model might "unintentionally" memorize its neighboring POI locations and the corresponding timestamp in the training data. Motivated by this, each time we query the models (e.g., the victim and shadow

models), we generate n_l random locations and n_t fixed-interval timestamps. To obtain stable and precise membership confidence scores, we first average the corresponding confidence scores at the target location by querying with n_l locations at the same timestamp. While the adversary does not possess the ground truth timestamp linked with the target POI for queries, the adversary aims to mimic a query close to the real training data. To achieve this, we repeat the same procedure of querying different locations for n_t timestamps and take the maximum confidence scores among the n_t averaged confidence scores as the final membership inference score for the target example. Algorithm 7 gives the outline of LiRA in terms of LocMIA, and the lines marked with red are specific to LocMIA.

TrajMIA The attack aims to determine whether a trajectory is used in the training data of the victim model. Unlike LocMIA, $X_{target} = (u, x_T)$ suffices to calculate the membership confidence score in LiRA, and we do not need any auxiliary inputs. To fully leverage information of the target example querying the victim model and improve the attack performance, we also utilize the $n-2$ intermediate outputs and the final output from the sequence x_T with a length of n to compute the membership confidence score, i.e., we take the average of all $n-1$ outputs. This change improves the attack performance as the intermediate outputs provide additional membership information for each point in the target trajectory. The purple lines in Algorithm 7 highlight steps specific to TrajMIA.

6.3.3 Practical Implications of the Attack Suite

Our attack suite is designed as an integrated framework focusing on the basic units of mobility data – locations and trajectories. It contains two prevalent types of privacy attacks: data extraction and membership inference attacks. Each attack in our attack suite targets a specific type of mobility data and could serve as a privacy auditing tool [JUO20]. They can also be used to infer additional sensitive information in mobility data:

- LocExtract extracts a user's most common location. Combined with the semantics of the POI, we may infer the user's address such as work address, which is closely related to user identity;
- TrajExtract can be further used to infer user trajectories and identify trip purposes by analyzing the POIs visited during a journey [MCH17];

Table 6.1: Top-k prediction accuracy of the three POI-recommendation victim models on the two datasets.

Dataset	Model	Top-1 ACC	Top-10 ACC
4SQ	GETNEXT	0.34	0.71
	LSTPM	0.25	0.67
	RNN	0.24	0.68
GOWALLA	GETNEXT	0.16	0.48
	LSTPM	0.15	0.39
	RNN	0.10	0.26

- LocMIA can determine the membership of multiple POIs, thereby facilitating the inference of a user's activity range and social connections in [CML11, RHL23];
- TRAJMIA infers if a user's trajectory is in the training dataset, which can serve as an auditing tool to examine the privacy leakage by assuming a worst-case adversary.

6.3.4 Attack Algorithms

TRAJEXTRACT To start with, we initialize β candidate trajectories with the same starting location l and the query time t given a user (lines 1-3). Next, we iteratively extend the candidate trajectories: to extract the i-th ($i \geq 1$) locations of $\hat{x}_T^{0:i-1}$ in β candidate trajectories, we query the model using $\hat{x}_T^{0:i-1}$ and compute the log perplexity for $\beta \times \mathcal{L}$ possible new trajectories with a length of $i+1$. We then choose the β trajectories with the lowest log perplexity as the new candidate trajectories in the next iteration (line 6). The iterations end until the length of the candidate trajectories reaches n. Lastly, we take the location sequences from the final trajectories. Note that both LOCEXTRACT and TRAJEXTRACT need the timestamp t to query the victim model, and we will show the effects of timestamp t in our experiments.

LiRA The key idea of LiRA [CCN22] is that models trained with or without X_{target} (i.e., f_{in} and f_{out}) would produce different loss distributions for X_{target}. Specifically, LiRA consists of four main steps:

(1) Train an equal number of shadow models *with* and *without* X_{target} using the shadow dataset owned by the attacker to simulate the behavior of the black-box victim model.

(2) Use X_{target} to query each shadow model to obtain loss values. The obtained loss values will be used to calculate two loss distributions, depending on whether the queried shadow model was trained on X_{target}.

(3) Query the victim model with X_{target} to obtain the output logits.

(4) Conduct a Λ hypothesis test to infer the membership of the X_{target}. The output score Λ indicates whether the output from the victim model is closer to one of the two loss distributions.

LiRA originally trained $2N$ shadow models for each target example. However, this approach suffers from computational inefficiency when the number of target examples is large. To address this issue, we employ the parallelized approach described in [CCN22], which reuses the same set of $2N$ shadow models for inferring the membership of multiple X_{target}. Our attacks further extend the key concept of LiRA and make it feasible for POI recommendation models by incorporating spatial-temporal information through unique designs, such as our spatial-temporal model query algorithm.

Algorithm 4 Common Location Extraction Attack

Require: Victim model: f_θ, target user: u, query budget: q, query timestamp: t, output size: k

Ensure: Top-k predictions: $[\hat{l}_{top1},\ldots,\hat{l}_{topk}]$

logits $\leftarrow \{\}$
for q times **do**
    $l \leftarrow$ RandomSample($\mathcal{L}$) ▷ Randomly generate a location from the location space
    logits $\cup\ f_\theta\big(u, \{(l,t)\}\big)$
end for
$\text{logits}_{\text{agg}}$ = Aggregate(logits) ▷ Aggregate confidence for all locations
return $\hat{l}_{top1},\ldots,\hat{l}_{topk} \leftarrow \text{Argmax}_k(\text{logits}_{\text{agg}})$

Algorithm 5 Training Trajectory Extraction Attack

Require: Victim model: f_θ, target user: u, starting location: l_0, target extraction length: n, query timestamp: t, beam width: β

Ensure: Top-β possible extraction results: $\hat{x}_{L_0}^{0:n}, \ldots, \hat{x}_{L_\beta}^{0:n}$

for $b \leftarrow 0$ to $\beta - 1$ **do**

  $\hat{x}_{T_b}^{0:0} \leftarrow (u, (l_0, t))$ ▷ Initialize the beam with l_0 and t

end for

for $i \leftarrow 1$ to $n - 1$ **do**

  for $\hat{x}_T^{0:i-1}$ in $\{\hat{x}_{T_0}^{0:i-1}, \ldots, \hat{x}_{T_\beta}^{0:i-1}\}$ **do**

    $\{\hat{x}_{T_0}^{0:i}, \ldots, \hat{x}_{T_\beta}^{0:i}\} \leftarrow$ UpdateBeam$_\beta(f_\theta(u, \hat{x}_T^{0:i-1}))$ ▷ Update the beam by keeping β trajectory with the smallest PPL from the query output and current beam

  end for

end for

$\hat{x}_{L_0}^{0:n-1}, \ldots, \hat{x}_{L_\beta}^{0:n-1} \leftarrow$ GetLoc$(\hat{x}_{T_0}^{0:n-1}, \ldots, \hat{x}_{T_\beta}^{0:n-1})$ ▷ Take the location sequence from $\hat{x}_T^{0:n-1}$ as result $\hat{x}_L^{0:n-1}$

return $\hat{x}_{L_0}^{0:n-1}, \ldots, \hat{x}_{L_\beta}^{0:n-1}$

6.4 Experiments

We empirically evaluate the proposed attack suite to answer the following research questions: (1) What's the performance of the proposed attacks in extracting or inferring the sensitive information from POI recommendation models (Sec. 6.4.3.1)? (2) What unique factors (e.g., user, location, trajectory) in mobility data correlate with the attack performance (Sec. 6.4.3.2)? (3) How do different attack designs (e.g., spatial-temporal querying for membership inference attacks) improve the attack performance (Sec. 6.4.3.3)?

6.4.1 Experimental Setup

We describe the datasets, models, and evaluation metrics used in our experiments below; full implementation details (preprocessing, victim-model training, and attack hyperparameters) follow

Algorithm 6 SpaTemQuery: Spatial-Temporal Model Query Algorithm for LocMIA

Require: Target model: f_{target}, number of query timestamps: n_t, number of query locations: n_l, target example: X_{target}

Ensure: Membership confidence score: $conf$

$u, l \leftarrow X_{target}$
$conf_{all} \leftarrow \{\}$
for $i \leftarrow 0$ to $n_t - 1$ **do**
    $conf_t \leftarrow \{\}$
    for $j \leftarrow 0$ to $n_l - 1$ **do**
        $t_i \leftarrow i/n_t$
        $l_j \leftarrow \text{RandomSample}(\mathcal{L})$
        $conf_t \leftarrow conf_t \cup f_{target}(u, (l_j, t_i))$ ▷ Query the model with random location and a synthetic timestamp
    end for
    $conf_{all} \leftarrow conf_{all} \cup \text{mean}(conf_t)$ ▷ Calculate average confidence from all queries for this timestamp
end for
return $conf \leftarrow \max(conf_{all})$ ▷ Take the confidence scores with largest confidence at position l as output

in Section 6.4.2.

Datasets *Following the literature* [YLZ22, KW18], we comprehensively evaluate four privacy attacks on two POI recommendation benchmarks: FourSquare (4sq) [YZZ14] and Gowalla [CML11].

Models We experiment with three representative POI recommendation models, including GETNext [YLZ22], LSTPM [SQC20], and RNN [WJJ21]. Note that GETNext and LSTPM are the state-of-the-art POI recommendation methods based on the transformer and hierarchical LSTM, respectively. We also include RNN since it is a commonly used baseline for POI recommendation.

Evaluation Metrics We use the top-k extraction attack success rate (ASR) to evaluate the effectiveness of data extraction attacks. For LocExtract, the top-k ASR is defined as $|U_{\text{extracted}}|/|\mathcal{U}|$,

Algorithm 7 Membership Inference Attack

Below, we demonstrate our location-level MIA and trajectory-level MIA algorithms. The lines marked in red are specific to LocMIA, while the lines marked in purple are specific to TrajMIA. Both attacks share the remaining lines.

Require: Victim model: f_θ, shadow data: D_s, number of shadow models: N, inference target: X_{target}, number of query timestamps: n_t, number of query locations: n_l

Ensure: The likelihood ratio to determine if we should reject the hypothesis that X_{target} is a member of f_θ: Λ

$conf_{in}, conf_{out} \leftarrow \{\}, \{\}$

$X_S \leftarrow \textsc{RandomSample}(\{X_S : X_{target} \in X_S\})$ ▷ Sample a location sequence and includes X_{target}

$X_S \leftarrow X_{target}$

for $i \leftarrow 0$ to N **do**

  $D_{in} \leftarrow \textsc{RandomSample}(D_s) \cup X_S$

  $D_{out} \leftarrow \textsc{RandomSample}(D_s) \setminus X_S$

  $f_{in}, f_{out} \leftarrow \textsc{Train}(D_{in}), \textsc{Train}(D_{out})$ ▷ Train f_{in} and f_{out}

  $conf_{in} \leftarrow conf_{in} \cup \phi(\textsc{SpaTemQuery}(f_{in}, n_t, n_l, X_{target}))$

  $conf_{out} \leftarrow conf_{out} \cup$

    $\phi(\textsc{SpaTemQuery}(f_{out}, n_t, n_l, X_{target}))$

  $conf_{in} \leftarrow conf_{in} \cup$

    $\phi\big(\text{mean}(\{f_{in}(X_S)^{0:0}, \ldots, f_{in}(X_S)^{0:n-1}\})\big)$

  $conf_{out} \leftarrow conf_{out} \cup$

    $\phi\big(\text{mean}(\{f_{out}(X_S)^{0:0}, \ldots, f_{out}(X_S)^{0:n-1}\})\big)$

end for

$\mu_{\text{in}}, \mu_{\text{out}} \leftarrow \text{mean}(conf_{\text{in}}), \text{mean}(conf_{\text{out}})$

$\sigma^2_{\text{in}}, \sigma^2_{\text{out}} \leftarrow \text{var}(conf_{\text{in}}), \text{var}(conf_{\text{out}})$

$conf_{\text{obs}} \leftarrow \phi(\textsc{SpaTemQuery}(f_\theta, n_t, n_l, X_{target}))$

$conf_{\text{obs}} \leftarrow \phi\big(\text{mean}(\{f_\theta(X_S)^{0:0}, \ldots, f_\theta(X_S)^{0:n-1}\})\big)$

return $\Lambda = \frac{p(conf_{\text{obs}} | \mathcal{N}(\mu_{\text{in}}, \sigma^2_{\text{in}}))}{p(conf_{\text{obs}} | \mathcal{N}(\mu_{\text{out}}, \sigma^2_{\text{out}}))}$ ▷ Hypothesis test

where $U_{\text{extracted}}$ is the set of users whose most visited locations are in the top-k predictions outputted by our attack; For TrajExtract the top-k ASR is |correct extractions|/|all (u, l_0) pairs|, where correct extractions are (u, l_0) pairs with top-k extracted results matching an exact location sequence in the training data.

For LocMIA and TrajMIA, we utilize the commonly employed metrics for evaluating membership inference attacks, namely the area under the curve (AUC), average-case "accuracy" (ACC), and true positive rate (TPR) versus false positive rate (FPR) in the low-false positive rate regime. Our primary focus is the TPR versus FPR metric in the low-false positive rate regime because evaluating membership inference attacks should prioritize the worst-case privacy setting rather than average-case metrics, as emphasized in [CCN22].

6.4.2 Implementation Details

6.4.2.1 Datasets

We conduct experiments on two POI recommendation benchmark datasets – FourSquare (4sq) [YZZ14] and Gowalla [CML11] datasets. Following the literature [YLZ22, KW18], we use the check-ins collected in NYC for both sources. The 4sq dataset consists of 76,481 check-ins during ten months (from April 12, 2012, to February 16, 2013). The Gowalla dataset comprises 35,674 check-ins collected over a duration of 20 months (from February 2009 to October 2010). In both datasets, a check-in record can be represented as [user ID, check-in time, latitude, longitude, location ID].

6.4.2.2 Data Preprocessing

We preprocess each dataset following the literature [YLZ22]: (1) We first filter out unpopular POIs and users that appear less than ten times to reduce noises introduced by uncommon check-ins. (2) To construct trajectories of different users in a daily manner, the entire check-in sequence of each user is divided into trajectories with 24-hour intervals. Then, we filter out the trajectories with only a single check-in. (3) We further normalize the timestamp (from 0:00 AM to 11:59 PM) in each check-in record into $[0, 1]$. After the aforementioned steps, the key statistics of the 4sq and Gowalla datasets

are shown in Table 6.2. (4) Lastly, we split the datasets into the training, validation, and test sets using the ratio of 8:1:1.

Table 6.2: Dataset statistics for the two POI-recommendation benchmarks.

	#POIs	#Check-ins	#Users	#Trajectories	Avg. Len.
4SQ	4,556	63,648	1,070	17,700	3.63
GOWALLA	2,559	32,633	1,419	7,256	4.46

6.4.2.3 Victim Model Training Settings

We use the official implementation of GETNEXT[2] and LSTPM[3] to train victim models. In particular, we train each model with a batch size 32 for 200 epochs by default. We use five random seeds in all experiments and report the average results.

6.4.2.4 Attack Settings

LOCEXTRACT Given a target user u, we extract the most visited location l_{top1} from the victim model f_θ with a query number $q = 50$. We set the query timestamp $t = 0.5$ (i.e., the middle of the day) by default, and we will present how the change of the query timestamp affects the attack performance in the ablation study.

TRAJEXTRACT In this attack, we experiment with $n = 4$ by default, though the attacker can potentially extract location (sub-)sequences with arbitrary length. We set the beam size $\beta = 50$ in the beam search to query the victim model and update candidate trajectories. For each query, we also have the default query timestamp $t = 0.5$.

LOCMIA In our experiments, since we randomly sample 80% of trajectories as the training dataset D_{tr} to build a victim model for MIA, we treat the remaining 20% data as non-members. For each target user u and the POI location l pair, we generate $N = 64$ synthesis trajectories using TRAJSYNTHESIS with the query timestamp $t_s = 0.5$. With the synthesis trajectories, we can also have

[2] https://github.com/songyangme/GETNext

[3] https://github.com/NLPWM-WHU/LSTPM

64 in-models (f_{in}) and 64 out-models (f_{out}). We also set $n_t = 10$ and $n_l = 10$. For evaluation, we conduct a hypothesis test on a balanced number of members and non-members.

TrajMIA We extract the membership information of some trajectory sequences with arbitrary lengths from the victim model. We also build $N = 64$ in-models (f_{in}) and $N = 64$ out-models (f_{out}) for a target trajectory sequence. For evaluation, we conduct a hypothesis test on a balanced number of members and non-members.

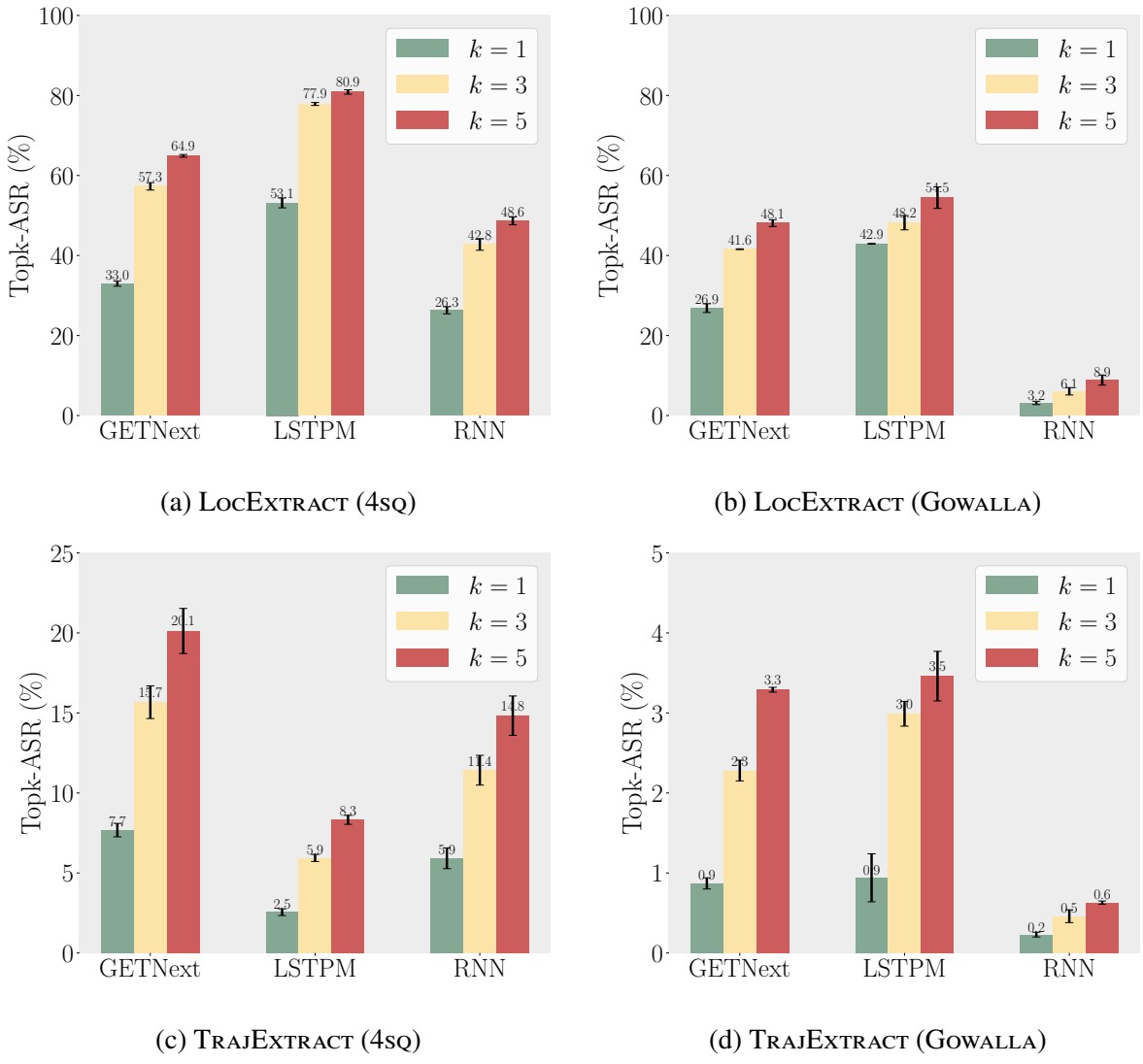
6.4.3 Experimental Results and Analysis

6.4.3.1 Attack performance (RQ1)

Figures 6.2 and 6.3 visualize the attack performance of data extraction and membership inference attacks, respectively. In Figure 6.2, we observe that LocExtract and TrajExtract can effectively extract users' most common locations and trajectories across various model architectures and datasets as the attack performance is significantly better than the random guess baseline, i.e., $1/|\mathcal{L}|$ (0.04% for LocExtract) and $1/|\mathcal{L}|^{n-1}$(10^{-8}% for TrajExtract). Likewise, as shown in Figure 6.3, LocMIA and TrajMIA successfully determine the membership of a specific user-location pair or trajectory, significantly outperforming the random guess baseline (represented by the diagonal line in both figures).

The attack performance also demonstrates that trajectory-level attacks are significantly more challenging than location-level attacks, evident from the better performance of LocExtract and LocMIA compared to TrajExtract and TrajMIA for data extraction and membership inference. We suspect this is because POI recommendation models are primarily designed to predict a single location. In contrast, our trajectory-level attacks aim to extract or infer a trajectory encompassing multiple consecutive locations. The experiment results also align with the findings that longer trajectories are less vulnerable to our attacks (see Figures 6.18 and 6.21 in Section 6.4.5).

The attack performance also differs across different model architectures and datasets. We see a general trend of privacy-utility trade-off in POI recommendation models based on the model performance of the victim model in Table 6.1: with better victim model performance comes better attack performance. While this is a common trend, it might not hold in some cases. For example, the



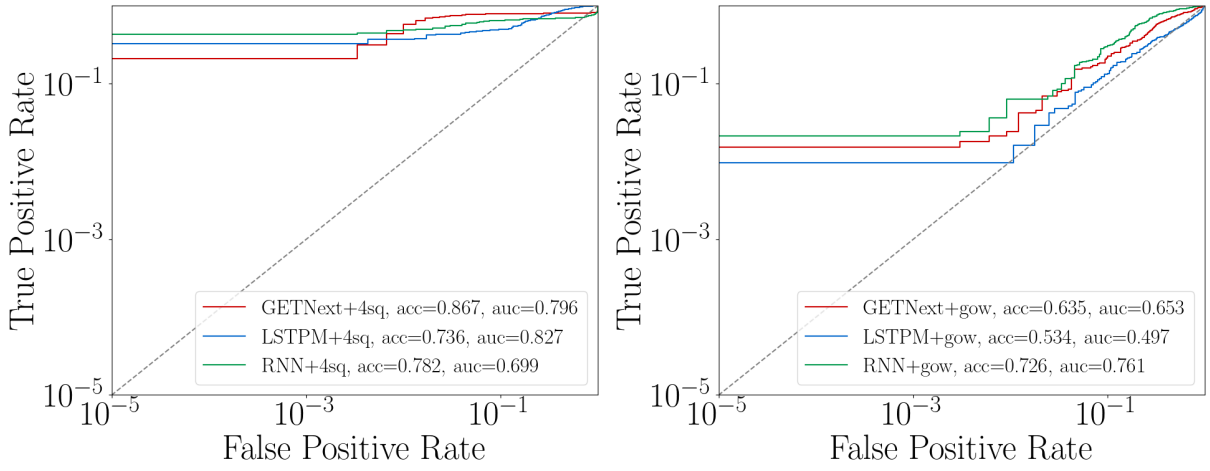
(a) LocExtract (4sq)

(b) LocExtract (Gowalla)

(c) TrajExtract (4sq)

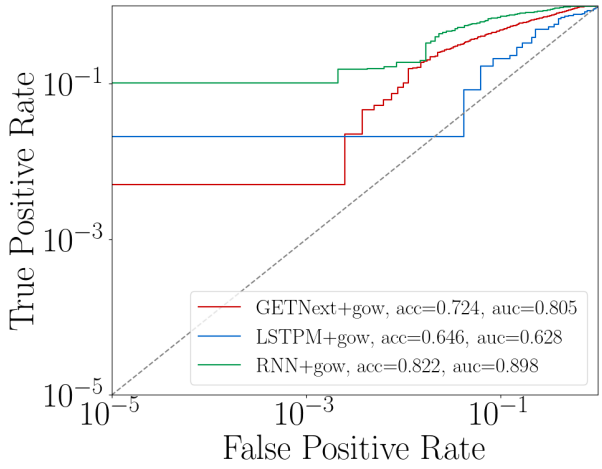
(d) TrajExtract (Gowalla)

Figure 6.2: Attack performance of data extraction attacks (LocExtract and TrajExtract) on three victim models and two mobility datasets.

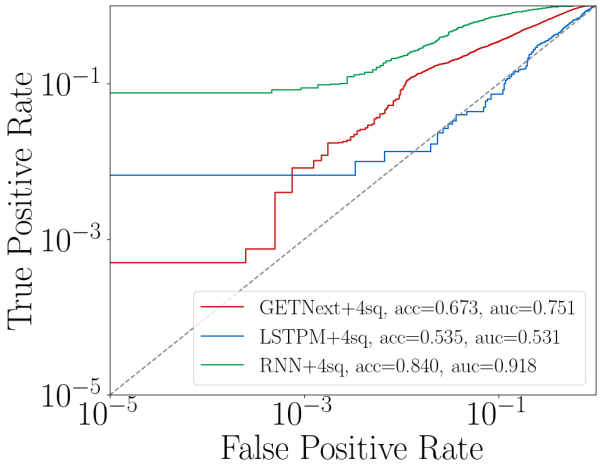


(a) LocMIA (4sq)

(b) LocMIA (Gowalla)



(c) TrajMIA (4sq)



(d) TrajMIA (Gowalla)

Figure 6.3: Attack performance of (LocMIA and TrajMIA) on three victim models and two POI recommendation datasets. The diagonal line indicates the random guess baseline.

MIA performance against RNN is sometimes better than GETNext and LSTPM performances. This might be because GETNext and LSTPM improve upon RNN by better leveraging spatial-temporal information in the mobility datasets. However, the adversary cannot use the exact spatial-temporal information in shadow model training since the adversary cannot access that information. This result can be inspiring in that even though spatial-temporal information can effectively improve attack performance, victim models that better utilize spatial-temporal information are still more resilient to MIAs. Future studies should also consider this characteristic when designing attacks or privacy-preserving POI recommendation models with better privacy-utility trade-offs.

6.4.3.2 Factors in mobility data that make it vulnerable to the attacks (RQ2)

Prior research demonstrates that data outliers are the most vulnerable examples to privacy attacks [CCN22, TSS22] in image and text datasets. However, it is unclear whether the same conclusion holds in mobility data and what makes mobility data as data outliers. To this end, we investigate which factors of the mobility datasets influence the attack's efficacy. In particular, we collect aggregate statistics of mobility data from three perspectives: user, location, and trajectory. We analyze which factors in these three categories make mobility data vulnerable to our attacks. We describe the selection process and the full list of aggregate statistics in Section 6.4.4.1. Our findings are as follows:

- For LocExtract, we do not identify any meaningful pattern correlated with its attack performance. We speculate that a user's most common location is not directly related to the aggregate statistics we study.
- For TrajExtract, our findings indicate that *users who have visited fewer unique POIs* are more vulnerable to this attack, as referenced in Figure 6.9 in Section 6.4.4. This can be explained by the fact that when users have fewer POIs, the model is less uncertain in predicting the next location due to the reduced number of possible choices that the model memorizes.
- For LocMIA, as shown in Figures 6.4a and 6.4b, we find that *locations visited by fewer users or have fewer surrounding check-ins* are more susceptible to LocMIA. We believe this is because

those locations shared with fewer users or surrounding check-ins make them training data outliers.

- For TRAJMIA, *users with fewer total check-ins* (Figure 6.5a), *unique POIs* (Figure 6.5b), and *fewer or shorter trajectories* (Figures 6.5c and 6.5d) are more susceptible. In Figures 6.6a and 6.6b, we also see that *trajectories intercepting less with others or with more check-ins* are more vulnerable to TRAJMIA. We believe these user-level and trajectory-level aggregate statistics make the target examples data outliers.

In summary, we conclude that the effect of data outliers also exists in privacy attacks against POI recommendations. In the context of POI recommendation, the mobility data outliers could be characterized from the perspectives of user, location, and trajectory. Different attacks in our attack suite might be vulnerable to particular types of data outliers, which are more unique and are vulnerable against our attacks compared to other data.

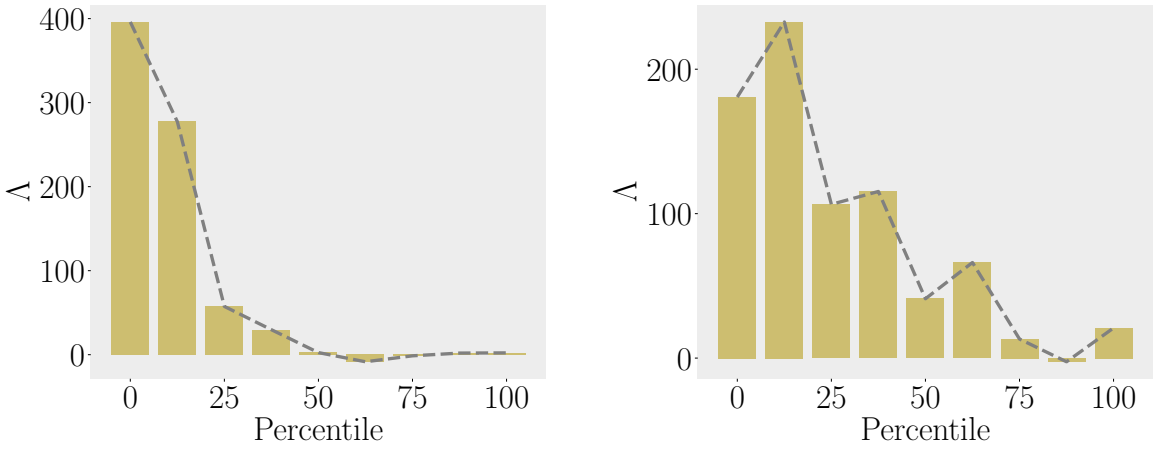
6.4.3.3 The impact of different attack designs (RQ3)

We explore different attack designs that may affect the performance of each attack. We present detailed results in Section 6.4.5 and summarize the key findings here:

- For LOCEXTRACT, as shown in Figures 6.7 and 6.17 in Section 6.4.5, we find that employing *appropriate query timestamp* and *soft voting mechanism* leads to better attack performance. The reason is that the POI recommendation relies on temporal information to make more accurate predictions.

- For TRAJEXTRACT, similar to LOCEXTRACT, we find that an appropriate query timestamp also promotes the attack performance. Moreover, Figure 6.18 in Section 6.4.5 shows that our attack is more effective when extracting location sequences of *shorter trajectories* as the influence of the starting location becomes weaker when the prediction moves forward.

- For LOCMIA, as shown in Figure 6.8, *utilizing more queries with different timestamps* in our spatial-temporal model query algorithm improves the results of inferring the membership of a target user-location pair (u, l). Since the adversary lacks information about real input sequences



(a) # of Users Visited

(b) # of Nearby Check-ins

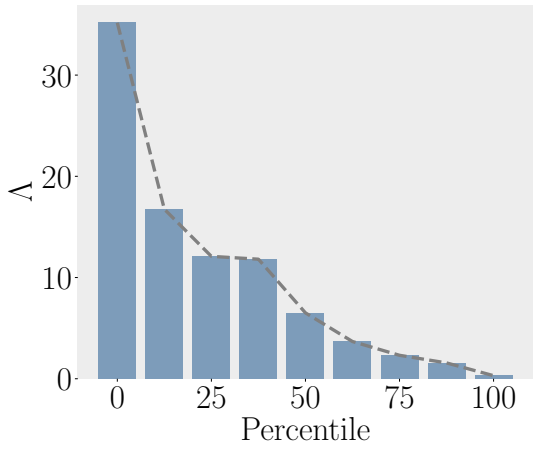
Figure 6.4: How location-level aggregate statistics are related to LocMIA. Locations visited by fewer different users or have fewer surrounding check-ins are more vulnerable to LocMIA.

that are followed by the target location l. Utilizing more queries helps to traverse the search space and promotes the attack performance.

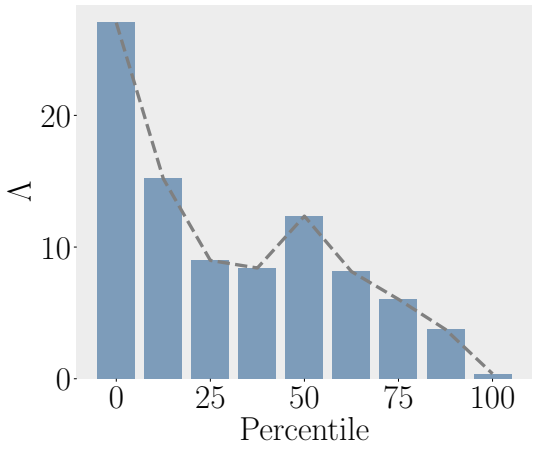
- For both LocMIA and TrajMIA, as shown in Figure 6.19 in Section 6.4.5, we find that *a larger number of shadow models* yields better attack performance since they provide more samples to simulate the loss distributions of in-samples and out-samples.

Our analysis also reveals additional intriguing findings: (1) Temporal information in the attacker's queries can significantly affect the attack results. For extraction attacks, we find that the extraction ASRs first increase and then decrease as the query timestamp increases, which peaks at 0.5 (using the middle of the day as the query timestamp) according to Figure 6.7. The reason is that most check-ins occur during the daytime rather than at night.

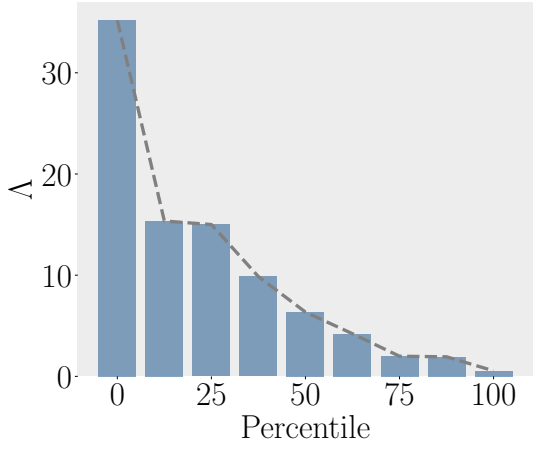
(2) Figure 6.14 in Section 6.4.5 demonstrates that our data extraction attacks remain effective even with a limited number of queries. In other words, our attacks are efficient in real-world scenarios.



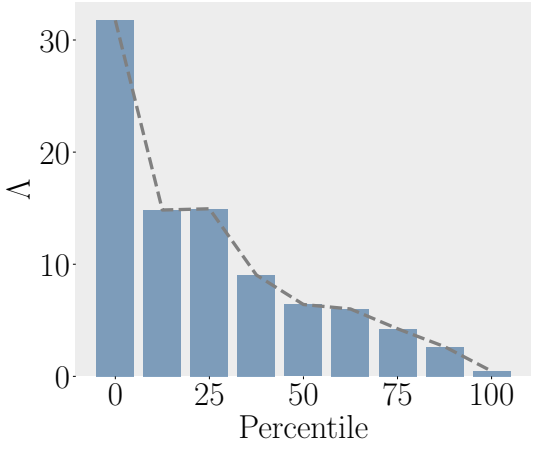
(a) # of User Check-ins



(b) # of User POIs

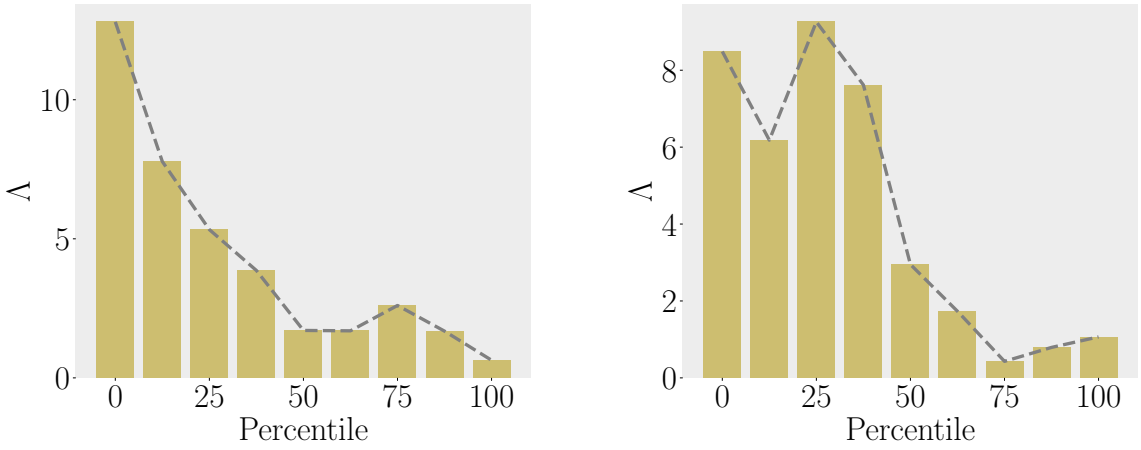


(c) # of User Trajectories



(d) Avg User Traj Length

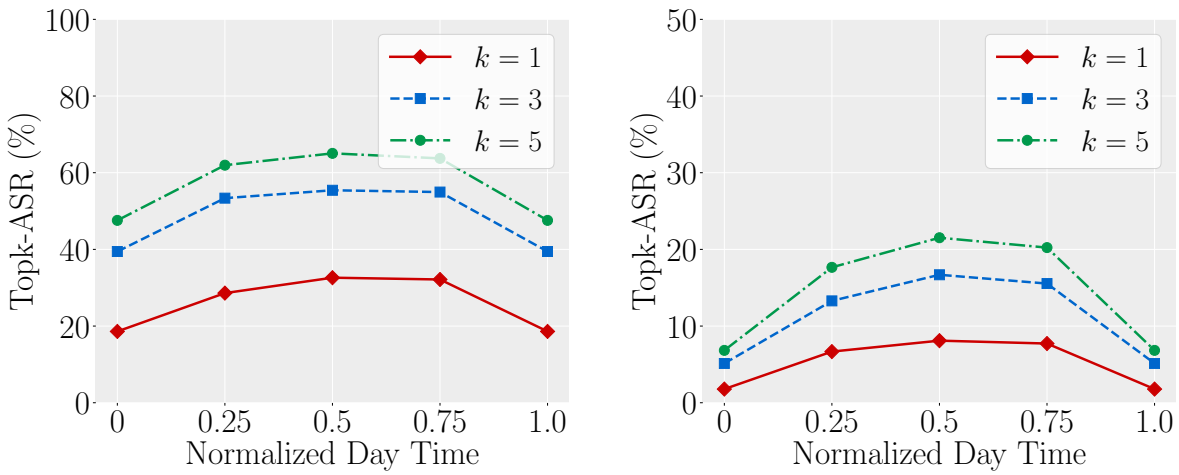
Figure 6.5: How user-level aggregate statistics are related to TrajMIA. *x-axis:* Percentile categorizes users/locations/trajectories into different groups according to their feature values. *y-axis:* Λ indicates the (averaged) likelihood ratio of training trajectories/locations being the member over non-member from the hypothesis test for each group, with a higher value indicating the larger vulnerability. The users with fewer total check-ins, fewer unique POIs, and fewer or shorter trajectories are more vulnerable to TrajMIA. (4sq)



(a) # of Intercepting Trajs

(b) # of POIs in Traj

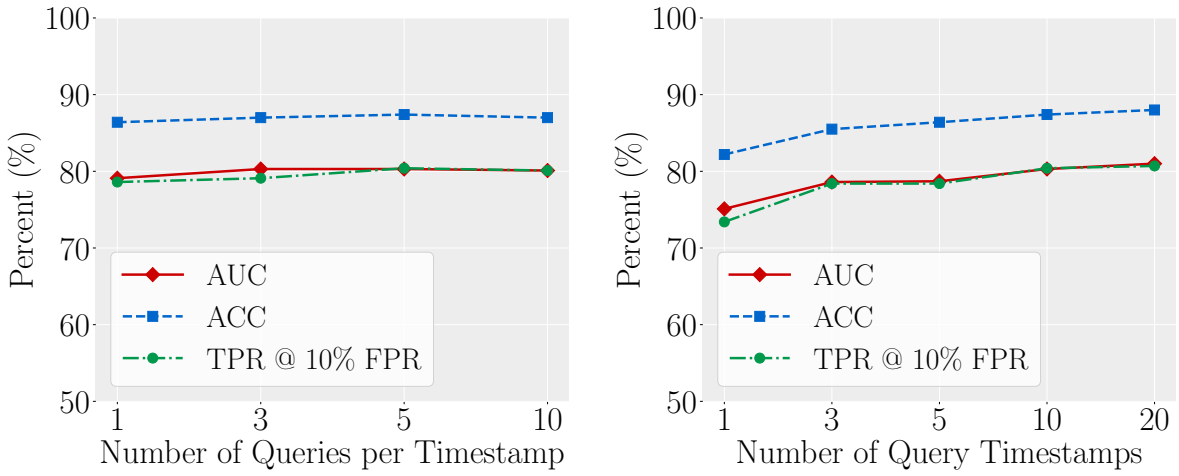
Figure 6.6: How trajectory-level aggregate statistics are related to TrajMIA. The trajectories with fewer intercepting trajectories or fewer POIs are more vulnerable to TrajMIA.



(a) LocExtract

(b) TrajExtract

Figure 6.7: The optimal query timestamp can significantly improve the performance of LocExtract and TrajExtract.(4sq)



(a) Impact of n_l on LocMIA

(b) Impact of n_t on LocMIA

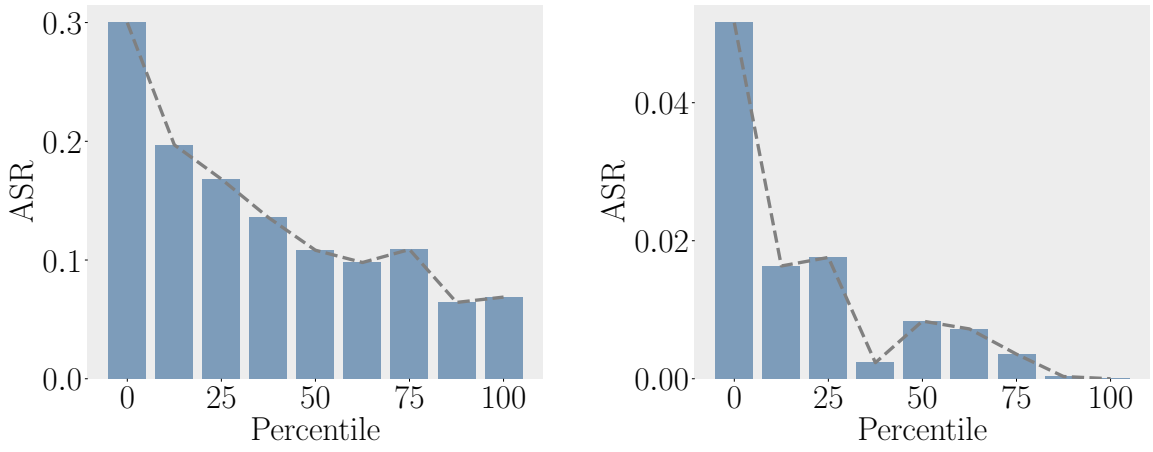
Figure 6.8: Both the number of query timestamps n_t and the number of query locations n_l affect the attack performance of LocMIA. We use the default $n_t = 10$ and $n_l = 10$ in (a) and (b), respectively. Given a few queries, LocMIA remains effective. (4sQ)

6.4.4 Vulnerability Factor Analysis: Methodology Details

6.4.4.1 How to Select Aggregate Statistics

This section outlines the basic principles and details for selecting representative aggregate statistics for analysis. For user-level aggregate statistics, we target the basic statistical information quantifying properties of locations and trajectories of a user. For location-level and trajectory-level aggregate statistics, we study their users, "neighboring" check-ins and trajectories, and the check-in time information. In summary, we select the following aggregate statistics:

- User-level aggregate statistics:
  1. Total number of check-ins;
  2. Number of unique visited POIs;
  3. Number of trajectories;
  4. Average trajectory length;
- Location-level aggregate statistics:



(a) # Unique POIs (4SQ)

(b) # Unique POIs (GOWALLA)

Figure 6.9: How user-level aggregate statistics are related to TRAJEXTRACT. The users who have fewer unique POIs are more vulnerable to TRAJEXTRACT.

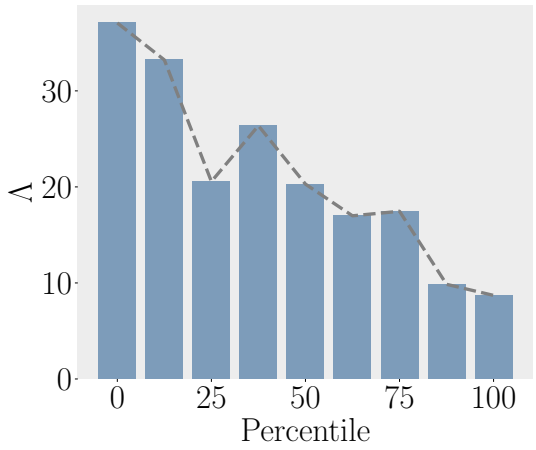
1. Number of users who have visited this POI;
2. Number of check-ins surrounding ($\leq$ 1km) this POI;
3. Number of trajectories sharing this POI;
4. Average time in a day for the visits to the POI;

- Trajectory-level aggregate statistics:

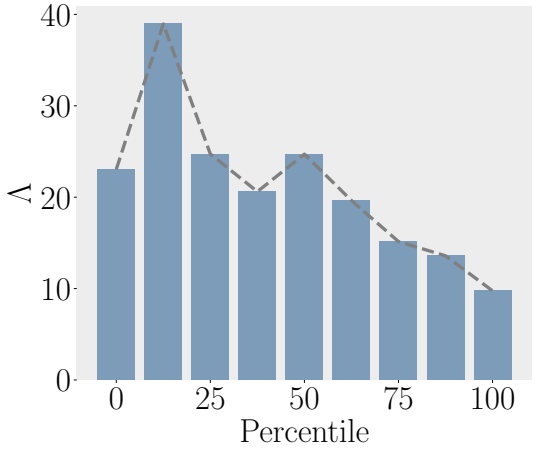
1. Number of users who have the same trajectories;
2. Number of check-ins surrounding ($\leq$ 1km) all POI in the trajectory;
3. Number of intercepting trajectories;
4. Average check-in time of the trajectory.

6.4.5 The Impact of Training and Attack Parameters

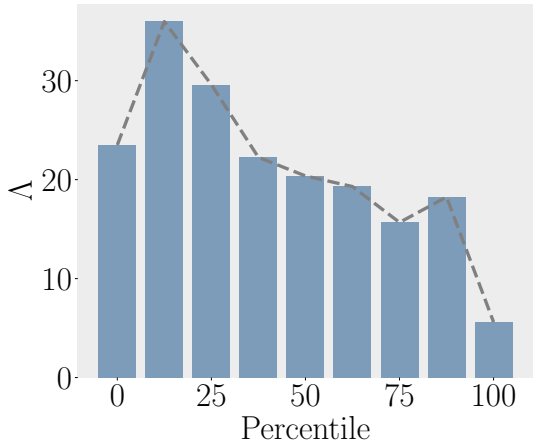
In this section, we analyze how training parameters (Sec. 6.4.5.1) and attack parameters affect the attack performance of data extraction attack (Sec. 6.4.5.2) and membership inference attack (Sec. 6.4.5.3).



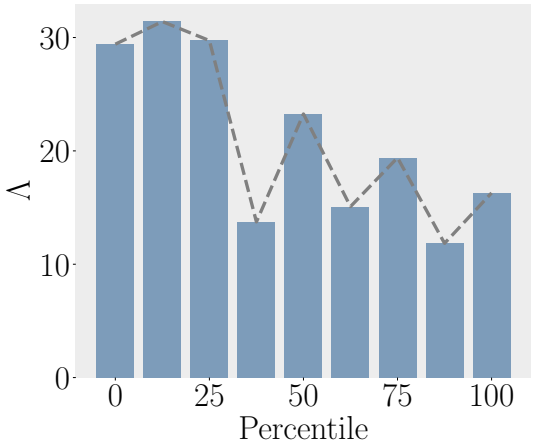
(a) # of User Check-ins



(b) # of User POIs

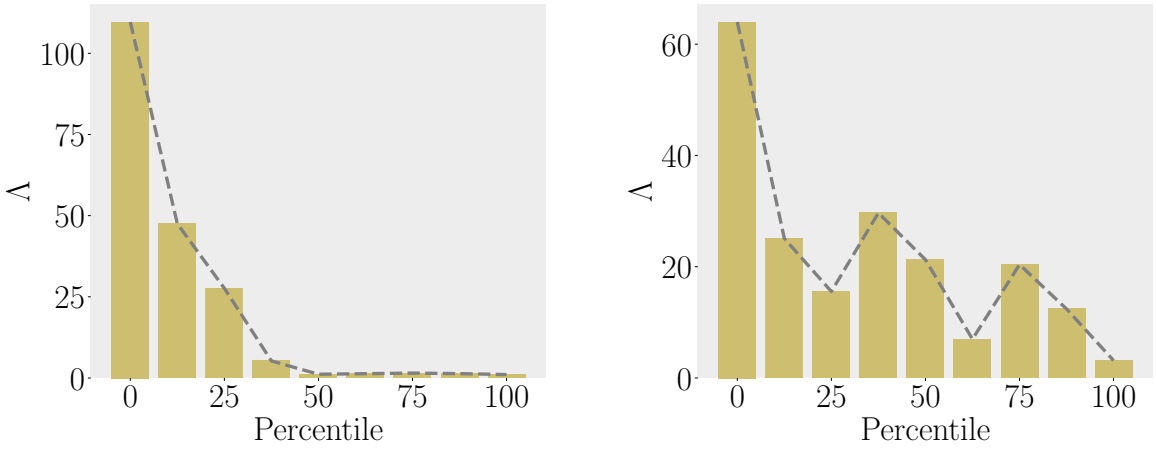


(c) # of User Trajectories



(d) Avg User Traj Length

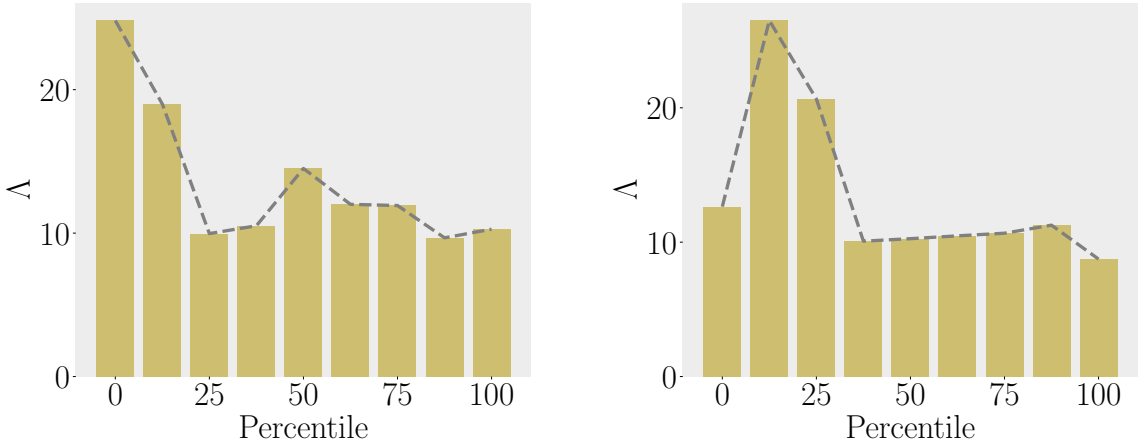
Figure 6.10: How user-level aggregate statistics are related to TrajMIA. The users with fewer total check-ins, fewer unique POIs, and fewer or shorter trajectories are more vulnerable to TrajMIA. (Gowalla)



(a) # of Users Visited

(b) # of Nearby Check-ins

Figure 6.11: How location-level aggregate statistics are related to LocMIA. The locations shared by fewer users or have fewer surrounding check-ins are more vulnerable to LocMIA. (Gowalla)



(a) # of Intercepting Trajs

(b) # of POIs in Traj

Figure 6.12: How trajectory-level aggregate statistics are related to TrajMIA. The trajectories with fewer intercepting trajectories or fewer POIs in the trajectory are more vulnerable to TrajMIA. (Gowalla)

6.4.5.1 The Impact of Training Parameters

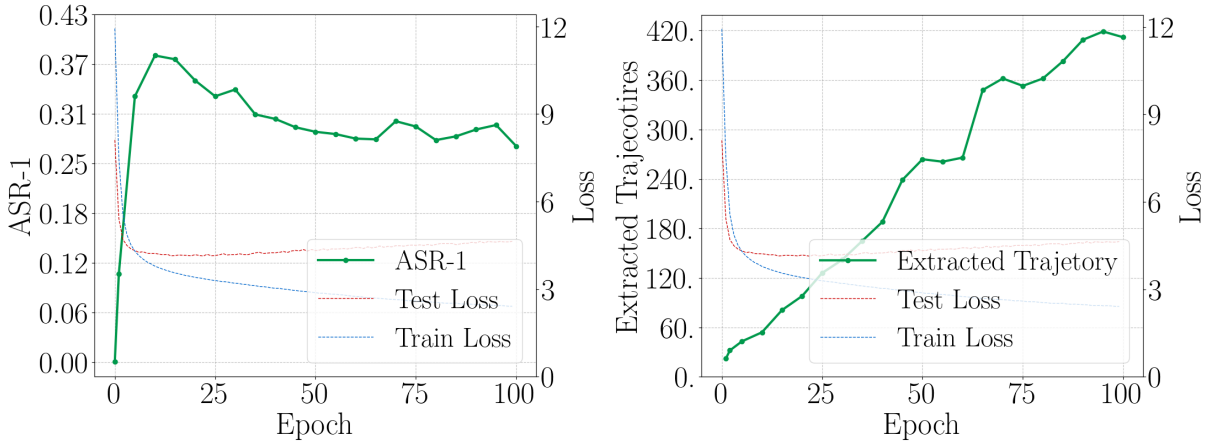
Our primary objective of this analysis is to understand whether the occurrence of overfitting, commonly associated with excessive training epochs, leads to heightened information leakage based on our proposed attacks. By showcasing the ASR at various model training stages, we aim to gain insights into the relationship between overfitting and ASR.

Based on the results presented in Figure 6.13, we observe that LocExtract achieves the best performance when the model is in the convergence stage. We speculate that continuing training beyond convergence leads to overfitting, causing a loss of generalization. Specifically, when the model is overfitted, it tends to assign higher confidence to the training data while disregarding the general rules present in the dataset. In contrast, our attack employs random queries to extract the general rules learned by the model from the dataset, resulting in better performance when applied to the optimally fitted model.

The attack performance of TrajExtract improves as the training process progresses, which can be attributed to the model becoming increasingly overfitted. The overfit model is more likely to output the exact training trajectory and generates more precise training trajectories than the best-fitted model when given the same number of queries. Similarly, the results of our membership inference attacks reveal a trend of attack performance consistently improving with the progression of the training process. This observation aligns with our expectations, as when the model undergoes more training iterations, the effects of training data are more emphasized. Consequently, the distribution of query results in our attack on the seen training data diverging further from the distribution derived from the unseen data. This growing disparity between the two distributions facilitates the membership inference task, particularly on overfitted models. The analysis of these three attacks indicates a consistent trend, highlighting the increased risk of privacy leakage due to overfitting with respect to the original training data.

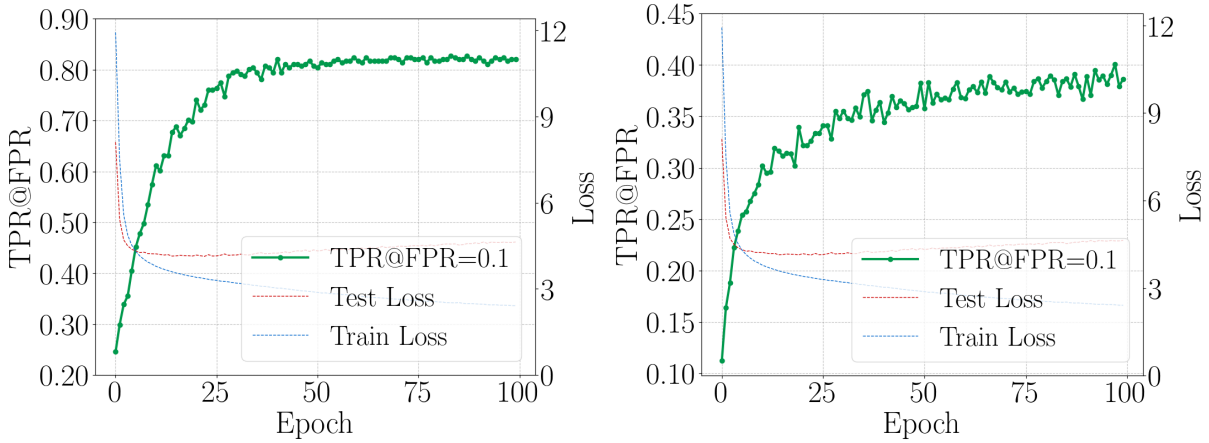
6.4.5.2 Ablation Study on Our Data Extraction Attacks

Data extraction attacks are effective given a limited number of queries In a realistic attack scenario, the adversary may encounter query limitations imposed by the victim model, allowing the



(a) LocExtract

(b) TrajExtract



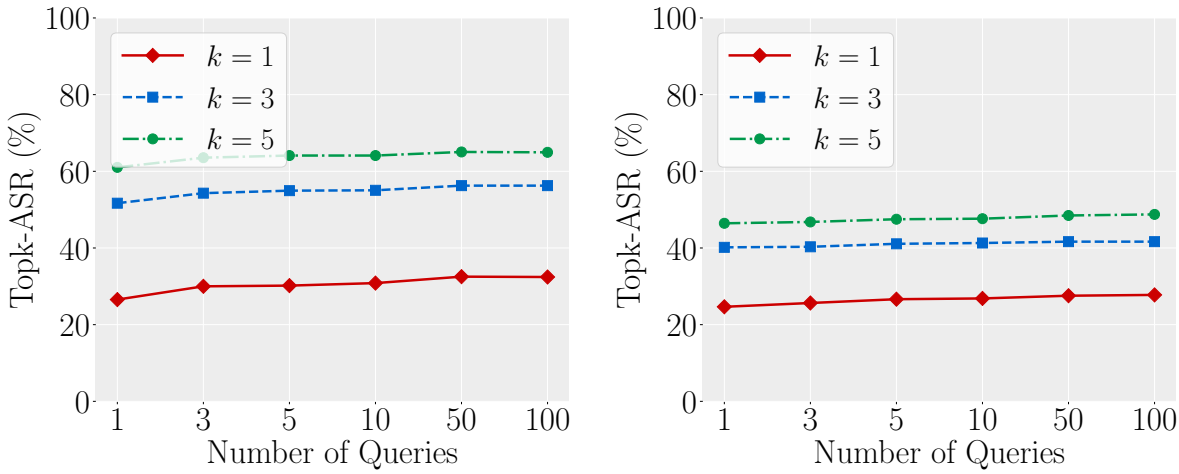
(c) LocMIA

(d) TrajMIA

Figure 6.13: The impact of the model generalization on the performance of four privacy attacks.

adversary to query the model for only a certain number of queries. Figure 6.14 illustrates that our data extraction attacks are effective given a limited number of queries. For example, as shown in Figure 6.14a, a mere $q = 50$ query is sufficient for the adversary to achieve a high ASR and infer a user's frequently visited location. In terms of TrajExtract attack (Figure 6.15a), the adversary can opt for a small beam width of $\beta = 10$, requiring only 1000 queries to extract a trajectory of length $n = 4$. This practicality of our data extraction attack holds true even when the query limit is very small.

Appropriate query timestamp improves the effectiveness of data extraction attacks POI recommendation models rely on temporal information to make accurate location predictions.



(a) LocExtract (4sq)

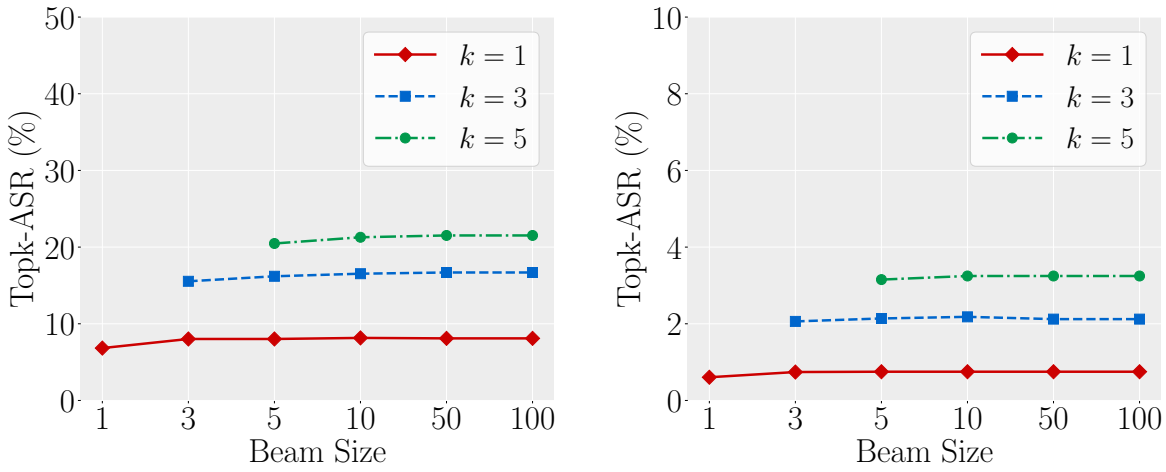
(b) LocExtract (Gowalla)

Figure 6.14: Our LocExtract is effective with a small number of queries (i.e., this attack is effective within a small query budget).

However, obtaining the same timestamps as training for attack can be challenging and is an unrealistic assumption. Therefore, in our data extraction attack setup, we set the query timestamp to $t = 0.5$ (i.e., the middle of the day).

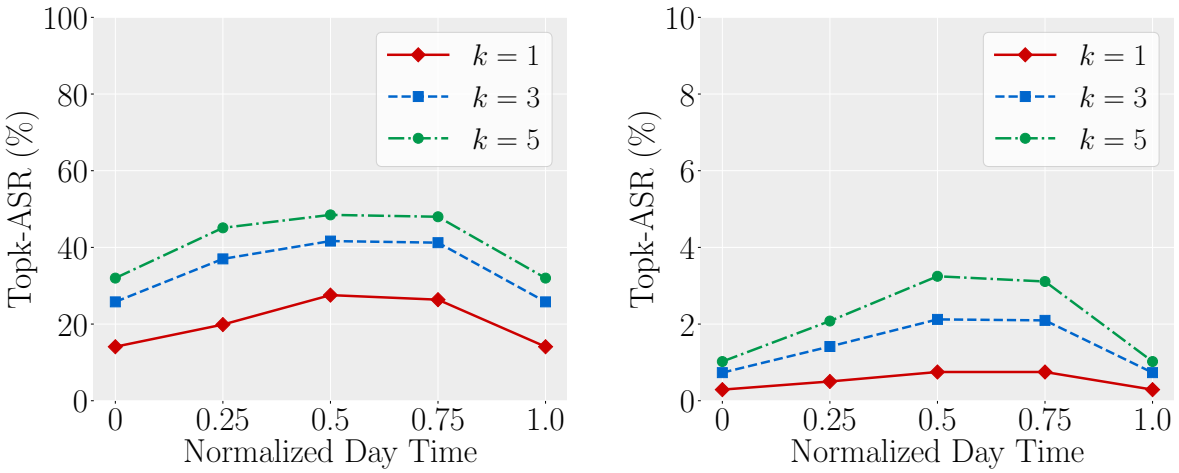
To analyze the effect of how different query timestamps affect data extraction attack performance, we conduct extraction attacks and vary different timestamps that represent various sections within a 24-hour window in the experiments. The results, illustrated in Figure 6.7 and Figure 6.16, indicate that utilizing timestamps corresponding to common check-in times, such as the middle of the day or late afternoon, yields better attack outcomes. This finding aligns with the rationale that users are more likely to engage in check-ins during the daytime or after work hours.

Soft voting improves LocExtract For LocExtract, we have the option to employ either hard voting or soft voting to determine the most frequently occurring location. Hard-voting ensembles make predictions based on a majority vote for each query while soft-voting ensembles consider the average predicted probabilities and select the top-k locations with the highest probabilities. From the experimental results depicted in Figure 6.17, we observe that there is not a substantial difference in ASR-1 when using hard voting or soft voting. However, employing soft voting yields better ASR-3 and ASR-5 results.



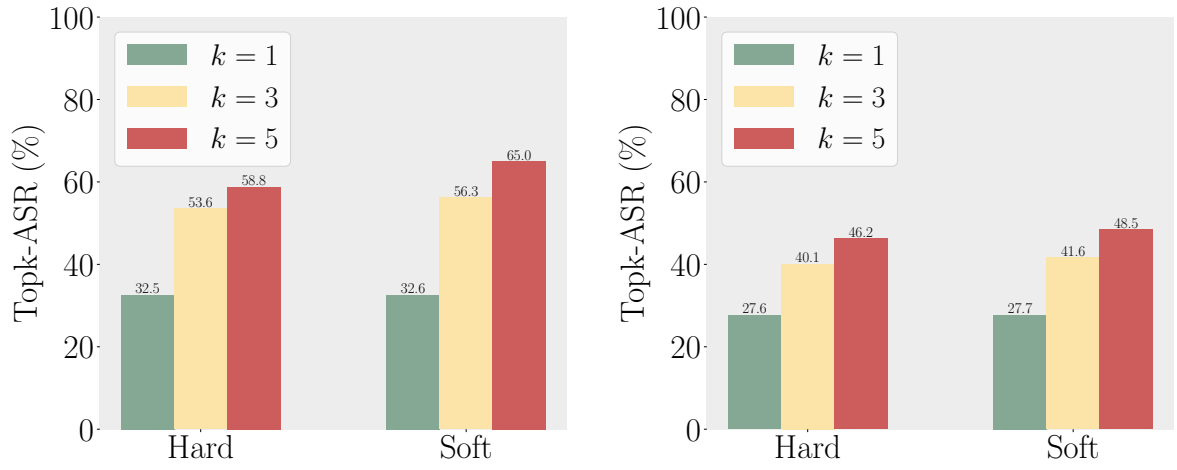
(a) TrajExtract (4sq) (b) TrajExtract (Gowalla)

Figure 6.15: Our TrajExtract is effective with a small beam size (i.e., this attack is effective within a small query budget).



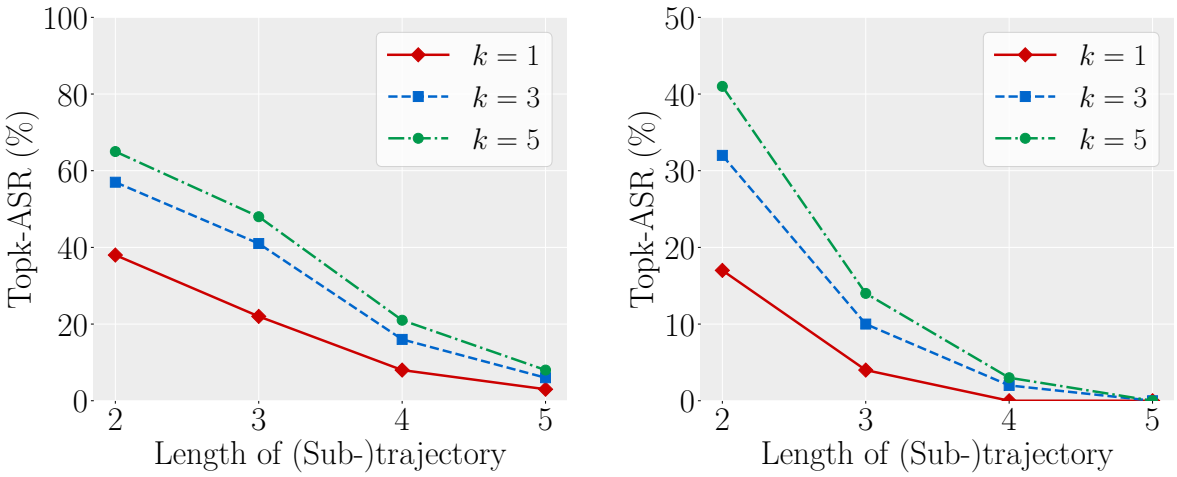
(a) LocExtract (Gowalla) (b) TrajExtract (Gowalla)

Figure 6.16: The optimal query timestamp can significantly improve the performance of LocExtract and TrajExtract (Gowalla).



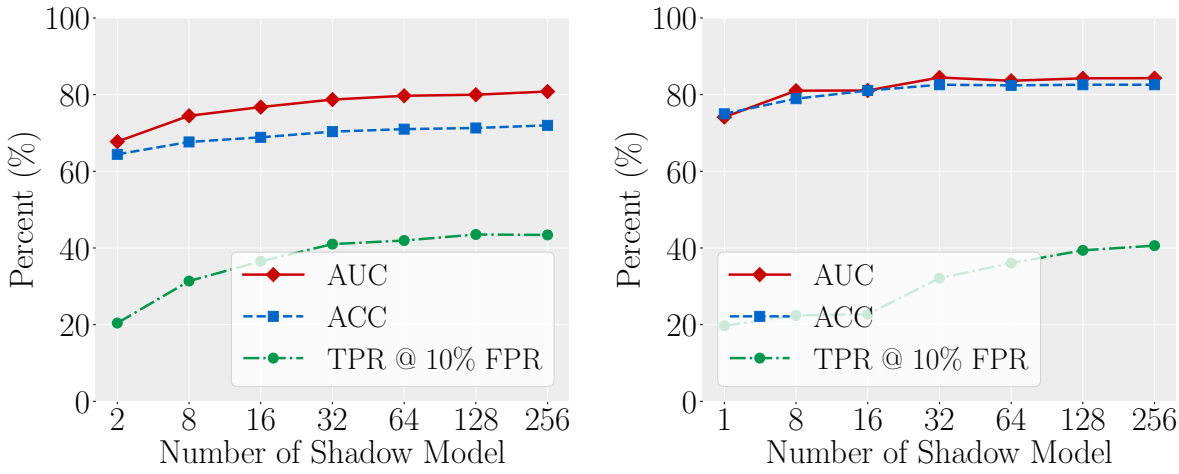
(a) 4SQ (b) GOWALLA

Figure 6.17: Comparing soft voting with hard voting for logits aggregation in LOCEXTRACT. Soft voting has larger improvements over hard voting as k increases.



(a) 4SQ (b) GOWALLA

Figure 6.18: The location sequences of shorter (sub-)trajectories are more vulnerable to TRAJEXTRACT.



(a) LocMIA (4sq)

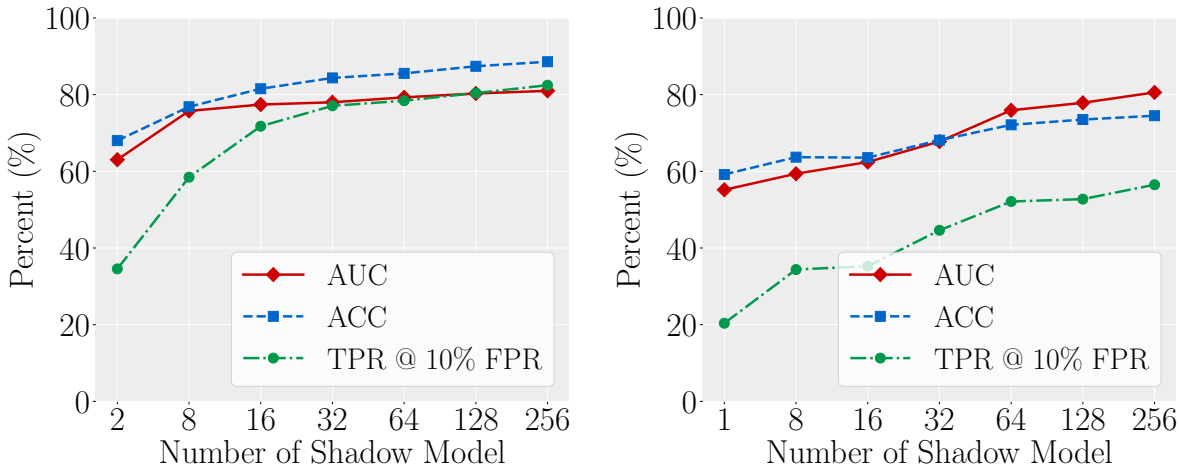
(b) LocMIA (Gowalla)

Figure 6.19: The attack performance of LocMIA significantly improves as the number of shadow model increases.

Location sequences of shorter trajectories are more vulnerable to TrajExtract For TrajExtract, we conduct an ablation study to extract trajectories of varying lengths n. The results, illustrated in Figure 6.18, indicate that the attack achieves higher ASR on shorter trajectories than longer ones. This observation can be attributed to our assumption that the attacker possesses prior knowledge of a starting location. As the prediction moves further away from the starting location, its influence on subsequent locations becomes weaker. Consequently, predicting locations farther from the starting point becomes more challenging, decreasing the attack's success rate for longer trajectories. Moreover, the extraction of long trajectories presents additional difficulties. With each step, the probability of obtaining an incorrect location prediction increases, amplifying the challenges the attack algorithm faces.

6.4.5.3 Ablation Study on Our Membership Inference Attacks

A larger number of shadow models improves the effectiveness of MIAs As mentioned in [CCN22], it has been observed that the attack's performance of LiRA tends to improve as the number of shadow models increases. Consistently, our attacks also follow this pattern, as depicted in Figure 6.19. Both location-level MIA and trajectory-level MIA show enhanced performance as we incorporate more



(a) TrajMIA (4sq) (b) TrajMIA (Gowalla)

Figure 6.20: The attack performance of TrajMIA significantly improves as the number of shadow model increases.

shadow models. This improvement is because an increased number of shadow models allows for a better approximation of the distributions for f_{in} and f_{out}, thereby simulating the victim model more accurately.

LocMIA is effective given a limited number of queries Since our LocMIA involves multiple queries to explore locations preceding the target location, as well as the corresponding timestamps, it is essential to consider potential limitations on the number of queries in real-world scenarios. Thus, we conduct experiments to investigate the impact of query limits on LocMIA. The results, depicted in Figure 6.8a, indicate that our attack remains effective even with a limited number of queries for different location choices. The further increase in query locations would not significantly improve attack results.

We also conduct experiments with different settings for the number of query timestamps, denoted as n_t. The rationale behind this step is that the adversary does not possess information about the real timestamp used to train the victim model. To simulate the effect of selecting the correct timestamp, we perform experiments with varying timestamps to identify the timestamp that yielded the highest confidence score for the targeted location. Based on empirical observations from our experiment on the 4sq dataset (see Figure 6.8b), increasing the number of query timestamps tends to yield better

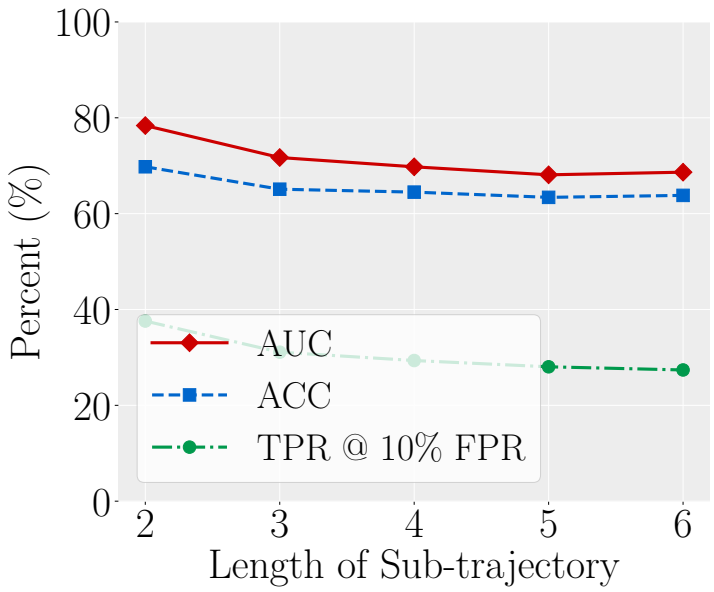


Figure 6.21: The longer trajectories are less vulnerable to TrajMIA.

overall results in practice.

TrajMIA is less effective as the length of target trajectory increases From Figure 6.21, we note that the attack performance drops as target trajectories are longer. This decline happens because all trajectory query scores influence the attack. Lengthier trajectories introduce increased randomness for the query, affecting the outcome of TrajMIA.

6.5 Defense

We evaluate existing defenses against privacy attacks on machine learning models. This section presents the defense mechanisms and key findings, with full experimental details in Section 6.5.3.

6.5.1 Defense Techniques

In particular, we evaluate two streams of defense mechanisms on proposed attacks, including (1) standard techniques to reduce overfitting (e.g., early stopping, l_2 regularization) and (2) differential privacy-based defenses (e.g., DP-SGD [ACG16]) for provable risk mitigation. The standard techniques reduce the victim model's memorization to some degree, but they are insufficient due to

the lack of statistical guarantees. To fill this gap, differential privacy [DR14] is also used to defend against our attacks, which can theoretically limit the impact of a single data point on the model's performance.

Specifically, we first experiment with DP-SGD [ACG16], the most representative DP-based defense, to train differentially-private POI recommendation models. The key idea of DP-SGD is to add Gaussian noises $\mathcal{N}(0, \sigma^2 C^2 I)$ to the clipped gradients g of the model during its training process. C is a clipping threshold that bounds the sensitivity of g by ensuring $\|g\| \leq C$. To achieve (ϵ, δ)-DP, we have $\sigma = \sqrt{2 \ln \frac{1.25}{\delta}}/\epsilon$. Despite that DP-SGD provides promising defense performance on language tasks [FDM19], we find that it can substantially sacrifice the model's utility on the POI recommendation task. Specifically, the top-10 accuracy is only 4.97% when the mechanism satisfies $(5, 0.001)$-DP, while the original top-10 accuracy without DP is 71%. The reason for this performance decrease is that POI recommendation aims to make accurate user-level predictions within a large output space (i.e., $> 4,000$ possible POIs). For different users, even the same location sequence may lead to a different result, which means that the model needs to capture user-specific behavior patterns from a relatively small user dataset. As a result, the training is quite sensitive to the noises introduced by DP-SGD, making it not applicable to POI recommendations.

However, we argue that DP-SGD provides undifferentiated protection for all the mobility data, while for POI recommendation, protecting more tailored sensitive information is more important. For example, a defender may only care about whether a list of check-ins about home addresses is protected or not. To this end, we introduce the notion of selective DP [SCL22] to relax DP and improve the model's utility-privacy trade-offs. Specifically, we apply the state-of-the-art selective DP method JFT [SCZ22] to protect different levels of sensitive information for each attack. The key idea of JFT is to adopt a two-phase training process: in the phase-I training, JFT redacts the sensitive information in the training dataset and optimizes the model with a standard optimizer; in the phase-II training, JFT applies DP-SGD to finetune the model on the original dataset in a privacy-preserving manner. Due to the phase-I training, we observe that the model's utility is significantly promoted. In addition to JFT, we also apply Geo-Indistinguishability (Geo-Ind) [ABC13] to protect common locations in LocExtract. We note that Geo-Ind is only applicable to LocExtract (but not LocMIA) because it requires modifying the training data and is incompatible with the notion of

Table 6.3: The exposure of sensitive information in each attack.

Attack	Sensitive Information
LocExtract	Most common location of each user
TrajExtract	Each location sequence/sub-sequence (x_L)
LocMIA	Each user-location pair (u, l)
TrajMIA	Each trajectory sequence/sub-sequence (x_T)

membership inference.

6.5.2 Takeaway Messages from the Defense

We evaluate different defense mechanisms in terms of their performance in preventing each attack from stealing the corresponding sensitive information. Table 6.3 summarizes the exposure of sensitive information in each attack. Section 6.5.3 illustrates our evaluation metrics, experimental setup, and the results. Recall that the check-ins within a mobility dataset are not equally important. Therefore, in our experiments, we comprehensively evaluate the defense mechanisms from two perspectives. Specifically, we measure their performance in (1) protecting all the sensitive information and (2) protecting a targeted subset of sensitive information from being attacked. Tables 6.22 and 6.23 show that existing defenses provide a certain degree of guarantee in mitigating privacy risks of ML-based POI recommendations, especially for the targeted subset of sensitive information. However, there is no such unified defense that can successfully defend against all the proposed attacks within a small utility drop. In other words, our exploration highlights the need for more advanced defenses.

6.5.3 Defense Implementation and Evaluation Details

We evaluate existing defenses against our privacy attacks, including standard techniques to reduce overfitting and differential privacy based defense. Detailed descriptions of defense techniques can be found in Section 6.5. Then, in 6.5.3.1, we describe the evaluation metrics to measure the defense performance from two perspectives. In 6.5.3.2, we compare different defenses and analyze the

numerical results with detailed explanations.

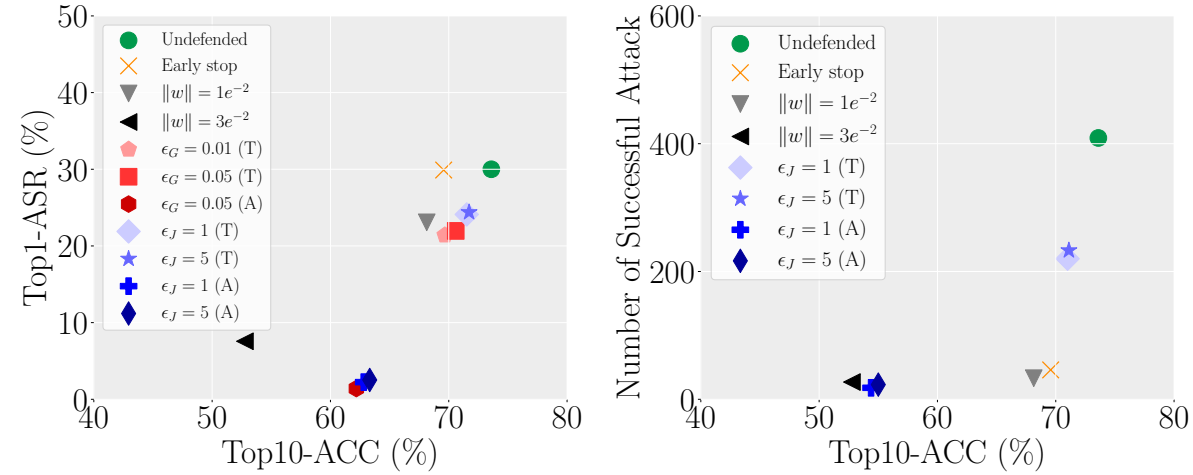
6.5.3.1 Defense Metrics

Our inference attacks extract different sensitive information about the training dataset from the victim model, as summarized in Table 6.3. To this end, we evaluate defense mechanisms in terms of their performance in preventing each attack from stealing the corresponding sensitive information. Specifically, we measure their defense performance on protecting *all the sensitive information* and *a targeted subset of sensitive information* for each attack, respectively. Here, we define the targeted subset of sensitive information as the mobility data a defender wants to protect in practice (e.g., some selected user-location pairs in LocMIA). We introduce this metric because not all the mobility data are sensitive or equally important. Take LocMIA as an example: Since the utility of POI recommendation is highly related to the model's memorization of user-location pairs, a user may want the model to recognize most of the POIs in his trajectory history while hiding those that are very likely to leak his personal identity (e.g., home). In other words, not all the mobility data need to be protected and it's more important to evaluate how defense mechanisms perform on the targeted subset of sensitive information.

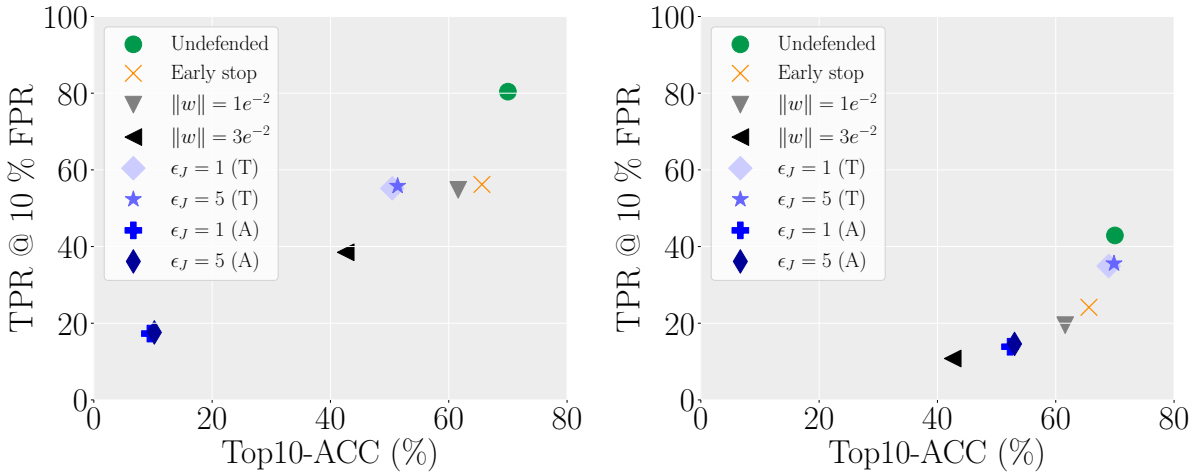
To this end, we jointly measure the defense performance in protecting all the sensitive information and the targeted subset of sensitive information for each attack. Based on different attack objectives, we construct a different targeted subset of sensitive information for measurement by randomly sampling a portion of (e.g., 30%) the most common locations in LocExtract, location sequences in TrajExtract, user-location pairs in LocMIA and trajectory sequences in TrajMIA. It is noted that we randomly sample 30% of sensitive information in each attack to construct the targeted subset for the ease of experiments. In practice, the defender may have more personalized choices based on user-specific requirements, which we leave as future work.

6.5.3.2 Defense Setup and Results

Setup GETNext models trained on the 4sq dataset are used for experiments. For L_2 regularization, we use weight decay $\|w\| = 1e^{-2}$ and $3e^{-2}$. For early stopping, we stop the training after 5 epochs.

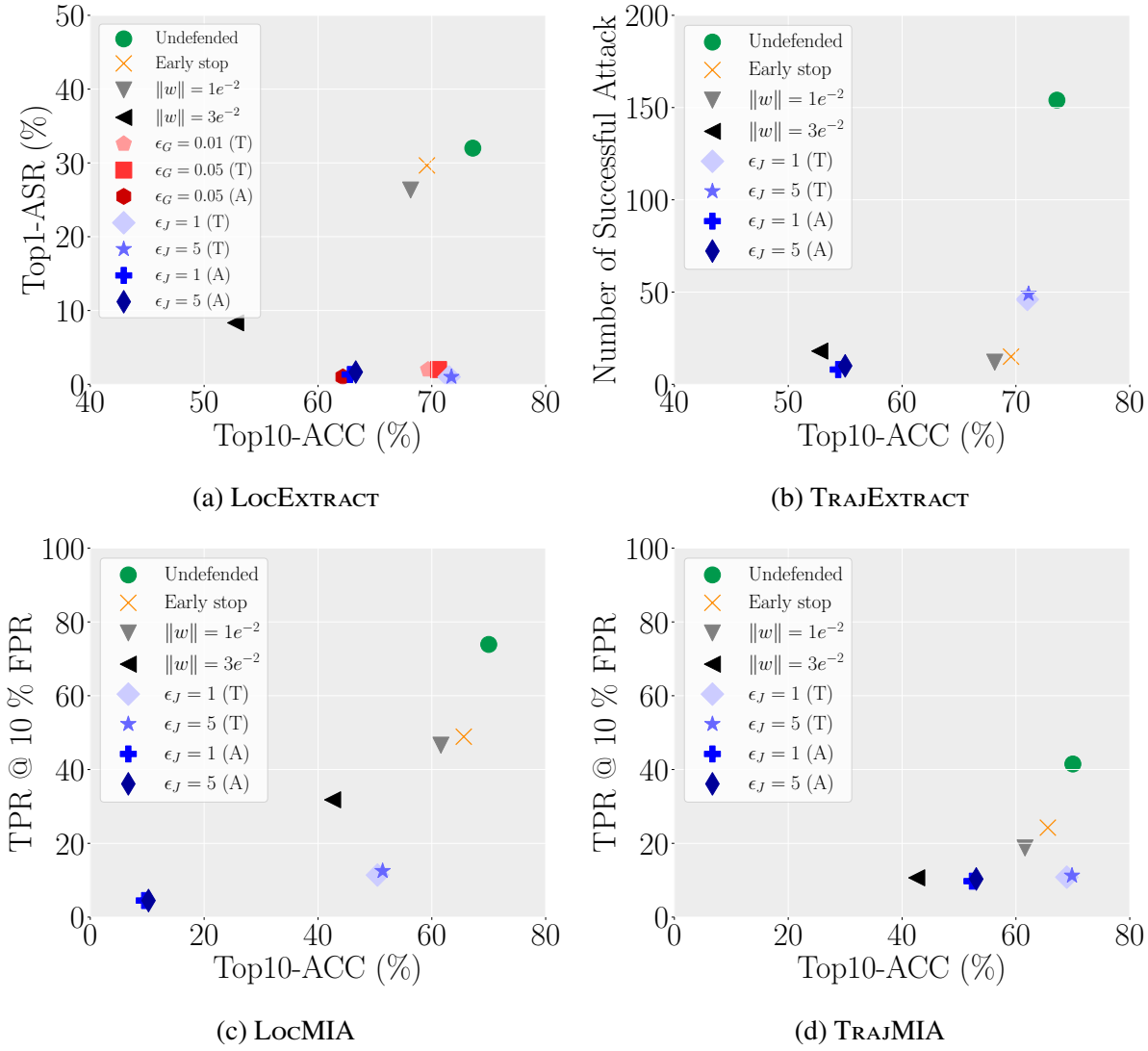


(a) LocExtract (b) TrajExtract



(c) LocMIA (d) TrajMIA

Figure 6.22: Defense performance on protecting all corresponding sensitive information for each attack.



(a) LocExtract

(b) TrajExtract

(c) LocMIA

(d) TrajMIA

Figure 6.23: Defense performance on protecting the targeted subset of sensitive information for each attack.

For JFT, we mask sensitive information that needs to be protected in phase-I. Then in phase-II, we use DP-SGD [ACG16] with different ϵ_J (1 and 5) to finetune the model. The C and δ are set to 10 and $1e^{-3}$. For Geo-Ind against LocExtract, we apply different ϵ_G (0.01 and 0.05) to replace each sensitive POI with its nearby location such that the original POI is indistinguishable from any location within $r = 400$ meters. Since both JFT and Geo-Ind can be used to protect different amounts of sensitive information, we either protect nearly all the sensitive information or only the targeted subset of sensitive information for each attack, denoted by suffixes (A) and (T).

LocExtract Figure 6.22a shows the defense results on protecting all the sensitive information for LocExtract. From the figure, we observe that DP-based defenses achieve better performance than standard techniques. Both JFT (A) and Geo-Ind (A) reduce ASR from 30% to 1% with only a 10% drop in utility. The reason is that these methods allow the defender to selectively protect common locations only. Besides, when protecting the same amount of sensitive information, JFT achieves slightly better accuracy than Geo-Ind because it involves phase-II training to further optimize the model. Moreover, although the ASR is still high for JFT (T) and Geo-Ind (T) in Figure 6.22a, we notice that they can substantially reduce the attack performance on the targeted subset of sensitive information, as shown in Figure 6.23a. This allows a defender to protect the targeted subset with negligible utility drop. Figure 6.24 further shows that Geo-Ind can predict nearby locations of a protected POI as prediction results to maintain its usage.

TrajExtract Figures 6.22b and 6.23b show that all the defenses can well protect location (sub-)sequences from being extracted by TrajExtract. This is because sequence-level extraction is a challenging task that pretty much relies on memorization.

LocMIA Figures 6.22c and 6.23c show that none of the existing defenses can be used to protect user-location membership information. While JFT (A) reduces the TPR@10%FPR to less than 20%, it significantly sacrifices the model's utility. The reason is that the defender needs to redact a large number of user-location pairs so as to protect them. As a result, the model may learn from wrong sequential information in the phase-I training, leading to a large utility drop. Even for protecting the targeted subset with 30% of total user-location pairs only, there's still a 20% drop in utility.

TrajMIA Figures 6.22d and 6.23d show the utility-privacy trade-off of different defenses against

TrajMIA We notice that JFT (T) can effectively mitigate the MIA on the targeted subset of trajectory sequences with a small degradation in accuracy. However, the utility drop is still large if a defender aims to protect the membership information of all trajectory sequences.

Summary Existing defenses provide a certain degree of guarantee in mitigating the privacy risks of ML-based POI recommendations. However, it is still challenging to remove all the vulnerabilities within a reasonable utility drop. This is because existing POI recommendation models heavily rely on memorizing user-specific trajectory patterns to make predictions, which lack semantic information as guidance. As a result, defense mechanisms such as DP-SGD can easily compromise the utility of the protected model due to the noises added to the gradients. Moreover, defenses such as JFT are not general for all inference attacks since each attack steals different sensitive information. To this end, our evaluation calls for more advanced mechanisms to defend against our attacks.

6.5.3.3 Additional Definitions in Defense

Definition 1 ((ϵ, δ)-DP)**.** *A randomized mechanism $\mathcal{A}$ satisfies (ϵ, δ)-DP if and only if for any two adjacent datasets D and D', we have:*

$$\forall O \in Range(\mathcal{A}) : Pr[\mathcal{A}(D) \in O] \leq e^{\epsilon} Pr[\mathcal{A}(D') \in O] + \delta$$

where $Range(A)$ indicates the set of all possible outcomes of mechanism $\mathcal{A}$ and δ indicates the possibility that plain ϵ-differential privacy is broken.

Definition 2 (geo-indistinguishability)**.** *A mechanism $\mathcal{A}$ satisfies ϵ-geo-indistinguishability iff for all l and l' , we have:*

$$d_{\mathcal{P}}(\mathcal{A}(l), \mathcal{A}(l')) \leq \epsilon d(l, l')$$

where d denotes the Euclidean metric and $d_{\mathcal{P}}$ denotes the distance between two output distributions. Enjoying ϵr-privacy within r indicates that for any l and l' such that $d(l, l') \leq r$, mechanism $\mathcal{A}$ satisfies ϵ-geo-indistinguishability.

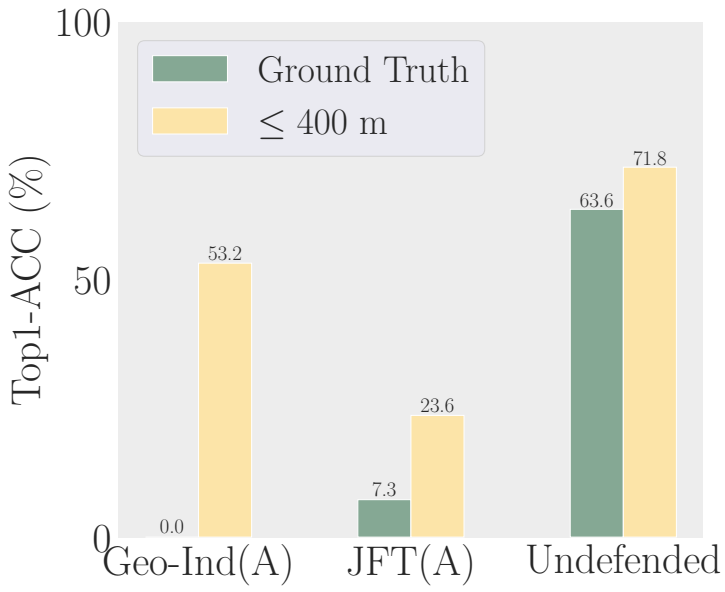


Figure 6.24: For LocExtract, when ground truths are the most common locations, Geo-Ind (A) can predict a nearby location ($\leq$ 400 m) of each protected location as the next POI with higher accuracy than JFT (A). The reason is that Geo-Ind applies the Laplacian mechanism to replace each protected POI with its nearby location.

6.6 Discussion

6.6.1 Broader Impacts

- Our attack suite unveils the vulnerability of POI recommendation models to privacy attacks. Our attack mechanisms (e.g., the Spatial-Temporal Query, which better utilizes spatial-temporal information to boost attack performance) and findings (e.g., identifying the types of data more likely to be memorized by the model) also benefit the measurement of privacy leakage for a broader range of learning-based models that involve spatio-temporal data, e.g., pre-trained trajectory models [DXW22] and spatio-temporal LLMs [ZAL23].

- The defense results indicate that selectively protecting sensitive information and using approaches like fine-tuning pre-trained models on a smaller private dataset might improve the utility-privacy trade-off. In other words, our study reveals that selective defense serves as a promising direction to reduce the privacy risks of these models.

6.6.2 Related Work

Privacy Attacks. Various types of privacy attacks, such as membership inference attacks [SSS17b, SZH18, CCN22, GAA23], training data extraction attacks [CLE19, CHN23], and model inversion attacks [FJR15a] have been proposed to infer sensitive information from model training data. Our attack suite contains membership inference and data extraction attacks. Existing data extraction and membership inference attacks [CLE19, CCN22] are insufficient for POI recommendation models due to the spatio-temporal nature of the data. Our work takes the first step to extracting sensitive location and trajectory patterns from POI recommendation models and solving unique challenges to infer the membership of both user-location pairs and user trajectories. As a final remark, our attacks differ from previous MIAs in mobility data [PTD17, ZZZ20], which focus on the privacy risks of data aggregation.

Mobility Data Privacy. Mobility data contain rich information that can reveal individual privacy such as user identity. Previous work utilizes side-channel attacks to extract sensitive information about mobility data from LBS, including social relationships [SH12, VFB11, XCZ14], trajectory history [Kru07, GP09, HGX06], network packets [VHB14, JWH07] and location embeddings [DXW22]. Despite the focus of previous work, deep neural networks (DNN) built on large volumes of mobility data have recently become state-of-the-art backbones for LBS, opening a new surface for privacy attacks. To the best of our knowledge, our work is the first of its kind to investigate the vulnerabilities of DNN models in leaking sensitive information about mobility data using inference attacks.

More Related Works. We discuss further related work on defenses against privacy attacks in Section 6.6.3.

6.6.3 Related Work on Defenses

Defenses against privacy attacks on mobility data. As discussed in Section 6.6.2, there have been various studies on stealing sensitive information from mobility data. Consequently, researchers have also explored various approaches to safeguard the privacy of mobility data, including K-Anonymity [GL05, GG03], which aims to generalize sensitive locations by grouping them with

other locations, Location Spoofing [BCP14, HSI16], which involves sending both real and dummy locations to deceive adversaries, Geo-indistinguishability [YXM22, ABC13], and local differential privacy (LDP) [XML23, BXZ21]. However, these prior defense mechanisms primarily focus on data aggregation and release processes and can not be directly used in the context of POI recommendation. In contrast, our work is the first to concentrate on protecting against privacy breaches originating from deep learning models such as POI recommendation models and that's why we are primarily focusing on testing defense mechanisms related to the model training process as mentioned in Section 6.5.3.

Defenses against privacy attacks on deep neural networks. There are also multiple works on protecting the privacy of deep learning models, with some notable examples including regularization and early stopping, which are commonly employed techniques to mitigate overfitting [GBC16]. Another approach is differentially private stochastic gradient descent (DP-SGD) [ACG16], which achieves differential privacy by introducing noise during the gradient descent process while training the model. Additionally, selective differential privacy (S-DP) has been proposed to safeguard the privacy of specific subsets of a dataset with a guarantee of differential privacy [SCL22, SCZ22]. However, these methods have primarily been tested on image or language-related models and require customization to fit into the usage of POI recommendation models. In our work, we focus on adapting these defense mechanisms to POI recommendation models by developing privacy definitions that are specifically tailored to the attacks we propose. In addition, we follow the concept of selective DP [SCZ22] to relax the original DP and selectively protect the sensitive information (e.g., most common locations) in mobility data.

6.7 Chapter Summary

This chapter demonstrated that state-of-the-art POI recommendation models are substantially vulnerable to model-centric privacy attacks. Our tailored attack suite—combining data extraction and Membership Inference Attacks at both the location and trajectory levels—revealed consistent and measurable leakage across representative model architectures and real-world datasets: the model exposes sensitive locations in 32.5% of random queries despite a training presence of only 18.7%,

and extraction attacks outperform random baselines by multiple orders of magnitude. A key finding is the "outlier effect": atypical user behaviors and rare trajectory patterns are disproportionately memorized by the model, creating asymmetric privacy risks for users whose mobility patterns fall outside the norm. Existing defense mechanisms, when tested, failed to meaningfully reduce attack success without unacceptable degradation of model utility.

These results establish a clear pattern: neural networks trained on sensitive sequential data memorize more than is necessary for their task. This memorization is not accidental noise—it is exploitable by a black-box adversary who has access only to the model's outputs. Chapter 7 extends this auditing framework to the generative side of the Cognitive Intelligence Pillar, examining zero-shot Text-to-Speech models. Generative models introduce distinct methodological challenges—continuous audio outputs, dual-conditioning architectures, and high generative variance—that require a new approach to membership inference.

CHAPTER 7

Auditing Generative Models: Privacy Boundaries in Biometric Content Generation

Building upon the empirical auditing of spatial-temporal predictive models in Chapter 6, this chapter extends our privacy assessment framework to the generative capabilities of the **Cognitive Intelligence Pillar**. In the Metaverse, synthesizing highly expressive and personalized digital avatars increasingly relies on the rapid advancement of zero-shot Text-to-Speech (TTS) foundation models [WCW23, SJT23]. Fine-tuning these centralized models on private speech corpora introduces a critical dual privacy risk: it threatens the exposure of both immutable biometric identities, which enable sophisticated voice clones, and sensitive linguistic content (e.g., private conversations or medical consultations) that directly exposes users' personal lives [Bac25, TWV22, WTC24]. Rigorously auditing the biometric privacy boundaries of these models is therefore a critical research imperative.

While Membership Inference Attacks (MIAs) served as a highly effective metric for quantifying spatial-temporal leakage in Chapter 6, evaluating zero-shot generative speech models presents unique, unaddressed challenges. Prior MIA frameworks have predominantly targeted discriminative tasks or large language models that output categorical token probabilities [WYL22, CCN22], whereas TTS is a generative task whose output is a continuous, variable-length acoustic waveform. Adapting MIA to this paradigm introduces three intertwined obstacles. First, an inherent input mismatch arises because modern zero-shot TTS is dual-conditioned on a text prompt and a reference audio: querying with generic inputs forces the model into an exploratory zero-shot adaptation state whose high-variance prosodic drift masks the subtle signals of training-time memorization. Second, raw acoustic features such as Mel-spectrograms and MFCCs lack the resolution to disentangle memorization traces from surface phonetic and prosodic content. Third, the continuous waveforms generated by

TTS carry no explicit confidence scores, and naive frame-level comparison is dominated by benign temporal misalignment, rendering point-to-point distance metrics hypersensitive to micro-prosodic variation.

To bridge this methodological gap, this chapter presents a systematic, black-box MIA framework tailored to zero-shot TTS. We formalize the auditing process as a one-bit transmission Markov chain and prove that the attack error is information-theoretically lower-bounded by the mutual information between the target record and the model's generated output (Theorem 1). The bound identifies, across the dual-modal query space, a reconstruction-based query (one that pairs the exact target text with the exact target reference audio) as the information-theoretically optimal choice, rather than the inference-style or training-style queries inherited from prior work. Recognizing the inherent heterogeneity of audio memorization, we further decompose the attack into two complementary granularities and quantify leakage by projecting deterministic generated outputs into robust representation spaces: speaker-level attacks isolate stable identity priors via temporally concatenated speaker-verification embeddings; record-level attacks anchor variable-length generated audio to the target time axis through a modified Dynamic Time Warping procedure that warps onto the target frame index, and aggregate the aligned multi-level WavLM hidden states into a sequence-specific memorization signal.

Extensive empirical evaluation across three representative zero-shot TTS architectures (CosyVoice2, F5-TTS, XTTS-v2) fine-tuned on two datasets (VCTK and British Dialect) demonstrates the framework's efficacy. speaker-level attacks reach AUC up to 1.000 in the strongest configurations and remain above 0.80 throughout; record-level attacks consistently achieve AUC between 0.80 and 0.90 and remain effective even when non-member records are drawn from speakers already seen during fine-tuning. A finer analysis identifies which characteristics of the underlying speech data disproportionately amplify vulnerability, namely low silence ratio, dense phoneme structure, rich high-frequency content, longer duration, and higher speech energy. By rigorously demarcating these biometric boundaries, this chapter establishes an essential foundation for privacy-preserving audio generation in spatial computing ecosystems.

In summary, the main contributions of this chapter are as follows:

- **An information-theoretic foundation for black-box TTS auditing.** We model membership inference as one-bit transmission through a dual-conditioned generative channel and prove (Theorem 1) that attack error is lower-bounded by the mutual information between the target record and the model's output. The bound identifies a reconstruction-based query as information-theoretically optimal across the dual-modal query space.

- **A dual-granularity black-box MIA pipeline for zero-shot TTS.** We instantiate the framework at Speaker- and record-level granularities with tailored representation extractors (temporally concatenated speaker-verification embeddings for global identity leakage, and multi-level WavLM hidden states for sequence-specific memorization), together with a modified DTW procedure that anchors variable-length generated outputs to the target time axis.

- **An empirical privacy audit across three TTS architectures and two datasets.** We show that fine-tuned zero-shot TTS systems exhibit severe membership leakage at both granularities, with speaker-level attacks reaching AUC up to 1.000 in the strongest configurations and record-level attacks consistently exceeding AUC 0.80. The record-level attack remains effective even when non-members are restricted to speakers seen during fine-tuning.

- **A characterization of which speech properties drive vulnerability.** Through systematic factor analyses we show that low-silence-ratio, phonemically dense, high-frequency-rich, longer, and higher-energy utterances are disproportionately memorized, providing actionable signals for privacy-aware curation of fine-tuning corpora.

7.1 Background

This section covers the background of our work: the fine-tuning paradigm of modern TTS models (Section 7.1.1), the black-box membership inference attack (MIA) framework (Section 7.1.2), and the threat model for our MIAs on TTS (Section 7.1.3).

7.1.1 Text-to-Speech (TTS) Models

TTS synthesis converts written text into natural-sounding speech with the voice characteristics of a target speaker, and it serves as a core component in voice assistants, audiobook narration, and accessibility tools [WMT18, TQS21, KMP24]. Early TTS systems, such as Tacotron2 [SPW18], NaturalSpeech2 [LZ20], and VITS [KKS21], typically synthesize speech for a closed set of pre-registered speakers. Modern TTS foundation models have moved beyond this limitation. By conditioning on a short reference utterance s_{ref}, these models can synthesize speech that matches the voice of an arbitrary speaker without dedicated speaker-specific fine-tuning. This capability has enabled broad applications in personalized voice generation, voice cloning, and content creation. Formally, given a target text sequence t and a reference utterance s_{ref}, it generates a waveform s_{gen} as follows:

$$s_{\text{gen}} = \mathcal{F}(t, s_{\text{ref}}; \theta) \tag{7.1}$$

where $\mathcal{F}$ denotes a generative model parameterized by θ that is conditioned on s_{ref} to preserve the acoustic characteristics of the target speaker.

Modern TTS architectures can be broadly categorized as autoregressive (AR), non-autoregressive (NAR), and hybrid models [XRZ25, AS25]. AR models, such as VALL-E [WCW23], XTTS-v2 [CDG24], and Qwen3-TTS-12Hz [HZH26], generate speech token by token. They are effective at modeling fine-grained prosody and natural expressiveness, but often incur higher inference latency. Non-autoregressive models usually employ diffusion or flow-matching decoders to directly map text to mel-spectrograms, with representative examples including Voicebox [LVS23], E2-TTS [EWT24], and F5-TTS [CNM25]. These models enable faster inference and stronger duration control, although they may reduce stylistic diversity in some settings. Hybrid models, such as CosyVoice [DCZ24, DWC24], Chatterbox TTS [Res25], and Qwen3-TTS-25Hz [HZH26], combine autoregressive and non-autoregressive components to balance expressiveness and efficiency. In this work, we evaluate the privacy vulnerabilities of fine-tuned TTS systems using three representative and widely-used models.

7.1.2 Membership Inference Attacks

Membership Inference Attack (MIA) is a prominent framework for auditing the privacy leakage of machine learning models. Its primary objective is to determine whether a specific data record x was included in the target model's training dataset. In practical deployment scenarios, adversaries typically operate under a black-box threat model. Under this setting, the adversary cannot observe the model's internal parameters or gradients, and must instead rely entirely on the observable outputs to perform inference.

Recent studies on conditional generative models often formulate black-box MIAs as a score-based decision process [PW23]. Rather than querying the target model $\mathcal{F}$ directly with the exact training record x, the adversary crafts an attack query q to prompt the model. The adversary then derives a scalar membership score $\mathcal{M}(x)$ to quantify the likelihood of x being in the training set. Formally, this score is computed via a feature extraction function $\phi(\cdot)$:

$$\mathcal{M}(x) = \phi\big(x, \mathcal{F}(q)\big) \tag{7.2}$$

where $\mathcal{F}(q)$ represents the model's generated output in response to the query. The function $\phi(\cdot)$ serves to capture the similarity, reconstruction error, or structural representation gap between the original target sample x and the generated output. The final membership decision is made by comparing $\mathcal{M}(x)$ against a predefined threshold τ. The effectiveness of the attack critically depends on designing a robust $\phi(\cdot)$ that captures the model's disparate behavior on seen versus unseen data.

Applying this framework to TTS is non-trivial due to three challenges. First, the dual-modality input space (text plus reference audio) makes it unclear how to construct a query that effectively triggers memorization. Second, TTS outputs are continuous waveforms in which linguistic content, speaker identity, and prosody are heavily entangled, making it difficult to isolate subtle membership traces from dominant generic content. Third, the generated waveforms are variable-length with stochastic timing, so the target record and the model's output cannot be directly compared via point-to-point distance metrics.

7.1.3 Threat Model

Target Scenario. We focus on the vulnerability of TTS models during the fine-tuning phase. Generative TTS models are frequently fine-tuned on private, proprietary, or copyrighted voice datasets ($\mathcal{D}_{\text{train}}$) to adapt them to specific domains, languages, or premium speaker identities. The target system is a fine-tuned TTS model $\mathcal{F}$, and the dataset $\mathcal{D}_{\text{train}}$ is kept confidential by the model provider.

Adversarial Goal. The adversary acts as an external auditor (e.g., a privacy regulator, a copyright holder, or a security researcher) aiming to assess the privacy leakage of $\mathcal{F}$. To achieve this, their primary objective is to conduct a Membership Inference Attack that determines whether a target record x or a target speaker identity was included in $\mathcal{D}_{\text{train}}$. Functionally, this attack is framed as a strict binary classification task (member versus non-member) aimed at verifying unauthorized data usage or compliance violations.

Adversary's Capabilities. We assume a restrictive, yet highly practical, black-box setting. The adversary has no knowledge of the model's internal weights, gradients, or the hyperparameters used during the fine-tuning process. The adversary can only interact with $\mathcal{F}$ through its public inference API. Specifically, the adversary can submit multiple attack queries $q = (t, s_{\text{ref}})$, consisting of a text prompt and a reference audio, and observe the generated continuous acoustic waveform $s_{\text{gen}} = \mathcal{F}(q)$. The adversary cannot access internal confidence scores, token-level probabilities, or intermediate representations. However, consistent with realistic auditing scenarios, we assume the adversary has access to standard computational resources and publicly available auxiliary tools. Specifically, the adversary can leverage open-source, pre-trained embedding models as surrogate feature extractors to process and analyze the generated acoustic outputs post-hoc.

Auditing Granularities. Auditing TTS relies on conditioning variables that often differ from the training audio. We formalize this threat model into two distinct levels, operating under different assumptions of adversarial knowledge (summarized in Table 7.1):

(1) speaker-level MIA. The adversary aims to determine if any data belonging to a target speaker $\text{spk}_{\text{target}}$ is in $\mathcal{D}_{\text{train}}$. Here, we assume the adversary possesses n distinct audio clips of $\text{spk}_{\text{target}}$, without knowing if they overlap with the training set. This represents a realistic scenario where a

public figure suspects unauthorized voice cloning and uses public recordings as queries to detect the memorization of their general acoustic profile.

(2) record-level MIA. The adversary aims to determine if a specific audio-text record $x = (t_{\text{target}}, s_{\text{target}})$ is in $\mathcal{D}_{\text{train}}$. We assume the adversary possesses the exact target record x. This is a more challenging, fine-grained task with assumptions aligning with standard MIA, representing a scenario where a user audits the unauthorized use of a specific private recording (e.g., a leaked voice message) during fine-tuning.

Table 7.1: Summary of threat model assumptions at different auditing granularities.

Attack	**Adversary Objective**	**Adversarial Knowledge**
speaker-level MIA	Detect if a target speaker $\text{spk}_{\text{target}} \in \mathcal{D}_{\text{train}}$.	A set of n reference utterances from $\text{spk}_{\text{target}}$.
record-level MIA	Detect if the target record $x \in \mathcal{D}_{\text{train}}$.	A target text-audio pair $x = (t_{\text{target}}, s_{\text{target}})$.

7.2 Methodology

In this section, we present our black-box MIA framework tailored for fine-tuned TTS models. We first identify the challenges of naïve adaptation (Section 7.2.1), then address them with a reconstruction-based query generator G_Q (Section 7.2.2) and an acoustic representation engineering pipeline $\phi(\cdot, \cdot)$ (Section 7.2.3).

7.2.1 Challenges of Naïve Adaptation

In standard generative models, a black-box membership inference attack (MIA) typically follows a three-step pipeline: (1) *Query Formulation* to probe the model, (2) *Representation Engineering* to extract and compare output features, and (3) *Scoring and Decision* to assert membership via a threshold. However, adapting this standard pipeline to Text-to-Speech (TTS) models is challenging, particularly in the first two stages:

In the first stage (*Query Formulation*), previous generative models benefit from a predictable

input-output mapping, which allows adversaries to easily design queries by mimicking standard inference. TTS models, however, rely on cross-modal dual-conditioning (text and reference audio). This creates a massive search space, making it unclear how to design an optimal query. For instance, an intuitive approach is to follow normal TTS inference: pairing the target audio with a mismatched text. However, our information-theoretic analysis shows this naive approach is sub-optimal for extracting membership signals. We detail the reasons for this failure and our optimal query design in Section 7.2.2.

In the second stage (*Representation Engineering*), extracting a clean membership signal from the model's raw output is inherently difficult for TTS because speech waveforms are highly entangled. Through our experiments, we found that standard acoustic representations like Mel-spectrograms [SVN37] mostly capture basic phonetic and voice information. Without higher-level abstraction, they fail to isolate the subtle acoustic artifacts needed to reliably separate members from non-members. Moreover, unlike text (discrete tokens) or images (fixed-size grids) that allow direct point-to-point comparisons, TTS models generate continuous, variable-length waveforms with inherent temporal stochasticity (e.g., varying durations and pacing). This misalignment makes naive frame-by-frame comparison impossible. Thus, accurately aligning these outputs while preserving the fragile acoustic membership artifacts is a critical challenge. We detail our joint representation extraction and alignment strategies in Section 7.2.3, and the framework throughout operates under an m-query sampling strategy, repeating each query m times to stabilize stochastic generation noise before scoring.

7.2.2 Query Generation (G_Q)

As highlighted by prior works, the effectiveness of membership inference depends not only on model memorization, but also on the conditions under which the model is queried [CZH24, NPS25]. However, it remains an open question how to systematically design and quantify such query conditions within the specific context of TTS models, which rely on a combination of audio and text inputs. In this section, we will ground our approach in information theory, providing a theoretical framework to systematically design and justify queries in this dual-modal setting.

7.2.2.1 Theory Foundation

To establish a unified framework for the information-theoretic analysis of diverse query-generation strategies, we model the membership inference process as a *Bits Transmission* process. By treating dual-modality query conditions as transmitted information, this formulation enables a principled analysis of how query constraints expose memorization in the model. In this paradigm, the i-th latent membership indicator of the evaluation set is modeled as a Bernoulli random variable $B_i \in \{0, 1\}$. The membership inference process is modeled as a Markov chain:

$$B_i \xrightarrow{H} X_i \xrightarrow{G_Q} Q_i \xrightarrow{\mathcal{F}} O_i \xrightarrow{D} \hat{B}_i \tag{7.3}$$

where H denotes the dataset generator that maps the latent bit B_i to an audit record $X_i = (t_{\text{target}}, s_{\text{target}})$; G_Q denotes the query generator that constructs the query Q_i from the audit record; $\mathcal{F}$ denotes the target TTS model that takes text and audio inputs and generates the output audio O_i; and D denotes the adversary decoder that produces the estimate $\hat{B}_i$.

In Theorem 1, we first show that the attack error rate is bounded by the mutual information (MI) between the target record X_i and the generated audio O_i.

Theorem 1 (Bit error lower bound)**.** *Based on the Markov modeling shown in Eq. 7.3, let* $B_i \in \{0, 1\}$ *be a balanced Bernoulli random variable with* $\Pr(B_i = 1) = \Pr(B_i = 0) = 1/2$*. Let*

$$P_i^e = \Pr(\hat{B}_i \neq B_i)$$

denote the membership inference error. Then the attack error is lower-bounded by the information preserved through the query generation stage:

$$P_i^e \geq h^{-1}(1 - MI(X_i; O_i)) \,, \tag{7.4}$$

where $h^{-1} : [0, 1] \to [0, 1/2]$ *denotes the inverse of the left branch of the binary entropy function.*

The proof combines two standard information-theoretic arguments: Fano's inequality, which relates the binary inference error to the information retained about the true membership bit, and the Data Processing Inequality, which bounds the information preserved along the audit chain.

Proof. The audit process forms the Markov chain in Eq. 7.3. By the Data Processing Inequality applied to this chain,

$$MI(B_i; \hat{B}_i) \leq MI(B_i; O_i) \leq MI(X_i; O_i).$$

Since $B_i \in \{0, 1\}$ is balanced, $H(B_i) = h(1/2) = 1$. The mutual information between B_i and its estimate $\hat{B}_i$ satisfies

$$\begin{aligned} MI(B_i; \hat{B}_i) &= H(B_i) - H(B_i \mid \hat{B}_i) \\ &= 1 - H(B_i \mid \hat{B}_i). \end{aligned}$$

Let $E_i = \mathbb{I}[\hat{B}_i \neq B_i]$ denote the error indicator. Given $\hat{B}_i$, the true bit B_i is in one-to-one correspondence with E_i via $B_i = \hat{B}_i \oplus E_i$, so $H(B_i \mid \hat{B}_i) = H(E_i \mid \hat{B}_i)$. Therefore,

$$\begin{aligned} MI(B_i; \hat{B}_i) &= 1 - H(E_i \mid \hat{B}_i) \\ &\geq 1 - H(E_i) \\ &= 1 - h(P_i^e), \end{aligned}$$

where the inequality follows from the fact that conditioning reduces entropy, and the last equality holds because $E_i \sim \text{Bernoulli}(P_i^e)$ so $H(E_i) = h(P_i^e)$, with h denoting the binary entropy function. Rearranging gives

$$P_i^e \geq h^{-1}\left(1 - MI(B_i; \hat{B}_i)\right),$$

where $h^{-1} : [0, 1] \to [0, 1/2]$ denotes the inverse of the left branch of the binary entropy function. Since h^{-1} is non-increasing on $[0, 1]$, combining both steps gives

$$P_i^e \;\geq\; h^{-1}\big(1 - MI(B_i; \hat{B}_i)\big) \;\geq\; h^{-1}\big(1 - MI(X_i; O_i)\big).$$

□

7.2.2.2 Optimizing the Mutual Information

As governed by the Markov chain established in Equation 7.3, the best MIA error is determined by $MI(X_i; O_i)$. Since the adversary lacks the capability to alter the target TTS model, optimizing the query generator G_Q is the sole mechanism to maximize this mutual information within the attack pipeline. This mutual information can be expressed as:

$$MI(X_i; O_i) = H(X_i) - H(X_i|O_i) \tag{7.5}$$

where $H(X_i)$ denotes the inherent entropy of the specific audit record. Because $H(X_i)$ is a fixed constant for any given target, maximizing the mutual information is mathematically equivalent to minimizing the conditional entropy $H(X_i|O_i)$. Therefore, the optimal query strategy must minimize the remaining uncertainty about the audit record X_i given the model's synthesized output O_i.

To determine what query strategy achieves this minimization, we decompose the audit record, the query, and the synthesized output into their respective textual (T) and acoustic components (S): $X_i = (T_x, S_x)$, $Q_i = (T_q, S_q)$, and $O_i = (T_o, S_o)$. Our goal is to evaluate the uncertainty across three distinct paradigms of input alignment.

$$H(X_i|O_i) = H(T_x, S_x|T_o, S_o) \tag{7.6}$$

Before evaluating these paradigms, we formalize an observation regarding the target TTS model's behavior: a capable TTS system faithfully aligns its synthesized output with both the semantic content of the input text and the acoustic style of the reference audio. This aligns with the primary training objective of modern TTS models, which are optimized to synthesize the provided text while preserving the acoustic attributes specified by the reference audio (e.g., speaker identity, prosody, emotion, and speaking style). Consequently, the generated output satisfies $T_o \approx T_q$ and $S_o \sim S_q$.

Paradigm 1: Mismatched Text ($T_q \neq T_x$ while $S_q = S_x$). This represents the standard use cases of TTS models during inference. Here, the output text matches the new query T_q ($T_o \approx T_q \neq T_x$) but it offers no useful clues about the original text T_x. Furthermore, because the model must pronounce different words, the generated audio features in S_o cannot match the original target audio S_x. Even when the global speaker identity is preserved, this semantic change distorts the fine-grained acoustic evidence needed for MIA. By losing both the correct text and acoustic voiceprint, the joint conditional entropy $H_{P1}(T_x, S_x|T_o, S_o)$ stays strictly large.

Paradigm 2: Mismatched Audio ($T_q = T_x$ while $S_q \neq S_x$). This mimics how TTS models are conditioned during training. The model outputs the correct text ($T_o \approx T_x$), but its generated audio inherits the style of the mismatched reference audio ($S_o \sim S_q \neq S_x$). While this successfully resolves the textual uncertainty $H(T_x|O_i)$, it distorts the audio match. The generated audio S_o adopts the noise or prosody of the perturbed prompt rather than the true target audio S_x. Because MIA requires a comparison of acoustic fingerprints, this stylistic degradation destroys the necessary evidence.

Consequently, even with the text resolved, the persistent acoustic uncertainty $H(S_x|O_i)$ keeps the joint conditional entropy $H_{P2}(T_x, S_x|T_o, S_o)$ large.

Paradigm 3: Exact Match ($T_q = T_x$ and $S_q = S_x$). This setup acts as a *reconstruction-based query*. Providing the exact text avoids the textual mismatch of Paradigm 1, while providing the exact acoustic components avoids the style loss of Paradigm 2. Because the fundamental objective of TTS training is to minimize reconstruction error, feeding the model with exactly matched inputs forces it to faithfully reproduce the training data. As a result, the output O_i becomes a close replica of the target X_i ($T_o \approx T_x$ and $S_o \sim S_x$). Because this replica preserves both the specific words and the granular vocal details, it provides highly reliable evidence about the target X_i. This preservation resolves most of the uncertainty, naturally driving the joint conditional entropy $H_{P3}(T_x, S_x|T_o, S_o)$ toward zero.

Synthesizing these observations, Paradigm 3 successfully reduces the conditional entropy to near zero, whereas the information loss in Paradigms 1 and 2 keeps their entropy strictly greater than zero. The following inequalities formalize this entropy gap:

$$H_{P3}(X_i|O_i) < H_{P1}(X_i|O_i), \tag{7.7}$$

$$H_{P3}(X_i|O_i) < H_{P2}(X_i|O_i). \tag{7.8}$$

Recall that $MI(X_i; O_i) = H(X_i) - H(X_i|O_i)$. Minimizing this conditional entropy directly maximizes the mutual information upper bound. By utilizing a reconstruction-based query to preserve the full dual-modality fingerprint, Paradigm 3 achieves this theoretical minimum. Therefore, we expect it to yield the highest MIA performance, which we empirically validate in Section 7.3.3.

7.2.3 Representation Engineering (ϕ)

In this section, we detail the design of our representation engineering module (ϕ). To enable successful MIAs, we address two fundamental challenges: first, we introduce tailored representation extractors to capture highly discriminative membership signals (Section 7.2.3.1); second, we design representation alignment techniques to resolve the inherent temporal misalignment of continuous audio (Section 7.2.3.2).

7.2.3.1 Representation Extraction

To overcome the limitations of low-level acoustic representations (e.g., Mel-spectrograms [SVN37] or MFCCs [DM80]) discussed earlier, we design tailored representation extractors aligned with the attack objectives: distilling stable global identity vectors via a speaker-verification (SV) model for speaker-level MIA, and preserving prosodic artifacts via a self-supervised model for record-level MIA.

Representation Extraction for speaker-level MIA. speaker-level MIA fundamentally relies on detecting a speaker's unique identity. This requires extracting a highly discriminative "identity voiceprint" from audio signals. Pre-trained SV models are well suited for this task, as they are optimized to condense high-dimensional audio into an identity-focused latent space [SGS18, DTD20]. Accordingly, our evaluation function $\phi_{\text{spk}}(\cdot)$ relies on a WavLM + ECAPA-TDNN SV encoder [CCW22] ($\mathcal{E}_{\text{SV}}$). Extracting features through this specialized architecture provides a robust representation for membership inference.

Because SV systems are known to perform poorly on short audio segments [LZ20], averaging embeddings from individual clips fails to capture stable speaker traits. To address this limitation, we leverage the adversary's access to n target samples as defined in our threat model. Specifically, we temporally concatenate all n audio sequences generated by querying the target model with these samples before extracting features. This allows the SV model to process the entire sequence jointly. Specifically, we compute the fixed-dimensional global speaker embeddings, $\mathbf{e}_{\text{target}}$ and $\mathbf{e}_{\text{gen}}$, by passing the concatenated queries through the SV encoder $\mathcal{E}_{\text{SV}}$:

$$\begin{aligned} \mathbf{e}_{\text{target}} &= \mathcal{E}_{\text{SV}}(\text{Concat}(s_1, \ldots, s_n)) \in \mathbb{R}^d, \\ \mathbf{e}_{\text{gen}} &= \mathcal{E}_{\text{SV}}(\text{Concat}(s_{\text{gen},1}, \ldots, s_{\text{gen},n})) \in \mathbb{R}^d \end{aligned} \tag{7.9}$$

where d denotes the embedding dimensionality. By translating the variable-length audio into these stable, fixed-dimensional vectors, we establish a reliable basis for computing the membership score in subsequent steps.

Representation Extraction for record-level MIA. Detecting record-level memorization is more challenging than the speaker attack. It requires capturing fine-grained, sequence-specific artifacts (e.g., exact phonetic alignment and micro-prosodic inflections) rather than a global identity invariant

like speaker identity. Conversely, representations from SV models are also insufficient for this task. Because these models are explicitly optimized to discard localized linguistic and prosodic variations to isolate an utterance-independent voiceprint, applying them at the record level inadvertently removes the precise trajectory artifacts we aim to detect.

To resolve this representation gap, $\phi_{\text{rec}}(\cdot)$ evaluates trajectory reconstruction with a self-supervised WavLM [CWC22] encoder. The suitability of WavLM for this task is twofold. First, its masked speech prediction objective forces the transformer backbone to model local dependencies frame by frame, preserving the continuous micro-prosodic trajectory characteristic of exact memorization. Second, WavLM encodes a structured feature hierarchy across its $L = 24$ transformer layers, with different layers capturing different types of information, as shown in the original paper. By extracting unpooled hidden states, we avoid the destructive abstraction of top-layer pooling and obtain robust multi-level representations that better capture membership signals. Formally, for an input utterance $x \in \{\text{target}, \text{gen}\}$, we construct the feature sequence as:

$$\mathbf{H}_x = \phi_{\text{rec}}(x) \in \mathbb{R}^{K_x \times |L_{\text{sel}}| \times d} \tag{7.10}$$

where K_x is the temporal frame length, L_{sel} is the set of selected layers, and d is the hidden state dimensionality ($d = 1024$ for WavLM). Specifically, we denote $\mathbf{h}_{x,i}^{(l)} \in \mathbb{R}^d$ as the specific embedding vector extracted at temporal frame i from layer l (where $l \in L_{\text{sel}}$).

7.2.3.2 Representation Temporal Alignment

In prior MIAs against text or image models, generated outputs are directly comparable. In contrast, TTS generates continuous, variable-length waveforms with inherently fluctuating durations and speaking rates, making naive frame-by-frame comparisons invalid. To resolve this temporal misalignment, we introduce distinct alignment strategies tailored to each of our proposed attacks. Specifically, for speaker-level MIA, our feature extractor collapses variable-length audio into fixed-dimensional global vectors for direct comparison. For record-level MIA, where fine-grained trajectory artifacts must be preserved, we design a modified Dynamic Time Warping (DTW) [SC03] mechanism to dynamically align the frame-level sequences.

Representation Alignment for speaker-level MIA. As previously mentioned, our feature extraction

process yields fixed-dimensional vectors ($\mathbf{e}_{\text{target}}$ and $\mathbf{e}_{\text{gen}}$) that explicitly represent the speaker's identity. This fixed dimensionality allows us to directly compare the target and generated audio mathematically. More importantly, because SV models are trained to separate speaker identities based on the angle between their feature vectors, cosine similarity naturally serves as the optimal metric to measure their overlap. Specifically, our evaluation function ϕ_{spk} quantifies this acoustic representation gap as:

$$v = \phi_{\text{spk}}(s_{\text{target}}^{\text{concat}}, s_{\text{gen}}^{\text{concat}}) = \frac{\mathbf{e}_{\text{target}} \cdot \mathbf{e}_{\text{gen}}}{\|\mathbf{e}_{\text{target}}\|_2 \|\mathbf{e}_{\text{gen}}\|_2} \tag{7.11}$$

Because our prior temporal concatenation step effectively stabilizes these features, the score v serves as a highly reliable measure of acoustic similarity and yields a discriminative membership indication to drive our threshold-based decision mechanism (τ).

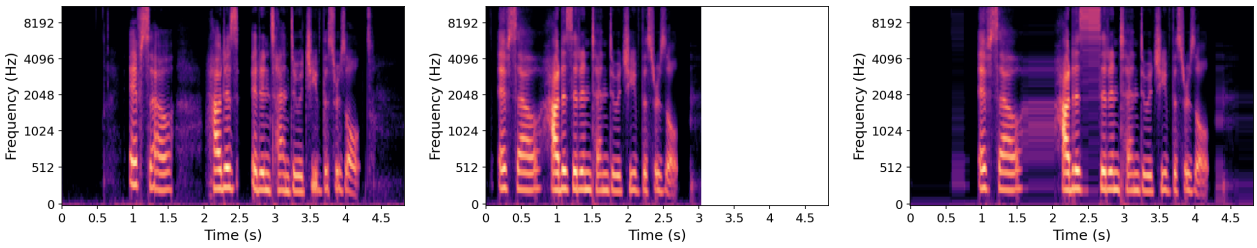
Representation Alignment for record-level MIA. While the unpooled WavLM hidden states provide the frame-level resolution required to detect record-level memorization, the extracted feature of target record ($\mathbf{H}_{\text{target}}$) and generated results ($\mathbf{H}_{\text{gen}}$) cannot be directly compared frame by frame. As illustrated in Figure 7.1, even under identical transcription, TTS models may synthesize the same phonetic sequence with different durations, speaking rates, and local prosodic timing. To obtain fine-grained temporal correspondence and enable consistent comparison across multiple queries, we align $\mathbf{H}_{\text{gen}}$ to $\mathbf{H}_{\text{target}}$ using a modified DTW, which warps the generated features to the time axis of the target utterance.

For each selected layer $l \in L_{\text{sel}}$ independently, we compute a per-layer optimal warping path using cosine distance:

$$\pi^{(l)\star} = \arg\min_{\pi \in \Pi} \sum_{(i,j)\in\pi} \left(1 - \frac{\hat{\mathbf{h}}_{\text{target},i}^{(l)} \cdot \hat{\mathbf{h}}_{\text{gen},j}^{(l)}}{\|\hat{\mathbf{h}}_{\text{target},i}^{(l)}\|_2 \|\hat{\mathbf{h}}_{\text{gen},j}^{(l)}\|_2}\right), \tag{7.12}$$

where $\hat{\mathbf{h}}$ denotes the L2-normalized hidden state and Π denotes the set of valid monotonic DTW paths. Unlike standard DTW, which extracts a similarity sequence along the warping path (producing a variable-length output), our modified DTW uses the path to warp the generated features onto the target time axis. Specifically, for each target frame i, we average all generated frames j that $\pi^{(l)\star}$ maps to:

$$\mathbf{h}_{\text{warped},i}^{(l)} = \frac{1}{\left|\{j : (i,j) \in \pi^{(l)\star}\}\right|} \sum_{\{j : (i,j) \in \pi^{(l)\star}\}} \mathbf{h}_{\text{gen},j}^{(l)}, \quad i = 1, \ldots, K_{\text{target}}. \tag{7.13}$$



(a) Target speech. **(b)** Generated speech. **(c)** Aligned speech.

Figure 7.1: DTW-based temporal alignment for record-level MIA. Even with matched transcriptions, the target speech (a) and the generated speech (b) differ in phoneme duration and prosodic timing. To enable fine-grained comparison, DTW warps the generated speech to produce (c), where corresponding spectral patterns are temporally aligned with the target speech (a).

The layer-wise cosine similarity between the target and the warped generated features is then:

$$s_i^{(l)} = \frac{\mathbf{h}_{\text{target},i}^{(l)} \cdot \mathbf{h}_{\text{warped},i}^{(l)}}{\|\mathbf{h}_{\text{target},i}^{(l)}\|_2 \|\mathbf{h}_{\text{warped},i}^{(l)}\|_2}, \quad i = 1, \ldots, K_{\text{target}}, \tag{7.14}$$

This produces a multivariate temporal similarity sequence $\mathbf{S} \in \mathbb{R}^{K_{\text{target}} \times |L_{\text{sel}}|}$, where K_{target} is the length of the target utterance. By anchoring the output to the target time axis, this warping strategy ensures that the downstream LSTM receives a fixed-length sequence directly aligned with the target acoustic trajectory, eliminating variable-length outputs across utterances and queries.

Having extracted the temporally aligned similarity sequence $\mathbf{S}$, our final objective is to distill this two-dimensional representation into a single record-level score. However, reducing this sequence requires capturing temporal dependencies. Simple frame-wise averaging is suboptimal, as consecutive frames of high hyperspherical similarity provide a substantially stronger memorization signal than isolated matches. To capture this structural continuity, we employ a lightweight Long Short-Term Memory (LSTM) network as a temporal aggregator. Given the similarity sequence $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{K_{\text{target}}}]$, the temporal aggregation to a final score v is formulated as:

$$\mathbf{m}_k = \text{LSTM}(\mathbf{s}_k, \mathbf{m}_{k-1}), \quad v = \sigma(\mathbf{W}^\top \mathbf{m}_{K_{\text{target}}} + b) \tag{7.15}$$

where $\mathbf{m}_k$ represents the final hidden state encompassing the entire sequence context, and σ denotes the projection mapping. Calibrated on an auxiliary dataset, this LSTM processes $\mathbf{S}$ and projects it

into a single trajectory reconstruction score v. By evaluating continuous structural alignment rather than discrete frame similarities, the LSTM robustly quantifies record-level memorization, yielding a high-fidelity scalar for the membership score $\mathcal{M}(x)$.

7.3 Experiment

In the following, we illustrate our experimental setup in Section 7.3.1. We empirically evaluate the proposed attack suite to answer the following research questions: (1) What is the performance of the proposed membership inference attacks against TTS models at both the speaker and record levels? (Section 7.3.2) (2) How do our attack designs including query, feature and alignment improve attack performance? (Sections 7.3.3.1 to 7.3.3.5) (3) How do the attack parameters affect attack performance? (Sections 7.3.3.6 to 7.3.3.9) (4) What speech data characteristics correlate with membership inference vulnerability? (Section 7.3.4).

7.3.1 Experiment Setup

Models. We conduct experiments using the official implementations of three representative TTS systems: F5-TTS [CNM25], XTTS-v2 [CDG24], and CosyVoice2 [DWC24]. These models are widely used in downstream fine-tuning settings and collectively cover diverse architectural paradigms, including flow-matching-based, autoregressive, and hybrid designs. For all experiments, we use publicly available pre-trained checkpoints and follow the official fine-tuning protocols with default hyperparameters. Specifically, F5-TTS uses the F5-TTS-v1-Base model trained on the Emilia dataset [HSW24], XTTS-v2 is pre-trained on a mixture of LibriTTS-R [KZK23], LibriLight [KRZ20], and internal audiobook-style datasets, while CosyVoice2 is initialized from the CosyVoice2-0.5B checkpoint, which is trained on large-scale in-house data.

Fine-tuning Datasets. To guarantee the integrity and effectiveness of our work, we utilize the VCTK [YVM19] and the British Dialects dataset [DKG20] for evaluation, ensuring that there is no overlap with the pre-training dataset.

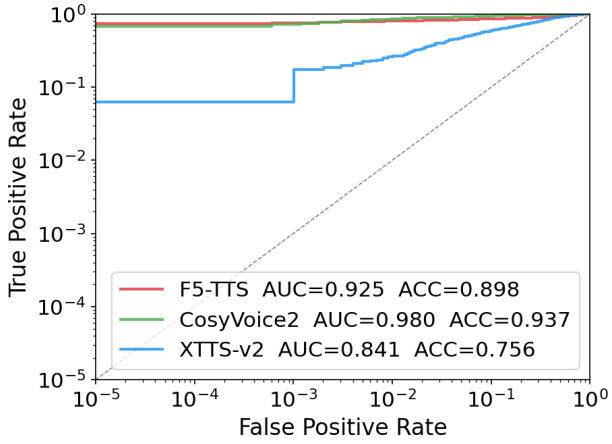
VCTK: This corpus is a multi-speaker dataset designed for the training and evaluation of TTS

systems, encompassing over 44,000 distinct audio samples that total approximately 44 hours of recordings collected from 110 English speakers with diverse regional accents. Within this corpus, each speaker recites approximately 400 distinct sentences sourced from newspaper publications, the Rainbow Passage, and a standard elicitation paragraph formulated to ensure comprehensive phonetic coverage. Consequently, this dataset serves as a standard benchmark for the fine-tuning of clean speech.

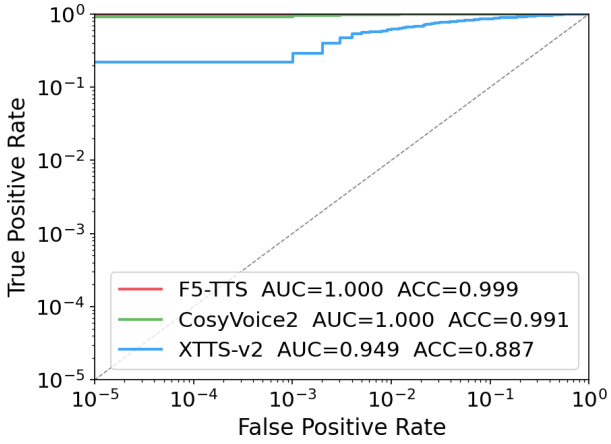
British Dialect Dataset: This collection encompasses 17,877 transcribed utterances, totaling 31 hours of high-quality audio recordings collected from 120 native speakers. These individuals represent six distinct regional varieties, namely, Southern England, Midlands, Northern England, Welsh, Scottish, and Irish English. The corpus utilizes scripts specifically formulated for accent elicitation to ensure comprehensive phonetic coverage. Furthermore, this dataset serves as an optimal use case for the fine-tuning of models on specific speech variations.

Dataset Split. We follow the standard data partitioning protocol in prior membership inference attacks [SSS17b, DKW23], and split the speakers into two disjoint halves for target model training and attack evaluation, respectively. Within each split, speakers are further divided into spk_{in} and spk_{out}. The TTS model is trained on N utterances sampled from spk_{in}. For speaker-level MIA, member samples are re-sampled from spk_{in} with a seed, while non-members are drawn from spk_{out}; for record-level MIA, the training set is treated as members and non-members are sampled from the remaining utterances. We set $N = 5000$ for VCTK and $N = 3000$ for British Dialects.

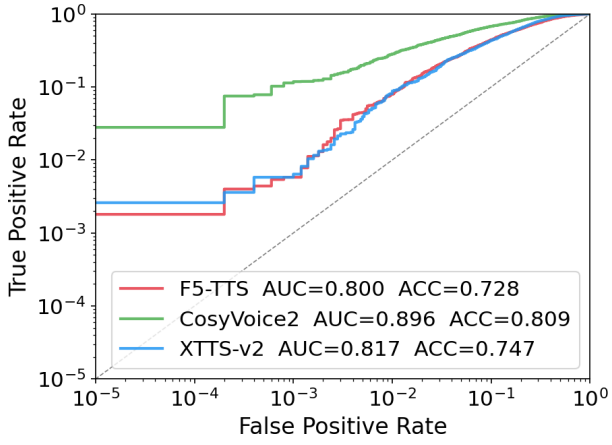
Evaluation Metrics. We adopt the standard metrics for evaluating membership inference attacks, including area under the curve (AUC), average-case "accuracy" (ACC), and true positive rate versus false positive rate in the low-false-positive-rate regime (TPR@1%FPR). Following prior work that argues that meaningful privacy evaluation should emphasize the worst case [CCN22], our primary focus is the TPR–FPR curve in the low false-positive regime, as it better reflects an adversary's ability to confidently identify training members.



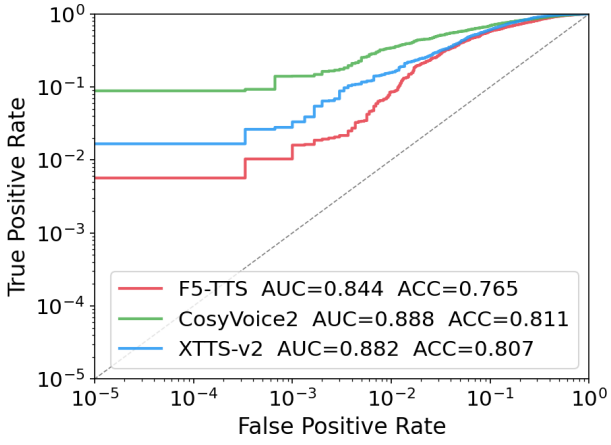
(a) Speaker MIA on VCTK.



(b) Speaker MIA on British Dialect.



(c) Record MIA on VCTK.



(d) Record MIA on British Dialect.

Figure 7.2: Attack performance of speaker-level MIA and record-level MIA on three victim TTS models and two fine-tuning datasets. The diagonal line indicates the random guess baseline.

7.3.2 Attack Performance

We evaluate the proposed attack suite on three TTS models and two datasets. The results show that fine-tuned TTS models leak membership information at both levels. We further show that record-level MIA remains effective in a more challenging setting where the model tries to distinguish records of the same speakers, confirming that the two attacks capture distinct forms of memorization.

7.3.2.1 TTS Models Are Vulnerable at Both Speaker and Record Levels

Figure 7.2 shows the main attack results. Overall, the attacks are highly effective across all three target models and both datasets. Figure 7.2a and Figure 7.2b show that speaker-level MIA is consistently strong, with all AUC values above 0.80 and near-perfect AUC in the strongest settings, namely CosyVoice2 and F5-TTS on the British Dialect dataset. The same trend is visible in the low-FPR region, where TPR remains substantially above chance even under strict false-positive constraints. As shown in Figure 7.2c and Figure 7.2d, record-level MIA is also effective across all models and datasets, with AUC values consistently well above the random-guess baseline of 0.5, although it is weaker than speaker-level MIA. This gap likely arises because speaker membership is reflected in stable identity patterns that persist across utterances, whereas record-level attacks must detect finer content-dependent memorization traces that are more easily weakened by variation in the generation process.

The attack performance also differs across datasets. In general, British Dialect yields stronger attack results than VCTK at both the speaker and record levels. This difference is especially clear for speaker-level MIA, where all three target models achieve higher AUC and TPR@1%FPR on British Dialect (see Figures 7.2a and 7.2b). At the record level, the gap is smaller, but CosyVoice2 and XTTS-v2 still show stronger low-FPR performance on British Dialect (see Figures 7.2c and 7.2d). One possible reason is the richer accent and speaking-style variation in British Dialect: more acoustically distinctive speakers may leave memorization signals that remain more separable after generation.

Attack effectiveness also varies across model architectures, and the vulnerability ranking differs between the two attack levels. For speaker-level MIA, XTTS-v2 is consistently the least vulnerable, with AUC of 0.819 on VCTK and 0.949 on British Dialect (see Figures 7.2a and 7.2b). We attribute this to the fact that XTTS-v2 fine-tuning does not directly adapt the acoustic decoder that maps latent representations to mel-spectrograms or raw waveforms, leaving fewer speaker-identity traces in the generated output. For record-level MIA, F5-TTS yields the lowest AUC on both datasets, achieving 0.800 on VCTK and 0.844 on British Dialect (see Figures 7.2c and 7.2d). A likely reason is that F5-TTS lacks autoregressive modeling, which means that it does not capture the sequential

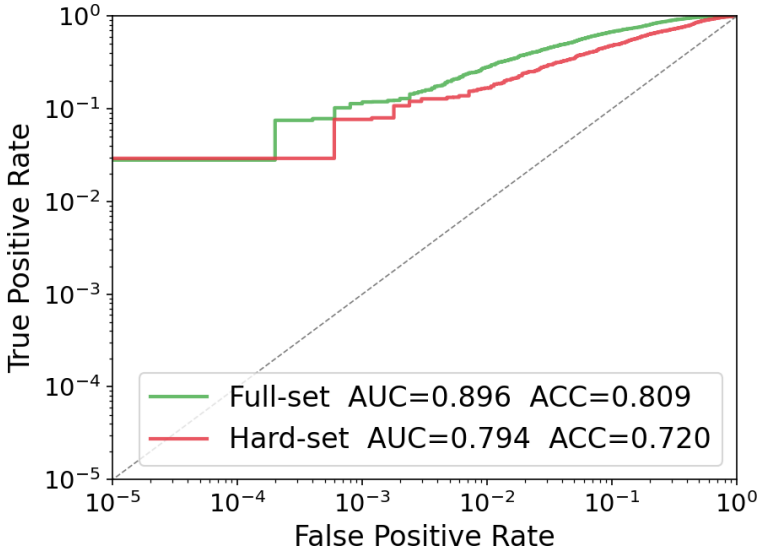
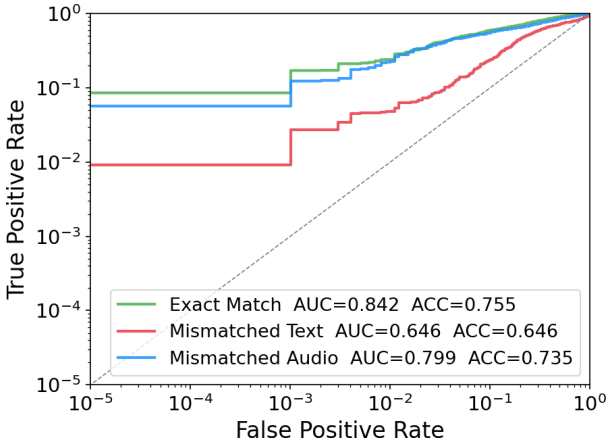


Figure 7.3: record-level MIA maintains comparable low-FPR performance even when non-members are drawn from seen speakers, confirming that the attack captures record-specific memorization beyond speaker identity.

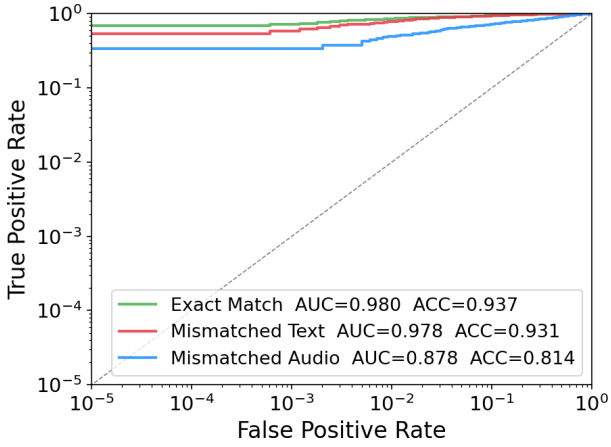
and local content dependencies required by record-level MIA to distinguish memorized records from non-members. Our ablation on the hybrid architecture of CosyVoice2 (Section 7.3.3.6) further supports this picture, showing that speaker- and record-level leakage are driven by different model components.

7.3.2.2 record-level MIA Captures Record-Specific Memorization Beyond Speaker Identity

We further evaluate a more challenging setting in which all non-member records are drawn from speakers who appear in the fine-tuning set. In this setting, the attack must distinguish between different records from the same speakers, rather than simply separating seen speakers from unseen ones. As shown in Figure 7.3, the ROC curve remains clearly above the random baseline, with low-FPR performance comparable to that on the full evaluation set. This result suggests that record-level MIA captures record-specific memorization cues beyond speaker identity.

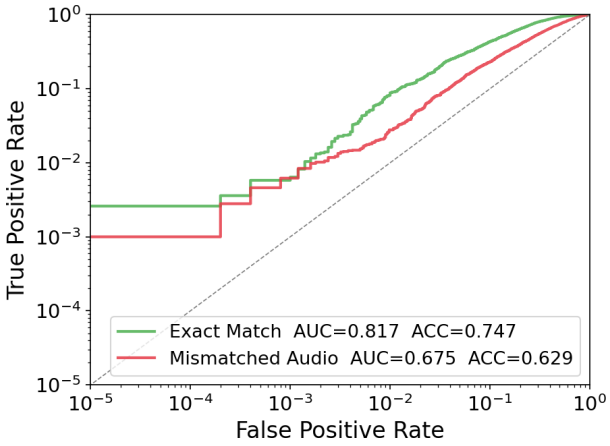


(a) XTTS-v2.

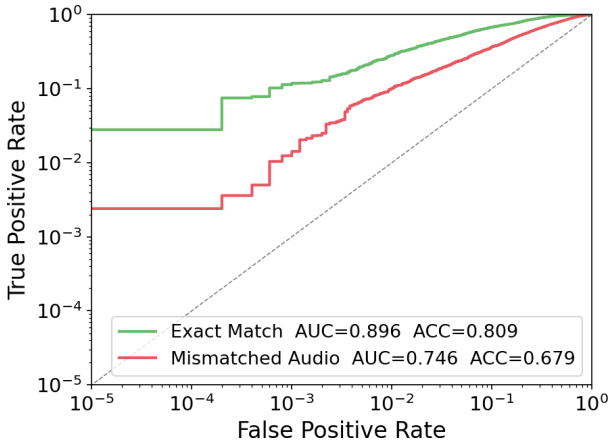


(b) CosyVoice2.

Figure 7.4: Exact match queries consistently outperform weaker strategies for speaker-level MIA, with the largest gains on the more challenging XTTS-v2 model.



(a) XTTS-v2.



(b) CosyVoice2.

Figure 7.5: Exact match queries substantially outperform mismatched audio for record-level MIA.

7.3.3 Ablation and Analysis

In this section, we provide the ablation results on the key design choices of the proposed attack suite, examining the effect of query strategy, feature representation, temporal alignment, fine-tuned model components, and query budget on attack performance.

7.3.3.1 Exact-Match Queries Substantially Improve Attack Performance

Both attack levels validate the information-theoretic principle from Section 7.2.2: a query that maximizes mutual information with the target record by providing matched text and matched audio achieves the lower bound of membership inference error. As shown in Figures 7.4 and 7.5, exact match queries consistently outperform weaker alternatives at both levels, with gains that grow as the memorization signal becomes finer-grained.

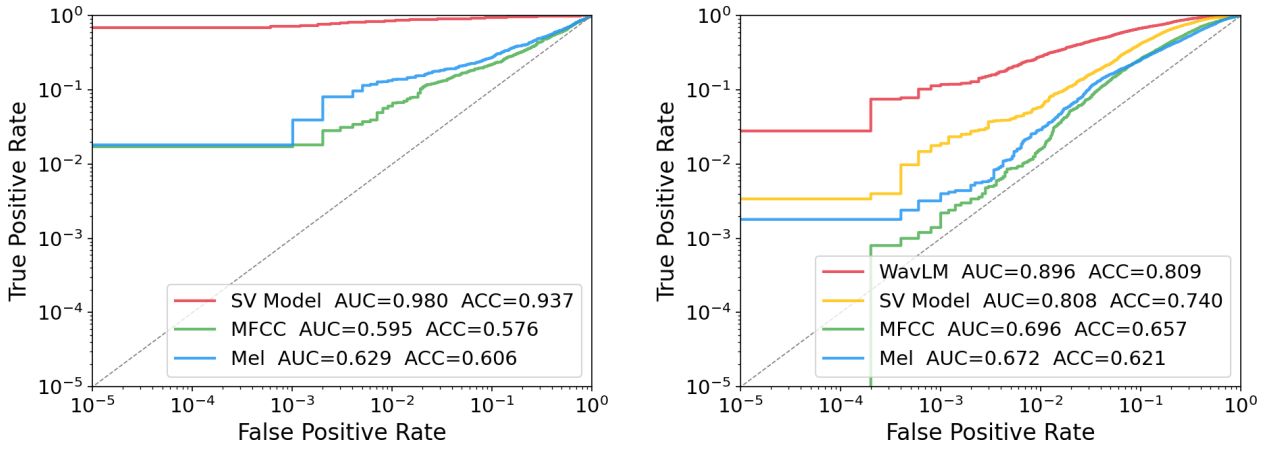
Figure 7.4 shows that for speaker-level MIA, exact match queries reach AUC 0.842 on XTTS-v2, outperforming mismatched audio at 0.799 and mismatched text at 0.646; the same ordering is reflected in TPR at low FPR. On CosyVoice2 the gap narrows because all strategies already perform well, but exact match still attains the highest AUC at 0.980. The advantage is largest on the weaker model, where tighter query constraints matter most.

Figure 7.5 shows an even clearer pattern for record-level MIA. Since mismatched text does not support fine-grained temporal alignment and detailed acoustic comparison, we only compare exact match queries against mismatched audio. The same principle produces even larger gains: on CosyVoice2, AUC increases from 0.746 to 0.896, and on XTTS-v2 from 0.675 to 0.817, with consistent improvements in TPR at low FPR.

7.3.3.2 Speaker- and record-level Attacks Require Features at Different Granularities

Figure 7.6 compares the features used to compute attack scores for speaker-level and record-level MIA. For speaker-level MIA, as shown in Figure 7.6a, the SV representation is the most effective, achieving an AUC of 0.9796. It substantially outperforms Mel and MFCC, which achieve AUC values of 0.6290 and 0.5948, respectively, with an even larger gap in the low-FPR region. This improvement is expected because the SV model is explicitly trained to suppress linguistic and prosodic variation while producing global, utterance-invariant embeddings that preserve speaker identity.

In contrast, record-level MIA requires more fine-grained features. As shown in Figure 7.6b, the SV representation alone is insufficient, achieving an AUC of 0.8079, whereas multi-level WavLM features aggregated by a shadow model improve the AUC to 0.8958. Both representations



(a) speaker-level MIA.

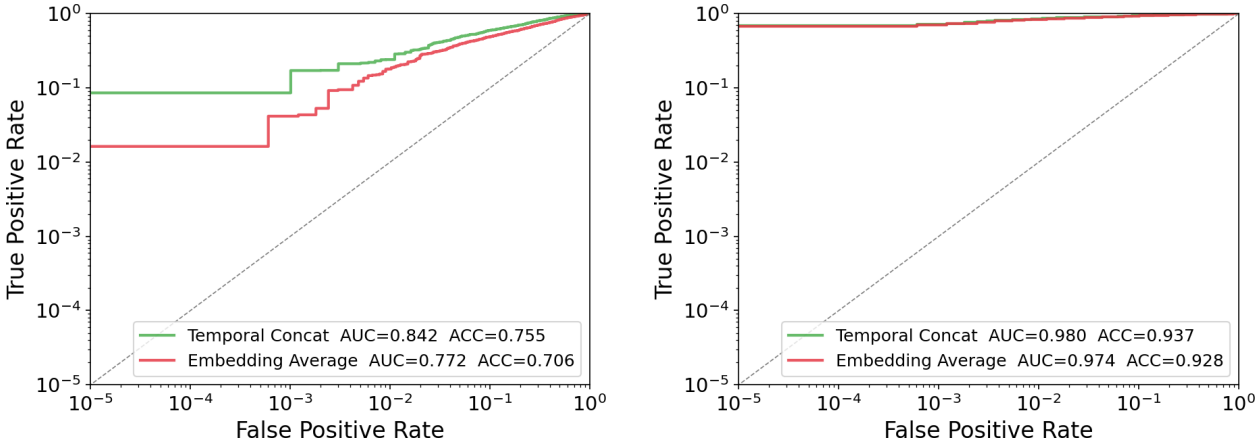
(b) record-level MIA.

Figure 7.6: SV features best expose speaker-level leakage, while WavLM best exposes record-level leakage.

substantially outperform Mel and MFCC, which achieve AUC values of 0.6962 and 0.6717, respectively. This gap indicates that record-level membership depends on local, content-dependent acoustic patterns that compact speaker embeddings discard by design. The contrast between the two panels further shows that speaker-level and record-level leakage arise at different granularities: global identity cues capture speaker-level leakage, whereas local sequence-specific cues capture record-level leakage.

7.3.3.3 Temporal Concatenation Captures More Stable Speaker Identity

Figure 7.7 shows that temporal concatenation yields more stable speaker representations than per-clip embedding averaging for speaker-level MIA. As shown in Figure 7.7a, on XTTS-v2, concatenation achieves an AUC of 0.842 and an ACC of 0.755, compared with 0.772 and 0.706 for the averaging baseline, respectively. The gap is especially prominent at low FPR, where concatenation maintains a higher TPR across the low false-positive regime. As shown in Figure 7.7b, on CosyVoice2, the margin narrows because both strategies already perform well, but concatenation still achieves a higher AUC of 0.980, compared with 0.974 for embedding averaging.



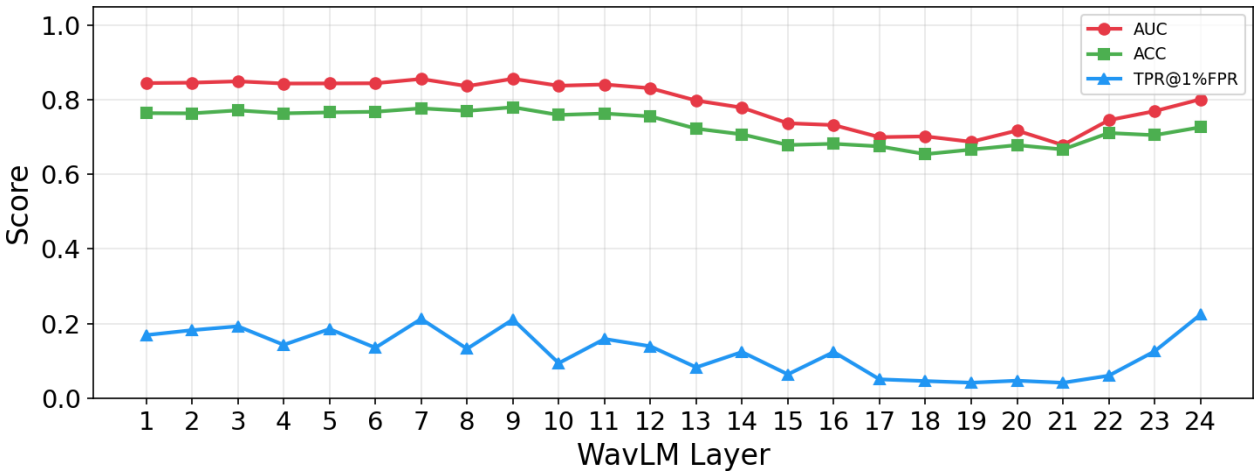
(a) XTTS-v2.

(b) CosyVoice2.

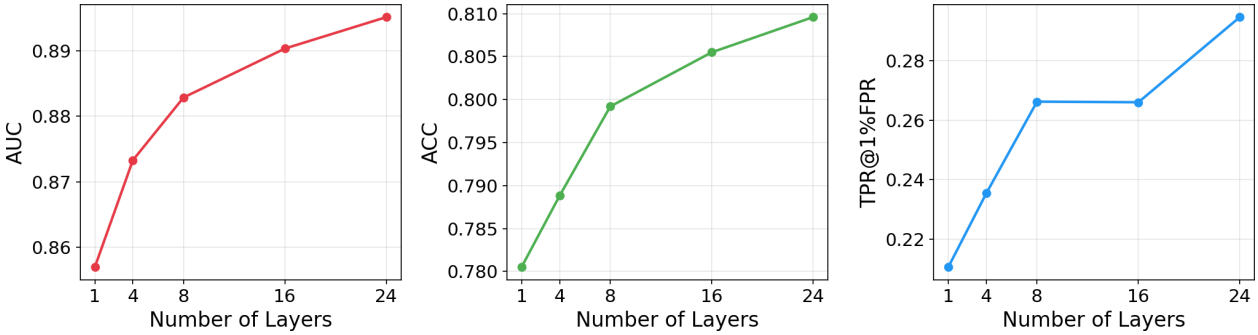
Figure 7.7: Temporal concatenation consistently outperforms embedding averaging for speaker-level MIA, with the largest gains on the more challenging XTTS-v2 model.

7.3.3.4 record-level MIA Requires Multi-Level WavLM Features

To further examine the role of multi-level information in record-level MIA, Figure 7.8 analyzes which WavLM layers are most useful and whether aggregating layers improves over using a single layer. The best single-layer results come from the lower WavLM hierarchy, where the first twelve layers consistently outperform most higher layers in AUC and ACC. TPR@1%FPR is noisier, but it also tends to peak in the lower layers. This suggests that record-level memorization relies more on fine-grained local acoustic matching than on the more abstract semantic representations encoded in upper layers. The signal, however, is not limited to one layer. Combining top-ranked layers steadily improves all reported metrics, with AUC increasing from roughly 0.857 for the best single layer to 0.873, 0.883, 0.890, and 0.895 when aggregating the top-4, top-8, top-16, and all 24 layers, respectively. ACC and TPR@1%FPR follow the same trend. These results suggest that different layers provide complementary information, and pooling them with the LSTM aggregator improves membership discrimination. This makes full-hierarchy aggregation a strong default choice for record-level MIA.



(a) MIA performance with each individual WavLM layer feature; lower layers are generally more discriminative than upper layers.



(b) Aggregating more top-ranked layers improves MIA performance monotonically, making full-hierarchy aggregation the best default.

Figure 7.8: Multi-level features improve record-level MIA.

7.3.3.5 record-level MIA Requires Fine-Grained Temporal Alignment

Figure 7.9 empirically validates the alignment design introduced in Section 7.2.3. A naive baseline that pads or truncates the generated sequence to match the reference length, without performing explicit alignment, yields near-random performance. This result confirms that temporal alignment is necessary for meaningful comparison. DTW achieves the strongest performance, with an AUC of 0.896 and an ACC of 0.809. It clearly outperforms shift-align, which corrects only a fixed temporal offset between the reference and generated sequences and reaches an AUC of 0.807 and an ACC of 0.737, as well as uniform-stretch, which globally rescales the generated sequence to match the reference duration and achieves an AUC of 0.697 and an ACC of 0.648. The large gap between DTW

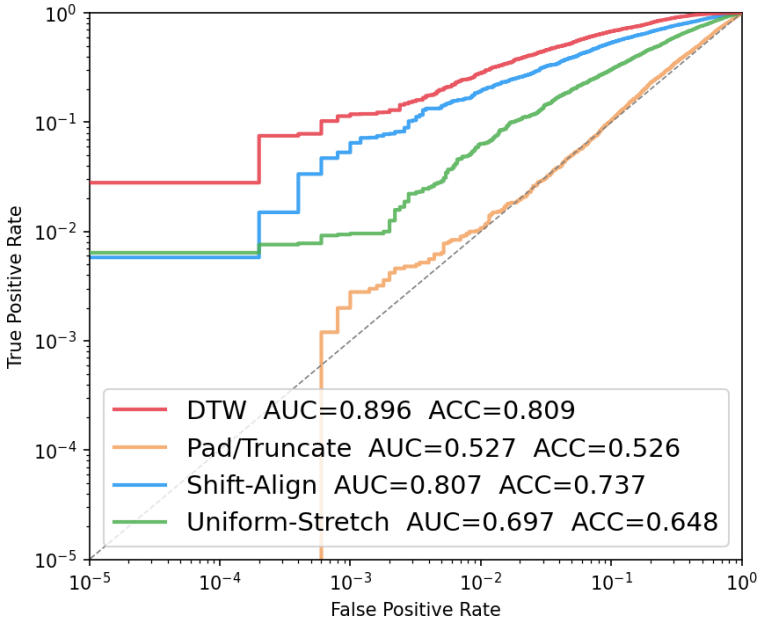


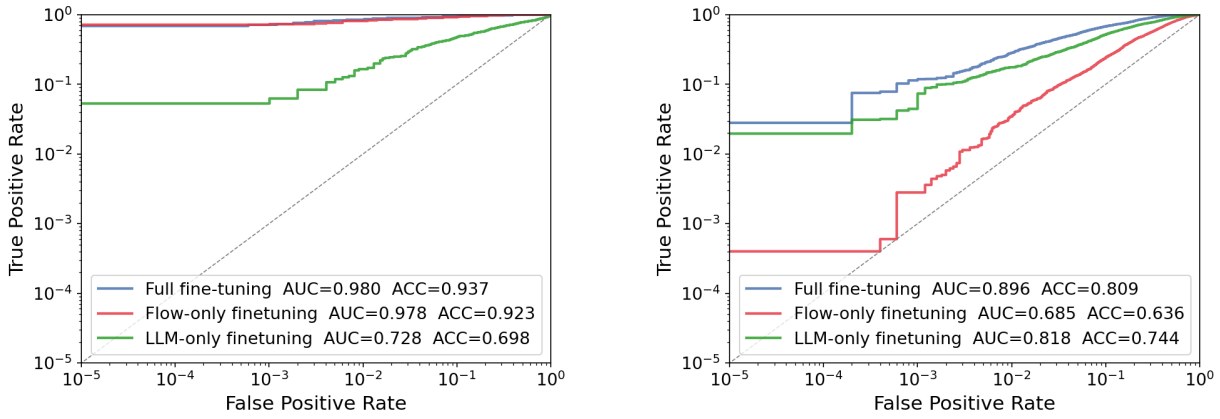
Figure 7.9: DTW substantially outperforms alternative alignment methods for record-level MIA, showing that fine-grained local alignment is important.

and these simpler alternatives indicates that record-level MIA cannot be handled by correcting only a global shift or a uniform temporal rescaling. Instead, the attack requires fine-grained alignment between phonetically corresponding regions so that local acoustic matches are preserved rather than treated as temporal mismatch.

7.3.3.6 Speaker and record-level Leakage Are Driven by Different Model Components

Since CosyVoice2 consists of an autoregressive language model over latent audio tokens and a flow-matching acoustic decoder that maps these tokens to mel-spectrograms, we examine the effect of fine-tuning each component on speaker-level and record-level MIA, as shown in Figure 7.10. For speaker-level MIA, Figure 7.10a shows that full and flow-only fine-tuning produce near-perfect AUC and high TPR at low FPRs. In contrast, LLM-only fine-tuning drops substantially to an AUC of 0.728, with a corresponding collapse in low-FPR TPR. The minimal gap between full and flow-only fine-tuning, together with the large drop under LLM-only fine-tuning, suggests that speaker-level leakage is mainly driven by the acoustic decoder.

For record-level MIA in Figure 7.10b, full fine-tuning again achieves the strongest performance,



(a) speaker-level MIA.

(b) record-level MIA.

Figure 7.10: speaker-level leakage is primarily driven by the flow-matching acoustic decoder, whereas record-level leakage relies more on the LLM component, suggesting different underlying memorization mechanisms.

with an AUC of 0.896, but the pattern under partial fine-tuning differs from the speaker-level case. LLM-only fine-tuning retains substantial leakage, with an AUC of 0.818, whereas flow-only fine-tuning drops further to an AUC of 0.685. The same ordering is also reflected in TPR in the low-FPR region. This trend is the reverse of the speaker-level result, where flow-only fine-tuning nearly matches full fine-tuning. One possible interpretation is that record-level memorization relies more on the autoregressive component, which captures prosodic and temporal traces, whereas the acoustic decoder primarily encodes identity information.

7.3.3.7 speaker-level MIA Requires Few Attacker Records for Strong Performance

Figure 7.11 shows that speaker-level MIA reaches strong performance with only a small number of records, as few as 3–5, available to the attacker. Increasing the number of records from 1 to 3 raises the AUC from approximately 0.87 to 0.96 and the TPR@1%FPR from about 0.47 to 0.82. Beyond 5 utterances, the gains become marginal, with the AUC approaching 1.0 and the TPR@1%FPR stabilizing near 0.95. These results indicate that the speaker-level attack does not require a large attacker record set. The speaker identity prior is sufficiently strong that a small number of reference clips can support a high-confidence membership decision. In practice, an auditor with access to only

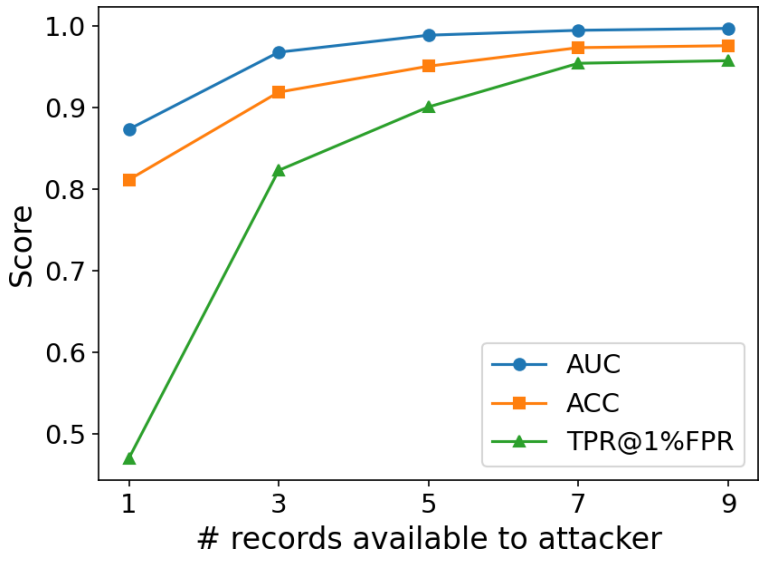
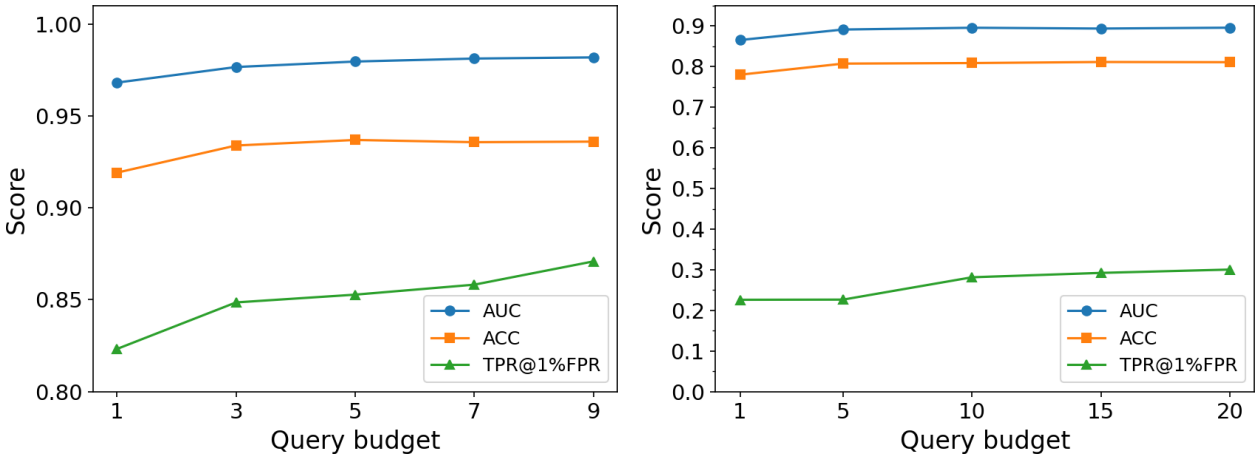


Figure 7.11: speaker-level MIA reaches strong performance with only a few attacker utterances; 3–5 recordings are sufficient for a high-confidence membership decision.



(a) speaker-level MIA.

(b) record-level MIA.

Figure 7.12: Effect of query budget, defined as the number of repeated queries with different random seeds, on speaker-level and record-level MIA.

3–5 public recordings of a target speaker can reliably determine whether the voice of that speaker was used for fine-tuning.

Table 7.2: Effect of different multi-query statistics on record-level MIA performance. The mean alone provides a modest gain over a single query, variance alone is weak, and combining mean with variance yields the best results.

Method	AUC	ACC	TPR@1%FPR
Single Query	0.866	0.780	0.226
Mean	0.869	0.784	0.265
Var	0.690	0.644	0.033
Mean+Var	0.896	0.809	0.281

7.3.3.8 Effect of Query Budget

Figure 7.12a shows that the performance of speaker-level MIA improves as the number of repeated queries increases, but most of the gain appears early. The AUC rises from 0.968 with one seed to 0.980 with five seeds and then plateaus, while TPR@1%FPR increases more gradually from 0.823 to 0.871 by nine seeds. The early saturation in AUC indicates that average-case classification becomes reliable with few repetitions; additional repetitions mainly refine the low-FPR operating point, where score variance matters most.

Figure 7.12b shows a similar but slower saturation pattern for record-level MIA. The AUC improves from roughly 0.863 with one seed to 0.895 with ten or more seeds, and TPR@1%FPR increases from about 0.23 to 0.30 by twenty seeds. Compared with speaker-level MIA, record-level scores are noisier for each query because content-specific memorization traces are weaker. More repeated queries are therefore needed before the low-FPR regime stabilizes.

7.3.3.9 Mean and Variance Both Contribute to the Aggregated Score

Given the gains from increasing the query budget, we further ablate how the multiple query scores should be aggregated into a single record-level score. Table 7.2 reports the contribution of different aggregation statistics. A single query already provides a strong baseline, with an AUC of 0.866 and a TPR@1%FPR of 0.226. Aggregating the mean score across multiple queries yields a modest

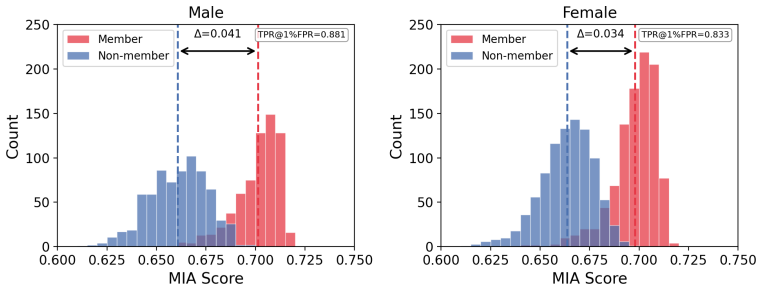


Figure 7.13: Male speakers are more vulnerable to speaker-level MIA than female speakers, possibly due to greater within-group acoustic variability in female speech.

but consistent improvement, increasing the AUC to 0.869 and the TPR@1%FPR to 0.265, likely because averaging reduces score noise from stochastic generation. In contrast, using only the score variance is substantially weaker, with an AUC of 0.690 and a TPR@1%FPR of 0.033, suggesting that variance alone is not a reliable membership signal. Combining the mean and variance gives the best overall performance, reaching an AUC of 0.896 and a TPR@1%FPR of 0.281. We attribute this improvement to the complementary membership signals captured by the two statistics: member samples tend to have higher similarity scores and lower variation across repeated queries.

7.3.4 Data Characteristics and Vulnerability

Beyond aggregate attack performance, we examine which speech properties affect vulnerability to membership inference, revealing what information the model memorizes from the training data. For speaker-level MIA, we find that male speakers are more vulnerable than female speakers. At the record level, utterances with lower silence ratio, higher phoneme density, greater high-frequency content, longer duration, and higher speech energy are more vulnerable. Unless otherwise noted, all analyses use CosyVoice2 on VCTK.

7.3.4.1 Male Speakers Are More Vulnerable to speaker-level MIA

Figure 7.13 shows that male speakers are more vulnerable to speaker-level MIA than female speakers. Specifically, the separation between the member and non-member score distributions is larger for male speakers, with a mean score gap of $\Delta = 0.041$ compared with $\Delta = 0.034$ for female speakers.

Table 7.3: Record-level MIA performance across silence ratio bins; lower silence ratio increases record-level risk.

Bin	AUC	ACC	TPR@1%FPR
0–20%	0.943	0.886	0.406
20–30%	0.936	0.859	0.427
30–40%	0.919	0.828	0.344
40–50%	0.886	0.793	0.258
50–60%	0.850	0.741	0.246
> 60%	0.748	0.625	0.129

This larger gap is also reflected in the low-FPR operating regime, where the TPR@1%FPR reaches 0.881 for male speakers but 0.833 for female speakers. A possible explanation is that female speech exhibits larger within-group acoustic variability, which leads to greater overlap between member and non-member score distributions.

7.3.4.2 Lower Silence Ratio Increases record-level Vulnerability

Silence ratio measures the fraction of an utterance occupied by non-speech or low-energy regions. Table 7.3 shows a clear monotonic decline in attack performance as the silence ratio increases. The AUC drops from 0.943 in the 0–20% bin to 0.748 in the > 60% bin, while TPR@1%FPR decreases from 0.406 to 0.129. We attribute this trend to the fact that silence provides little discriminative acoustic information for comparison. As the silence ratio increases, the amount of informative content available to distinguish members from non-members correspondingly decreases.

7.3.4.3 Denser Phoneme Structure Increases record-level Vulnerability

Phoneme density measures how many phonemes on average appear per syllable in an utterance and serves as a proxy for articulatory complexity. Table 7.4 shows that utterances with denser

Table 7.4: Record-level MIA performance across phoneme density bins; higher phoneme density can increase risk.

Bin	AUC	ACC	TPR@1%FPR
<3	0.852	0.759	0.184
3–3.5	0.888	0.794	0.262
3.5–4	0.907	0.807	0.335
4–4.5	0.927	0.828	0.372
> 4.5	0.912	0.819	0.340

phoneme structure are generally more vulnerable to membership inference. AUC rises from 0.852 in the < 3 bin to a peak of 0.927 in the 4–4.5 bin, and TPR@1%FPR increases from 0.184 to 0.372 over the same range. Phonemically denser utterances contain richer acoustic transitions and more fine-grained local structure, which provide stronger signals for record-level comparison. The highest-density bin (> 4.5) shows a slight decline to AUC 0.912 and TPR@1%FPR 0.340. One possible explanation for this drop is that extremely high phoneme density is associated with fast or heavily clustered articulation, where the synthesis model may smooth over fine-grained detail regardless of membership status, reducing the separability of member and non-member outputs.

7.3.4.4 Richer High-Frequency Content Increases record-level Vulnerability

High-frequency energy ratio measures the proportion of spectral energy above 4 kHz, which often captures fricatives, sibilants, and other fine acoustic details. Prior work suggests that TTS models have greater difficulty reproducing such high-frequency detail [BLB23, JWS24], which can make these regions more informative for distinguishing member and non-member utterances. Table 7.5 shows that utterances with higher high-frequency energy ratios are consistently more vulnerable to membership inference. The AUC rises from 0.879 in the 0–1% bin to 0.937 in the > 12% bin, and TPR@1%FPR increases from 0.248 to 0.438. One possible explanation is that faithful reproduction

Table 7.5: Record-level MIA performance across high-frequency energy ratio bins; richer high-frequency content consistently increases vulnerability.

Bin	AUC	ACC	TPR@1%FPR
0–1%	0.879	0.775	0.248
1–3%	0.892	0.795	0.295
3–6%	0.894	0.800	0.298
6–12%	0.911	0.824	0.293
> 12%	0.937	0.853	0.438

Table 7.6: Record-level MIA performance across record-duration bins; longer records show higher vulnerability.

Bin	AUC	ACC	TPR@1%FPR
< 2s	0.857	0.736	0.282
2–3s	0.881	0.788	0.271
3–4s	0.902	0.813	0.261
4–5s	0.912	0.819	0.320
> 5s	0.916	0.823	0.395

of these details may require stronger memorization of the specific recording, which increases the separation between member and non-member outputs.

7.3.4.5 Longer Recordings Are More Vulnerable

Utterance duration is the total length of the speech sample in seconds, including both voiced regions and pauses. Table 7.6 shows a monotonically increasing relationship between utterance duration

Table 7.7: Record-level MIA performance across speech energy bins; higher energy recordings are more vulnerable.

Bin	AUC	ACC	TPR@1%FPR
<0.02	0.845	0.713	0.242
0.02–0.04	0.865	0.763	0.221
0.04–0.06	0.901	0.814	0.308
0.06–0.08	0.922	0.838	0.358
>0.08	0.915	0.824	0.330

and attack performance. AUC rises steadily from 0.857 for utterances under 2 seconds to 0.916 for those over 5 seconds, and TPR@1%FPR increases from 0.282 to 0.395 over the same range. A likely explanation is that longer utterances provide more acoustic content for the DTW-based scoring pipeline to work with, yielding more reliable frame-level alignment between the target and generated audio and thus a stronger membership signal.

7.3.4.6 Higher Speech Energy Increases record-level Vulnerability

Speech energy measures the average RMS amplitude over voiced regions of an utterance. Table 7.7 shows a clear increasing relationship between speech energy and attack performance. AUC rises from 0.845 in the lowest bin (< 0.02) to 0.922 in the 0.06–0.08 bin, and TPR@1%FPR increases from 0.242 to 0.358 over the same range. We attribute this trend to the fact that higher-energy recordings contain richer harmonic and formant structure, providing more discriminative acoustic evidence for the DTW-based scoring pipeline to detect memorization.

7.4 Related Work

Membership Inference Attacks on Generative Models. With the rapid growth of generative models, privacy auditing has become an increasingly important research topic [NST21, XWW25, SNJ23]. Membership inference attacks (MIAs), as a central auditing tool, have been extensively studied for generative models such as large language models and text-to-image generation models, moving from white-box settings toward more realistic black-box access. White-box attacks use internal model signals such as model parameters and gradients [PWK25], losses [HP23, MMY23], or denoising trajectories [ZCD24, KDM23]. In contrast, black-box attacks infer membership solely from observable input–output behavior, such as generated-sample similarity [PW23, DZC25, LDH25] or token logits [SAX23, ZSY24]. Despite this progress, existing generative-model MIAs do not directly transfer to TTS. Modern TTS models use cross-modal dual conditioning on both text and reference audio, and their outputs are continuous, variable-length waveforms rather than discrete tokens or fixed-dimensional images. These differences make it challenging to construct memorization-triggering queries and to compare generated outputs with target records, motivating a TTS-specific black-box MIA framework.

Speech Privacy. Speech carries rich sensitive information, including speaker identity, linguistic content, emotion, and other paralinguistic traits, which has motivated growing interest in speech privacy [LKY22, ZLH25, TMC26, TMV25, TKL22]. In particular, MIAs have been investigated for speech understanding tasks, such as automatic speech recognition and automatic speaker recognition [SSM21, TPO24, CZS23]. However, the privacy risks of generative TTS systems remain underexplored. The most relevant prior work [KDM23] studies white-box membership inference for diffusion models, including limited experiments on early TTS systems conditioned only on text. Our work targets a fundamentally different setting: modern generative TTS systems that support text–audio dual conditioning for voice cloning, cover diverse model families beyond diffusion, and provide no access to internal states, gradients, or losses. As a result, the adversary must infer membership solely from synthesized waveforms, making black-box TTS membership inference substantially more challenging and largely unexplored.

Table 7.8: Attack performance under early stopping defense for speaker-level and record-level MIA.

Level	AUC	ACC	TPR@1%FPR
Speaker	0.900	0.836	0.543
Record	0.748	0.689	0.036

7.5 Discussion

Broader Impact on Multimodal Generative Models. The success of our attack stems from three key principles: matched queries that preserve information and maximize the membership signal, multi-level representations that capture identity and content traces, and temporal alignment for fine-grained comparison of variable-length outputs. These principles also apply to a broader class of generative models with multimodal inputs and continuous, variable-length outputs, including video editing systems [KWR] and omni-modal models [XGH25, XW24] that jointly process speech, vision, and text.

7.5.1 Potential Defenses and Their Limitations

To mitigate our proposed MIA, defenders might consider empirical mitigations at both the training and inference stages. We evaluate three candidate defenses: early stopping during training, input perturbation at inference time, and DP-SGD as a provable training-time defense.

7.5.1.1 Early Stopping

As shown in Table 7.8, early stopping reduces record-level MIA substantially (AUC drops to 0.748, TPR@1%FPR to 0.036), since the model has less opportunity to overfit fine-grained content details. However, speaker-level MIA remains strong (AUC 0.900, TPR@1%FPR 0.543), indicating that speaker identity is memorized early in training and is hard to suppress effectively by this approach.

Table 7.9: Attack performance under perturbation-based defenses.

Perturb	Level	Param	AUC	ACC	TPR@1%FPR
Noise	Speaker	40 dB	0.970	0.925	0.814
		25 dB	0.923	0.878	0.559
	Record	40 dB	0.838	0.761	0.157
		25 dB	0.808	0.732	0.126
Truncation	Speaker	50%	0.878	0.814	0.490
		20%	0.695	0.663	0.024
	Record	50%	0.746	0.679	0.100
		20%	0.657	0.616	0.065

7.5.1.2 Input Perturbation

Table 7.9 evaluates two inference-time perturbations applied to the reference audio before it is passed to the TTS model. For noise addition, we add white noise at signal-to-noise ratios of 40 dB and 25 dB. A 40 dB SNR corresponds to barely perceptible noise and largely preserves perceived audio quality, while a 25 dB SNR causes clearly audible degradation but remains intelligible. These two levels cover a practical range that a deployed API might use without causing major quality concerns for regular users. Even under the stronger 25 dB noise setting, the attacks remain effective: speaker-level MIA still achieves a high AUC, and record-level MIA remains well above random. This suggests that the proposed attacks are robust to substantial additive noise.

For temporal truncation, we keep only the first 50% or 20% of the original audio duration, corresponding to mild and aggressive truncation, respectively. Truncation reduces attack performance more strongly than additive noise, especially under the aggressive 20% setting, where much of the reference speech is removed. However, the attacks still remain largely above random, indicating that

Table 7.10: Attack performance under DP-SGD defense for speaker-level and record-level MIA at two privacy budgets.

Level	ε	AUC	ACC	TPR@1%FPR
Speaker	1	0.535	0.551	0.007
	10	0.543	0.555	0.015
Record	1	0.519	0.518	0.010
	10	0.534	0.528	0.006

membership signals can persist even when only a shortened reference is available. Removing a large portion of the reference audio is therefore necessary to substantially weaken the attack, which would likely reduce the utility of the TTS system for regular users.

7.5.1.3 DP-SGD

The insufficiency of the two empirical mitigations above motivates the necessity of rigorous, albeit costly, provable defenses. We evaluate DP-SGD [ACG16] for privacy-preserving TTS fine-tuning on CosyVoice2 with VCTK. As shown in Table 7.10, DP-SGD trained with privacy budgets $\varepsilon = 1$ and $\varepsilon = 10$ reduces both speaker-level and record-level MIA to near-random performance (AUC ≈ 0.52–0.54, TPR@1% < 0.02), confirming that formal differential privacy effectively mitigates the membership leakage exposed by our attack. However, this formal privacy guarantee typically comes with a non-trivial trade-off in generation quality, which may be undesirable in practice.

7.6 Chapter Summary

This chapter presented a systematic black-box MIA framework for zero-shot TTS models, addressing the methodological challenges of auditing high-variance generative systems. By formalizing audit as a one-bit transmission Markov chain, we identified a reconstruction-based query as information-

theoretically optimal, and introduced a modified DTW procedure that anchors variable-length generated audio to the target time axis for reliable representation-level comparison. Empirical results across three representative zero-shot TTS architectures and two datasets confirm strong memorization during fine-tuning: speaker-level attacks reach AUC up to 1.000 and remain above 0.80 throughout, and record-level attacks consistently achieve AUC between 0.80 and 0.90 even when non-members are restricted to speakers seen during fine-tuning.

Together with the findings from Chapter 6, these results complete the empirical audit of the Cognitive Intelligence Pillar. Predictive and generative Metaverse AI systems memorize substantially more sensitive sequential user data than their task requires, and this memorization is recoverable through black-box access alone. The memorized information, namely precise physical trajectories and immutable biometric voice signatures, is among the most sensitive user data the Metaverse collects. Chapter 8 synthesizes the findings from both research lines and outlines directions for a Metaverse that is trustworthy by design.

CHAPTER 8

Conclusion and Future Work

This dissertation has examined the Metaverse as a substantially more complex computational paradigm than the 2D internet that came before. It is one in which embodied immersive interaction (VR/AR-driven continuous sensing), persistent multi-user shared state, and tightly-integrated algorithmic inference operate jointly and in real time. Whereas traditional computing systems consume discrete, intentional user input, the Metaverse runs on persistent, multi-user-visible streams of multi-modal human signals that are immediately ingested by predictive and generative ML models. This three-way coupling of embodied broadcast, persistent shared state, and integrated ML inference is what makes immersive social presence possible. It is also what significantly raises the security and privacy stakes of each underlying technology and turns securing the Metaverse into a problem distinct from securing any one of them in isolation.

The central argument of this work is that Metaverse security and privacy cannot be addressed threat by threat, nor by importing security models designed for any single layer (VR in isolation, generic ML, or 2D networking) without accounting for how the layers reinforce one another. The vulnerabilities we have studied span remote kinematic side-channels, trajectory memorization in predictive models, and biometric leakage in zero-shot avatar voice synthesis, and they are not isolated bugs. They are structural consequences of two architectural choices that the Metaverse paradigm currently treats as defaults: centralized collection of interaction data over persistent shared spaces, and the deployment of ML models whose memorization behavior over Metaverse-specific data (sequential trajectories, identity-bearing biometric signals) is not routinely audited. Addressing these problems requires changing the architecture and the evaluation methodology, not just patching individual weaknesses one application at a time.

8.1 Summary of Contributions

This dissertation contributed to this goal along two research lines.

8.1.1 The Spatial Interaction Pillar

We began by identifying a concrete, exploitable vulnerability whose severity is significantly raised by the Metaverse setting: the kinematic data that VR/AR systems already produce as part of normal immersive operation is, in a multi-user Metaverse application, broadcast across co-present peers for avatar synchronization, and can be parsed from network traffic by a remote adversary to reconstruct user keystrokes with over 97% top-1 accuracy. The Metaverse's persistent, multi-user broadcast regime turns a baseline VR signal into a remote, network-visible attack surface accessible to any peer in the same virtual space, with no malware and no physical proximity required. The result was validated across multiple VR platforms and led to deployed countermeasures by both Rec Room and SteamVR following responsible disclosure.

To understand why such vulnerabilities persist, we conducted an empirical developer study with 23 professional XR developers. The findings reveal two structural root causes specific to the Metaverse development ecosystem: awareness gaps that prevent developers from recognizing spatially-specific threats whose attack surface is the very broadcast pipeline the platform requires, and a diffusion of responsibility amplified by opaque game engine tooling whose data-handling pipelines were not originally designed with multi-user, privacy-preserving telemetry as a first-class concern. This diagnosis points to the inadequacy of awareness campaigns alone: the development environment itself must change.

In response, we reframed decentralization not as a binary architectural switch but as a function-by-function redistribution of distinct operator roles (identity, shared state, behavior auditing), grounding the framework's design mechanism by mechanism in user-stated needs surfaced by a 22-participant interview study. These needs include the Digital Doorbell permission-gated access intuition (articulated without prompt by 10 of 22 participants) and the Accountability Paradox between privacy and evidence. The resulting serverless framework combines a Standardized State Abstraction Layer over Named Data Networking, a peer-side semantic-correctness pipeline, and

quorum-based behavior auditing. A 20-participant within-subjects double-blind validation against a centralized baseline returned absence of evidence for a perceptual gap on the three constructs participants themselves nominated as load-bearing, identification at chance, semantic-correctness rollbacks unnoticed in every decentralized session, and per-update round-trip latency within 2 ms of the centralized baseline. Together these results place the framework inside the small-group, dense-collaboration regime that participants themselves drew as decentralization's high-value Metaverse deployment surface.

8.1.2 The Cognitive Intelligence Pillar

We then turned to the ML models that process Metaverse data. For predictive spatial-temporal models, we focused on the algorithmic backbone of Metaverse location-based services, where the data is sequential, semantically annotated, and tied to a persistent avatar identity rather than isolated GPS pings. We developed a tailored attack suite of four attacks (LocExtract, TrajExtract, LocMIA, TrajMIA) that extracts sensitive user location information and trajectory membership from POI recommendation systems. Our results show that these models memorize training data well beyond what their task requires: models expose sensitive locations in 32.5% of random queries despite a training presence of only 18.7%, and we identify an "outlier effect" that makes rare trajectories disproportionately vulnerable. Existing defenses failed to address the attacks without unacceptable utility loss.

For generative models, we addressed the harder problem of auditing zero-shot TTS systems. The Metaverse deploys these systems to synthesize the avatar voices that carry a user's vocal identity across persistent social spaces, a setting in which leaking the trained voice signal is functionally equivalent to leaking the avatar identity itself. By formalizing the audit process as a one-bit transmission Markov chain, we identified a reconstruction-based query as the information-theoretically optimal probe and developed the first black-box MIA framework for this setting. speaker-level attacks reach AUC up to 1.000 in the strongest configurations and remain above 0.80 throughout, while record-level attacks consistently achieve AUC between 0.80 and 0.90. These results show that biometric voice data is memorized and recoverable from deployed TTS models,

even through black-box access.

8.1.3 The Broader Argument: From Reactive Patching to Structural Trust

Together, these contributions advance a single coherent argument: trust in the Metaverse must be a structural property, not a policy commitment. The Metaverse is the regime in which spatial interaction, ML inference, and persistent multi-user state operate as one, and the trust failures we have documented are failures of that coupling, not of any single layer in isolation. The transition demonstrated here, from reactive patching to architectural redesign and from theoretical privacy claims to empirical auditability, offers a practical template for how future immersive systems can be built and evaluated. It shows that embodied digital experience does not require surrendering control of one's data, and that the ML models powering these experiences can and should be held to quantifiable privacy standards.

8.2 Future Research Directions

This dissertation establishes empirical grounding for several research directions that remain open.

8.2.1 Privacy-Preserving ML for Spatial-Temporal Data

Our auditing results show that existing defenses (e.g., differential privacy applied to POI models) degrade utility too sharply to be practical. A key open problem is developing privacy mechanisms that account for the multimodal, sequential nature of spatial-temporal data, where individual locations are not independent records but components of a behaviorally coherent trajectory. This is a particularly load-bearing problem in the Metaverse, where mobility data is no longer a sparse stream of GPS pings but a continuous, semantically annotated trajectory tied to a persistent avatar identity, and where many sessions per user pass through the same predictive backbone. Techniques that offer meaningful privacy guarantees without discarding temporal structure remain an open challenge, and represent a prerequisite for deploying privacy-safe LBS in the Metaverse. Synthetic-trajectory generation that preserves spatial-temporal structure while severing membership signal—e.g., coarse-to-fine

frameworks such as GeoGen [XCJ26]—is one promising direction in this space.

8.2.2 Expanding ML Robustness: From Privacy to Adversarial Security

While this dissertation focuses on privacy leakage via inference attacks, future work should address the broader adversarial threat landscape. The Metaverse setting raises new robustness questions that 2D or text-only adversarial ML literature does not yet cover, because adversarial inputs in the Metaverse are themselves embodied and persistent: a malicious spatial prompt or perceptual perturbation propagates through the user's continuous interaction loop rather than landing as a single discrete query. Key directions therefore include developing robust learning frameworks for multi-modal spatial data and establishing defenses against novel attack modalities specific to immersive AI, such as spatial prompt injection against world-model agents, adversarial inputs designed to exploit perceptual vulnerabilities inherent to embodied interfaces, and supply-chain attacks on the increasingly common practice of merging or composing pre-trained Metaverse models (e.g., backdoor attacks against model merging [ZCL24]). Extending the auditing frameworks developed here to cover these adversarial dimensions would provide a more complete picture of ML trustworthiness in the Metaverse.

8.2.3 Trustworthy World Models for Spatial Computing

As the Metaverse evolves toward predictive world models that reason about physical environments and user behavior, ensuring their trustworthiness becomes critical. Future work should investigate *certified generation*, where safety and physical plausibility constraints are embedded directly into the generative process, and *predictive integrity*, ensuring that model outputs remain consistent with both ground-truth physics and user intent. Developing evaluation benchmarks and auditing protocols tailored to these multi-modal world models is an important next step for the community; early causal-benchmark efforts for embodied world models [CSZ26] begin to map this evaluation surface.

8.2.4 Generative Model Auditing at Scale

The TTS auditing framework introduced here targets the fine-tuning phase. Extending this methodology to models trained from scratch at scale, and to other generative modalities (e.g., motion generation, facial animation, and avatar synthesis), is a natural next step. In the Metaverse, each of these modalities does the same kind of identity-bearing work that TTS does for voice: the synthesized avatar body, face, and motion serve as a user's persistent embodied identity in shared social spaces. Auditing them with the same rigor as the voice channel is therefore directly load-bearing for Metaverse privacy. Developing auditing frameworks that are both computationally tractable and methodologically rigorous for large-scale generative systems remains an important open problem for the ML privacy community.

8.2.5 Human-Centric Security at the Intersection of HCI and ML

Security in the Metaverse is inherently human-centric. Future research should address the interplay between system-level protections and human perception, including the design of developer tools and game engine APIs that guide practitioners toward privacy-preserving defaults, the design of end-user interfaces that communicate risk without disrupting immersion, and defenses against adversarial interaction designs that exploit cognitive biases in immersive environments.

8.2.6 Toward Principled Trust Protocols

Achieving strong, structural guarantees of trustworthiness requires formalizing the properties we want immersive systems to satisfy. The Metaverse setting makes this particularly urgent: because user identity, spatial state, and ML inference are coupled across a persistent, multi-user surface, an informal promise from any single platform is structurally insufficient as a trust anchor. Concrete directions include formalizing spatial interaction logic to reason about what information is and is not derivable from transmitted kinematic streams, and designing verifiable interaction protocols that make data flow guarantees auditable by external parties. Grounding trust in principled, auditable system design, rather than implicit platform commitments, is the longer-term direction toward which this dissertation points.

8.3 Final Remarks

The Metaverse is not simply the next version of the Internet. It is a redefinition of how people inhabit digital systems, one where the boundary between user and data, between interaction and observation, is structurally blurred. The work presented in this dissertation argues that securing this paradigm requires going beyond patching individual vulnerabilities. It requires understanding the systemic causes of insecurity, changing the architectural defaults that produce it, and building the empirical tools needed to hold intelligent systems accountable.

The five research contributions here represent steps toward that goal across the two foundational layers of the Metaverse stack. Whether and how quickly the broader ecosystem moves in this direction will depend on collaboration across systems security, machine learning, human-computer interaction, and policy. This dissertation aims to contribute to the technical foundation that makes such progress possible.

References

[ABB18] Devon Adams, Alseny Bah, Catherine Barwulor, Nureli Musaby, Kadeem Pitkin, and Elissa M Redmiles. "Ethics emerging: the story of privacy and security perceptions in virtual reality." In SOUPS, 2018.

[Abb24] Eric Abbruzzese. "Virtual Reality Statistics 2024 – Data and Facts!" https://www.demandsage.com/virtual-reality-statistics/, March 16 2024.

[ABC13] Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. "Geo-indistinguishability: Differential privacy for location-based systems." In ACM SIGSAC Conference on Computer and Communications Security, 2013.

[AC18] Hala Assal and Sonia Chiasson. "Security in the software development lifecycle." In SOUPS, 2018.

[ACG16] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. "Deep learning with differential privacy." In ACM SIGSAC Conference on Computer and Communications Security, 2016.

[AFM16] Yasemin Acar, Sascha Fahl, and Michelle L Mazurek. "You are not your developer, either: A research agenda for usable security and privacy research beyond end users." SecDev, 2016.

[AGA21] Abdullah Al Arafat, Zhishan Guo, and Amro Awad. "Vr-spy: A side-channel attack on virtual key-logging in vr headsets." In 2021 IEEE Virtual Reality and 3D User Interfaces (VR), pp. 564–572. IEEE, 2021.

[Alh24] A'aeshah Alhakamy. "Extended reality (XR) toward building immersive solutions: the key to unlocking industry 4.0." ACM Computing Surveys, 2024.

[Alt] AltspaceVR. "wireshark-photon-dissector." https://github.com/AltspaceVR/wireshark-photon-dissector/tree/master.

[App24] Apple Inc. "Apple Vision Pro Privacy Overview.", February 2024.

[AS25] Hussam Azzuni and Abdulmotaleb El Saddik. "Voice cloning: Comprehensive survey." arXiv preprint arXiv:2505.00579, 2025.

[ASA21] Zainab Rashid Alkindi, Mohamed Sarrab, and Nasser Alzeidi. "User Privacy and Data Flow Control for Android Apps: Systematic Literature Review." Journal of Cyber Security and Mobility, 2021.

[ASD24] SB Abhinaya, Aafaq Sabir, and Anupam Das. "Enabling Developers, Protecting Users: Investigating Harassment and Safety in VR." arXiv e-prints, pp. arXiv–2403, 2024.

[ASM22] Melvin Abraham, Pejman Saeghe, Mark Mcgill, and Mohamed Khamis. "Implications of xr on privacy, security and behaviour: Insights from experts." In Nordic Human-Computer Interaction Conference, pp. 1–12, 2022.

[awe25] "AWE USA 2025: The World's #1 XR Event." In Augmented World Expo (AWE), 2025. Immersive spatial computing expo, June 2025.

[AZH21] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions." Journal of big Data, **8**:1–74, 2021.

[Bac25] Tom Bäckström. "Privacy in speech technology." Proceedings of the IEEE, 2025.

[bat23] battleye. "BattlEye." `https://www.battleye.com/`, October 2023.

[BCP14] Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. "Optimal geo-indistinguishable mechanisms for location privacy." In ACM SIGSAC Conference on Computer and Communications Security, 2014.

[BE09] Andrew J Blumberg and Peter Eckersley. "On locational privacy, and how to avoid losing it forever." 2009.

[BEE19a] Lindsay Blackwell, Nicole Ellison, Natasha Elliott-Deflo, and Raz Schwartz. "Harassment in social virtual reality: Challenges for platform governance." CSCS, 2019.

[BEE19b] Lindsay Blackwell, Nicole Ellison, Natasha Elliott-Deflo, and Raz Schwartz. "Harassment in Social Virtual Reality: Challenges for Platform Governance." In Proceedings of the ACM on Human-Computer Interaction (CSCW), volume 3, pp. 1–25. ACM, 2019.

[BFF14] Peter Bailis, Alan Fekete, Michael J Franklin, Ali Ghodsi, Joseph M Hellerstein, and Ion Stoica. "Coordination avoidance in database systems." Proceedings of the VLDB Endowment, **8**(3):185–196, 2014.

[BLB23] Taejun Bak, Junmo Lee, Hanbin Bae, Jinhyeok Yang, Jae-Sung Bae, and Young-Sun Joo. "Avocodo: Generative adversarial network for artifact-free vocoder." In Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, pp. 12562–12570, 2023.

[BLL24] Carlo Alberto Bono, Lucio La Cava, Luca Luceri, and Francesco Pierri. "An exploration of decentralized moderation on mastodon." In Proceedings of the 16th ACM Web Science Conference, pp. 53–58, 2024.

[Boo] XR Bootcamp. "Comparing Unity vs Unreal for VR, MR or AR Development Projects." `https://xrbootcamp.com/unity-vs-unreal-engine-for-xr-development/#:~:text=Popular%20VR%20Games%20and%20Social%20Platforms%20made%20with%20Unity,-Oculus%20is%20the&text=This%20is%20one%20of%20the,platform%20are%20made%20by%20Unity.`

[BQS25] Lamiaa Basyoni, Adnan Qayyum, Khaled Shaban, Ezieddin Elmahjub, Abdulaziz Al-Ali, Osama Halabi, and Junaid Qadir. "Generative ai-driven metaverse: The promises and challenges of ai-generated content." IEEE Open Journal of the Computer Society, 2025.

[BSL23] Gianni Bremer, Niklas Stein, and Markus Lappe. "Machine learning prediction of locomotion intention from walking and gaze data." International Journal of Semantic Computing, **17**(01):119–142, 2023.

[BTC18] Irina Brass, Leonie Tanczer, Madeline Carr, Miles Elsden, and Jason Blackstock. "Standardising a moving target: The development and evolution of IoT security standards." In Living in the Internet of Things: Cybersecurity of the IoT. IET, 2018.

[BTM23] Elise Bonnail, Wen-Jie Tseng, Mark Mcgill, Eric Lecolinet, Samuel Huron, and Jan Gugenheimer. "Memory manipulations in extended reality." In CHI, 2023.

[BXZ21] Ting Bao, Lei Xu, Liehuang Zhu, Lihong Wang, and Tielei Li. "Successive point-of-interest recommendation with personalized local differential privacy." **70**(10):10477–10488, 2021.

[BYL20] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. "Adaptive graph convolutional recurrent network for traffic forecasting." In International Conference on Neural Information Processing Systems, 2020.

[Cal18] California State Legislature. "California Consumer Privacy Act.", 2018.

[CAL23] Derin Cayir, Abbas Acar, Riccardo Lazzeretti, Marco Angelini, Mauro Conti, and Selcuk Uluagac. "Augmenting security and privacy in the virtual realm: An analysis of extended reality devices." IEEE Security & Privacy, 2023.

[Cas22] Daniel Castro. "AR/VR Poses New Content Moderation Challenges That Policymakers Should Address." Technical report, ITIF, February 2022.

[CBA24] Yasra Chandio, Noman Bashir, and Fatima M Anwar. "Stealthy and Practical Multimodal Attacks on Mixed Reality Tracking." In AIxVR. IEEE, 2024.

[CBL24] 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.

[CBY19] Peter Casey, Ibrahim Baggili, and Ananya Yarramreddy. "Immersive virtual reality attacks and the human joystick." IEEE Transactions on Dependable and Secure Computing, 2019.

[CBY21] Peter Casey, Ibrahim Baggili, and Ananya Yarramreddy. "Immersive Virtual Reality Attacks and the Human Joystick." IEEE Transactions on Dependable and Secure Computing, **18**(2):550–562, 2021.

[CCN22] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. "Membership inference attacks from first principles." In IEEE Symposium on Security and Privacy, 2022.

[CCW22] Zhengyang Chen, Sanyuan Chen, Yu Wu, Yao Qian, Chengyi Wang, Shujie Liu, Yanmin Qian, and Michael Zeng. "Large-scale self-supervised speech representation learning for automatic speaker verification." In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6147–6151. IEEE, 2022.

[CDG24] Edresson Casanova, Kelly Davis, Eren Gölge, Görkem Göknar, Iulian Gulea, Logan Hart, Aya Aljafari, Joshua Meyer, Reuben Morais, Samuel Olayemi, et al. "Xtts: a massively multilingual zero-shot text-to-speech model." arXiv preprint arXiv:2406.04904, 2024.

[CG16] Kate Crawford and Tarleton Gillespie. "What is a Flag For? Social Media Reporting Tools and the Vocabulary of Complaint." New Media & Society, **18**(3):410–428, 2016.

[CHN23] Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. "Extracting training data from diffusion models." In 32nd USENIX security symposium (USENIX Security 23), pp. 5253–5270, 2023.

[CIJ22] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. "Quantifying memorization across neural language models." 2022.

[Cit22] Citi Global Insights. "Metaverse and Money: Decrypting the Future." Report, Citigroup, 2022.

[CJL23] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. "Executing your commands via motion diffusion in latent space." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 18000–18010, 2023.

[CLE19] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. "The Secret Sharer: Evaluating and testing unintended memorization in neural networks." In USENIX Security Symposium, 2019.

[CLL23] Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S Yu, and Lichao Sun. "A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt." arXiv preprint arXiv:2303.04226, 2023.

[CLY14] Meng Chen, Yang Liu, and Xiaohui Yu. "Nlpmm: A next location predictor with markov modeling." In Advances in Knowledge Discovery and Data Mining: Pacific-Asia Conference, 2014.

[CML11] Eunjoon Cho, Seth A Myers, and Jure Leskovec. "Friendship and mobility: user movement in location-based social networks." In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2011.

[CML25] Derin Cayir, Reham Mohamed, Riccardo Lazzeretti, Marco Angelini, Abbas Acar, Mauro Conti, Z Berkay Celik, and Selcuk Uluagac. "Speak Up, I'm listening: Extracting speech from zero-permission VR sensors." In NDSS, 2025.

[CNA20] Souti Chattopadhyay, Nicholas Nelson, Audrey Au, Natalia Morales, Christopher Sanchez, Rahul Pandita, and Anita Sarma. "A tale from the trenches: cognitive biases and software development." In ICSE, 2020.

[CNM25] Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, JianZhao JianZhao, Kai Yu, and Xie Chen. "F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching." In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 6255–6271, 2025.

[CSZ26] Kunlin Cai, Rui Song, Jinghuai Zhang, Kaiyuan Zhang, Pranav Bodapati, Alicia Yu, Fnu Suya, Mohammad Rostami, Jiaqi Ma, and Yuan Tian. "What-If World: A Causal Benchmark for General World Models in Embodied Scenarios." arXiv preprint arXiv:2605.27589, 2026.

[CTK23] 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.

[CTW21] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. "Extracting training data from large language models." In 30th USENIX Security Symposium (USENIX Security 21), pp. 2633–2650, 2021.

[CV95] Corinna Cortes and Vladimir Vapnik. "Support-vector networks." Machine learning, **20**:273–297, 1995.

[CWC22] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. "Wavlm: Large-scale self-supervised pre-training for full stack speech processing." IEEE Journal of Selected Topics in Signal Processing, **16**(6):1505–1518, 2022.

[CWE18] Wei Cai, Zehua Wang, Jason B Ernst, Zhen Hong, Chen Feng, and Victor CM Leung. "Decentralized applications: The blockchain-empowered software system." IEEE access, **6**:53019–53033, 2018.

[Cyb21] Cyber XR Coalition. "Immersive Technology Standards for Privacy, Safety, and Security." Technical report, Cyber XR Coalition, 2021. Accessed July 22, 2025.

[CYL13] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. "Where you like to go next: Successive point-of-interest recommendation." In International Joint Conference on Artificial Intelligence, 2013.

[CZH24] Kunlin Cai, Jinghuai Zhang, Zhiqing Hong, William Shand, Guang Wang, Desheng Zhang, Jianfeng Chi, and Yuan Tian. "Where have you been? A study of privacy risk for point-of-interest recommendation." In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 175–186, 2024.

[CZL25] Kunlin Cai, Jinghuai Zhang, Ying Li, Zhiyuan Wang, Xun Chen, Tianshi Li, and Yuan Tian. "From Perception to Protection: A Developer-Centered Study of Security and Privacy Threats in Extended Reality (XR)." arXiv preprint arXiv:2509.06368, 2025.

[CZS23] Guangke Chen, Yedi Zhang, and Fu Song. "Slmia-sr: Speaker-level membership inference attacks against speaker recognition systems." arXiv preprint arXiv:2309.07983, 2023.

[DAR23] DARPA. "Intrinsic Cognitive Security (ICS).", Oct 2023.

[Das24] Sarina Dastgerdy. "Virtual Reality and Augmented Reality Security: A Reconnaissance and Vulnerability Assessment Approach." arXiv preprint arXiv:2407.15984, 2024.

[DBG13] John David N. Dionisio, William G. Burns III, and Richard Gilbert. "3D Virtual Worlds and the Metaverse: Current Status and Future Possibilities." In ACM Computing Surveys, volume 45, pp. 1–38, 2013.

[DBW19] John Dudley, Hrvoje Benko, Daniel Wigdor, and Per Ola Kristensson. "Performance envelopes of virtual keyboard text input strategies in virtual reality." In 2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 289–300. IEEE, 2019.

[DCA20] Deeksha Dangwal, Meghan Cowan, Armin Alaghi, Vincent T Lee, Brandon Reagen, and Caronline Trippel. "SoK: Opportunities for software-hardware-security codesign for next generation secure computing." In Proceedings of the 9th International Workshop on Hardware and Architectural Support for Security and Privacy, 2020.

[DCV23] Anuj Dubey, Rosario Cammarota, Avinash Varna, Raghavan Kumar, and Aydin Aysu. "Hardware-software co-design for side-channel protected neural network inference." In HOST. IEEE, 2023.

[DCZ24] Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, et al. "Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens." arXiv preprint arXiv:2407.05407, 2024.

[DDK14] Tamara Denning, Zakariya Dehlawi, and Tadayoshi Kohno. "In situ with bystanders of augmented reality glasses: Perspectives on recording and privacy-mediating technologies." In CHI, 2014.

[DFH23] Sauvik Das, Cori Faklaris, Jason I. Hong, and Laura A. Dabbish. Now Foundations and Trends, 2023.

[DKG20] Isin Demirsahin, Oddur Kjartansson, Alexander Gutkin, and Clara Rivera. "Open-source multi-speaker corpora of the english accents in the british isles." In Proceedings of the twelfth language resources and evaluation conference, pp. 6532–6541, 2020.

[DKW23] Jinhao Duan, Fei Kong, Shiqi Wang, Xiaoshuang Shi, and Kaidi Xu. "Are diffusion models vulnerable to membership inference attacks?" In International Conference on Machine Learning, pp. 8717–8730. PMLR, 2023.

[DL68] John M Darley and Bibb Latané. "Bystander intervention in emergencies: diffusion of responsibility." Journal of personality and social psychology, 1968.

[DM80] Steven B. Davis and Paul Mermelstein. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences." IEEE Transactions on Acoustics, Speech, and Signal Processing, **28**(4):357–366, 1980.

[DR14] Cynthia Dwork, Aaron Roth, et al. "The algorithmic foundations of differential privacy." 2014.

[dsk] dsky. "VR tech 411 : 6DoF, XYZ + YPR, position + orientation in 3space." https://blog.dsky.co/2015/05/13/vr-tech-411-6dof-xyz-ypr-position-orientation-in-3space/.

[DTD20] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification." In Helen Meng, Bo Xu, and Thomas Fang Zheng, editors, 21st Annual Conference of the International Speech Communication Association, Interspeech 2020, Virtual Event, Shanghai, China, October 25-29, 2020, pp. 3830–3834. ISCA, 2020.

[DWC24] Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, et al. "Cosyvoice 2: Scalable streaming speech synthesis with large language models." arXiv preprint arXiv:2412.10117, 2024.

[DXW22] Jiaxin Ding, Shichuan Xi, Kailong Wu, Pan Liu, Xinbing Wang, and Chenghu Zhou. "Analyzing sensitive information leakage in trajectory embedding models." In International Conference on Advances in Geographic Information Systems, 2022.

[DZC25] Linkang Du, Zheng Zhu, Min Chen, Zhou Su, Shouling Ji, Peng Cheng, Jiming Chen, and Zhikun Zhang. "ArtistAuditor: Auditing Artist Style Pirate in Text-to-Image Generation Models." In Proceedings of the ACM on Web Conference 2025, pp. 2500–2513, 2025.

[El 24] Mohammed El-Hajj. "Cybersecurity and Privacy Challenges in Extended Reality: Threats, Solutions, and Risk Mitigation Strategies." In Virtual Worlds. MDPI, 2024.

[EPI23] EPIC. "Easy Anti-Cheat." https://easy.ac/en-us/, October 2023.

[Eur16] European Parliament and Council of the European Union. "Regulation (EU) No 2016/679: General Data Protection Regulation (GDPR)." Official Journal of the European Union, May 2016.

[EWT24] Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, et al. "E2 tts: Embarrassingly easy fully non-autoregressive zero-shot tts." In 2024 IEEE spoken language technology workshop (SLT), pp. 682–689. IEEE, 2024.

[fac23] "FACEIT Client Anit-cheat." https://www.faceit.com/en/anti-cheat, October 2023.

[FDM19] Natasha Fernandes, Mark Dras, and Annabelle McIver. "Generalised differential privacy for text document processing." In Principles of Security and Trust: 8th International Conference, POST 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, Prague, Czech Republic, April 6–11, 2019, Proceedings 8, pp. 123–148. Springer International Publishing, 2019.

[Fel20] Vitaly Feldman. "Does learning require memorization? a short tale about a long tail." In Proceedings of the 52nd annual ACM SIGACT symposium on theory of computing, pp. 954–959, 2020.

[FHE12] Adrienne Porter Felt, Elizabeth Ha, Serge Egelman, Ariel Haney, Erika Chin, and David Wagner. "Android permissions: User attention, comprehension, and behavior." In SOUPS, 2012.

[FHF18] Daniel Freeman, Polly Haselton, Jason Freeman, Bernhard Spanlang, Sameer Kishore, Emily Albery, Megan Denne, Poppy Brown, Mel Slater, and Alecia Nickless. "Automated psychological therapy using immersive virtual reality for treatment of fear of heights: a single-blind, parallel-group, randomised controlled trial." The Lancet Psychiatry, 2018.

[FJR15a] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. "Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures." In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, p. 1322–1333, New York, NY, USA, 2015.

[FJR15b] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. "Model inversion attacks that exploit confidence information and basic countermeasures." In ACM SIGSAC Conference on Computer and Communications Security, 2015.

[FLD18] Angela Fan, Mike Lewis, and Yann Dauphin. "Hierarchical neural story generation." 2018.

[FMN23] 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), pp. 877–894, 2023.

[FWS21] Michael Fröhlich, Maurizio Raphael Wagenhaus, Albrecht Schmidt, and Florian Alt. "Don't stop me now! exploring challenges of first-time cryptocurrency users." In Proceedings of the 2021 ACM designing interactive systems conference, pp. 138–148, 2021.

[FWT22] Michael Fröhlich, Franz Waltenberger, Ludwig Trotter, Florian Alt, and Albrecht Schmidt. "Blockchain and cryptocurrency in human computer interaction: a systematic literature review and research agenda." In Proceedings of the 2022 ACM Designing Interactive Systems Conference, pp. 155–177, 2022.

[FZM22] Guo Freeman, Samaneh Zamanifard, Divine Maloney, and Dane Acena. "Disturbing the peace: Experiencing and mitigating emerging harassment in social virtual reality." CSCW, 2022.

[GAA23] Maziar Gomrokchi, Susan Amin, Hossein Aboutalebi, Alexander Wong, and Doina Precup. "Membership Inference Attacks Against Temporally Correlated Data in Deep Reinforcement Learning." IEEE Access, 2023.

[gal24] "Galaxity." https://www.meta.com/experiences/2337801553010648/, April 10 2024.

[Gama] Gabriel Gambetta. "Fast-Paced Multiplayer (Part III): Entity Interpolation." https://www.gabrielgambetta.com/entity-interpolation.html.

[Gamb] Exit Games. "Photon: The Networking Library for Unity.".

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. "Deep Learning." In MIT Press, pp. 221–265, 2016.

[GDL24] Hanyang Guo, Hong-Ning Dai, Xiapu Luo, Zibin Zheng, Gengyang Xu, and Fengliang He. "An Empirical Study on Oculus Virtual Reality Applications: Security and Privacy Perspectives." In Proceedings of the IEEE/ACM 46th International Conference on Software Engineering, pp. 1–13, 2024.

[GDP25] GDPR Advisor. "GDPR and Augmented Reality Advertising: Ensuring Consumer Privacy." https://www.gdpr-advisor.com/gdpr-and-augmented-reality-advertising-ensuring-consumer-privacy/, 2025.

[GG03] Marco Gruteser and Dirk Grunwald. "Anonymous usage of location-based services through spatial and temporal cloaking." In International Conference on Mobile Systems, Applications and Services, 2003.

[Gia24] Alberto Giaretta. "Security and privacy in virtual reality: a literature survey." Virtual Reality, 2024.

[GIS22] Peter Leo Gorski, Luigi Lo Iacono, and Matthew Smith. "Eight lightweight usable security principles for developers." IEEE Security & Privacy, 2022.

[GKP14] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. "De-anonymization attack on geolocated data." 2014.

[GL05] Bugra Gedik and Ling Liu. "Location privacy in mobile systems: A personalized anonymization model." In IEEE International Conference on Distributed Computing Systems, 2005.

[GMA23] Siem Ghirmai, Daniel Mebrahtom, Moayad Aloqaily, Mohsen Guizani, and Merouane Debbah. "Self-Sovereign Identity for Trust and Interoperability in the Metaverse.", 2023.

[GNS23] Gonzalo Munilla Garrido, Vivek Nair, and Dawn Song. "Sok: Data privacy in virtual reality." arXiv preprint arXiv:2301.05940, 2023.

[Goo24] Google LLC. "ARCore Geospatial API: Build Global-Scale, Location-Based AR Experiences." Technical report, Google, 2024. Google for Developers. Accessed: 2026-04-13.

[Goo25] Google. "Android Security Best Practices." `https://source.android.com/docs/security/best-practices`, 2025. Accessed: 2025-04-20.

[GP09] Philippe Golle and Kurt Partridge. "On the anonymity of home/work location pairs." In Pervasive Computing: 7th International Conference, Pervasive 2009, Nara, Japan, May 11-14, 2009. Proceedings 7, pp. 390–397. Springer, 2009.

[GR24] Alvin Wang Graylin and Louis Rosenberg. Our next reality: How the AI-powered metaverse will reshape the world. Nicholas Brealey, 2024.

[Gra23] Grand View Research, Inc. "Spatial Computing Market Size, Share & Trends Analysis Report, 2030." Technical report, Grand View Research, 2023. Accessed: 2026-04-13.

[Gre18] Jason Gregory. Game engine architecture. AK Peters/CRC Press, 2018.

[GS87] Hector Garcia-Molina and Kenneth Salem. "Sagas." ACM Sigmod Record, **16**(3):249–259, 1987.

[GST21] Jaybie Agullo de Guzman, Aruna Seneviratne, and Kanchana Thilakarathna. "Unravelling spatial privacy risks of mobile mixed reality data." IMWUT, 2021.

[GSW23] Sindhu Reddy Kalathur Gopal, Diksha Shukla, James David Wheelock, and Nitesh Saxena. "Hidden Reality: Caution, Your Hand Gesture Inputs in the Immersive Virtual World are Visible to All!" In 32nd USENIX Security Symposium (USENIX Security 23), pp. 859–876, 2023.

[GTM22] Jan Gugenheimer, Wen-Jie Tseng, Abraham Hani Mhaidli, Jan Ole Rixen, Mark McGill, Michael Nebeling, Mohamed Khamis, Florian Schaub, and Sanchari Das. "Novel challenges of safety, security and privacy in extended reality." In CHI Extended Abstracts, 2022.

[GTZ24] Roberto Gallotta, Graham Todd, Marvin Zammit, Sam Earle, Antonios Liapis, Julian Togelius, and Georgios N Yannakakis. "Large language models and games: A survey and roadmap." IEEE Transactions on Games, 2024.

[GZK21] Jan Gogoll, Niina Zuber, Severin Kacianka, Timo Greger, Alexander Pretschner, and Julian Nida-Rümelin. "Ethics in the software development process: from codes of conduct to ethical deliberation." Philosophy & Technology, 2021.

[HDG24] Stefan Albert Horstmann, Samuel Domiks, Marco Gutfleisch, Mindy Tran, Yasemin Acar, Veelasha Moonsamy, and Alena Naiakshina. ""Those things are written by lawyers, and programmers are reading that." Mapping the Communication Gap Between Software Developers and Privacy Experts." PETS, 2024.

[HGW23] Thien Huynh-The, Thippa Reddy Gadekallu, Weizheng Wang, Gokul Yenduri, Pasika Ranaweera, Quoc-Viet Pham, Daniel Benevides da Costa, and Madhusanka Liyanage. "Blockchain for the Metaverse: A Review." Future Generation Computer Systems, **143**:401–419, 2023.

[HGX06] Baik Hoh, Marco Gruteser, Hui Xiong, and Ansaf Alrabady. "Enhancing security and privacy in traffic-monitoring systems." 2006.

[HHB18] Wajih Ul Hassan, Saad Hussain, and Adam Bates. "Analysis of privacy protections in fitness tracking social networks-or-you can run, but can you hide?" In USENIX Security Symposium, 2018.

[HHL24] Sihao Hu, Tiansheng Huang, Gaowen Liu, Ramana Rao Kompella, Fatih Ilhan, Selim Furkan Tekin, Yichang Xu, Zachary Yahn, and Ling Liu. "A survey on large language model-based game agents." arXiv preprint arXiv:2404.02039, 2024.

[HLL17] Jing He, Xin Li, and Lejian Liao. "Category-aware next point-of-interest recommendation via listwise bayesian personalized ranking." In International Joint Conference on Artificial Intelligence, 2017.

[HM16] Eszter Hargittai and Alice Marwick. ""What Can I Really Do?" Explaining the Privacy Paradox with Online Apathy." International Journal of Communication, **10**:3737–3757, 2016.

[HNS25] Sohyeon Hwang, Priyanka Nanayakkara, and Yan Shvartzshnaider. "Trust and Friction: Negotiating How Information Flows through Decentralized Social Media." In Proceedings of the ACM on Human-Computer Interaction (CSCW). ACM, 2025.

[HP23] Hailong Hu and Jun Pang. "Membership inference of diffusion models." arXiv preprint arXiv:2301.09956, 2023.

[HRR24] Emmie Hine, Isadora Neroni Rezende, Huw Roberts, David Wong, Mariarosaria Taddeo, and Luciano Floridi. "Safety and privacy in immersive extended reality: An analysis and policy recommendations." Digital Society, 2024.

[HSI16] Takahiro Hara, Akiyoshi Suzuki, Mayu Iwata, Yuki Arase, and Xing Xie. "Dummy-based user location anonymization under real-world constraints." 2016.

[HSS13] Benjamin Henne, Christian Szongott, and Matthew Smith. "SnapMe if you can: Privacy threats of other peoples' geo-tagged media and what we can do about it." In ACM Conference on Security and Privacy in Wireless and Mobile Networks, 2013.

[HSW24] Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, et al. "Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation." In 2024 IEEE Spoken Language Technology Workshop (SLT), pp. 885–890. IEEE, 2024.

[HWN24] Hilda Hadan, Derrick M. Wang, Lennart E. Nacke, and Leah Zhang-Kennedy. "Privacy in Immersive Extended Reality: Exploring User Perceptions, Concerns, and Coping Strategies." In Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI). ACM, 2024.

[HZH26] Hangrui Hu, Xinfa Zhu, Ting He, Dake Guo, Bin Zhang, Xiong Wang, Zhifang Guo, Ziyue Jiang, Hongkun Hao, Zishan Guo, et al. "Qwen3-TTS Technical Report." arXiv preprint arXiv:2601.15621, 2026.

[iee25] "IEEE Conference on Virtual Reality and 3D User Interfaces." In IEEE VR, 2025. 32nd annual conference held in Saint-Malo, France.

[IMD20] Md. Ashraful Islam, Mir Mahathir Mohammad, Sarkar Snigdha Sarathi Das, and Mohammed Eunus Ali. "A survey on deep learning based Point-Of-Interest (POI) recommendations.", 2020.

[imm25] "XR Community – ImmersiveUnity." `https://xrcommunity.immersiveunity.com/`, 2025.

[Ind25] Anduril Industries. "Anduril and Meta Team Up to Transform XR for the American Military.", 2025. Accessed: 2025-07-14.

[inp] "Input System." `https://docs.unity3d.com/Packages/com.unity.inputsystem@1.0/manual/index.html`.

[JDT23] Ismat Jarin, Yu Duan, Rahmadi Trimananda, Hao Cui, Salma Elmalaki, and Athina Markopoulou. "BehaVR: User Identification Based on VR Sensor Data." arXiv preprint arXiv:2308.07304, 2023.

[JF20] Pascal Jansen and Fabian Fischbach. "The social engineer: An immersive virtual reality educational game to raise social engineering awareness." In Extended Abstracts of the 2020 Annual Symposium on Computer-Human Interaction in Play, 2020.

[JST09] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard. "Networking Named Content." In Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies (CoNEXT). ACM, 2009.

[JUO20] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. "Auditing differentially private machine learning: How private is private SGD?" 2020.

[JWH07] Tao Jiang, Helen J Wang, and Yih-Chun Hu. "Preserving location privacy in wireless LANs." In International Conference on Mobile Systems, Applications and Services, 2007.

[JWS24] Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Yanqing Liu, Yichong Leng, Kaitao Song, Siliang Tang, et al. "Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models." arXiv preprint arXiv:2403.03100, 2024.

[KB17] Martin Kleppmann and Alastair R Beresford. "A conflict-free replicated JSON datatype." IEEE Transactions on Parallel and Distributed Systems, **28**(10):2733–2746, 2017.

[KBO21] Veronika Krauß, Alexander Boden, Leif Oppermann, and René Reiners. "Current practices, challenges, and design implications for collaborative ar/vr application development." In CHI, 2021.

[KDF15] Ruogu Kang, Laura Dabbish, Nathaniel Fruchter, and Sara Kiesler. ""My Data Just Goes Everywhere": User Mental Models of the Internet and Implications for Privacy and Security." In Proceedings of the Eleventh Symposium on Usable Privacy and Security (SOUPS). ACM, 2015.

[KDM23] Fei Kong, Jinhao Duan, RuiPeng Ma, Hengtao Shen, Xiaofeng Zhu, Xiaoshuang Shi, and Kaidi Xu. "An efficient membership inference attack for the diffusion model by proximal initialization." arXiv preprint arXiv:2305.18355, 2023.

[KFG24] Martin Kleppmann, Paul Frazee, Jake Gold, Jay Graber, Daniel Holmgren, Devin Ivy, Jeromy Johnson, Bryan Newbold, and Jaz Volpert. "Bluesky and the AT Protocol: Usable Decentralized Social Media." In Proceedings of the ACM CoNEXT Workshop on the Decentralization of the Internet. ACM, 2024.

[KKS21] Jaehyeon Kim, Jungil Kong, and Juhee Son. "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech." In International conference on machine learning, pp. 5530–5540. PMLR, 2021.

[KMF17] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. "Lightgbm: A highly efficient gradient boosting decision tree." Advances in neural information processing systems, **30**, 2017.

[KMP24] Preeti Kathiria, Sapan H Mankad, Jitali Patel, Mayank Kapadia, and Neel Lakdawala. "Assistive systems for visually impaired people: A survey on current requirements and advancements." Neurocomputing, **606**:128284, 2024.

[KMR19] Sai Anirudh Karre, Neeraj Mathur, and Y Raghu Reddy. "Is Virtual Reality product development different? An empirical study on VR product development practices." In ISEC, 2019.

[Kru94] Richard A Krueger. Focus Groups: A Practical Guide for Applied Research. SAGE Publications, 1994.

[Kru07] John Krumm. "Inference attacks on location tracks." In Pervasive Computing: 5th International Conference, PERVASIVE 2007, Toronto, Canada, May 13-16, 2007. Proceedings 5, pp. 127–143. Springer, 2007.

[KRZ20] Jacob Kahn, Morgane Riviere, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al. "Libri-light: A benchmark for asr with limited or no supervision." In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7669–7673. IEEE, 2020.

[KW18] Dejiang Kong and Fei Wu. "HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction." In International Joint Conference on Artificial Intelligence, 2018.

[KWC24] Christopher Katins, Paweł W Woźniak, Aodi Chen, Ihsan Tumay, Luu Viet Trinh Le, John Uschold, and Thomas Kosch. "Assessing User Apprehensions About Mixed Reality Artifacts and Applications: The Mixed Reality Concerns (MRC) Questionnaire." In Proceedings of the CHI Conference on Human Factors in Computing Systems, pp. 1–13, 2024.

[KWH19] Martin Kleppmann, Adam Wiggins, Peter van Hardenberg, and Mark McGranaghan. "Local-First Software: You Own Your Data, in spite of the Cloud." In Proceedings of the 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward!), pp. 154–178. ACM, 2019.

[KWR] Max Ku, Cong Wei, Weiming Ren, Huan Yang, and Wenhu Chen. "AnyV2V: A Tuning-Free Framework For Any Video-to-Video Editing Tasks." Transactions on Machine Learning Research.

[KZK23] Yuma Koizumi, Heiga Zen, Shigeki Karita, Yifan Ding, Kohei Yatabe, Nobuyuki Morioka, Michiel Bacchiani, Yu Zhang, Wei Han, and Ankur Bapna. "LibriTTS-R: A Restored Multi-Speaker Text-to-Speech Corpus." In Proc. Interspeech 2023, pp. 5496–5500, 2023.

[LAH18] Tianshi Li, Yuvraj Agarwal, and Jason I Hong. "Coconut: An IDE plugin for developing privacy-friendly apps." IMWUT, 2018.

[Lam23] Alyssa Lamberti. "What is Acceptable Packet Loss? 10`https://obkio.com/blog/acceptable-packet-loss/`, Mar 31 2023.

[LBB98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, **86**(11):2278–2324, 1998.

[LBZ21] Lik-Hang Lee, Tristan Braud, Pengyuan Zhou, Lin Wang, Dianlei Xu, Zijun Lin, Abhishek Kumar, Carlos Bermejo, and Pan Hui. "All One Needs to Know about Metaverse: A Complete Survey on Technological Singularity, Virtual Ecosystem, and Research Agenda." CoRR, **abs/2110.05352**, 2021.

[LDH25] Jingwei Li, Jing Dong, Tianxing He, and Jingzhao Zhang. "Towards Black-Box Membership Inference Attack for Diffusion Models." In International Conference on Machine Learning, pp. 34271–34287. PMLR, 2025.

[LGT23] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. "Magic3d: High-resolution text-to-3d content creation." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 300–309, 2023.

[LHY22] Shiqing Luo, Xinyu Hu, and Zhisheng Yan. "Holologger: Keystroke inference on mixed reality head mounted displays." In 2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), pp. 445–454. IEEE, 2022.

[LKR16] 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, 2016.

[LKY22] Jaemin Lim, Kiyeon Kim, Hyunwoo Yu, and Suk-Bok Lee. "Overo: Sharing private audio recordings." In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, pp. 1933–1946, 2022.

[LLD21] Tianshi Li, Elizabeth Louie, Laura Dabbish, and Jason I Hong. "How developers talk about personal data and what it means for user privacy: A case study of a developer forum on reddit." CSCW, 2021.

[LLK22] Geoffrey Litt, Sarah Lim, Martin Kleppmann, and Peter Van Hardenberg. "Peritext: A crdt for collaborative rich text editing." Proceedings of the ACM on Human-Computer Interaction, **6**(CSCW2):1–36, 2022.

[LLL17] Yanchi Liu, Chuanren Liu, Xinjiang Lu, Mingfei Teng, Hengshu Zhu, and Hui Xiong. "Point-of-Interest demand modeling with human mobility patterns." In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017.

[LLL21] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. "Stan: Spatio-temporal attention network for next location recommendation." In Web Conference, 2021.

[LMH22] Shiyong Lan, Yitong Ma, Weikang Huang, Wenwu Wang, Hongyu Yang, and Pyang Li. "Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting." In International Conference on Machine Learning, 2022.

[LNE23] Dennis Lawo, Thomas Neifer, Margarita Esau-Held, and Gunnar Stevens. "Digital Sovereignty: What it is and why it matters for HCI." In Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems, pp. 1–7, 2023.

[LNF24] Shiqing Luo, Anh Nguyen, Hafsa Farooq, Kun Sun, and Zhisheng Yan. "Eavesdropping on controller acoustic emanation for keystroke inference attack in virtual reality." In NDSS, 2024.

[LRA22] Tianshi Li, Kayla Reiman, Yuvraj Agarwal, Lorrie Faith Cranor, and Jason I Hong. "Understanding challenges for developers to create accurate privacy nutrition labels." In CHI, 2022.

[LRK18] Kiron Lebeck, Kimberly Ruth, Tadayoshi Kohno, and Franziska Roesner. "Arya: Operating system support for securely augmenting reality." IEEE S&P, 2018.

[LSL22] Jacob Logas, Ari Schlesinger, Zhouyu Li, and Sauvik Das. "Image DePO: towards gradual decentralization of online social networks using decentralized privacy overlays." Proceedings of the ACM on Human-Computer Interaction, **6**(CSCW1):1–28, 2022.

[LTS23] Minzhao Lyu, Rahul Dev Tripathi, and Vijay Sivaraman. "MetaVRadar: Measuring Metaverse Virtual Reality Network Activity." POMACS, 2023.

[LVS23] Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. "Voicebox: Text-guided multilingual universal speech generation at scale." Advances in neural information processing systems, **36**:14005–14034, 2023.

[LWG20] Defu Lian, Yongji Wu, Yong Ge, Xing Xie, and Enhong Chen. "Geography-aware sequential location recommendation." In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2020.

[LWW16] Q. Liu, Shu Wu, Liang Wang, and Tieniu Tan. "Predicting the next location: A recurrent model with spatial and temporal contexts." In AAAI Conference on Artificial Intelligence, 2016.

[Lyo] Kim Lyons. "Rec Room rides uptick in users during the pandemic to become a VR unicorn." https://www.theverge.com/2021/3/25/22350421/rec-room-teenagers-gaming-users-pandemic-virtual-reality.

[LYY23] Taotao Li, Changlin Yang, Qinglin Yang, Siqi Zhou, Huawei Huang, and Zibin Zheng. "MetaOpera: A Cross-Metaverse Interoperability Protocol.", 2023.

[LZ20] Kai Liu and Huan Zhou. "Text-independent speaker verification with adversarial learning on short utterances." In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6569–6573. IEEE, 2020.

[LZK25] Zhehui Liao, Hanwen Zhao, Ayush Kulkarni, Shaan Singh Chattrath, and Amy X. Zhang. "Building Proactive and Instant-Reactive Safety Designs to Address Harassment in Social Virtual Reality." In Proceedings of the ACM on Human-Computer Interaction (CSCW). ACM, 2025.

[LZX14] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. "GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation." In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014.

[Mag21] Security Magazine. "The Next Frontier in Cybersecurity: Mitigating Normalcy Bias.", 2021. Accessed: 2025-04-16.

[Mb14] Alice E Marwick and danah boyd. "Networked Privacy: How Teenagers Negotiate Context in Social Media." New Media & Society, **16**(7):1051–1067, 2014.

[McH12] Mary L McHugh. "Interrater reliability: the kappa statistic." Biochemia medica, 2012.

[MCH17] Chuishi Meng, Yu Cui, Qing He, Lu Su, and Jing Gao. "Travel purpose inference with GPS trajectories, POIs, and geo-tagged social media data." In IEEE International Conference on Big Data, 2017.

[McK22] McKinsey & Company. "Value creation in the metaverse: The real business of the virtual world." Report, McKinsey & Company, 2022.

[MDD24] Samuel Mayworm, Michael Ann DeVito, Daniel Delmonaco, Hibby Thach, and Oliver L. Haimson. "Content Moderation Folk Theories and Perceptions of Platform Spirit among Marginalized Social Media Users." ACM Transactions on Social Computing, **7**(1):1–27, 2024.

[mem23] "Memorable Password Generator." https://springhole.net/writing_roleplaying_randomators/memorable-password.htm, August 2023.

[met] "Meta Community Forums." https://communityforums.atmeta.com/. Accessed: 2025-04-20.

[Met21] Meta Platforms, Inc. "Building the Metaverse Responsibly." https://about.fb.com/news/2021/09/building-the-metaverse-responsibly/, 2021. Accessed: 2025-04-20.

[Met22] Meta Platforms, Inc. "Meta Quest Pro Privacy: What You Need to Know." https://www.meta.com/blog/meta-quest-pro-privacy/, October 2022. Accessed: 2026-04-13.

[Met24a] Meta. "Hand and Body Privacy Notice.", 2024.

[Met24b] Meta. "Meta Quest Virtual Reality Check (VRC) Guidelines." https://developer.oculus.com/resources/publish-quest-req/, 2024.

[met25] "Developer Category - Meta Community Forums." https://communityforums.atmeta.com/category/developer, 2025. Accessed: 2025-07-17.

[MHJ20] Mark Roman Miller, Fernanda Herrera, Hanseul Jun, James A Landay, and Jeremy N Bailenson. "Personal identifiability of user tracking data during observation of 360-degree VR video." Scientific Reports, **10**(1):17404, 2020.

[Mik21] Mike Boland. "Foursquare's power play continues with relaunched places and new API.", 2021.

[Mik24] Mikadumont. "Code analysis using .net compiler platform (roslyn) analyzers - visual studio (windows).", 2024.

[mir] "Mirror Networking." https://mirror-networking.gitbook.io/docs/.

[MMY23] Tomoya Matsumoto, Takayuki Miura, and Naoto Yanai. "Membership inference attacks against diffusion models." In 2023 IEEE Security and Privacy Workshops (SPW), pp. 77–83. IEEE, 2023.

[Mor13] David L Morgan. "Paired depth interviews: A comment on Bjornholt and Farstad." Qualitative Research, **16**(1):109–112, 2013.

[MR09] Joseph Meyerowitz and Romit Roy Choudhury. "Hiding stars with fireworks: location privacy through camouflage." In Annual International Conference on Mobile Computing and Networking, 2009.

[MSF19] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. "Reliability and inter-rater reliability in qualitative research: Norms and guidelines for CSCW and HCI practice." CSCW, 2019.

[MTU95] 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, pp. 282–292. Spie, 1995.

[Mys22] Stylianos Mystakidis. "Metaverse." Encyclopedia, **2**(1):486–497, 2022.

[MZF20] Divine Maloney, Samaneh Zamanifard, and Guo Freeman. "Anonymity vs. familiarity: Self-disclosure and privacy in social virtual reality." In VRST, 2020.

[ND24] Naheem Noah and Sanchari Das. "Privacy and Security in Extended Reality: Exploring the Risks of External Biometric Data Collection." Available at SSRN 4780358, 2024.

[NDT17] Alena Naiakshina, Anastasia Danilova, Christian Tiefenau, Marco Herzog, Sergej Dechand, and Matthew Smith. "Why do developers get password storage wrong? A qualitative usability study." In ACM CCS, 2017.

[net] "Source Multiplayer Networking." `https://developer.valvesoftware.com/wiki/Source_Multiplayer_Networking`.

[net23] "Unity Netcode." `https://unity.com/products/netcode`, October 2023.

[New22] Daniel Newman. "NVIDIA GTC 2022 Debuts New Omniverse Metaverse Tools." `https://futurumgroup.com/insights/nvidia-gtc-2022-debuts-new-omniverse-metaverse-tools/`, September 2022. The Futurum Group. Accessed: 2026-04-13.

[NGM23] 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.

[NGS22] Vivek Nair, Gonzalo Munilla Garrido, and Dawn Song. "Going incognito in the metaverse." arXiv preprint arXiv:2208.05604, 2022.

[Nie93] Jakob Nielsen. Usability Engineering. Morgan Kaufmann, San Francisco, CA, 1993.

[Nis04] Helen Nissenbaum. "Privacy as Contextual Integrity." Washington Law Review, **79**(1):119–158, 2004.

[NKF23] Matthew N Nicholson, Brian C Keegan, and Casey Fiesler. "Mastodon rules: characterizing formal rules on popular Mastodon instances." In Companion Publication of the 2023 Conference on Computer Supported Cooperative Work and Social Computing, pp. 86–90, 2023.

[NPS25] Ali Naseh, Yuefeng Peng, Anshuman Suri, Harsh Chaudhari, Alina Oprea, and Amir Houmansadr. "Riddle me this! stealthy membership inference for retrieval-augmented generation." In Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security, pp. 1245–1259, 2025.

[NRO23] Vivek Nair, Louis Rosenberg, James F. O'Brien, and Dawn Song. "Truth in Motion: The Unprecedented Risks and Opportunities of Extended Reality Motion Data.", 2023.

[NSA24] NSA's Research Directorate. "Ghidra." https://ghidra-sre.org/, 2024.

[NSH18] Milad Nasr, Reza Shokri, and Amir Houmansadr. "Comprehensive privacy analysis of deep learning." In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), volume 2018, pp. 1–15, 2018.

[NST21] Milad Nasr, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlin. "Adversary instantiation: Lower bounds for differentially private machine learning." In 2021 IEEE Symposium on security and privacy (SP), pp. 866–882. IEEE, 2021.

[Nur25] Let's Nurture. "How Augmented Reality (AR) is Transforming Healthcare in 2025: Benefits and Applications.", 2025. Accessed: 2025-07-14.

[Nvi23] Nvidia. "Nvidia VCR." https://info.nvidia.com/xr-vcr-reg-page.html, March 2023.

[nvi25] "XR & Spatial Computing – NVIDIA Developer Forums." https://forums.developer.nvidia.com/c/omniverse/xr-spatial-computing/707, 2025. Accessed: 2025-07-17.

[Off25a] Office of Research Cyberinfrastructure, University of Central Florida. "Research Data Privacy." Online, 2025.

[Off25b] Office of the Vice Provost for Research, Lehigh University. "Guidance: Data Security and Privacy." Online, 2025.

[OO00] Gary M. Olson and Judith S. Olson. "Distance Matters." Human-Computer Interaction, **15**(2–3):139–178, 2000.

[Ope21] Open Worldwide Application Security Project (OWASP). "OWASP Top Ten." https://owasp.org/www-project-top-ten/, 2021. Accessed: 2025-04-22.

[Ope22] OpenAI. "ChatGPT based on GPT-4." https://www.openai.com/, 2022.

[Ope24] OpenAI. "Expanding on How Voice Engine Works and Our Safety Research." Technical report, OpenAI, June 2024. Accessed: 2026-04-13.

[Opp22] Jonas Oppenlaender. "The perception of smart contracts for governance of the metaverse." In Proceedings of the 25th international academic mindtrek conference, pp. 1–8, 2022.

[ovr] "Map Controllers." https://developer.oculus.com/documentation/unity/unity-ovrinput/#unity-ovrinput.

[ovr24] "oVRshot." https://www.meta.com/experiences/pcvr/1379022322193957/, April 10 2024.

[PB23] Jovan Powar and Alastair R Beresford. "SoK: Managing risks of linkage attacks on data privacy." 2023.

[PBS18] Nuno Preguiça, Carlos Baquero, and Marc Shapiro. "Conflict-free replicated data types (CRDTs)." arXiv preprint arXiv:1805.06358, 2018.

[Per24] Charles Perez. "Analyzing transaction networks of popular wearables within the decentraland metaverse." In Proceedings of the 2024 11th Multidisciplinary International Social Networks Conference, pp. 131–138, 2024.

[Pho] "Photon fusion." https://doc.photonengine.com/fusion/current/getting-started/fusion-intro#hosted_mode___server_mode.

[PJB22] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. "Dreamfusion: Text-to-3d using 2d diffusion." arXiv preprint arXiv:2209.14988, 2022.

[POC23] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. "Generative agents: Interactive simulacra of human behavior." In UIST, 2023.

[Pok24] PokeMap. "PokeMap: Pokemon GO Live Radar and Interactive Map." Technical report, PokeMap, 2024. Accessed: 2026-04-13.

[PRC22] Thiago Porcino, Derek Reilly, Esteban Clua, and Daniela Trevisan. "A guideline proposal for minimizing cybersickness in VR-based serious games and applications." In SeGAH. IEEE, 2022.

[PS23] Suchismita Pahi and Calli Schroeder. "Extended Privacy for Extended Reality: XR Technology Has 99 Problems and Privacy Is Several of Them." Notre Dame Journal on Emerging Technologies, apr 2023.

[PTD17] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. "Knock knock, who's there? Membership inference on aggregate location data." 2017.

[PW23] Yan Pang and Tianhao Wang. "Black-box membership inference attacks against fine-tuned diffusion models." arXiv preprint arXiv:2312.08207, 2023.

[PWK25] Yan Pang, Tianhao Wang, Xuhui Kang, Mengdi Huai, and Yang Zhang. "White-box Membership Inference Attacks against Diffusion Models." Proceedings on Privacy Enhancing Technologies, **2025**(2), 2025.

[rec23] "Rec Room." https://recroom.com/, June 15 2023.

[rec24] "Rec Room Ship Notes." https://recroom.com/ship-notes/2024/7/29/rec-room-update-the-my-little-template-edition, July 29 2024.

[Res25] Resemble AI. "Chatterbox-TTS." https://github.com/resemble-ai/chatterbox, 2025. GitHub repository.

[RHL23] Lingfei Ren, Ruimin Hu, Dengshi Li, Zheng Wang, Junhang Wu, Xixi Li, and Wenyi Hu. "Who is your friend: inferring cross-regional friendship from mobility profiles." 2023.

[RK21] 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.

[RKM14] Franziska Roesner, Tadayoshi Kohno, and David Molnar. "Security and privacy for augmented reality systems." Communications of the ACM, 2014.

[Ros58] Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." Psychological review, **65**(6):386, 1958.

[RSL20] Drummond Reed, Manu Sporny, Dave Longley, Christopher Allen, Ryan Grant, Markus Sabadello, and Jonathan Holt. "Decentralized identifiers (dids) v1. 0." Draft Community Group Report, 2020.

[RWP23] Louis Rosenberg, Catriona Wallace, Kavya Pearlman, and Bhanujeet Choudhary. "The Metaverse and Standards." 05 2023.

[SAD24] Mona M Soliman, Eman Ahmed, Ashraf Darwish, and Aboul Ella Hassanien. "Artificial intelligence powered Metaverse: analysis, challenges and future perspectives." Artificial Intelligence Review, **57**(2):36, 2024.

[Sal21] Johnny Saldaña. "The coding manual for qualitative researchers." 2021.

[SAX23] Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. "Detecting pretraining data from large language models." arXiv preprint arXiv:2310.16789, 2023.

[SAY25] Abhinaya SB, Abhishri Agrawal, Yaxing Yao, Yixin Zou, and Anupam Das. "" What are they gonna do with my data?": Privacy Expectations, Concerns, and Behaviors in Virtual Reality." PETS, **2025**, 2025.

[SC03] Hiroaki Sakoe and Seibi Chiba. "Dynamic programming algorithm optimization for spoken word recognition." IEEE transactions on acoustics, speech, and signal processing, **26**(1):43–49, 2003.

[SCB24] 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." arXiv preprint arXiv:2405.14036, 2024.

[SCE23] Ahmed Salem, Giovanni Cherubin, David Evans, Boris Köpf, Andrew Paverd, Anshuman Suri, Shruti Tople, and Santiago Zanella-Béguelin. "Sok: Let the privacy games begin! a unified treatment of data inference privacy in machine learning." In 2023 IEEE Symposium on Security and Privacy (SP), pp. 327–345. IEEE, 2023.

[SCG24] Niloofar Sayadi, Sadie Co, and Diego Gomez-Zara. ""Feeling that I was Collaborating with Them": A 20-year Scoping Review of Social Virtual Reality Leveraging Collaboration." arXiv preprint arXiv:2412.20266, 2024.

[Sch02] Ralph Schroeder. "Social Interaction in Virtual Environments: Key Issues, Common Themes, and a Framework for Research." In The Social Life of Avatars: Presence and Interaction in Shared Virtual Environments, pp. 1–18. Springer, 2002.

[SCL22] Weiyan Shi, Aiqi Cui, Evan Li, Ruoxi Jia, and Zhou Yu. "Selective Differential Privacy for Language Modeling." In Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2022.

[SCT23] Nazanin Sabri, Bella Chen, Annabelle Teoh, Steven P Dow, Kristen Vaccaro, and Mai Elsherief. "Challenges of moderating social virtual reality." In CHI, 2023.

[SCZ22] Weiyan Shi, Si Chen, Chiyuan Zhang, Ruoxi Jia, and Zhou Yu. "Just fine-tune twice: Selective differential privacy for large language models." 2022.

[SFL23] Kelsea Schulenberg, Guo Freeman, Lingyuan Li, and Catherine Barwulor. "" Creepy Towards My Avatar Body, Creepy Towards My Body": How Women Experience and Manage Harassment Risks in Social Virtual Reality." CSCW, 2023.

[SGS18] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. "X-vectors: Robust dnn embeddings for speaker recognition." In 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp. 5329–5333. IEEE, 2018.

[SH12] Mudhakar Srivatsa and Mike Hicks. "Deanonymizing mobility traces: Using social network as a side-channel." In ACM Conference on Computer and Communications Security, 2012.

[Sho14] Adam Shostack. Threat modeling: Designing for security. John wiley & sons, 2014.

[Shu22] Shubham Sharma. "How Foursquare helps enterprises drive positive results with geospatial technology.", 2022.

[sin24] "Sing Together: VR Karaoke." https://store.steampowered.com/app/1781140/Sing_Together_VR_Karaoke/, April 10 2024.

[SJT08] Karen A Scarfone, Wayne Jansen, and Miles Tracy. "Sp 800-123. guide to general server security.", 2008.

[SJT23] Kai Shen, Zeqian Ju, Xu Tan, Yanqing Liu, Yichong Leng, Lei He, Tao Qin, Sheng Zhao, and Jiang Bian. "Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers." arXiv preprint arXiv:2304.09116, 2023.

[SLF23] Kelsea Schulenberg, Lingyuan Li, Guo Freeman, Samaneh Zamanifard, and Nathan J McNeese. "Towards leveraging AI-based moderation to address emergent harassment in social virtual reality." In Proceedings of the 2023 CHI conference on human factors in computing systems, pp. 1–17, 2023.

[SLS22] Manu Sporny, Dave Longley, Markus Sabadello, Drummond Reed, Orie Steele, and Christopher Allen. "Decentralized Identifiers (DIDs) v1.0." W3c recommendation, World Wide Web Consortium (W3C), 2022.

[SMF22] Philipp Sykownik, Divine Maloney, Guo Freeman, and Maic Masuch. "Something personal from the metaverse: goals, topics, and contextual factors of self-disclosure in commercial social VR." In CHI, 2022.

[Smi24] Matthew S. Smith. "Privacy Is Just No Longer a Thing in Augmented Reality?" IEEE Spectrum, 2024. Accessed: 2025-04-23.

[SNJ23] Thomas Steinke, Milad Nasr, and Matthew Jagielski. "Privacy auditing with one (1) training run." Advances in Neural Information Processing Systems, **36**:49268–49280, 2023.

[SPB11] Marc Shapiro, Nuno Preguica, Carlos Baquero, and Marek Zawirski. "Conflict-Free Replicated Data Types." In Proceedings of the 13th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS), pp. 386–400. Springer, 2011.

[SPW18] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions." In 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp. 4779–4783. IEEE, 2018.

[SQC20] Ke Sun, Tieyun Qian, Tong Chen, Yile Liang, Quoc Viet Hung Nguyen, and Hongzhi Yin. "Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation." In AAAI Conference on Artificial Intelligence, 2020.

[SSG23] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. "Diffusion art or digital forgery? investigating data replication in diffusion models." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 6048–6058, 2023.

[SSM21] Muhammad A Shah, Joseph Szurley, Markus Mueller, Athanasios Mouchtaris, and Jasha Droppo. "Evaluating the vulnerability of end-to-end automatic speech recognition models to membership inference attacks." In Proc. Interspeech 2021, pp. 891–895, 2021.

[SSS17a] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. "Membership inference attacks against machine learning models." In IEEE Symposium on Security and Privacy, 2017.

[SSS17b] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. "Membership inference attacks against machine learning models." In IEEE Symposium on Security and Privacy, 2017.

[Sta25a] Staff, Faculty of Archaeology, Leiden University. "Personal and Sensitive Data." Online, 2025.

[Sta25b] Statista. “AR & VR Market - Worldwide.”, 2025. Statista Market Insights. Accessed: 2026-04-13.

[ste23] “Steam Overlay.” https://partner.steamgames.com/doc/features/overlay, October 2023.

[ste24a] “SteamVR.” https://store.steampowered.com/app/250820/SteamVR/, May 13 2024.

[ste24b] “SteamVR Beta Updated - 2.7.1.” https://store.steampowered.com/news/app/250820/view/6829226410396566972, June 19 2024.

[STP13] Reza Shokri, George Theodorakopoulos, Panos Papadimitratos, Ehsan Kazemi, and Jean-Pierre Hubaux. “Hiding in the mobile crowd: Location privacy through collaboration.” 2013.

[SVN37] Stanley Smith Stevens, John Volkmann, and Edwin Broomell Newman. “A scale for the measurement of the psychological magnitude pitch.” The journal of the acoustical society of america, **8**(3):185–190, 1937.

[SW20] Jiacheng Shang and Jie Wu. “Secure voice input on augmented reality headsets.” IEEE Transactions on Mobile Computing, 2020.

[SZA23] 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), pp. 159–174, 2023.

[SZH18] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. “Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models.” 2018.

[SZS24] 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 USENIX Security, 2024.

[TAM23] Shahroz Tariq, Alsharif Abuadbba, and Kristen Moore. “Deepfake in the metaverse: Security implications for virtual gaming, meetings, and offices.” In Proceedings of the 2nd Workshop on Security Implications of Deepfakes and Cheapfakes, 2023.

[TBM22] 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 Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, pp. 1–15, 2022.

[Tec] Unity Technologies. “Unity XR Input.” https://docs.unity3d.com/Manual/xr_input.html.

[TFV21] Mohammad Tahaei, Alisa Frik, and Kami Vaniea. “Privacy champions in software teams: Understanding their motivations, strategies, and challenges.” In CHI, 2021.

[TKL22] Wei-Cheng Tseng, Wei-Tsung Kao, and Hung-yi Lee. "Membership Inference Attacks Against Self-supervised Speech Models." In Proc. Interspeech 2022, pp. 5040–5044, 2022.

[TLC22] 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), pp. 3789–3806, 2022.

[TMC26] Natalia Tomashenko, Xiaoxiao Miao, Pierre Champion, Sarina Meyer, Michele Panariello, Xin Wang, Nicholas Evans, Emmanuel Vincent, Junichi Yamagishi, and Massimiliano Todisco. "The Third VoicePrivacy Challenge: Preserving Emotional Expressiveness and Linguistic Content in Voice Anonymization." arXiv preprint arXiv:2601.11846, 2026.

[TMV25] Natalia Tomashenko, Xiaoxiao Miao, Emmanuel Vincent, and Junichi Yamagishi. "The first voiceprivacy attacker challenge." In ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–2. IEEE, 2025.

[TPO24] Francisco Teixeira, Karla Pizzi, Raphael Olivier, Alberto Abad, Bhiksha Raj, and Isabel Trancoso. "Improving membership inference in asr model auditing with perturbed loss features." arXiv preprint arXiv:2405.01207, 2024.

[TQS21] Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. "A survey on neural speech synthesis." arXiv preprint arXiv:2106.15561, 2021.

[TRG22] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. "Human motion diffusion model." arXiv preprint arXiv:2209.14916, 2022.

[TSS22] Florian Tramèr, Reza Shokri, Ayrton San Joaquin, Hoang Le, Matthew Jagielski, Sanghyun Hong, and Nicholas Carlini. "Truth serum: Poisoning machine learning models to reveal their secrets." In ACM SIGSAC Conference on Computer and Communications Security, 2022.

[TWV22] Natalia Tomashenko, Xin Wang, Emmanuel Vincent, Jose Patino, Brij Mohan Lal Srivastava, Paul-Gauthier Noé, Andreas Nautsch, Nicholas Evans, Junichi Yamagishi, Benjamin O'Brien, et al. "The voiceprivacy 2020 challenge: Results and findings." Computer Speech & Language, **74**:101362, 2022.

[Uni23] Unity. "Unity Documentation: Transform." `https://docs.unity3d.com/ScriptReference/Transform.html`, October 2023.

[USnd] U.S. Department of Veterans Affairs. "VA Publications." `https://www.va.gov/vapubs/`, n.d.

[VFB11] Carmen Ruiz Vicente, Dario Freni, Claudio Bettini, and Christian S Jensen. "Location-related privacy in geo-social networks." 2011.

[VFP20] Daniel Votipka, Kelsey R Fulton, James Parker, Matthew Hou, Michelle L Mazurek, and Michael Hicks. "Understanding security mistakes developers make: Qualitative analysis from build it, break it, fix it." In USENIX Security, 2020.

[VHB14] Nevena Vratonjic, Kévin Huguenin, Vincent Bindschaedler, and Jean-Pierre Hubaux. "A location-privacy threat stemming from the use of shared public IP addresses." 2014.

[Vir22] Linde VirtualAcademy. "Is your frame rate affecting your VR experience." https://vr.linde.com/2022/10/06/is-your-frame-rate-affecting-your-vr-experience/, October 6 2022.

[VRC23] VRChat. "VRChat." https://hello.vrchat.com/, June 15 2023.

[VWM21] Artemij Voskobojnikov, Oliver Wiese, Masoud Mehrabi Koushki, Volker Roth, and Konstantin Beznosov. "The u in crypto stands for usable: An empirical study of user experience with mobile cryptocurrency wallets." In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1–14, 2021.

[WA18] Chamila Wijayarathna and Nalin Asanka Gamagedara Arachchilage. "Am I Responsible for End-User's Security? A Programmer's Perspective." arXiv preprint arXiv:1808.01481, 2018.

[Wan18] Yang Wang. "Inclusive security and privacy." IEEE Security & Privacy, 2018.

[Was10] Rick Wash. "Folk Models of Home Computer Security." In Proceedings of the Sixth Symposium on Usable Privacy and Security (SOUPS). ACM, 2010.

[WCW23] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. "Neural codec language models are zero-shot text to speech synthesizers." arXiv preprint arXiv:2301.02111, 2023.

[Wei22] Markus Weinberger. "What Is Metaverse?—A Definition Based on Qualitative Meta-Synthesis." Future Internet, **14**(11):310, 2022.

[wir23] "Wireshark." https://www.wireshark.org/, June 15 2023.

[WJF22] Xiaoying Wei, Xiaofu Jin, and Mingming Fan. "Communication in Immersive Social Virtual Reality: A Systematic Review of 10 Years' Studies.", 2022.

[WJJ21] Jingyuan Wang, Jiawei Jiang, Wenjun Jiang, Chao Li, and Wayne Xin Zhao. "LibCity: An open library for traffic prediction." In International Conference on Advances in Geographic Information Systems, 2021.

[WLS24] Han Wang, Hui Li, Abla Smahi, Feng Zhao, Yao Yao, Ching Chuen Chan, Shiyu Wang, Wenyuan Yang, and Shuo-Yen Robert Li. "MIS: A multi-identifier management and resolution system in the metaverse." ACM Transactions on Multimedia Computing, Communications and Applications, **20**(7):1–25, 2024.

[WMT18] Sarah G Wood, Jerad H Moxley, Elizabeth L Tighe, and Richard K Wagner. "Does use of text-to-speech and related read-aloud tools improve reading comprehension for students with reading disabilities? A meta-analysis." Journal of learning disabilities, **51**(1):73–84, 2018.

[WQK22] Matthew Weidner, Huairui Qi, Maxime Kjaer, Ria Pradeep, Benito Geordie, Yicheng Zhang, Gregory Schare, Xuan Tang, Sicheng Xing, and Heather Miller. "Collabs: A flexible and performant crdt collaboration framework." arXiv preprint arXiv:2212.02618, 2022.

[WSZ22] Yuntao Wang, Zhou Su, Ning Zhang, Rui Xing, Dongxiao Liu, Tom H Luan, and Xuemin Shen. "A survey on metaverse: Fundamentals, security, and privacy." IEEE communications surveys & tutorials, **25**(1):319–352, 2022.

[WSZ23a] Yuntao Wang, Zhou Su, Ning Zhang, Rui Xing, Dongxiao Liu, Tom H. Luan, and Xuemin Shen. "A Survey on Metaverse: Fundamentals, Security, and Privacy." IEEE Communications Surveys & Tutorials, **25**(1):319–352, 2023.

[WSZ23b] 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), pp. 3382–3398. IEEE Computer Society, 2023.

[WTC24] Kevin Warren, Tyler Tucker, Anna Crowder, Daniel Olszewski, Allison Lu, Caroline Fedele, Magdalena Pasternak, Seth Layton, Kevin Butler, Carrie Gates, et al. "" Better Be Computer or I'm Dumb": A Large-Scale Evaluation of Humans as Audio Deepfake Detectors." In Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, pp. 2696–2710, 2024.

[WYL22] Yixin Wu, Ning Yu, Zheng Li, Michael Backes, and Yang Zhang. "Membership inference attacks against text-to-image generation models." arXiv preprint arXiv:2210.00968, 2022.

[WZS24] 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 ACM CCS, 2024.

[XCJ26] Rongchao Xu, Kunlin Cai, Lin Jiang, Zhiqing Hong, Yuan Tian, and Guang Wang. "GeoGen: A Two-stage Coarse-to-Fine Framework for Fine-grained Synthetic Location-based Social Network Trajectory Generation." In AAAI Conference on Artificial Intelligence, 2026.

[XCZ14] Dan Xu, Peng Cui, Wenwu Zhu, and Shiqiang Yang. "Graph-based residence location inference for social media users." 2014.

[XGH25] Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. "Qwen3-omni technical report." arXiv preprint arXiv:2509.17765, 2025.

[XLX21] Haoran Xin, Xinjiang Lu, Tong Xu, Hao Liu, Jingjing Gu, Dejing Dou, and Hui Xiong. "Out-of-town recommendation with travel intention modeling.", 2021.

[XML23] Chonghuan Xu, Xinyao Mei, Dongsheng Liu, Kaidi Zhao, and Austin Shijun Ding. "An efficient privacy-preserving point-of-interest recommendation model based on local differential privacy." **9**(3):3277–3300, 2023.

[XR 23] XR Association. "XR Association Releases State of the Industry Report.", 2023.

[XR425] XR4Europe. "European XR Industry Report 2025." Tech. report, XR4Europe, May 2025.

[XRS23] XRSI. "The metaverse and standards.", Jul 2023.

[XRZ25] Tianxin Xie, Yan Rong, Pengfei Zhang, Wenwu Wang, and Li Liu. "Towards controllable speech synthesis in the era of large language models: A systematic survey." In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pp. 764–791, 2025.

[XW24] Zhifei Xie and Changqiao Wu. "Mini-omni2: Towards open-source gpt-4o with vision, speech and duplex capabilities." arXiv preprint arXiv:2410.11190, 2024.

[XWW25] Zihang Xiang, Tianhao Wang, and Di Wang. "Privacy audit as bits transmission:(im) possibilities for audit by one run." In 34th USENIX Security Symposium (USENIX Security 25), pp. 2693–2711, 2025.

[YAI24] Mingyu Yang, Tanvir Ahmed, Saya Inagaki, Kazuo Sakiyama, Yang Li, and Yuko Hara-Azumi. "Hardware/software cooperative design against power side-channel attacks on IoT devices." IEEE Internet of Things Journal, 2024.

[YFR20] Dingqi Yang, Benjamin Fankhauser, Paolo Rosso, and Philippe Cudre-Mauroux. "Location prediction over sparse user mobility traces using rnns." In International Joint Conference on Artificial Intelligence, 2020.

[YK13] Amir Yahyavi and Bettina Kemme. "Peer-to-peer architectures for massively multiplayer online games: A survey." ACM Computing Surveys (CSUR), **46**(1):1–51, 2013.

[YLB24] Zhuolin Yang, Cathy Yuanchen Li, Arman Bhalla, Ben Y. Zhao, and Haitao Zheng. "Inception Attacks: Immersive Hijacking in Virtual Reality Systems." In Proceedings of the 33rd USENIX Security Symposium. USENIX Association, 2024. arXiv:2403.05721.

[YLC25] Chuan Yan, Zeng Li, Kunlin Cai, Liuhuo Wan, Ruomai Ren, Yiran Shen, and Guangdong Bai. "A Multi-Store Privacy Measurement of the Virtual Reality App Ecosystem." arXiv preprint arXiv:2510.23024, 2025.

[YLZ22] Song Yang, Jiamou Liu, and Kaiqi Zhao. "GETNext: Trajectory flow map enhanced transformer for next POI recommendation." In ACM SIGIR Conference on Research and Development in Information Retrieval, 2022.

[YMP24] Tianyuan Yu, Xinyu Ma, Varun Patil, Yekta Kocaogullar, and Lixia Zhang. "Exploring the Design of Collaborative Applications via the Lens of NDN Workspace." In 2024 IEEE International Conference on Metaverse Computing, Networking, and Applications (MetaCom), pp. 89–96. IEEE, 2024.

[YMW25] Zhengkun Ye, Ahmed Tanvir Mahdad, Yan Wang, Cong Shi, Yingying Chen, and Nitesh Saxena. "BPSniff: Continuously surveilling private blood pressure information in the metaverse via unrestricted inbuilt motion sensors." In IEEE Security and Privacy, 2025.

[YSD24] Yaman Yu, Tanusree Sharma, Sauvik Das, and Yang Wang. ""Don't put all your eggs in one basket": How Cryptocurrency Users Choose and Secure Their Wallets." In Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, pp. 1–17, 2024.

[YSH23] Zhuolin Yang, Zain Sarwar, Iris Hwang, Ronik Bhaskar, Ben Y Zhao, and Haitao Zheng. "Can Virtual Reality Protect Users from Keystroke Inference Attacks?" arXiv preprint arXiv:2310.16191, 2023.

[YVM19] Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonald. "CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92)." The Rainbow Passage which the speakers read out can be found in the International Dialects of English Archive:(http://web. ku. edu/˜ idea/readings/rainbow. htm)., 2019.

[YXM22] Yan Yan, Fei Xu, Adnan Mahmood, Zhuoyue Dong, and Quan Z Sheng. "Perturb and optimize users' location privacy using geo-indistinguishability and location semantics." 2022.

[YZZ14] Dingqi Yang, Daqing Zhang, Vincent W Zheng, and Zhiyong Yu. "Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs." 2014.

[ZAB14] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, kc claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and Beichuan Zhang. "Named Data Networking." ACM SIGCOMM Computer Communication Review, **44**(3):66–73, 2014.

[ZAL23] Zheng Zhang, Hossein Amiri, Zhenke Liu, Andreas Züfle, and Liang Zhao. "Large Language Models for Spatial Trajectory Patterns Mining." 2023.

[ZCD24] Shengfang Zhai, Huanran Chen, Yinpeng Dong, Jiajun Li, Qingni Shen, Yansong Gao, Hang Su, and Yang Liu. "Membership inference on text-to-image diffusion models via conditional likelihood discrepancy." Advances in Neural Information Processing Systems, **37**:74122–74146, 2024.

[ZCL14] Jia-Dong Zhang, Chi-Yin Chow, and Yanhua Li. "Lore: Exploiting sequential influence for location recommendations." In ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2014.

[ZCL24] Jinghuai Zhang, Jianfeng Chi, Zheng Li, Kunlin Cai, Yang Zhang, and Yuan Tian. "BadMerging: Backdoor Attacks Against Model Merging." In ACM Conference on Computer and Communications Security (CCS), 2024.

[ZJC22] Yuheng Zhao, Jinjing Jiang, Yi Chen, Richen Liu, Yalong Yang, Xiangyang Xue, and Siming Chen. "Metaverse: Perspectives from graphics, interactions and visualization." Visual Informatics, **6**(1):56–67, 2022.

[ZLH25] Jingmiao Zhang, Suyuan Liu, Jiahui Hou, Zhiqiang Wang, Haikuo Yu, and Xiang-Yang Li. "SpeechGuard: Recoverable and customizable speech privacy protection." In 34th USENIX Security Symposium (USENIX Security 25), pp. 5931–5948, 2025.

[ZSC23] 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.

[ZSK18] André Zenner, Marco Speicher, Sören Klingner, Donald Degraen, Florian Daiber, and Antonio Krüger. "Immersive notification framework: Adaptive & plausible notifications in virtual reality." In Extended abstracts of CHI, 2018.

[ZSY24] Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. "Min-k%++: Improved baseline for detecting pre-training data from large language models." arXiv preprint arXiv:2404.02936, 2024.

[ZXW23] Qingxiao Zheng, Shengyang Xu, Lingqing Wang, Yiliu Tang, Rohan C Salvi, Guo Freeman, and Yun Huang. "Understanding safety risks and safety design in social VR environments." CSCW, 2023.

[ZYM23] 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 ACM CCS, 2023.

[ZZW24] Zhilin Zhang, Jun Zhao, Ge Wang, Samantha-Kaye Johnston, George Chalhoub, Tala Ross, Diyi Liu, Claudine Tinsman, Rui Zhao, Max Van Kleek, et al. "Trouble in Paradise? Understanding Mastodon Admin's Motivations, Experiences, and Challenges Running Decentralised Social Media." Proceedings of the ACM on Human-Computer Interaction, **8**(CSCW2):1–24, 2024.

[ZZY16] Shenglin Zhao, Tong Zhao, Haiqin Yang, Michael Lyu, and Irwin King. "STELLAR: Spatial-temporal latent ranking for successive point-of-interest recommendation." In AAAI Conference on Artificial Intelligence, 2016.

[ZZZ20] Guanglin Zhang, Anqi Zhang, and Ping Zhao. "Locmia: Membership inference attacks against aggregated location data." 2020.