Priv360: Application-Oriented QoE-Optimized Client-Side Protection for 360-Viewer Identification

Sheyda Mirzakhani
Linköping University
Sweden

Niklas Carlsson
Linköping University
Sweden

Abstract

Head movement data in virtual reality (VR), particularly during 360° video streaming, can reveal uniquely identifying behavioral patterns, posing serious privacy risks. While noise injection can obscure these signals, it often degrades the user's Quality of Experience (QoE), creating a challenging privacy–utility tradeoff. We introduce Priv360, a client-side defense that injects carefully tuned noise into the transmitted 6-DoF head pose while preserving the user's actual viewing experience. Before sending metadata to the server, the client reconstructs a stable predicted viewport from the noisy pose using a fast AR(2) model fused with a constant-jerk Kalman filter (with an optional LSTM-enhanced variant). Only this predicted viewport is transmitted for quality adaptation; the client continues to render the true viewport locally using the unperturbed pose. Using real 6-DoF datasets, including a newly collected Meta Quest 3 dataset, and a leave-one-video-out evaluation, we show that Priv360 substantially reduces re-identification accuracy while maintaining high visual quality across noise levels, prediction horizons, bandwidth settings, and attacker architectures. We further compare multiple prediction filters and show that combining learned and model-based predictors yields the most favorable privacy–QoE tradeoff. Our results provide a practical privacy defense and actionable insights for deploying privacy-aware VR streaming.

Keywords

Privacy, PET, Client-side protection, QoE, VR, 360 video

1 Introduction

Virtual reality (VR) systems are rapidly expanding into education, healthcare, entertainment, and telepresence. However, the same continuous motion tracking needed to deliver these immersive experiences—especially head movement signals used in 360° video streaming—carry behavioral signatures that can uniquely identify users. A growing body of work shows that even limited head (or hand) motion can support reliable re-identification [4, 22, 24, 27], and that such signals can remain linkable across sessions and contexts. As noted by Miller et al., the resulting behavioral biometrics are so distinctive that "a private browsing mode is in principle impossible" [22]. This raises an urgent question: how can VR systems preserve user privacy without sacrificing the responsiveness and visual stability required for comfortable immersion?

Unfortunately, there has been very limited work on creating defenses and providing answers to this question. Some propose perturbation or differential-privacy mechanisms but have not been evaluated under viewport-adaptive conditions. Simple noise injection can obfuscate identifying patterns but often degrades head-tracking accuracy and causes visible quality loss, making the privacy-utility tradeoff particularly challenging in 360° streaming, where accurate viewport prediction is central to Quality of Experience (QoE).

In this paper, we present **Priv360**, a client-side, privacy-preserving framework for 360° video streaming that obfuscates identifying signals while maintaining, and in many cases improving, the user's immersive experience. At the core of Priv360 is a client-side architecture that injects controlled perturbations into positional head-pose data before it is transmitted to the server, ensuring that only privacy-preserving, noise-modulated telemetry is visible to the service provider. To counteract the QoE degradation that noise would otherwise introduce, Priv360 applies a lightweight prediction pipeline on the client: a short-horizon AR(2) head-movement forecaster followed by a constant-jerk Kalman filter. This predictor operates solely on the noisy motion signal and is intentionally limited in expressiveness so that it smooths perturbations *without* reconstructing identity-bearing micro-movements. Finally, The client renders the scene using the true head orientation, while using the locally corrected (noisy) position only to guide tile selection; the server receives only perturbed, privacy-preserving tile requests and never the user's true motion.

As part of the broader contribution of Priv360, we offer the following key contributions:

- **A Privacy-Preserving Prediction Framework for 360° VR Streaming:** We present the first system that integrates privacy-preserving perturbation directly into the client-side viewport-prediction pipeline. Priv360 requires no server changes, is compatible with tile-based adaptive streaming, and applies prediction locally to preserve QoE while substantially reducing behavioral linkability from server-visible motion telemetry.
- **Effective Privacy-Utility Tradeoffs with Client-Side Filtering:** Through trace-based evaluation, we show that Priv360 significantly reduces user re-identification accuracy while preserving or improving QoE. It consistently outperforms both server-side prediction and noise-only approaches, demonstrating that *where* prediction is applied matters as much as *whether* it is applied.
- **Support for Multiple Predictors and Model-Agnostic Extensibility:** Priv360 supports a flexible prediction stack. We evaluate five forecasting methods—LSTM+KF, PF+KF, AR(2)+KF, ES+KF, and GPR+KF—and find that while AR(2) model provides the best overall performance, several alternatives provide meaningful tradeoffs too.

Proceedings on Privacy Enhancing Technologies YYYY(X), 1–18

https://doi.org/XXXXXXX.XXXXXXX

- **Empirical Insights into Realistic Operating Conditions:** Using real-world 6-DoF motion datasets, including a newly collected Meta Quest 3 dataset, we quantify how telemetry quantity, session duration, hardware generation, attacker architecture, noise strength, buffer size, bandwidth budget, and tile granularity jointly affect privacy and QoE. We further analyze viewport prediction error, tile-switching stability, mixed clean/protected attacker training, and subjective QoE, showing that Priv360 provides robust privacy protection across settings while preserving acceptable viewing quality.

Combined, our work advances the state of privacy-aware VR streaming by offering both a deployable defense—Priv360—and practical guidance for how prediction and perturbation should be combined to balance privacy and immersive quality. Beyond the Priv360 design, our findings also shed light on the broader design space of prediction under privacy constraints, emphasizing that selective, localized filtering can play a central role in improving privacy protection without undermining the user experience.

Next, we present background and related works (Section 2), the system model (Section 3), the attack scenario (Section 4), and the Priv360 defense framework (Section 5). We then describe the datasets and data collection methodology (Section 6), the privacy evaluation and attacker models (Section 7), and the QoE evaluation methodology (Section 8). Sections 9 presents the primary defense instantiations considered, including a basic request perturbation and our client-side perturbation with prediction, and Section 10 presents the evaluation results. Finally, we discuss limitations (Section 11) and conclude (Section 12).

2 Background and Related Work

Virtual reality systems rely on fine-grained motion sensing to create immersive experiences across gaming, education, and healthcare. However, these same sensors—tracking head and hand motion, gaze, and body posture also capture biometric signatures that can be exploited to re-identify users or infer sensitive traits [11, 32, 34, 38].

Attacks: Most attacks rely on rich multimodal telemetry, including head, controller, gaze, facial, and full-body motion. BehaVR [17] constructs hundreds of engineered features from these streams for high-accuracy re-identification. Earlier works such as ReAvatar [6], Miller et al. [22], and Olade et al. [27] demonstrate that full 6-DOF motion alone enables near-perfect identification. Other studies exploit non-motion signals such as network traffic [4] or mixed telemetry including handedness or gaze [23], and large-scale analyses confirm persistent identifiability across sessions [24]. Together, these works show that VR identifiability can arise from multiple signal sources, including motion, gaze, controllers, and network behavior. Priv360 focuses on a narrower but practical signal path that is central to viewport-adaptive 360° streaming: head-pose telemetry used for viewport prediction and quality adaptation. This setting is important because it captures the pose-derived information that must leave the client in many adaptive streaming pipelines, while excluding richer sensors that are not required for 360° video delivery. Because prior open-source attacks do not target this restricted input, we implement baseline models limited to head position and orientation. Their performance matches reported accuracies, providing a reliable baseline for evaluating our defenses.

Defenses: Compared to the large body of attack research, defenses for VR motion telemetry remain limited. Prior efforts propose high-level ideas such as motion jitter or randomized response but rarely provide empirical validation. MetaGuard ("Going Incognito") [26] applies differential privacy to dozens of anthropometric and system-level features, but it assumes access to rich multimodal sensors and does not operate in real-time streaming settings. Deep Motion Masking [25] learns privacy-preserving motion transformations but relies on full multimodal motion streams and introduces non-trivial distortions. No prior defense works consider viewport adaptive streaming and the impact the defenses have on the QoE.

Our work differs in both scope and threat model: we target a minimal and practical setting where the server receives only 6D head pose, and we design a lightweight client-side defense that injects controlled noise paired with a short-horizon prediction and filtering pipeline that locally corrects the perturbed signal, allowing the client to recover an accurate viewport while the server observes only privacy-preserving motion. This integration enables substantially better privacy-QoE tradeoffs than noise alone, as evaluated end-to-end under realistic viewport-adaptive streaming conditions.

Viewport-Adaptive and Tile-Based Streaming: Viewport-adaptive delivery is already standard in modern 360° video systems. Commercial systems such as the KPN/TNO/TiledMedia tiling framework [18] and academic prototypes demonstrate rapid tile switching and substantial bandwidth savings (50–65%) [8, 18, 28]. Oculus's offset-cubemap projection further confirms that real-world pipelines already adapt tile quality to the predicted viewport [40]. These systems typically use recent head orientation or viewport history to predict visible tiles, while rate adaptation may also use bandwidth, buffer occupancy, segment duration, and tile bitrate information. Priv360 is designed to operate within this standard architecture and requires no server-side modification.

Head-Pose and Viewport Prediction: Prediction is a standard technique in 360° streaming and cloud-rendered VR, used to prefetch likely viewports and mitigate motion-to-photon latency. Prior approaches span AR models and Kalman filters [10], LSTM-based and attention models [7, 20, 31, 41], deep-learning approaches for viewport prediction [9, 14], and systems for predictive prefetching and latency reduction [1, 2, 15]. Recent work demonstrates that accurate forecasts are possible even seconds ahead [7, 41].

Importantly, these prediction methods assume access to accurate, unmodified motion telemetry. In Priv360, prediction serves a different role: we intentionally operate on *noisy*, privacy-preserving head-pose signals and use a short-horizon predictor (e.g., AR(2) + Kalman filter) to locally adjust the injected noise so to achieve a more desirable privacy-QoE tradeoff.

Distinction from Prior Work: Priv360 is the first defense to address privacy in viewport-adaptive 360° VR while explicitly accounting for its impact on QoE. Unlike prior work that either (i) improves QoE assuming clean, trusted motion data or (ii) adds privacy noise without supporting prediction or adaptive streaming, Priv360 integrates controlled perturbation directly into the client-side prediction loop. This unified design enables private viewport signaling while preserving the user's true viewing experience. Our work also provides the first systematic analysis of how noise level, buffer size, and bandwidth affect the privacy–QoE tradeoff, offering practical guidance for deploying privacy-aware 360° streaming.

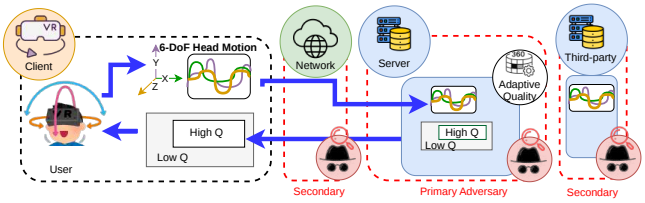


Figure 1: Use case scenario and attack model overview.

3 Use Case Scenarios and System Variations

Modern 360° video platforms increasingly use viewport-aware streaming, including tile-based adaptation and projection formats such as pyramid or offset-cube layouts [29]. These systems allocate higher quality near the predicted viewing direction and lower quality elsewhere to improve QoE under bandwidth constraints. We model an interactive 360° service in which the client sends pose-derived viewport information to a remote renderer or video server, which then adapts visual quality accordingly.

For clarity and reproducibility, our implementation uses a standard tile-based formulation, as it makes the spatial structure of quality allocation explicit and reflects many research prototypes and early commercial deployments. Importantly, Priv360 is not tied to tiling. The same client-side privacy mechanism applies to projection-based and full-frame systems as long as the server adapts quality according to a predicted viewport. Thus, Priv360 targets the underlying principle common to modern viewport-aware architectures, independent of the specific adaptation strategy.

Figure 1 provides a high-level overview of a typical system. Here, the user's pose, which consists of three positional coordinates and three orientation angles, is represented using color-coded components to indicate how the client determines the viewing direction. Furthermore, high-quality tiles within the predicted viewport are shown in white, lower-quality peripheral tiles are shown in gray, arrows illustrate what is sent in each direction, and boxes of different colors are used to separate the trusted user environment (blue) from the non-trusted network (red) and third-party server (red).

System Model Variations: For our analysis, we consider three system models of increasing client-side complexity:

- **Zero-delay Baseline:** The server immediately delivers requested tiles, modeling negligible delay and little or no buffering. This isolates the privacy effect of positional perturbation without viewport prediction.
- **Server-Side Prediction:** The client sends its current (or perturbed) head pose and buffer status (time until additional frames are needed to maintain stall-free playback) to the server, who performs head-movement prediction and selects which tiles—and at what quality—to deliver accordingly. This scenario may be most suitable for weak clients, where prediction is offloaded to the server.
- **Client-Side Prediction:** In this scenario, the client predicts its future head pose, before sending this information to the server, who delivers the most suitable tiles (or renders a personalized 360-frame) based on this information. Like in the server-side case, this model assumes a playback buffer, but prediction is done entirely to the client.

Deployment Context: These scenarios are especially relevant when pose-derived telemetry is processed separately from account identifiers, as in shared-device deployments, QoE analytics, debugging pipelines, outsourced rendering, or de-identified telemetry analysis. In such settings, the server or analytics component may not receive explicit user identifiers, but can still observe motion-derived signals sufficient for behavioral linkability.

4 Threat Model and Attacker Goals

Unfortunately, the head-pose information sent from the client to the server for viewport adaptation also contains rich behavioral structure. Prior work shows that such trajectories can act as biometric signatures that identify or link users with high accuracy, even across applications or recording sessions [4, 22, 24, 27].

Threat Scope: Motion-Based Behavioral Linkability: We study behavioral linkability from head-motion telemetry in settings where stronger identifiers such as logins, device IDs, cookies, or billing records are unavailable, shared, removed, or separated from the telemetry stream. Prior work shows that VR motion traces can act as persistent behavioral biometrics across users, sessions, and contexts [4, 17, 22, 24, 27]. In these settings, motion telemetry can distinguish individuals behind shared accounts, re-link de-identified sessions, or connect pseudonymous activity across contexts. This is complementary to, rather than a replacement for, account- or device-level identification.

Application Scenarios: This threat arises in several realistic deployments. Viewport-adaptive 360° streaming systems already use pose-derived signals to predict the user's viewport and allocate visual quality accordingly [8, 18, 28, 40]. In shared-device environments such as households, classrooms, VR arcades, museums, public installations, and enterprise training labs, multiple individuals may use the same headset, account, or organizational profile. Account-level identifiers then identify the device or organization, but not the individual user. In analytics, QoE monitoring, debugging, research, and third-party processing pipelines, explicit identifiers may be removed or separated from pose logs, while motion telemetry is retained for viewport adaptation and performance analysis.

Honest-but-Curious or Compromised Backend (Primary Adversary): Our primary adversary is an honest-but-curious backend or analytics provider, or compromised infrastructure component, that receives pose-derived telemetry during normal viewport-adaptive streaming. The adversary may have labeled historical telemetry from authenticated or attributable sessions and later use it to link protected, pseudonymous, shared-device, or de-identified sessions to prior users. This models realistic logging pipelines where telemetry collected for debugging, QoE adaptation, personalization, or analytics is repurposed to train behavioral linkability models.

Attack Goals and Harms: The attacker need not assign a real-world identity to a single trace. The core risk is persistent behavioral linkability: linking sessions from the same individual over time, distinguishing users behind shared devices, deanonymizing telemetry datasets, associating pseudonymous activity across applications, building long-term behavioral profiles, or misattributing actions and preferences in shared-device environments. These harms remain meaningful when conventional identifiers are unavailable, removed, or shared across multiple users.

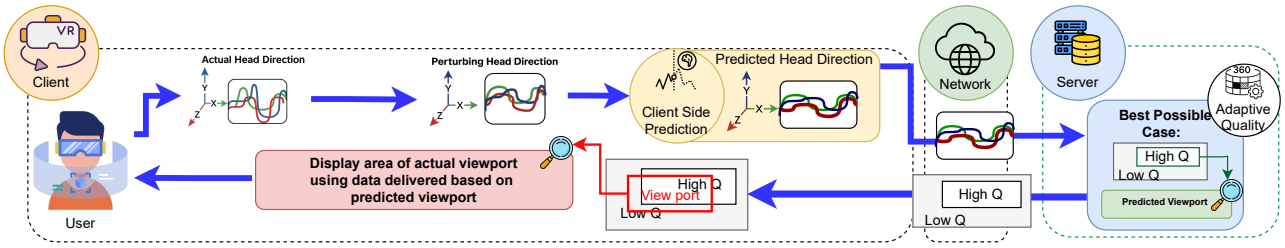


Figure 2: Simplified high-level overview of the Priv360 framework. The client locally perturbs head-pose information, predicts the upcoming viewport, and sends only the predicted viewport metadata to the server. The server adapts tile quality accordingly, while the client renders the true viewport using unperturbed local pose tracking.

Other Entities with Access to the Telemetry Stream (Secondary Adversaries): Other entities, including third-party analytics processors, outsourced rendering components, or infrastructure services, may observe subsets of the telemetry stream and represent weaker variants of the primary adversary. We therefore evaluate the backend/analytics adversary, which has the richest motion telemetry, strongest learning capability, and clearest ability to accumulate labeled traces over time.

Out-of-Scope Threats and Metadata Leakage: Priv360 targets leakage from the motion signal itself. It does not claim to prevent side channels based on packet timing, packet sizes, traffic volume, or other network metadata. Such channels are complementary to motion-signal leakage and may require network-layer defenses such as padding, batching, traffic shaping, or anonymity systems. Similarly, Priv360 does not protect against a fully compromised client device that bypasses or disables the perturbation pipeline.

Evaluation Using Primary Adversary: Our empirical evaluation therefore targets the primary backend/analytics adversary. This choice is conservative for motion-signal leakage because the backend has access to the most complete telemetry and can train adaptive models on historical data. The evaluation further considers defense-aware attackers trained on protected telemetry and mixed clean/protected training scenarios, modeling both fully informed deployments and transitional settings in which historical clean telemetry may remain available.

Behavioral Linkability: Behavioral linkability is the concrete operational threat evaluated in this paper: matching an unlabeled or pseudonymous motion trace to a previously observed user, or linking multiple sessions belonging to the same individual. Prior work shows that even short head-motion traces can support high re-identification accuracy [4, 22, 24, 27]. Priv360 therefore aims to reduce the identity-bearing structure embedded in pose-derived telemetry that leaves the client, limiting the backend's ability to construct or apply such linkability models.

5 Priv360 Defense Framework

Priv360 is designed to limit the identifying information contained in head-pose signals used for viewport adaptation without sacrificing users quality of experience (QoE). The key idea is simple: perturb what the server sees, but preserve what the user sees.

Modern viewport-adaptive systems already send pose-derived metadata (e.g., predicted viewport or tile-quality mask) to the server. Priv360 intercepts this pipeline by modifying only the data that leaves the client, ensuring that the server receives a privacy-preserving, uncertainty-tolerant version of the signal. Figure 2 shows the overall architecture, while Figure 3 walks through a single frame.

Design Principles: Our design follows three principles. First, all pose perturbation remains local to the headset: the true head pose is used for rendering and user comfort, while only perturbed, predicted viewport metadata is sent to the server. Second, the system leverages the inherent structure of viewport-adaptive streaming, which already fetches a neighborhood of high-quality content around the predicted viewport; this built-in redundancy naturally masks small positional perturbations without affecting perceived visual quality. Third, we tune privacy via controllable perturbations that are structured so that: (1) the noise meaningfully reduces identifiability and (2) prediction models (e.g., AR(2)+Kalman) compensate for noise before tile requests are sent. Together, these principles enable controllable privacy–QoE tradeoffs that can be adapted to different platforms, users, and content types.

Step-by-Step Example (Client-Prediction Case): In each rendering cycle (Figure 3), the headset obtains the user's 6-DoF pose, keeps the raw pose local to the device, and locally applies carefully tuned positional perturbations to reduce biometric identifiability. The perturbed pose is then passed through a lightweight predictor (e.g., AR(2)+Kalman), which estimates the viewport that will be relevant at the time of playback (network + decoding delay). Next, the client transmits only this predicted viewport information or equivalent server-actionable streaming metadata, such as visible-tile indices or tile-quality requests, derived from the perturbed signal (or similar info) to the server, which simply performs standard quality adaptation based on this metadata. Thus, the server does not receive the raw head-pose trace; it observes only the transmitted prediction/request sequence. Without Priv360, this sequence would be derived from the user's unprotected motion and could preserve user-specific temporal patterns useful for re-identification. Finally, at playback time, the client renders the actual viewport using the true, unperturbed pose tracked locally. Because existing viewport-adaptive systems already deliver a high-quality neighborhood around the predicted direction, these perturbations remain visually hidden, ensuring that privacy noise affects only what the server can infer, not what the user sees.

Server-Side Prediction Variation: Priv360 also supports platforms where prediction is performed on the server. In this case: (1) the client sends perturbed pose signals, (2) the server runs the prediction model and selects tiles, and (3) the client still renders using true local pose. Section 10.3 compares the privacy-QoE tradeoffs associated with the two approaches.

Summary: Priv360 provides a practical defense mechanism that: (1) perturbs only the server-visible signals, (2) keeps all real-time rendering on the device, (3) exploits existing viewport-adaptive redundancy to mask privacy noise, and (4) requires no server-side changes (though noise-aware server-side prediction can further help in server-prediction deployments). This hybrid of local perturbation, local prediction, and standard server-side adaptation enables meaningful privacy protection against behavioral linkability while maintaining high QoE and staying compatible with current adaptive streaming infrastructures [8, 18, 28, 40].

6 Datasets and Data Collection

Our evaluation uses two complementary 6-DoF VR datasets: a large public HTC Vive dataset for systematic privacy and QoE analysis, and a newly collected Meta Quest 3 dataset to validate the findings on modern standalone hardware.

6.1 Primary 6-DoF Dataset

We use the *Spherical Video Streaming* dataset [37] as our primary dataset because it is one of the few public datasets with precise 6-DoF motion tracking suitable for positional privacy analysis. It contains approximately 70 hours of HTC Vive viewing data from 48 participants (24 male, 24 female) across 18 360° videos, including world-space position (x, y, z) and unit-quaternion orientation sampled at 100 Hz. The videos span five content categories—Performance (8), Sport (5), Talkshow (2), Documentary (2), and Film (1)—and range from approximately 164 to 655 seconds, capturing diverse visual content, scene dynamics, and head-motion patterns.

This combination of high-fidelity positional tracking, large and balanced participant demographics, and shared video content across users remains uncommon in datasets but is important for evaluating biometric linkability and cross-video identifiability. Although collected in 2017, the motion signatures reflect stable human motor-control patterns rather than hardware-era artifacts, as supported by our Meta Quest 3 validation; the dataset is therefore well suited for controlled positional perturbation and privacy–QoE analysis.

We also include a small cross-dataset sanity check using a rotation–gaze dataset lacking positional data. Since position is central to our threat model, these supplemental results appear in Appendix C.

6.2 Modern Meta Quest 3 Dataset Collection

To validate that the observed privacy risks also arise on modern VR hardware, we collected a new 6-DoF dataset using a Meta Quest 3 headset, capturing natural head-motion behavior across diverse users and 360° videos under realistic viewing conditions.

Instrumentation: We developed a custom Unity-based application deployed directly on a Meta Quest 3 headset to collect synchronized 6-DoF head-motion telemetry during video playback. The application provided experimenter controls for session management, playback control, and telemetry export, enabling controlled and repeatable collection of high-frequency motion traces on a modern standalone VR platform.

Participants: We recruited 28 participants (9 female, 19 male), ages 23–39 (μ = 29.11, σ = 4.29), from a local university using email advertisements and in-person recruitment. Participants had varying levels of VR experience (3 never, 15 once or twice, 7 sometimes, and 3 frequently). All participants provided written consent prior to the study. Sessions lasted approximately 30–45 minutes, and participants received a pastry for their participation.

Video Dataset: Participants viewed 14 immersive 360° videos selected from the original set of 18 videos used in the 2017 dataset. Videos were selected based on content diversity to ensure a representative range of scene dynamics and viewing behaviors. To maintain consistent recording conditions and avoid duration-related bias, all videos were trimmed to a uniform duration of 90 seconds.

Session Design: Participants first completed a short warm-up task to familiarize themselves with the headset and VR environment. They were then instructed to watch each 360° video naturally and freely explore the scene using head movements, as during normal VR viewing. To encourage attentive viewing without constraining exploration behavior, participants were informed that they would answer a simple content-related question after each clip. Short pauses were provided between videos to reduce fatigue.

Counterbalancing: Each participant viewed all 14 videos under a Williams-design counterbalancing scheme, reducing presentation-order and first-order carryover effects. This design balances video position and predecessor/successor relationships across participants; participant count was chosen to satisfy these requirements.

6.3 Distinct Videos for Training and Testing

To prevent the classifier from overfitting to video-specific motion patterns, in addition to ensuring that the training dataset is entirely distinct from the testing dataset, we made sure that the actual set of videos differed too. Specifically, we adopt a leave-one-video-out (LOVO) evaluation across the full set of recordings. See Figure 4. In each iteration, all sessions of one video (orange) are held out exclusively for testing, while all sessions of the remaining videos (green) are used for training. This process is repeated over all videos, and the reported accuracy reflects the average across these folds.

This protocol ensures that neither the viewing conditions nor the content seen during testing appears in training; i.e., evaluation occurs on entirely unseen videos, not just unseen time segments. This is substantially more stringent than common practice in prior work, where models are often trained and evaluated on disjoint segments of the same video [4, 24].

7 Privacy Evaluation and Attacker Models

This section defines the attacker models and evaluation protocol, establishes the identifiability of clean telemetry, and then tests whether this risk persists across telemetry quantities, session durations, hardware generations, and attacker architectures.

7.1 Attacker Models

We evaluate three attacker families: Random Forest, LSTM, and Transformer classifiers trained on 6-DoF head-motion telemetry. The Random Forest operates on standardized frame-level pose features and provides a classical non-sequential baseline. The LSTM and Transformer operate on short motion sequences of length 10, using the seven pose features available in each frame: quaternion orientation (q_x, q_y, q_z, q_w) and 3D position (x, y, z). The LSTM captures recurrent temporal patterns, while the Transformer uses

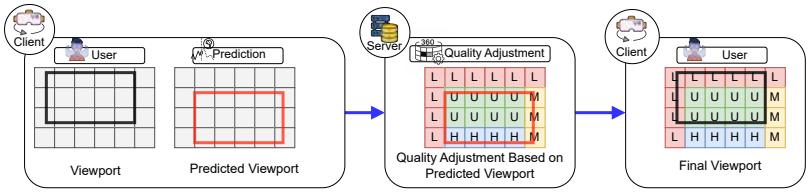


Figure 3: Conceptual illustration of client-side prediction in Priv360. The client first perturbs the current head pose and predicts the future viewport. Based on this prediction, the server delivers quality-adapted tiles, prioritizing high-quality regions around the predicted view. At playback, the client renders the true viewport using the unperturbed pose, while the delivered tiles mask the injected noise. Tile labels: U = ultra, H = high, M = medium, L = low quality.

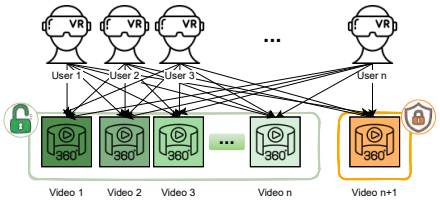


Figure 4: Training (green) and testing (orange) split. In each fold, all sessions of one video (orange) are reserved for testing, while all sessions of the remaining videos (green) form the training set. The model is retrained for every held-out video, and results are averaged across folds.

self-attention to model sequence-level dependencies. Architecture and training details are provided in Appendix B.

Adaptive Defense-Aware Attackers: Unless otherwise stated, defense evaluations use perturbation-matched training: attackers are trained on telemetry perturbed with the same parameters used at test time. This models a defense-aware adversary and avoids overstating privacy gains from train–test mismatch.

7.2 Baseline Accuracy and Privacy Metric

Baseline Accuracy: On the original, non-perturbed primary dataset, the LSTM model achieved an accuracy of 84.13% and Random Forest 81.3%. These results reflect the strong identifiability of fine-grained positional motion under our primary threat model. Baseline results for the secondary dataset—used only as a qualitative cross-dataset check—are included in Appendix C.

Positional vs. Directional Features: To isolate which parts of the head-pose signal leak the most identity information, we evaluated classifiers using different feature subsets. Position alone yields 72.1% accuracy, while orientation-only features yield substantially lower accuracy than positional features (12.4% versus 72.1%), indicating that spatial translation carries considerably stronger biometric structure in our evaluated setting. However, orientation still contains residual behavioral information that can contribute to re-identification, particularly when combined with temporal sequence models. Combining both modalities increases accuracy to 91.1%, reflecting modest complementarity but a clear dominance of positional cues. This distinction motivates our emphasis on positional perturbation, which provides the largest reduction in identifiability while preserving favorable QoE characteristics. For completeness, we also evaluate orientation perturbation and combined 6-DoF perturbation, showing that combined protection provides the strongest privacy protection under adaptive attackers.

Privacy Metric: Consistent with prior work on behavioral biometrics (e.g., head-motion and gait-based re-identification), we evaluate privacy empirically through the reduction in attacker classification accuracy. This operational metric directly captures how much identity-bearing structure remains available under our threat model. It is not a formal privacy guarantee, but provides a practical and interpretable measure of behavioral linkability for the adversarial goals considered here.

Table 1: Re-identification accuracy under varying telemetry availability and session duration. Unless otherwise stated, experiments use 48 users, 18 videos, and 60-second traces.

(a) Number of videos

Videos	Accuracy (%)
18	73.95
14	71.58
10	51.87
6	28.97

(b) Number of users

Users	Accuracy (%)
48	73.95
36	78.16
24	65.37
12	57.76

(c) Session duration

Duration (s)	Acc. (%)
30	69.27
60	71.71
90	73.59
120	74.31

7.3 Attacker Capability Based on Telemetry Quantity and Session Duration

To evaluate the realism of telemetry-based attacker training, we measure how LSTM re-identification accuracy changes with the amount and duration of available telemetry, modeling platforms that accumulate motion traces during ordinary VR use.

Table 1 summarizes attacker performance under varying numbers of videos, users, and session durations. Unless otherwise specified, experiments use 48 users, 18 videos, and 60-second traces.

First, varying the number of training videos per user shows that accuracy rises sharply as telemetry accumulates, from 28.97% with six videos to 73.95% with all 18 videos, indicating that behavioral models strengthen with routine viewing history.

Second, we vary the size of the identification pool. As expected, accuracy generally decreases with more candidate users, but remains well above random guessing, indicating that the motion signals remain discriminative at larger scales.

Third, we vary session duration. Even 30-second traces yield strong re-identification accuracy (69.27%), with longer traces providing modest gains, showing that identity-bearing motion patterns emerge quickly during normal VR viewing.

Together, these results demonstrate that telemetry collected during routine VR usage can support effective behavioral re-identification attacks, with attacker performance depending on the quantity and diversity of historical traces available for training.

7.4 Consistency Across Hardware Generations

We next test whether the privacy risks observed on the 2017 HTC Vive dataset also arise on modern VR hardware. The new Meta

Quest 3 dataset differs in headset generation, participant pool, collection period, and session duration, providing a useful cross-hardware validation of the observed behavioral signatures.

Despite these differences, the results are consistent across datasets. Under the same leave-one-video-out protocol, the LSTM attacker achieves 88.18% mean accuracy ($\sigma = 7.18$) on the Meta Quest 3 dataset, compared with 75.05% on the HTC Vive dataset under matched user, video, and duration settings. This confirms that user-specific motion patterns are present on both hardware platforms.

The new dataset also confirms that short sessions are sufficient for re-identification: despite using only 90-second clips, all evaluated attackers achieve high accuracy, showing that identity-bearing motion patterns emerge quickly during natural VR viewing.

Finally, the results show that our conclusions are stable across attacker architectures. LSTM, Random Forest, and Transformer models achieve comparable accuracies on both the Meta Quest 3 dataset (88.18%, 86.71%, and 87.42%) and the 2017 dataset (75.05%, 70.79%, and 71.70%), indicating that the observed trends are not artifacts of a single model family. Section 10.5 later shows that this robustness extends to the defense: Priv360 substantially reduces attacker accuracy on both datasets.

Overall, the Meta Quest 3 dataset confirms and extends the findings from the original 2017 HTC Vive dataset: modern 6-DoF VR telemetry can remain highly identifying, while carefully designed perturbation mechanisms can substantially reduce this privacy risk.

8 QoE Evaluation Methodology

To evaluate utility, we simulate viewport-adaptive 360° streaming using time-aligned head-pose traces. Tile qualities are selected from the server-visible pose signal, while QoE is measured over the viewport induced by the user's true local pose. This captures Priv360's key separation: perturbation affects server-side quality allocation, not the user's locally rendered viewing direction.

8.1 QoE Evaluation Simulations

To quantify how privacy-preserving noise affects QoE, we simulate playback frame by frame using time-aligned head-pose traces, preserving the user's true viewing direction while varying the server-visible pose used for tile-quality allocation.

For each frame $t \in \{1, 2, \ldots, T\}$, we maintain two synchronized pose streams: the true pose, representing the viewport the user actually sees, and the server-visible pose, which may be perturbed or predicted and is used for tile-quality allocation. The server then assigns higher quality near the predicted viewing region and lower quality elsewhere under a bandwidth constraint.

Quality Adaptation: In our simulations, each 360° frame is divided into spatial tiles, typically using a 3×2 or 6×4 layout. As per typical design, the server assigns higher quality to tiles closer to the viewing direction (based on the head pose claimed to the server) and lower quality elsewhere, subject to a per-frame bandwidth budget [36]. Such tile-based VR architectures allow a diverse set of techniques to be used for enhancing the video quality while keeping bandwidth consumption under control [39].

Here, for the tile-quality assignment, we used a greedy method, where each tile i was assigned quality levels $q_i \in \{1, 2, 3, 4\}$ greedily until the bandwidth cap was reached [13]. Specifically, the server computes $d_i(t) = \|c_i - \tilde{p}(t)\|_2$ for each tile i, where c_i is the tile center and $\tilde{p}(t)$ is the server-visible head direction, and assigns quality levels $q_i \in \{1, 2, 3, 4\}$ using a greedy water-filling allocation:

$$\sum_{i=1}^{N} w(d_i(t)) \cdot u(q_i), \tag{1}$$

where $w(d_i(t))$ is a weight given to each tile based on its relative distance from the client viewing direction and $u(q_i)$ is a (typically concave) utility function, conditioned on the tile quality assignment adhering to a per-frame bitrate constraint:

$$\sum_{i=1}^{N} b(q_i) \leq B. \tag{2}$$

Here, $b(q_i)$ is the bitrate cost for level q_i and B is the total bandwidth budget for the frame.

Viewport Extraction: Once tile qualities are assigned, lower-quality tiles are downsampled to simulate compression and visual degradation, while higher-quality tiles are preserved. Finally, for QoE measurements, each frame is cropped around the user's true viewing direction to capture the user-visible viewport.

8.2 QoE Metric Calculations

While our system uses head pose to determine the user's instantaneous viewport for tile-quality adaptation, the end-to-end delays and the actual viewport orientations are mostly unaffected by our obfuscation mechanism. Our approach therefore influences only the visual fidelity of the streamed 360° content, not the latency or stability of pose-driven rendering.

Frame quality Metrics: To evaluate the impact of the noise on the user's perceived visual quality, we first used three standard image quality metrics used in prior 360° works (e.g., [21]): Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR) [16], and Structural Similarity Index (SSIM) [35]. These metrics were calculated by comparing the the video frame obtained when the perturbed head positions were used versus when the actual head positions were used. Here, the MSE measures the average squared differences between the pixel values and provides a direct indication of distortion. In contrast, the PSNR, calculated from MSE, expresses image quality in decibels with higher values representing better quality. Finally, the SSIM assesses the similarity in structure, luminance, and contrast between the two viewports.

Perceived Visual Quality (PVQ): Objective metrics such as MSE, PSNR, and SSIM quantify pixel-level distortions but do not fully reflect how users perceive quality during 360° playback. We therefore introduce a perceptually oriented metric, the Perceived Visual Quality (PVQ), which provides a user-centric complement to these distortion measures. PVQ is computed as a visibility-weighted average of the quality levels of tiles within the viewport:

$$PVQ = \frac{\sum_{i=1}^{N} w_i \cdot q_i}{\sum_{i=1}^{N} w_i}, \tag{3}$$

where N is the number of tiles, $q_i \in \{1, 2, 3, 4\}$ is the quality of tile i (Ultra = 4, High = 3, Medium = 2, Low = 1), and w_i denotes the tile's relative visibility within the final viewport (e.g., see final viewport in Figure 3). The PVQ thus complements pixel-based metrics by reflecting the overall visual quality delivered to the user under different bandwidth and privacy conditions.

9 Defense Instantiations

We evaluate two concrete instantiations of the privacy mechanism. The first is a direct Gaussian perturbation baseline that exposes the basic privacy–QoE tradeoff of noise alone. The second is the full Priv360 design, which combines perturbation with short-horizon client-side prediction to stabilize viewport-adaptive streaming while preserving privacy.

9.1 Basic Gaussian Noise Baseline

We first consider a simple direct-perturbation baseline in which the client adds temporally smoothed Gaussian noise to the server-visible head-pose signal, but does not apply additional prediction or filtering before tile-quality selection. This baseline isolates the effect of noise alone: stronger perturbation can suppress behavioral linkability, but also makes the server's viewport estimate less accurate and degrades tile allocation.

Basic Gaussian Noise Module: The baseline adds temporally coherent Gaussian perturbations to the transmitted head-pose signal using an exponentially weighted moving average (EWMA) process. At each discrete time step t, the perturbation δ_t is

$$\delta_t = (1-\alpha)\,\delta_{t-1} + \alpha\,\varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0,\sigma^2), \tag{4}$$

where $\delta_1 = 0$ is initialized independently for each user. The smoothing factor $\alpha \in [0,1]$ controls the temporal correlation of the noise: smaller values produce smoother perturbations, while larger values introduce faster fluctuations that more strongly disrupt identifiable motion patterns. The variance σ^2 controls the perturbation magnitude, with larger values providing stronger privacy protection at greater potential QoE cost.

Overall, direct perturbation confirms the expected privacy–QoE tension: stronger and less temporally correlated noise reduces attacker accuracy, but degrades viewport quality because the server allocates high-quality tiles around increasingly noisy viewport estimates. This motivates the prediction-assisted design below, which aims to retain the privacy benefit of perturbation while restoring temporal coherence for adaptive streaming. Detailed baseline sweeps are reported in Appendix D.

9.2 Client-Side Perturbation with Prediction

Prediction plays a central role in viewport-adaptive 360° streaming: by anticipating the user's future viewing direction, the client (or server) can prefetch high-quality tiles ahead of time. However, prediction is inherently imperfect even with clean motion signals, and its accuracy typically drops sharply beyond short look-ahead windows. Privacy noise increases this uncertainty, raising concerns about whether prediction-based prefetching remains viable. Priv360 is designed for exactly this regime: it does not depend on long-term precision, but instead follows established practice by (i) using conservative quality allocation that already hedges against prediction error, and (ii) applying a lightweight short-horizon predictor with Kalman filtering that compensates for injected noise.

Importantly, the pose used for tile-quality decisions is not the viewport shown to the user. The client renders using the true head orientation and locally corrected position, while the server sees only the noise-perturbed trajectory. This separation ensures that perturbations influence only tile-quality allocation—not visual stability. The result is a controllable privacy-QoE tradeoff in which stronger noise yields higher privacy and more conservative tile selection, and lighter noise yields higher QoE, without compromising the feasibility of prediction in the underlying streaming pipeline.

9.3 AR(2)-Based Prediction with Kalman Filter

Building on the design principles outlined above, Priv360 employs a simple short-horizon predictor to stabilize the noisy positional signal before it is used for tile-quality decisions. The goal is not to recover the original head-motion trajectory, but to smooth the injected perturbations just enough to provide temporal coherence for viewport selection. To achieve this, we combine a second-order autoregressive model (AR(2)) with a constant-velocity Kalman filter. The AR(2) model predicts the current motion value using the two most recent samples, providing a simple temporal forecast of the perturbed trajectory, while the Kalman filter further smooths short-term fluctuations. Importantly, the predictor operates strictly on the perturbed trajectory and is intentionally lightweight, limiting its ability to reconstruct fine-grained identity-bearing motion patterns while still producing a stable privacy-preserving estimate suitable for conservative prediction-aware tiling.

To formalize this process, consider the head-position sequence: $\mathbf{P} = [p(1), p(2), \ldots, p(N)], \quad p(t) \in \mathbb{R}^3$. We first apply the temporally correlated perturbation introduced in Section 9.1: $\tilde{\mathbf{P}} = \mathbf{P}+\mathrm{EWMA}(\alpha,\sigma)$, and all subsequent processing is performed strictly on the perturbed signal $\tilde{\mathbf{P}}$.

AR(2) Prediction: Each spatial coordinate (x, y, z) is predicted using a ridge-regularized AR(2) model [12]:

$$\hat{p}(t) = a_1\,\tilde{p}(t-1) + a_2\,\tilde{p}(t-2),$$

where the coefficients (a_1, a_2) are updated online to track recent motion. This provides a short-horizon prediction without reintroducing identity-linked dynamics. We use the following configuration: `win_len` = 128, `update_stride` = 8, and `L2` = 1×10^{-3}.

Kalman Filtering: To further stabilize the noisy AR(2) predictions and ensure smooth viewport behavior, we pass the AR(2) output through a constant-velocity Kalman filter [3, 5]. Consistent with motion models used in prior head-tracking systems [10, 30], the filter represents the state as $\mathbf{x}_t = [p, \dot{p}]^\top \in \mathbb{R}^6$, capturing both position and velocity. The AR(2) estimate is treated as a pseudo-measurement and drives the correction step:

$$\mathbf{x}_{t|t} = \mathbf{x}_{t|t-1} + K_t\big(\hat{p}(t) - H\mathbf{x}_{t|t-1}\big),$$

from which the filtered position is obtained as $\hat{p}(t) = H\mathbf{x}_{t|t}$. This filtering stage smooths local perturbations without attempting to reconstruct high-frequency motion, yielding a stable position estimate suitable for client-side viewport selection.

Privacy-Preserving Effect: The Kalman filter operates only on the perturbed trajectory $\tilde{\mathbf{P}}$ and does not attempt to recover the original motion signal $\mathbf{P}$. As a result: (i) identity-bearing micro-movements remain obfuscated, (ii) smoothing is strictly local rather than reconstructive, and (iii) the server receives only tile requests derived from $\hat{p}(t)$. Thus, the AR(2)+Kalman filter improves QoE through stable viewport prediction while preserving the empirical privacy protection provided by the perturbation.

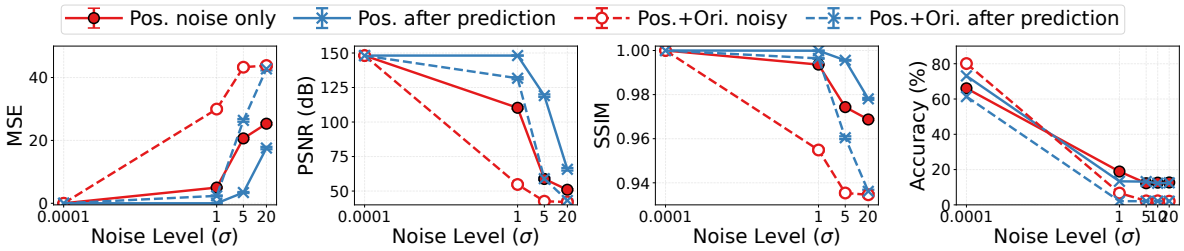


Figure 5: Test case evaluating QoE and privacy under varying positional noise levels. The red curves show metrics computed directly on the noisy positional inputs, while the blue curves show metrics after applying the client-side AR(2)+Kalman prediction. Results are shown for $\alpha = 0.5$, bandwidth = 36 kbps, and a 6×4 tile configuration, with σ varied across noise levels.

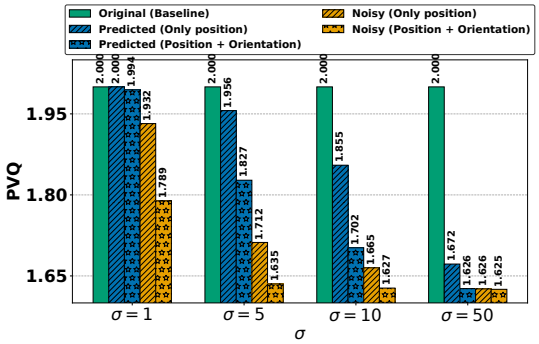


Figure 6: Average PVQ across noise levels σ for a representative user with 6×4 tiling and a 36-unit frame bandwidth. Bars compare the baseline without noise, predicted positions, and noisy observed positions.

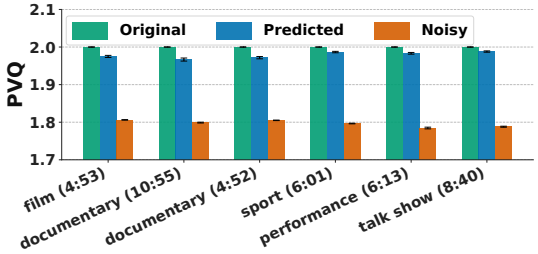


Figure 7: Weighted average PVQ per video across users for *Original*, *Predicted*, and *Noisy*, where noise is added to both position and orientation using $\alpha = 0.5$ and $\sigma = 1$. Video names are annotated with their durations in parentheses.

10 Performance Evaluation

We now evaluate the end-to-end privacy–QoE tradeoffs of Priv360, focusing on how client-side prediction changes both attacker accuracy and viewport quality under perturbation.

10.1 End-to-End Privacy–QoE Tradeoff

We begin with a representative setting using 6×4 tiles and a per-frame bandwidth budget of 36 units.

QoE-Tradeoff: First, Figure 5 compares the performance of raw noisy data (red) and denoised model predictions (blue) across the three QoE metrics (MSE, PSNR, SSIM) and the attack accuracy (%) for different noise levels σ. As noise increases, the QoE deteriorates significantly (e.g., MSE rises; PSNR and SSIM decline) while protection improves. With exception for very small noise levels, the QoE metrics (MSE, PSNR, and SSIM) consistently improve after applying our prediction model, confirming the model's strong denoising performance across all noise levels. Furthermore, we find that the identification accuracy after prediction is consistently lower than that of the raw noisy data across all noise levels σ, reflecting that our AR(2)+KF filter have reduced the value of some hidden features that the attack model otherwise could leverage for user identification.

Perceived Visual Quality (PVQ): Second, we examine the PVQ metric for representative users. Figure 6 shows that PVQ declines as the noise level σ increases, reflecting stronger privacy perturbations. However, across all noise levels, the predicted head positions consistently produce higher PVQ than the raw noisy signal, indicating that the prediction module effectively counteracts much of the QoE degradation introduced by the noise. These results confirm that our AR(2)+KF pipeline not only stabilizes viewport selection but also preserves the user's perceived visual quality under bandwidth constraints, even when substantial privacy noise is applied.

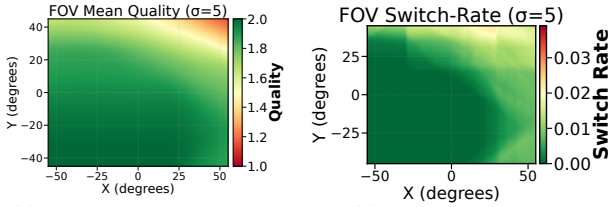
Overall, these results illustrate that while our model significantly enhances QoE, it also suppresses identity-revealing artifacts—leading to a more desirable QoE-privacy tradeoff.

Consistency Across Video Categories: To examine whether our client-side prediction consistently improves QoE across different types of content, we computed user-weighted PVQ scores for six videos spanning sports, documentary, performance, talk show, and film. Figure 7 shows the distribution of PVQ for the three conditions—*Original*, *Predicted*, and *Noisy*—over all users.

Across all videos, the *Predicted* condition yields significantly higher PVQ scores than the *Noisy* condition. This consistency is reflected in multiple paired tests (paired t-test, Wilcoxon, and permutation), all of which remain significant after Benjamini–Hochberg FDR correction. The effect sizes (Cohen's $d_z > 6$ and Cliff's $d \approx 1.0$) indicate improvements that are both statistically reliable and practically meaningful (e.g., all p-values below $1.57 \cdot 10^{-49}$).

Combined, these findings demonstrate that (i) prediction reliably mitigates the QoE loss introduced by noise, (ii) this improvement is robust across diverse video categories and user viewing behavior, and (iii) privacy protection remains strong (low attacker accuracy), as the underlying perturbations are not removed. Figure 8 provides a complementary spatial view for one representative user, illustrating how the delivered quality and tile-switching patterns vary within the viewport under the same representative operating conditions.

Sensitivity to Operating Conditions: The trends above are not specific to the representative setting. Across additional sweeps,



(a) Average quality per pixel **(b) Average switch rate**

Figure 8: Heatmap of average quality and switch rate within the viewport (FoV) for example user when using $\sigma = 5$, a 6×4 tiling layout, and a bandwidth budget of 36 units.

higher bandwidth reduces the QoE cost of perturbation, intermediate buffer sizes provide the best prediction-assisted tradeoff, and larger prediction windows offer limited additional benefit. Full sweeps over bandwidth, tile-quality distribution, prediction window size, and buffer size appear in Appendix E.

10.2 Comparison with Other Predictors

To understand the effect of different prediction strategies in our framework, we compare the AR(2)+Kalman Filter approach with four alternative predictors that we implemented and evaluated. All methods operate on the perturbed positional signal and are smoothed using Kalman filtering to stabilize motion [3].

- **Particle Filter + Kalman Filter (PF+KF):** The posterior $p(\mathbf{x}_t \mid \tilde{p}_{1:t})$ is approximated using importance-weighted particles. The mean particle state $\bar{\mathbf{x}}_t^{PF}$ is fused with the Kalman state $\hat{\mathbf{x}}_t^{KF}$ using covariance-based information fusion: $\mathbf{x}_t = (P_{PF}^{-1} + P_{KF}^{-1})^{-1}(P_{PF}^{-1}\bar{\mathbf{x}}_t^{PF} + P_{KF}^{-1}\hat{\mathbf{x}}_t^{KF})$.
- **Exponential Smoothing + Kalman Filter (ES+KF):** Estimates position using $\hat{p}(t) = \alpha p(t) + (1-\alpha)\hat{p}(t-1)$, where the smoothing factor α controls the tradeoff between responsiveness and stability [5]. The smoothed output is then filtered using the same Kalman structure.
- **Gaussian Process Regression + Kalman Filter (GPR+KF):** We fit a Gaussian process with an RBF kernel to a sliding window of past samples and use its posterior mean as the position estimate, followed by Kalman smoothing.
- **Long Short-Term Memory + Kalman Filter (LSTM+KF):** Train LSTM to model temporal dependencies in the recent pose history, $\hat{p}(t) = f(p(t-k), \ldots, p(t-1))$, and applies a Kalman filter to smooth the LSTM output and provide a stable position estimate.

All predictors yield substantially smoother and more stable trajectories once paired with Kalman filtering [10, 30]. This stability allows the client to allocate higher-quality tiles around the predicted viewport, improving visual quality under bandwidth limits.

Figure 9 compares the predictors under three representative noise levels ($\sigma = 1, 5, 50$). Across all methods and noise settings, prediction consistently improves MSE, PSNR, and SSIM relative to using the raw noisy signal, demonstrating that prediction effectively counteracts much of the QoE loss introduced by the privacy noise. At the same time, identification accuracy is further reduced after prediction, indicating that predictive smoothing removes some residual motion cues that the attacker would otherwise exploit. This is an important observation, as it shows that our prediction with filtering approach can simultaneously improve QoE and privacy.

The benefits naturally diminish when the injected noise becomes extremely large (e.g., $\sigma = 50$), where prediction has less meaningful structure to recover and QoE may degrade. This highlights the importance of choosing noise and prediction parameters jointly to balance privacy and quality.

Overall, while the AR(2)+KF model offers a strong balance between QoE improvement and privacy preservation, several of the tested predictors produce comparable tradeoffs. We therefore position Priv360 as a general and configurable privacy-QoE framework, rather than one tied to a single prediction architecture.

10.3 Client-Side vs. Server-Side Prediction

We have found that client-side prediction offers the best overall tradeoff between QoE and privacy. For example, as illustrated in Figure 10, across all noise levels (σ), client-side prediction (green triangles) consistently yields the lowest attacker accuracy—indicating stronger privacy protection—while maintaining QoE comparable to server-side prediction (blue squares). In contrast, the noise-only approach (red circles), which lacks any prediction, results in both higher attacker accuracy and degraded QoE.

This gives rise to a clear ranking of system design options, from best to worst: (1) client-side prediction, (2) server-side prediction, and (3) noise only. Although server-side prediction improves QoE relative to noise-only input, it does so by exposing the unfiltered signal to the server—offering no privacy gain. Thus, despite similar visual quality, server-side prediction entails the same re-identification risk as noise-only transmission. These findings underscore the value of performing head movement prediction on the client: it minimizes exposure of sensitive signals while preserving viewing quality, making it the most privacy-preserving and balanced solution.

Modern standalone and tethered headsets (e.g., Quest-class or PC-tethered HMDs) already execute per-frame pose fusion, reprojection, and viewport prediction at 90–120 Hz. The additional computation introduced by our AR(2)+KF predictor is negligible compared to this existing tracking pipeline, consisting only of a few lightweight floating-point matrix updates per frame. Thus, client-side prediction fits well within the capabilities of current VR hardware, whereas extremely low-end mobile-shell systems may instead rely on the server-side variant described above.

Devices that cannot run the lightweight AR(2)+KF pipeline may offload prediction to the server, but must accept this reduced privacy protection, reflecting the inherent tradeoff between computational offloading and motion-privacy protection.

10.4 Tile-Switching Stability

While the previous results characterize average viewport quality, they do not capture temporal stability effects caused by rapid tile-quality changes within the viewport. We therefore next analyze tile-switching frequency under noisy telemetry conditions.

Figure 11a compares viewport prediction error for raw noisy telemetry and AR(2)+Kalman prediction under $\sigma \in \{1, 5, 20\}$. Error increases with perturbation strength, but AR(2)+Kalman consistently stabilizes the trajectory and reduces prediction error relative to raw noisy telemetry across all noise levels.

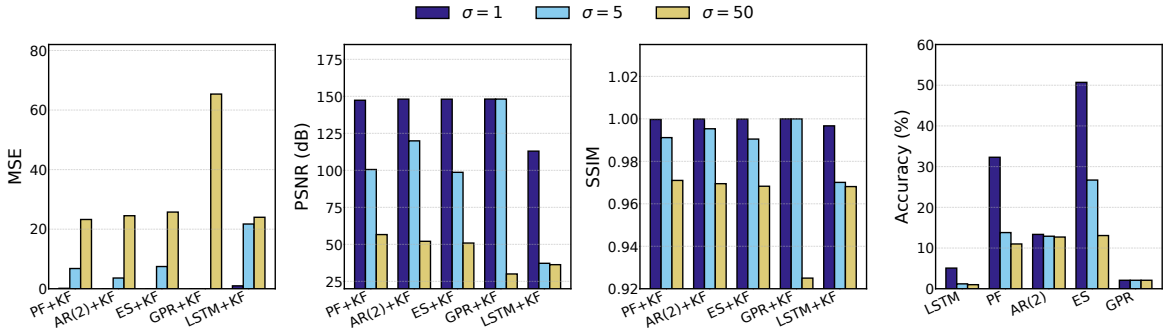
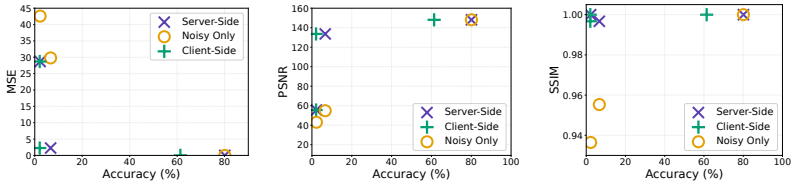
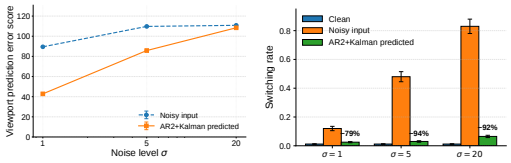


Figure 9: Comparison of the noise impact (σ) when using five different prediction methods combined with a Kalman filter, after first adding noise. Results are aggregated from four random video samples. Three noise levels are presented: low ($\sigma = 1$), medium ($\sigma = 5$), and high ($\sigma = 50$). Highlighted markers indicate the best-performing methods. Note that even the highest accuracy after prediction remains below the accuracy on the noisy input without prediction (shown in the previous figure).



(a) MSE vs. accuracy **(b) PSNR vs. accuracy** **(c) SSIM vs. accuracy**

Figure 10: Scatter-plots of the QoE vs accuracy, comparing three prediction modes: Client-side , Server-side, and Noise-only . ($\alpha = 0.5$, buffer of 11 ms. noisy observations based on noisy head-position and head-direction data



(a) Prediction error **(b) Switching rate**

Figure 11: Viewport prediction error and switching rate under different noise levels ($\sigma \in \{1, 5, 20\}$).

Our filters are designed to smooth submitted head-movement patterns and reduce frequent tile-quality changes. To evaluate the combined effects, we measure switching frequency during viewport-adaptive playback. Figure 11b compares clean, noisy, and AR(2)+Kalman-predicted telemetry. Raw noisy estimates substantially increase tile switching relative to the clean baseline, with instability growing under stronger perturbation as the system repeatedly reallocates high-quality tiles across viewport regions.

AR(2)+Kalman substantially reduces this instability at all evaluated noise levels, lowering switching frequency relative to raw noisy telemetry by approximately 79%, 94%, and 92% for $\sigma = 1$, $\sigma = 5$, and $\sigma = 20$, respectively. The tight confidence intervals indicate that these stability gains are consistent across participants.

10.5 Cross-Hardware Defense Consistency

To evaluate whether Priv360's privacy behavior generalizes across hardware generations, we compare results for the 2017 HTC Vive and Meta Quest 3 datasets. Table 2 summarizes attacker performance on clean, noisy, and prediction-assisted telemetry across all three attacker architectures.

Consistent with the baseline results in Section 7.4, clean Meta Quest 3 telemetry yields somewhat higher re-identification accuracy, confirming that modern high-fidelity 6-DoF traces remain highly distinctive. Although the smaller candidate pool partly contributes to this difference (28 vs. 48 users), Priv360 remains effective on both datasets: under noisy telemetry ($\sigma = 5$), attacker accuracy drops from 70–88% on clean traces to near-random levels across datasets and attacker architectures.

Importantly, the prediction-assisted telemetry does not substantially restore attacker performance. While AR(2)+Kalman slightly increases accuracy relative to directly perturbed traces, accuracies remain far below the clean baseline, indicating that prediction improves streaming stability without reconstructing identity-bearing trajectory structure.

The trends are also consistent across attacker architectures, with LSTM, Random Forest, and Transformer attackers showing similar reductions on both datasets. Overall, Priv360's effectiveness is not tied to a particular classifier family and appears to generalize across older and modern 6-DoF VR platforms while maintaining strong protection against behavioral re-identification.

10.6 Orientation vs. Position Perturbation

To further evaluate which motion components contribute most strongly to residual leakage under Priv360, we compare three perturbation settings: orientation-only, position-only, and combined orientation–position perturbation. Table 3 reports attacker accuracies across both datasets and attacker architectures.

Across both datasets, orientation-only perturbation remains insufficient, with attacker accuracies still above 50-58% in several cases. Position perturbation is substantially stronger, typically reducing accuracy to about 9–18%. Combining orientation and position provides the strongest protection, reducing accuracy to near-random levels on both datasets.

These results suggest that positional motion carries the dominant identifying structure in our 6-DoF VR telemetry, while orientation contributes residual behavioral information if left unprotected. The

Table 2: Re-identification accuracy on the 2017 HTC Vive dataset (48 users; random: 2.08%) and Meta Quest 3 dataset (28 users; random: 3.57%) using LSTM, Random Forest, and Transformer attackers. Results report mean ± standard deviation; noisy and predicted results use $\sigma = 5$.

Dataset	LSTM Accuracy (%)			Random Forest Accuracy (%)			Transformer Accuracy (%)		
	Clean	Noisy	Predicted	Clean	Noisy	Predicted	Clean	Noisy	Predicted
Primary Dataset (2017)	75.05 ± 8.90	2.080 ± 0.009	2.082 ± 0.009	70.79 ±8.28	2.42 ± 0.07	3.30 ± 0.30	71.70 ± 10.55	2.083 ± 0.005	2.084 ± 0.002
Validation Dataset (2026)	88.18 ± 7.18	3.63 ± 0.10	4.74 ± 1.36	86.71 ± 6.73	5.13 ± 0.27	7.30 ± 0.65	87.42 ± 7.99	3.59 ± 0.07	3.82 ± 0.44

Table 3: Impact of perturbation target on attacker accuracy (predicted). Results shown for 2017 dataset (48 users; random: 2.08%**) and the Meta Quest 3 dataset (28 users; random:** 3.57%**).**

Noise Components	Primary Dataset (2017)		Validation Dataset (2026)	
	LSTM	Transformer	LSTM	Transformer
Orientation-only	53.87%	54.56%	57.56%	50.37%
Position-only	9.48%	9.49%	18.02%	12.53%
Both (Orient. + Pos.)	2.082%	2.084%	4.74%	3.82%

Table 4: Impact of mixed clean/protected attacker training on average re-identification accuracy (%) under $\sigma = 5$ (predicted). Here, p denotes the clean-training fraction; evaluation uses protected test traces. Shown for 2017 (48 users; random: 2.08%**) and Meta Quest 3 dataset (28 users; random:** 3.57%**).**

Clean Ratio (p)	Primary Dataset (2017)		Validation Dataset (2026)	
	LSTM	Transformer	LSTM	Transformer
$p = 100\%$	3.06%	3.07%	5.71%	4.55%
$p = 75\%$	2.84%	2.72%	7.50%	6.41%
$p = 50\%$	3.26%	3.01%	7.98%	7.03%
$p = 25\%$	3.54%	3.46%	8.44%	7.55%
$p = 0\%$	2.082%	2.084%	4.74%	3.82%

consistency across the 2017 HTC Vive and Meta Quest 3 datasets further supports the conclusion that orientation-only protection is insufficient against adaptive attackers.

10.7 Impact of Training Data Mix

Thus far, we have assumed a strong adaptive attacker that knows the defense and trains on perturbation-matched telemetry using the test-time noise parameters. This removes train–test mismatch and gives the defense-aware adversary favorable conditions.

To test the impact of this assumption, we consider a heterogeneous setting where the attacker may not know which users employ Priv360. The attacker therefore trains on mixed clean/protected telemetry collected before and after protection is enabled. Let p denote the fraction of clean training sessions available to the attacker, with the remaining $(1-p)$ protected by Priv360. We evaluate $p \in \{0\%, 25\%, 50\%, 75\%, 100\%\}$, always using protected test traces.

As shown in Table 4, combined protection keeps attack accuracy low across datasets, even when large fractions of the training data are clean. Both LSTM and Transformer attackers remain near random on the 2017 dataset and only slightly above random on the Meta Quest 3 dataset, typically around 4–8%, despite partial clean-telemetry access. Importantly, the Transformer does not substantially outperform the LSTM under any training mix, indicating that Priv360's effectiveness is not tied to weaknesses of a particular classifier family.

To further examine why perturbation remains effective against stronger adaptive attackers, Appendix F analyzes the low-dimensional structure of clean and protected telemetry. The PCA visualizations show that Priv360 suppresses separable user-specific structure, rather than exploiting classifier-specific weaknesses, and that AR(2)+Kalman does not recover the original trajectory distribution.

10.8 Subjective QoE Validation

While our evaluation primarily uses objective QoE metrics (PSNR, SSIM, and PVQ), we also conducted a small subjective study to check whether these trends align with perceived viewing quality and comfort in VR. The goal is not to establish a comprehensive VR QoE benchmark, but to validate that the objective trends reflect user experience under privacy-preserving perturbation.

Using our implemented Unity-based 360° VR streaming prototype, we evaluated the three operating modes used throughout the paper: *Clean*, *Noisy*, and *Predicted*. *Clean* uses original viewport telemetry, *Noisy* uses perturbed telemetry directly for streaming adaptation, and *Predicted* applies AR(2)+Kalman prediction to perturbed telemetry before adaptation.

Six participants took part in the study. After each viewing condition, participants filled out a short subjective QoE questionnaire adapted from [33], covering overall experience (Q0), perceptual video quality (Q1), viewing acceptability (Q2), dizziness/nausea (Q3), headset comfort (Q4), and perceived effect of field-of-view limitations (Q5). All questions used a five-point Likert scale, with higher scores indicating better perceived experience. To reduce ordering effects, the three conditions were presented using a Williams-design counterbalancing strategy.

Figure 12 summarizes mean subjective QoE scores. Consistent with the objective metrics, *Noisy* receives the lowest ratings across most dimensions, reflecting degradation from perturbation alone. In contrast, *Predicted* improves perceived quality and acceptability, approaching the *Clean* baseline despite using perturbed telemetry. These gains align with the lower viewport error and tile-switching frequency in Section 10.4, suggesting that improved streaming stability directly benefits perceived QoE.

Importantly, participants did not report substantial discomfort or motion-sickness increases under *Predicted*, suggesting that prediction and filtering preserve temporal viewing stability despite perturbation. Overall, the subjective trends align with the objective QoE and tile-stability metrics reported above.

Overall, while limited in scale, the study provides qualitative support for the objective QoE metrics used throughout our privacy-preserving viewport-adaptive VR evaluation.

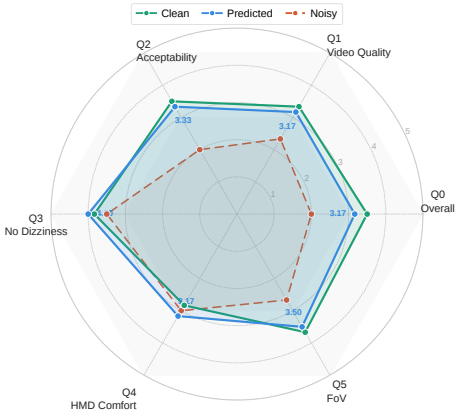


Figure 12: Mean subjective QoE scores across six perceptual dimensions (Q0–Q5) for *Clean*, *Noisy* (σ=1, α=0.5), and *Predicted*. Higher scores indicate better perceived experience.

11 Discussion

While Priv360 provides empirical protection against behavioral linkability in our evaluated 360° streaming setting, it does not provide a formal privacy guarantee, and several limitations remain. Ethical considerations are discussed in Appendix A.

Model Generalization and Personalization: Our prediction models are trained on motion data from a fixed cohort of users. Although the evaluation shows strong performance across diverse conditions, it is unclear how well these models generalize to unseen users, new content genres, or evolving user behavior over time. In particular, the predictor may benefit from personalization to accommodate user-specific motion patterns. To more easily adapt to new privacy challenges, our framework was designed to allow the use of a wide range of noise functions and prediction filters.

Dependency on Client Resources: Priv360 assumes the client can perform real-time prediction and filtering, which is reasonable for modern standalone headsets but may be challenging on low-power mobile devices. Our experiments show that the framework is not tied to any specific model class: lightweight predictors (e.g., AR models or exponential smoothing) remain effective under short horizons, while more expressive filters offer additional robustness when resources allow. Future deployments could adaptively select the predictor based on device capabilities, ensuring broad compatibility across platforms.

Latency and Real-Time Rendering Considerations: Priv360 runs entirely on the client and does not modify the headset's rendering pipeline or motion-to-photon path. Its added overhead is limited to lightweight prediction and tile-quality update logic for viewport adaptation. On a Meta Quest 3 implementation, we measured the elapsed time from head-motion update detection to tile-quality assignment in Unity. Across repeated measurements, this overhead was typically below 1 ms, well within a single VR frame budget. These measurements capture application-level adaptation overhead, not full end-to-end rendering latency, which is dominated by the VR runtime, display pipeline, and network conditions.

Fixed Noise Calibration: Our current approach uses static noise levels to perturb positional data. Although effective, this fixed strategy may either under-protect users in sensitive scenarios or unnecessarily degrade QoE in benign environments. An adaptive noise mechanism that responds to context (e.g., scene complexity, content sensitivity, or user preferences) could improve both robustness and personalization of the defense. Incorporating user-controlled privacy sliders or learning-based noise adaptation may offer additional flexibility.

Assumed Trust in the Client: The framework assumes that the client device is trustworthy and executes the prediction and noise injection faithfully. While common in privacy architectures, this assumption may fail if the client is compromised. Hardware-based attestation could help enforce integrity of the pipeline.

Privacy vs. Utility in High-Noise Regimes: Prediction effectively offsets noise at low to moderate levels, but QoE degrades at higher noise (e.g., $\sigma \geq 50$), revealing an inherent privacy–utility tradeoff. Rather than collapsing abruptly, the system degrades progressively as noise increases, reflecting the expected shift along the QoE–privacy tradeoff curve. This behavior is not evidence of perfect robustness; instead, it shows that the system remains functional under stress and avoids pathological failure modes observed in naïve perturbation schemes.

Network Metadata Leakage: Our system does not address packet-size, timing, or other network-metadata side channels. Prior work has shown such channels can reveal behavioral or identity information even when payloads are encrypted. Priv360 is complementary to such defenses: it mitigates the privacy risks arising directly from motion-signal content, while metadata protection remains an important avenue for future work.

12 Conclusion

In this work, we presented Priv360, a client-side framework designed to reduce behavioral re-identification risks in viewport-adaptive 360° video streaming. Priv360 perturbs outgoing pose signals to limit the identifying information available to the server, while locally correcting these perturbations through a lightweight prediction and filtering pipeline that preserves the user's true viewport. This design aligns with modern VR architectures and maintains visual stability by ensuring that injected noise affects only server-side quality allocation, not what the user sees.

Across evaluated real-world datasets and a range of bandwidth and buffer conditions, Priv360 consistently reduces empirical cross-session linkability under the evaluated attacker models while retaining high QoE. Client-side prediction outperforms server-side and noise-only baselines, and filtering pipelines such as AR(2) + Kalman or LSTM + Kalman provide strong privacy–utility tradeoffs even under moderate noise. Our results also show that positional perturbation is the most effective privacy lever, enabling meaningful empirical privacy gains with limited perceptual impact.

Overall, our results suggest that privacy-preserving pose obfuscation and practical viewport-adaptive streaming can be compatible goals in the evaluated setting. The framework provides concrete guidance for deploying privacy-aware 360° streaming systems, and opens future opportunities for adaptive noise schedules, personalized filters, and integration with broader privacy mechanisms.

Artifact Availability: Selected analysis/evaluation code will be released at https://github.com/sheysheyM/Priv360-PETs-artifact; datasets/prototype implementations excluded from initial release.

Acknowledgments

This work was funded by the Swedish Research Council (VR) and the Graduate School in Computer Science (CUGS) at Linköping University. We thank Eleanor Brunskog for help with the dataset collection, and acknowledge the use of ChatGPT-4o/5.1 to assist with revising the text, including improving grammar, correcting typos, and fixing awkward phrasings.

References

[1] Jayasingam Adhuran and Maria G Martini. 2024. Efficient viewport prediction and tiling schemes for 360 degree video streaming. In *Proceedings of the 15th ACM Multimedia Systems Conference*. 374–380.

[2] Mathias Almquist, Viktor Almquist, Vengatanathan Krishnamoorthi, Niklas Carlsson, and Derek Eager. 2018. The prefetch aggressiveness tradeoff in 360 video streaming. In *Proceedings of the 9th ACM Multimedia Systems Conference*.

[3] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. 2002. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on signal processing* 50, 2 (2002), 174–188.

[4] Sara Baldoni, Salim Benhamadi, Federico Chiariotti, Michele Zorzi, and Federica Battisti. 2025. Movement-and Traffic-based User Identification in Commercial Virtual Reality Applications: Threats and Opportunities. *arXiv preprint arXiv:2501.16326* (2025).

[5] Robert Goodell Brown. 1959. Statistical forecasting for inventory control. *(No Title)* (1959).

[6] Brandon Falk, Yan Meng, Yuxia Zhan, and Haojin Zhu. 2021. Poster: Reavatar: Virtual reality de-anonymization attack through correlating movement signatures. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 2405–2407.

[7] Baoqi Gao, Daoxu Sheng, Lei Zhang, Qi Qi, Bo He, Zirui Zhuang, and Jingyu Wang. 2024. Star-vp: Improving long-term viewport prediction in 360 videos via space-aligned and time-varying fusion. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 5556–5565.

[8] Mario Graf, Christian Timmerer, and Christopher Mueller. 2017. Towards bandwidth efficient adaptive streaming of omnidirectional video over http: Design, implementation, and evaluation. In *Proceedings of the 8th ACM on Multimedia Systems Conference*. 261–271.

[9] Quentin Guimard, Lucile Sassatelli, Francesco Marchetti, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. 2022. Deep variational learning for multiple trajectory prediction of 360 head movements. In *Proceedings of the 13th ACM Multimedia Systems Conference*. 12–26.

[10] Serhan Gül, Sebastian Bosse, Dimitri Podborski, Thomas Schierl, and Cornelius Hellge. 2020. Kalman filter-based head motion prediction for cloud-based mixed reality. In *Proceedings of the 28th ACM international conference on multimedia*.

[11] Ayah Hamad and Bochen Jia. 2022. How virtual reality technology has changed our lives: an overview of the current and potential applications and limitations. *International journal of environmental research and public health* 19, 18 (2022), 11278.

[12] Andrew C Harvey. 1990. Forecasting, structural time series models and the Kalman filter. (1990).

[13] Jeroen Van der Hooft, Maria Torres Vega, Stefano Petrangeli, Tim Wauters, and Filip De Turck. 2019. Tile-based adaptive streaming for virtual reality video. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 15, 4 (2019), 1–24.

[14] Xueshi Hou and Sujit Dey. 2020. Motion prediction and pre-rendering at the edge to enable ultra-low latency mobile 6DoF experiences. *IEEE Open Journal of the Communications Society* 1 (2020), 1674–1690.

[15] Gazi Karam Illahi, Ashutosh Vaishnav, Teemu Kämäräinen, Matti Siekkinen, and Mario Di Francesco. 2023. Learning to predict head pose in remotely-rendered virtual reality. In *Proceedings of the 14th Conference on ACM Multimedia Systems*.

[16] Anil K Jain. 1989. *Fundamentals of digital image processing*. Prentice-Hall, Inc.

[17] Ismat Jarin, Yu Duan, Rahmadi Trimananda, Hao Cui, Salma Elmalaki, and Athina Markopoulou. 2025. Behavr: User identification based on VR sensor data. *PoPETs* (2025).

[18] Mattis Jeppsson, Håvard N Espeland, Tomas Kupka, Ragnar Langseth, Andreas Petlund, Qiaoqiao Peng, Chuansong Xue, Dag Johansen, Konstantin Pogorelov, Håkon Stensland, et al. 2019. Efficient live and on-demand tiled HEVC 360 VR video streaming. *International Journal of Semantic Computing* 13, 03 (2019), 367–391.

[19] Yili Jin, Junhua Liu, Fangxin Wang, and Shuguang Cui. 2022. Where are you looking? A large-scale dataset of head and gaze behavior for 360-degree videos and a pilot study.. In *Proceedings of the ACM International Conference on Multimedia*. 1025–1034. https://doi.org/10.1145/3503161.3548200

[20] Dongwon Lee, Minji Choi, and Joohyun Lee. 2021. Prediction of head movement in 360-degree videos using attention model. *Sensors* 21, 11 (2021), 3678.

[21] Eric Lindskog and Niklas Carlsson. 2021. Reeft-360: Real-time emulation and evaluation framework for tile-based 360 streaming under time-varying conditions. In *Proceedings of the 12th ACM Multimedia Systems Conference*. 307–313.

[22] Mark Roman Miller, Fernanda Herrera, Hanseul Jun, James A Landay, and Jeremy N Bailenson. 2020. Personal identifiability of user tracking data during observation of 360-degree VR video. *Scientific Reports* 10, 1 (2020), 17404.

[23] Vivek Nair, Gonzalo Munilla Garrido, Dawn Song, and James F O'Brien. 2022. Exploring the privacy risks of adversarial VR game design. *arXiv preprint arXiv:2207.13176* (2022).

[24] Vivek Nair, Wenbo Guo, Justus Mattern, Rui Wang, James F O'Brien, Louis Rosenberg, and Dawn Song. 2023. Unique identification of 50,000+ virtual reality users from head & hand motion data. In *32nd USENIX Security Symposium (USENIX Security 23)*. 895–910.

[25] Vivek Nair, Wenbo Guo, James F O'Brien, Louis Rosenberg, and Dawn Song. 2023. Deep Motion Masking for Secure, Usable, and Scalable Real-Time Anonymization of Virtual Reality Motion Data. *arXiv preprint arXiv:2311.05090* (2023).

[26] Vivek C Nair, Gonzalo Munilla-Garrido, and Dawn Song. 2023. Going incognito in the metaverse: Achieving theoretically optimal privacy-usability tradeoffs in VR. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–16.

[27] Ilesanmi Olade, Charles Fleming, and Hai-Ning Liang. 2020. Biomove: Biometric user identification from human kinesiological movements for virtual reality systems. *Sensors* 20, 10 (2020), 2944.

[28] Ozgur Oyman, Mauricio Aracena, Tom De Koninck, Igor DD Curcio, Thierry Fautier, and Mick O'Doherty. 2020. VRIF Guidelines on Live VR Services. *SMPTE Motion Imaging Journal* 129, 8 (2020), 108–114.

[29] Sohee Park, Minh Hoai, Arani Bhattacharya, and Samir R Das. 2021. Adaptive streaming of 360-degree videos with reinforcement learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1839–1848.

[30] Juan Antonio Pérez-Ortiz, Felix A Gers, Douglas Eck, and Juergen Schmidhuber. 2003. Kalman filters improve LSTM network performance in problems unsolvable by traditional recurrent nets. *Neural Networks* 16, 2 (2003), 241–250.

[31] Jashanjot Singh Sidhu and Abdelhak Bentaleb. 2024. LCR360: Efficient Head Movement Prediction and Viewport Sharing in 360° Video Streaming. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. 2084–2090.

[32] Mel Slater and Maria V Sanchez-Vives. 2014. Transcending the self in immersive virtual reality. *Computer* 47, 7 (2014), 24–30.

[33] Huyen TT Tran, Nam P Ngoc, Cuong T Pham, Yong Ju Jung, and Truong Cong Thang. 2019. A subjective study on user perception aspects in virtual reality. *Applied sciences* 9, 16 (2019), 3384.

[34] Dimiter Velev and Plamena Zlateva. 2017. Virtual reality challenges in education and training. *International Journal of Learning and Teaching* 3, 1 (2017), 33–37.

[35] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. 2003. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, Vol. 2. Ieee, 1398–1402.

[36] Wenjia Wei, Jiangping Han, Yitao Xing, Kaiping Xue, Jianqing Liu, and Rui Zhuang. 2021. MP-VR: An MPTCP-based adaptive streaming framework for 360-degree virtual reality videos. In *ICC 2021-IEEE International Conference on Communications*. IEEE, 1–6.

[37] Chenglei Wu, Zhihao Tan, Zhi Wang, and Shiqiang Yang. 2017. A dataset for exploring user behaviors in VR spherical video streaming. In *Proceedings of the 8th ACM on Multimedia Systems Conference*. 193–198.

[38] Yi Wu, Cong Shi, Tianfang Zhang, Payton Walker, Jian Liu, Nitesh Saxena, and Yingying Chen. 2023. 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)*. IEEE, 3382–3398.

[39] Shou-Cheng Yen, Ching-Ling Fan, and Cheng-Hsin Hsu. 2019. Streaming 360 videos to head-mounted virtual reality using DASH over QUIC transport protocol. In *Proceedings of the 24th ACM Workshop on Packet Video*. 7–12.

[40] Chao Zhou, Zhenhua Li, and Yao Liu. 2017. A measurement study of oculus 360 degree video streaming. In *Proceedings of the 8th ACM on Multimedia Systems Conference*. 27–37.

[41] Huiyi Zhou, Feng Zhao, and Chunhai Li. 2025. Multi-scale Historical Trajectory Decomposition for Viewport Prediction in 360-degree Videos. *ACM Transactions on Multimedia Computing, Communications and Applications (TOMM)* (2025).

A Ethical Statement

This work is strictly defense-oriented. Our goal is to reduce the privacy risks of behavioral re-identification in VR, not to advance attack capabilities. All experiments use either public, consented datasets or our Meta Quest 3 data collection, for which participants provided informed consent. We evaluate only realistic, previously documented adversaries and intentionally avoid designing stronger

identification methods than necessary to assess our defense. Priv360 operates entirely on the user's device and is intended to minimize the motion telemetry exposed to external parties, contributing to safer and more privacy-aware VR streaming systems.

B Additional Attack Model Details

To evaluate the robustness of Priv360 against realistic behavioral linkability attacks, we consider multiple attacker architectures spanning both classical and modern sequence-learning models. Specifically, we evaluate Random Forest (RF), LSTM, and Transformer-based classifiers trained on 6-DoF head-motion telemetry. Together, these models represent a broad range of adversarial capabilities previously shown effective for VR motion re-identification.

- **LSTM:** We design a deep sequential model to identify users based on motion data recorded from VR headsets. With our LSTM model, the input data is segmented into overlapping sequences of ten time steps, each labeled with the user ID corresponding to its final time step. All features are standardized to zero mean and unit variance, producing input tensors shaped as `(num_sequences, 10, 7)`. The model comprises three stacked LSTM layers with 128, 64, and 32 units, each followed by a 30% dropout to mitigate overfitting. This is followed by a fully connected dense layer with 64 ReLU-activated units and 20% dropout, ending with a softmax layer that outputs user class probabilities. Training utilizes the Adam optimizer with sparse categorical cross-entropy loss, mini-batches of size 32, for up to ten epochs, with early stopping based on validation loss.
- **Random Forest:** Unlike the LSTM, the Random Forest does not process sequential inputs. First, we standardize all features. Next, we encode each user ID as an integer label. Finally, we fit a Random Forest classifier with 100 decision trees on these standardized feature vectors to predict the user ID from a single feature vector. The model was trained for 500 epochs to optimize classification performance.
- **Transformer.** The Transformer classifier takes head-motion sequences of length 10, where each frame contains seven features: quaternion rotation (q_x, q_y, q_z, q_w) and 3D position (x, y, z). Each input frame is projected to a 64-dimensional embedding and combined with a learned positional embedding to preserve temporal order. The encoder consists of two Transformer blocks, each using four-head self-attention, residual connections, layer normalization, and a feed-forward network with 128 hidden units. The encoded sequence is summarized using global average pooling, followed by dropout and dense layers, and a final softmax layer predicts the participant identity.

On More Complex Attack Models: In general, we have found that more powerful sequence models (e.g., deep RNNs or transformers) offer little advantage in our setting: the available datasets are small, and Priv360's perturbation explicitly injects stochasticity that limits exploitable structure. Since the defense reduces the mutual information between the transmitted signal and the true motion trajectory, any attacker—regardless of model capacity—is fundamentally constrained in how much identity-bearing structure can be recovered.

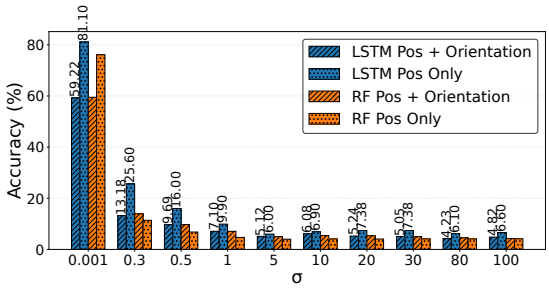


Figure 13: Impact of the noise variance (σ) on the attack accuracy of LSTM and Random Forest classifiers using our secondary dataset (Head and Gaze Behavior [19]). Here, we use $\alpha = 0.5$.

C Secondary Dataset and Cross-Dataset Sanity Check

Secondary Dataset: To assess whether the qualitative privacy–utility trends we observe in our main experiments remain consistent across sensing modalities and hardware platforms, we include complementary experiments using the*Head and Gaze Behavior* dataset [19]. This dataset includes 100 participants (43 male, 57 female) and offers 3-DOF rotational data (unit quaternions) and eye gaze vectors, recorded at 120 Hz using the VIVE Pro Eye headset.

Use as a Sanity Check: Because this dataset lacks world-space positional data, it cannot be used to evaluate positional identifiability directly. However, it provides continuous gaze-based signals that correlate with head-motion behavior. We therefore use these signals only to construct a coarse proxy for translational movement and to test whether perturbation yields qualitatively similar privacy–utility trends under a different sensing pipeline. The goal is not to re-evaluate positional privacy, but to confirm that our findings are not artifacts of a single dataset, hardware platform, or motion-capture configuration.

Summary of Results: As shown in Figure 13, adding privacy noise reduces identification accuracy from 81.1% to 6–7% for LSTM and to 4% for Random Forests, mirroring the trends observed on the primary dataset. This cross-dataset consistency supports our claim that Priv360's perturbation mechanism degrades identifiable structure even when sensing modalities differ.

D Performance with Basic Noise for Next Frame

This section examines the simplest instantiation of our approach: direct perturbation combined with next-frame prediction—to illustrate its limitations. As we show below, this baseline (outlined in Section 9.1) performs poorly under moderate noise levels, motivating the use of client-side filtering mechanisms (Section 9.2).

D.1 Privacy Protection with Basic Noise

We first evaluate how effectively the Gaussian-noise module reduces user-identification accuracy. Following the same attack setup as in Section 7, we measure the performance of the two classifiers (LSTM and RF) when noise is injected into the head-pose stream.

Impact of Noise Level: Figure 14 shows the identification accuracy of two classifiers (LSTM and Random Forest) under varying

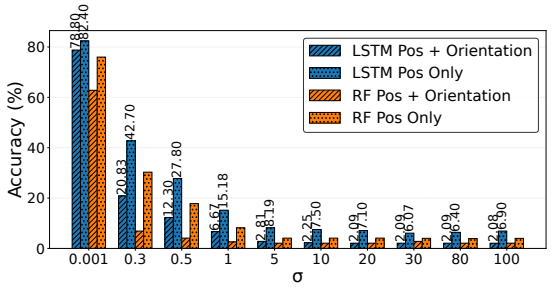
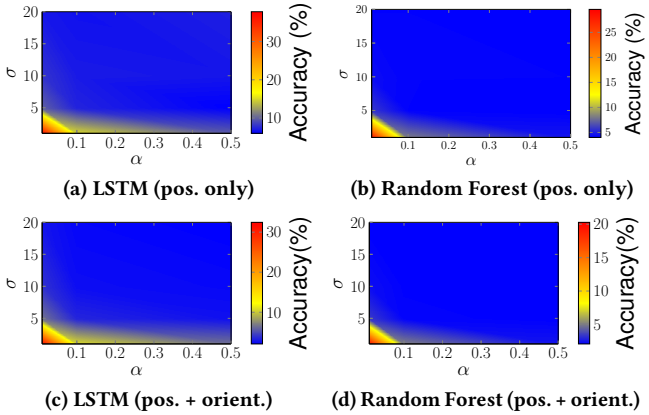


Figure 14: Impact of the noise variance (σ) on the attack accuracy of LSTM and Random Forest classifiers using our primary dataset (Spherical Video Streaming [37]) and $\alpha = 0.5$.



(a) LSTM (pos. only) **(b) Random Forest (pos. only)**

(c) LSTM (pos. + orient.) **(d) Random Forest (pos. + orient.)**

Figure 15: Impact of noise parameters (α and σ) of the defense on the attack accuracy (%) of LSTM (a, c) and Random Forest (b, d) classifiers. Two cases based on where noise is applied: positional only (a, b) and position + orientation (c, d).

noise levels (σ) and with $\alpha = 0.5$ for the primary dataset. (Results for the seconday dataset are available in Appendix C.) We include results for when noise is applied only to positional features, as well as when applied to both positional and orientation features.

We note that both models show strong identification performance at low noise levels ($\sigma = 0.001$), with LSTM achieving 82.4% and Random Forest 76.0%. As noise increases, accuracy drops sharply: LSTM falls to 15.2% at $\sigma = 1$, and both models drop below 7% by $\sigma = 30$. This highlights how even moderate noise can effectively disrupt identifying patterns in positional data.

Finally, we note that the LSTM model consistently outperforms the Random Forest, confirming our chose to select this as our primary attack model (that will be used exclusively in later sections).

Interplay Between Noise Variance (σ) and Smoothing Factor (α): Figure 15 shows how the identification accuracy depends jointly on σ and α. As α increases, privacy improves for all values of σ. Two trends emerge clearly. First, increasing σ reduces accuracy because larger perturbations obscure more discriminative motion cues. Second, for any fixed σ, higher α further improves privacy by making the noise fluctuate more rapidly, limiting the attacker's ability to exploit temporal structure. Intuitively, σ controls the amplitude of each perturbation, whereas α controls how quickly those perturbations evolve. Effective defenses therefore require balancing these two parameters to achieve strong accuracy reductions while preserving acceptable QoE.

D.2 QoE Results with Basic Noise

In our QoE experiments, each frame is divided into 6×4 tiles (24 total), the bandwidth per frame is fixed at 36 units, and the bandwidth b_i for a tile is equal to its quality index $q_i =\in \{1, 2, 3, 4\}$.

Figure 16 shows the QoE metrics for different noise levels. As observed, low noise levels (e.g., $\sigma = 0.001$) have minimal impact on visual quality, yielding very high SSIM and PSNR values. However, as the noise level increases, visual degradation becomes more significant. The MSE rises sharply, indicating increasing pixel distortion, while PSNR and SSIM values decline, confirming noticeable differences in image structure and quality.

Accuracy vs. QoE Tradeoff: Figure 17 illustrates the tradeoff introduced by different noise levels (hidden variable in the figure), ranging from minimal perturbations ($\sigma = 0.001$) to strong noise ($\sigma = 10$). As the defenses become more effective and identification accuracy drops (left side of the plots), QoE degrades accordingly: MSE increases while PSNR and SSIM decline. This confirms that stronger privacy protection comes at a clear visual-quality cost. We also observe that the LSTM consistently achieves higher identification accuracy than the Random Forest across all noise levels, indicating that it is the more challenging attack model to defend. For this reason, subsequent sections uses LSTM as attacker model and explore defense parameter settings in the high-noise regime, where privacy is strongest but QoE degradation is more pronounced.

E Additional Parameter Sensitivity Results

This appendix reports additional parameter-sensitivity results for bandwidth, tile allocation, predictor window size, and playback buffer size. These sweeps support the main evaluation in Section 10.1: client-side prediction consistently improves QoE relative to raw noisy telemetry while preserving low attacker accuracy, with the largest QoE cost appearing in bandwidth-limited regimes.

E.1 Impact of Bandwidth

The effect of our privacy mechanism depends strongly on the available bandwidth. At low bandwidth budgets, the system must make strict decisions about which tiles receive high quality. In this regime, perturbations in the head-position signal can shift the predicted viewport enough to change which tiles are prioritized, leading to noticeable differences from the baseline.

As bandwidth increases, these constraints relax. With a larger budget, the server can allocate higher-quality tiles across a broader region of the panorama, making the quality-selection process less sensitive to small viewport shifts. Consequently, the gap between the privacy-preserving system and the baseline steadily narrows.

To illustrate this trend, Figures 18 and 19 shows the average PVQ and the actual tile quality distribution as visible in the viewport across different bandwidth budgets when using the noisy head-pose data vs. the perturbation based client-side prediction. At lower budgets (e.g., 24), the difference is small as all tiles are delivered at the lowest quality. The same is true for the highest budgets (e.g., 96) as all tiles can be delivered at the highest quality. The intermediate region is more interesting. Here, we see the value of

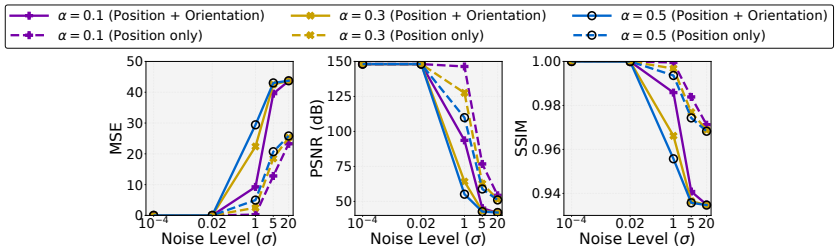


Figure 16: The impact of the noise (σ) on the primary QoE metrics (MSE, PSNR, and SSIM) for different α (0.1. 0.3, 0.5).

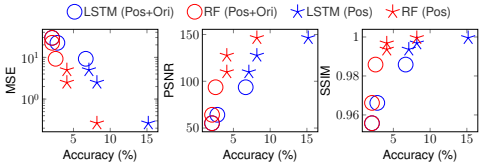


Figure 17: Comparison of accuracy vs. QoE tradeoff when using LSTM and Random Forest attack models. Tradeoff achieved using four different noise levels (σ = 1, 5, 20).

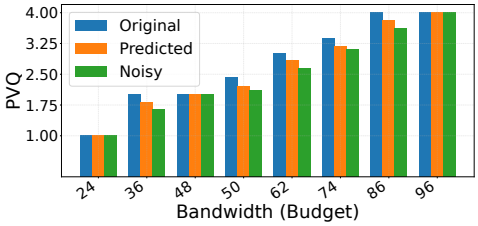


Figure 18: Impact of the available bandwidth on the PVQ scores.

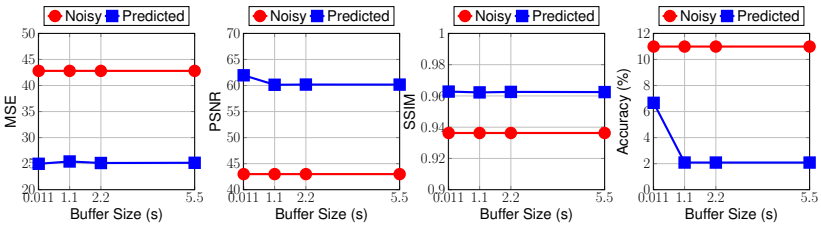


Figure 20: Impact of window size on QoE performance and accuracy with noise added to both position and orientation. (α = 0.5, σ = 1.0 and 10.0; bw = 24.)

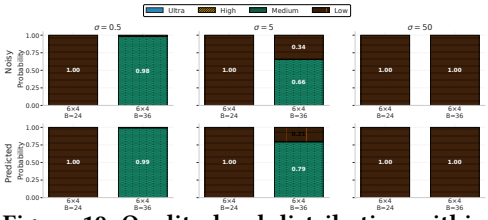


Figure 19: Quality level distribution within the FoV for different bandwidths: noise only vs. with prediction.

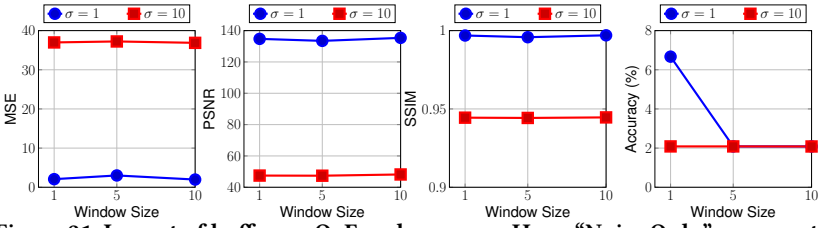


Figure 21: Impact of buffer on QoE and accuracy. Here, "Noise Only" represents data with added noise without prediction, and "After Prediction" shows the improvement after applying our prediction algorithm. (α = 0.5, σ = 5.0.)

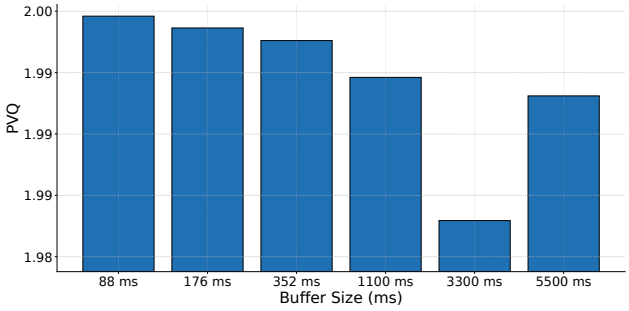


Figure 22: Impact of buffer size on PVQ when applying noise to both position and orientation (σ = 0.5) for different prediction horizons.

our predictive method, as we are able to achieve higher fraction of higher-quality tiles within the actual viewport. This shows the value of our approach.

These results show that the cost of privacy is concentrated in bandwidth-limited settings where tile-quality decisions are tight. Once the system has enough budget, QoE becomes effectively identical to the baseline, while the user's motion remains privacy-protected. In other words, at high bandwidth budgets, our defense provides privacy with essentially no QoE penalty.

E.2 Impact of Window Size

Thus far we have used a default window size of 5 frames. To better understand the impact and sensitivity of this choice, Figure 20 shows how window size affects performance under different noise levels. For low noise (σ = 1), increasing the window size steadily improves quality metrics (lower MSE, higher PSNR and SSIM), with a noticeable boost at size 10. However, accuracy remains mostly unchanged. Under high noise (σ = 10), the improvements are less consistent—some QoE metrics improve slightly, while others fluctuate or worsen. Accuracy increases only marginally. Overall, longer windows can help to some extent, but the benefit depends on the specific metric and noise level.

E.3 Buffers and Non-Real-Time Conditions

In real systems, to protect against network delays and jitter, clients may need to maintain a playback buffer and prefetch tiles several steps ahead. We next consider the impact of maintaining such buffer.

Figure 21 shows that client-side prediction (red curves) consistently improves both QoE and privacy compared to using noise alone (blue curve). As buffer size increases, QoE metrics exhibit a clear rise-then-fall trend: performance improves up to around 2 seconds (e.g., lower MSE, higher PSNR and SSIM), but degrades beyond that point. Privacy, on the other hand, remains stable: user

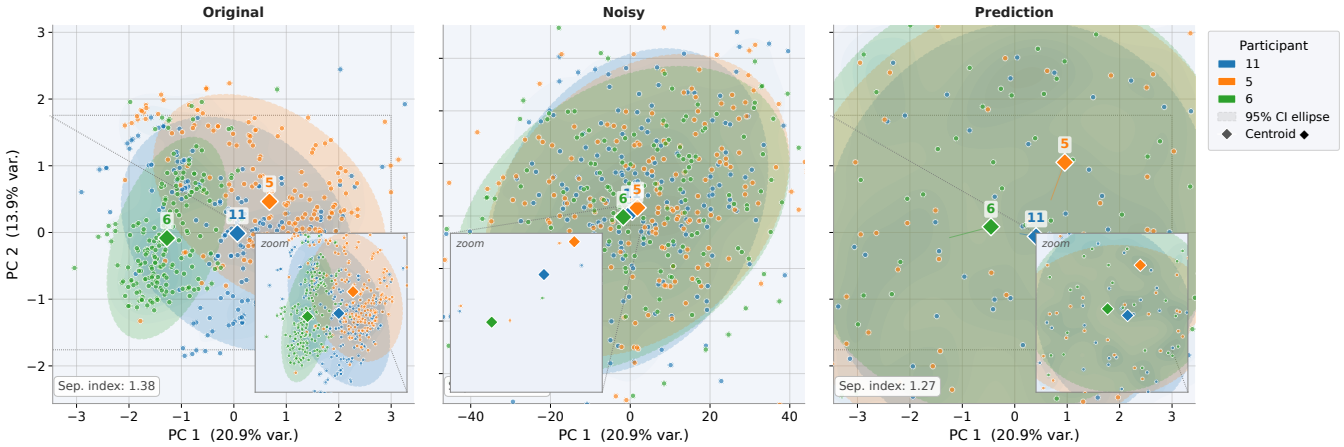


Figure 23: PCA projection of clean (Original), perturbed (Noisy), and AR(2)+Kalman-filtered (Predicted) viewport trajectories. Colors indicate users, diamond markers show user centroids, and shaded ellipses denote 95% confidence regions. Perturbation substantially increases overlap between user clusters, while the prediction pipeline smooths the trajectories without recovering the original identity-bearing structure.

identification accuracy drops from 8.1% with raw noise to about 7–7.5% with prediction, regardless of buffer size.

These results suggest that intermediate buffer sizes (around 2 seconds or 180 frames) offer the best QoE-privacy tradeoff. They enable accurate prediction while minimizing latency and avoiding issues like visual lag or substantial motion mismatch, which are critical for maintaining immersion in 360° video experiences.

F Low-Dimensional Analysis of Perturbed Trajectories

To better understand why Priv360 remains effective against stronger and adaptive attackers, we analyze the low-dimensional structure of the telemetry trajectories under clean, perturbed, and predicted conditions. For this analysis, we project the high-dimensional viewport trajectories onto the first two principal components (PC1 and PC2) using Principal Component Analysis (PCA).

Figure 23 compares three conditions: Original (clean telemetry), Noisy (Priv360-perturbed telemetry), and Predicted (AR(2)+Kalman-filtered telemetry). Each point represents a trajectory sample, colors indicate users, diamond markers show per-user centroids, and shaded ellipses denote 95% confidence regions.

The Original trajectories exhibit relatively compact and separable user-specific clusters, consistent with the strong re-identification performance observed on unprotected telemetry. In contrast, the Noisy trajectories show substantially increased overlap and dispersion in the projected space, indicating that perturbation suppresses separable behavioral structure and reduces the distinctiveness of user motion patterns.

Importantly, the Predicted trajectories do not collapse back onto the Original distribution. Although the AR(2)+Kalman prediction pipeline partially smooths the perturbed signals and restores some temporal coherence, the resulting trajectories remain substantially separated from the clean clusters. This observation suggests that the prediction stage does not reconstruct the original identity-bearing motion structure, but instead produces a stabilized privacy-preserving representation suitable for viewport adaptation.

Together, these results support the interpretation that Priv360 reduces re-identification performance by suppressing recoverable behavioral structure in the telemetry traces rather than merely introducing classifier-specific noise.