NDSS SYMPOSIUM/2026

Side-channel Inference of User Activities in AR/VR using GPU Profiling

Feb 24th, 2026

Seonghun Son[1] ,Chandrika Mukherjee[2], Reham Mohamed Aburas[3],
Berk Gulmezoglu[1], Z.Berkay Celik[2]

Iowa State University[1], Purdue University[2], American University of Sharjah[3]

Motivation

- Issue
  - AR/VR devices use significant GPU resources to render 3D objects in the scene.

Motivation

- Issue
  - AR/VR devices use significant GPU resources to render 3D objects in the scene.

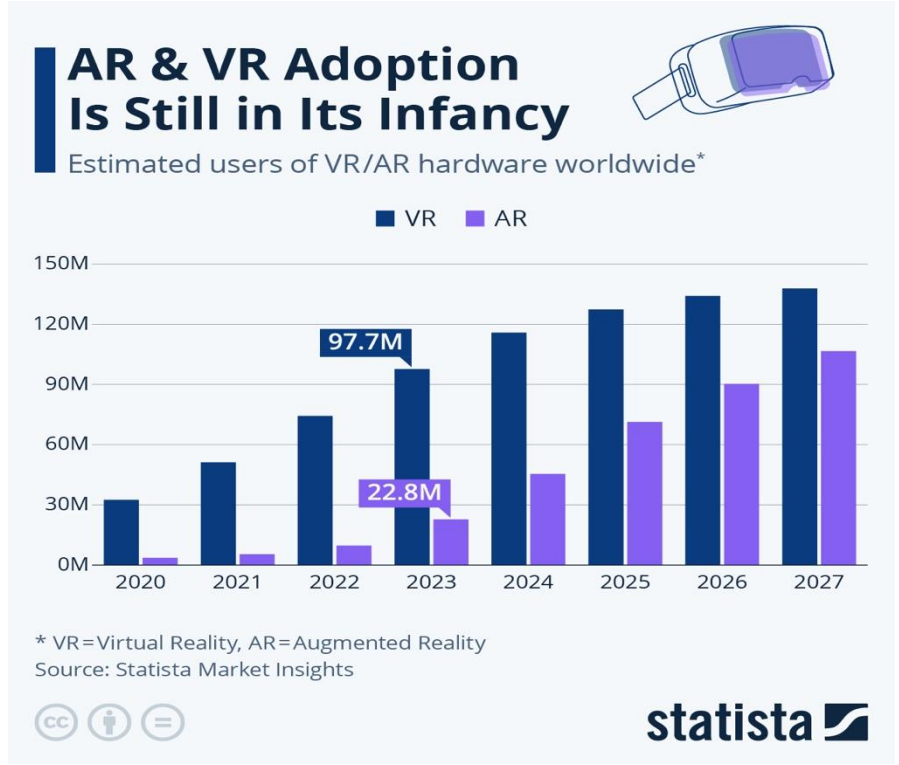
- Usage

Medical

Educations

Industry

Entertainments

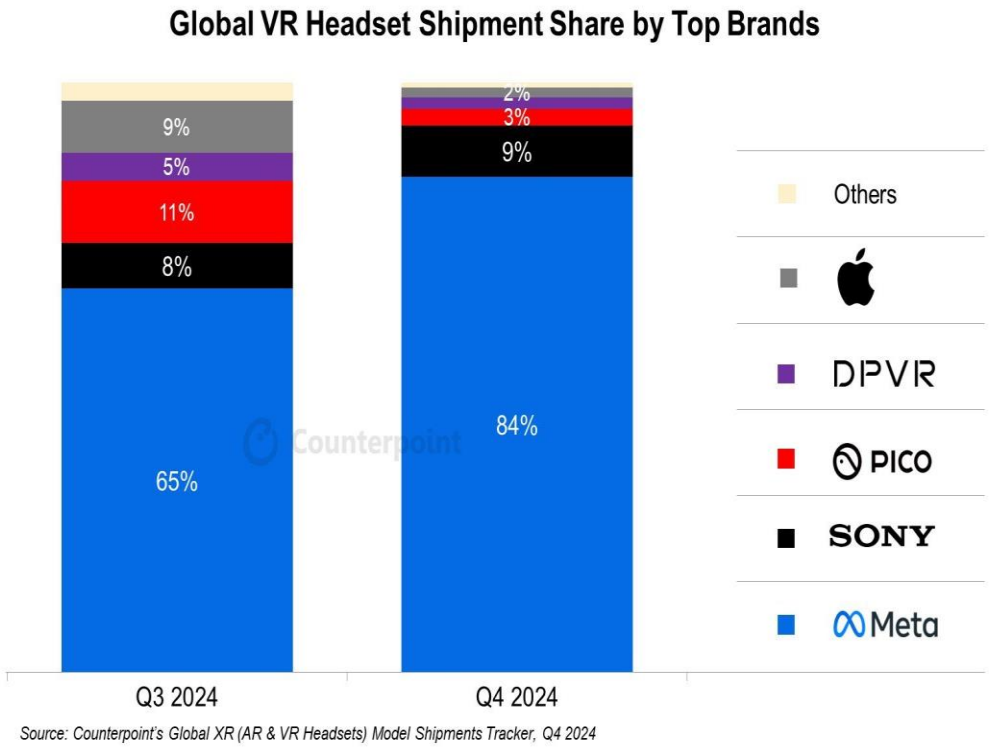


Estimated User Growth

Motivation

- Issue
  - AR/VR devices use significant GPU resources to render 3D objects in the scene.
- Usage

AR/VR Market Map

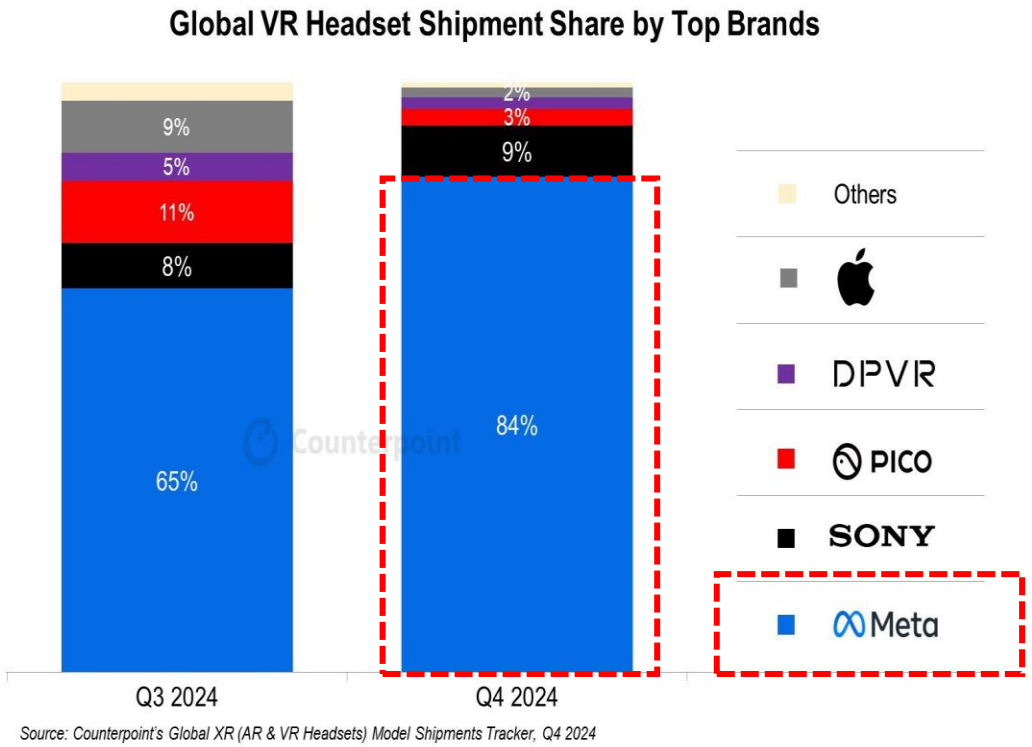


Market Share of AR/VR devices

Motivation

- Issue
  - AR/VR devices use significant GPU resources to render 3D objects in the scene.
- Usage

AR/VR Market Map



Market Share of AR/VR devices

Motivation

- Issue
  - AR/VR devices use significant GPU resources to render 3D objects in the scene.
- Usage

Meta Quest with
Qualcomm Snapdragon XR2 Gen2 chip

OVR Metrics Tool:
GPU profiler provided by 3rd party app

Motivation

- Issue
  - AR/VR devices use significant GPU resources to render 3D objects in the scene.

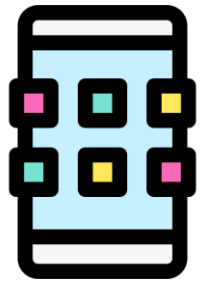
- Why does it matter?
  - By profiling GPU usage, an attacker can conduct side-channel attacks on AR/VR devices
  - The GPU profiler cannot be easily removed because it enhances the user experience when rendering 3D objects in the scene.

Motivation

- Issue
  - AR/VR devices use significant GPU resources to render 3D objects in the scene.
- Why does it matter?
  - By profiling GPU usage, an attacker can conduct side-channel attacks on AR/VR devices
  - The GPU profiler cannot be easily removed because it enhances the user experience when rendering 3D objects in the scene.
- What has been addressed
  - Common mitigations are lowering the GPU counter resolution and blocking concurrent apps running in the foreground.

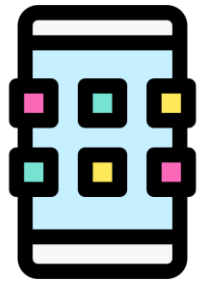
Motivation

- Issue
  - AR/VR devices use significant GPU resources to render 3D objects in the scene.
- Why does it matter?
  - By profiling GPU usage, an attacker can conduct side-channel attacks on AR/VR devices.
  - The GPU profiler cannot be easily removed because it enhances the user experience when rendering 3D objects in the scene.
- What has been addressed
  - Common mitigations are lowering the GPU counter resolution and/or blocking concurrent apps running in the foreground.



Motivation

- Issue
  - AR/VR devices use significant GPU resources to render 3D objects in the scene.
- Why does it matter?
  - By profiling GPU usage, an attacker can conduct side-channel attacks on AR/VR devices.
  - The GPU profiler cannot be easily removed because it enhances the user experience when rendering 3D objects in the scene.
- What has been addressed
  - Common mitigations are lowering the GPU counter resolution and/or blocking concurrent apps running in the foreground.



Motivation

- Issue
  - AR/VR devices use significant GPU resources to render 3D objects in the scene.
- Why does it matter?
  - By profiling GPU usage, an attacker can conduct side-channel attacks on AR/VR devices.
  - The GPU profiler cannot be easily removed because it enhances the user experience when rendering 3D objects in the scene.
- What has been addressed
  - Common mitigations are lowering the GPU counter resolution and/or blocking concurrent apps running in the foreground.

Motivation

- Issue
  - AR/VR devices use significant GPU resources to render 3D objects in the scene.
- Why does it matter?
  - By looking at GPU usage, an attacker might conduct side-channel attacks on AR/VR devices
  - The GPU profiler cannot be easily removed to enhance the user experience of rendering 3D objects in the scene.
- What has been addressed
  - Common mitigations are lowering counter resolution and blocking concurrent apps running in the foreground.

Our Objective

Reveal user activity by **built-in GPU profiler** with **1 Hz sampling rate**, running a **background** script

Related works

- Comparison with different side-channel attacks

Side-Channel Type	Side-channel primitive	Extracted Attributes (# Labels)	Resolution (Hz)	Standalone	AR/VR
Physical	Facial vibration monitoring belt	Gender (2), User(27) Body fat (-)	203	x	VR
	Power monitoring device	App (10), Website(10), Audio (5)	100	✓	VR
Montion/ Gesture Sensors	Controller	Keystrokes (38)	60	✓	VR
	IMU	Keystrokes (60)	72	x	VR
	Camera	Keystrokes (4)	30	✓	VR
	IMU (Accelerometer/Gyro)	Digits (10)	1000	x	VR
System-level APIs	Performance Counter	Gesture (5), Voice (5), Digits (10), App (12) †	60	x	AR/VR
OVRWatcher	**GPU profiler (in-built)**	**VR Object (35), App (100), Website (100), Avatar (10)**	**1**	✓	**AR/VR**

Comparison of related AR/VR side-channel attacks and OVRWatcher

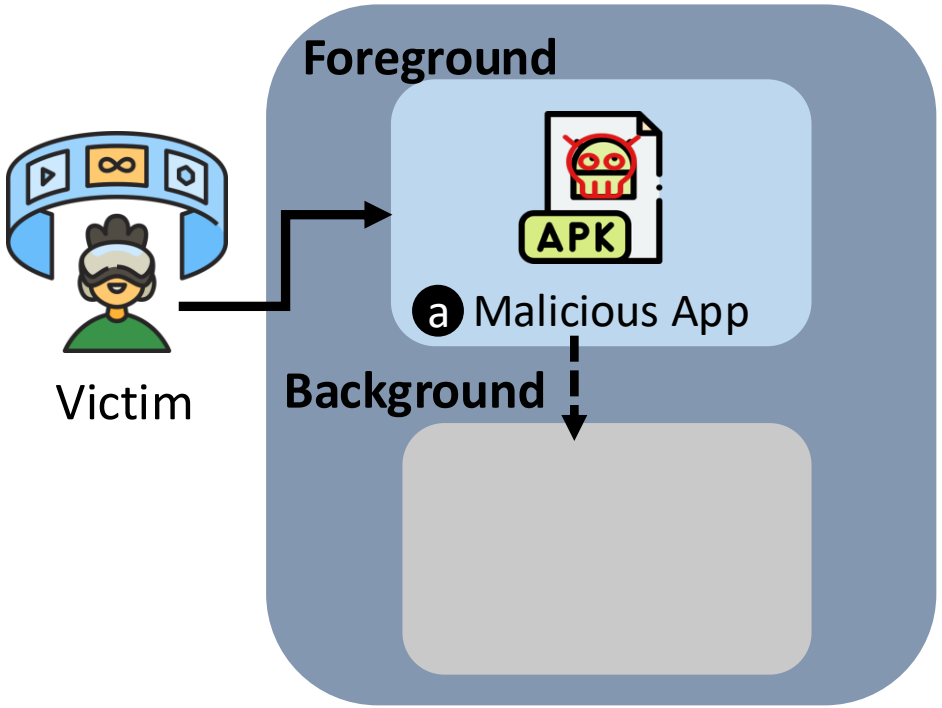
Related works

- Comparison with different side-channel attacks

Side-Channel Type	Side-channel primitive	Extracted Attributes (# Labels)	Resolution (Hz)	Standalone	AR/VR
Physical	Facial vibration monitoring belt	Gender (2), User(27) Body fat (-)	203	x	VR
	Power monitoring device	App (10), Website(10), Audio (5)	100	✓	VR
Montion/ Gesture Sensors	Controller	Keystrokes (38)	60	✓	VR
	IMU	Keystrokes (60)	72	x	VR
	Camera	Keystrokes (4)	30	✓	VR
	IMU (Accelerometer/Gyro)	Digits (10)	1000	x	VR
System-level APIs	Performance Counter	Gesture (5), Voice (5), Digits (10), App (12) †	60	x	AR/VR
OVRWatcher	**GPU profiler (in-built)**	**VR Object (35), App (100), Website (100), Avatar (10)**	**1**	✓	**AR/VR**

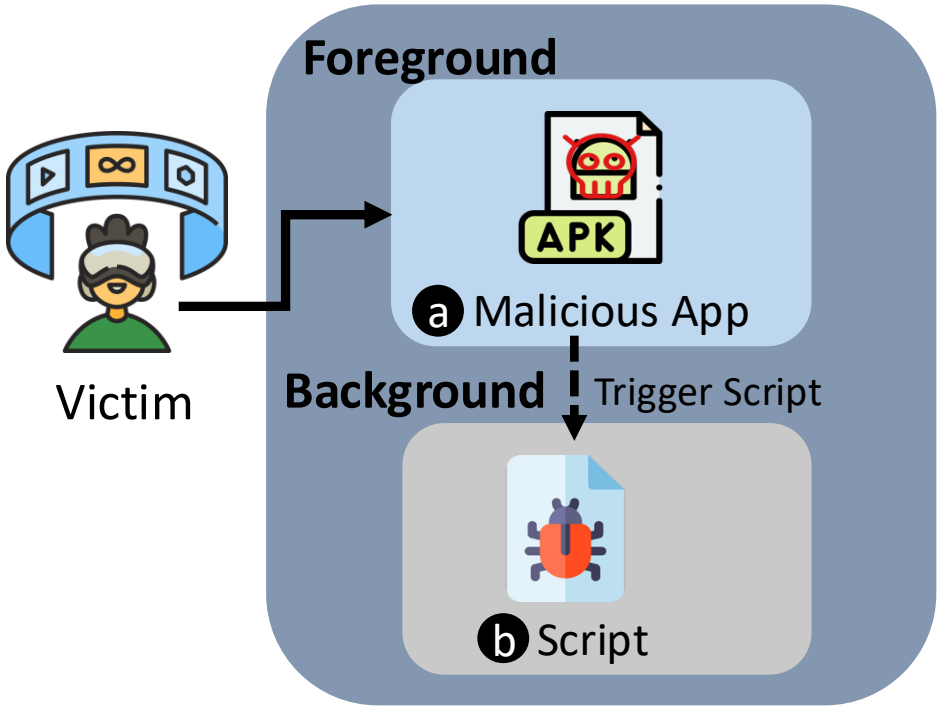
Comparison of related AR/VR side-channel attacks and OVRWatcher

Threat Model



(a) The user installs a normal-looking malicious app that can trigger a script.

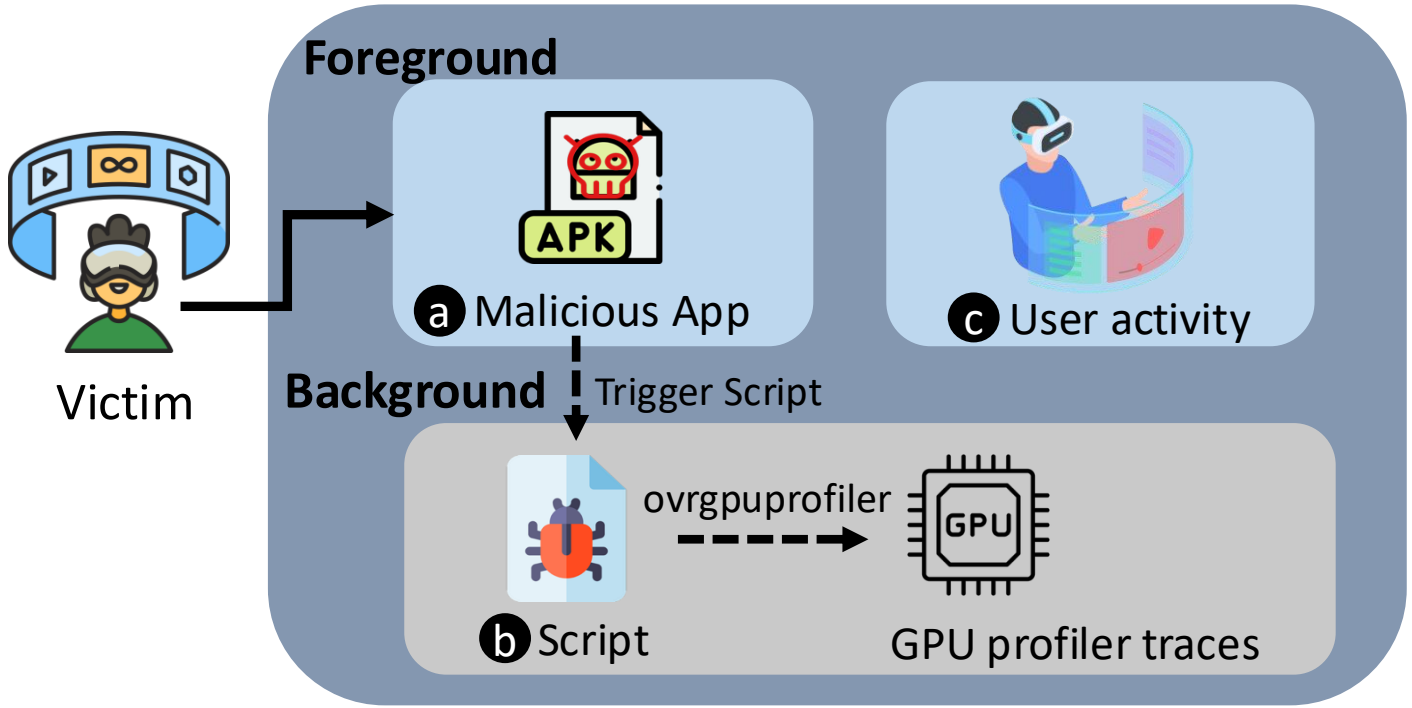
Threat Model



(a) The user installs a normal-looking malicious app that can trigger a script.

(b) Malicious app triggers a script or binary file before it terminates and runs in the background.

Threat Model

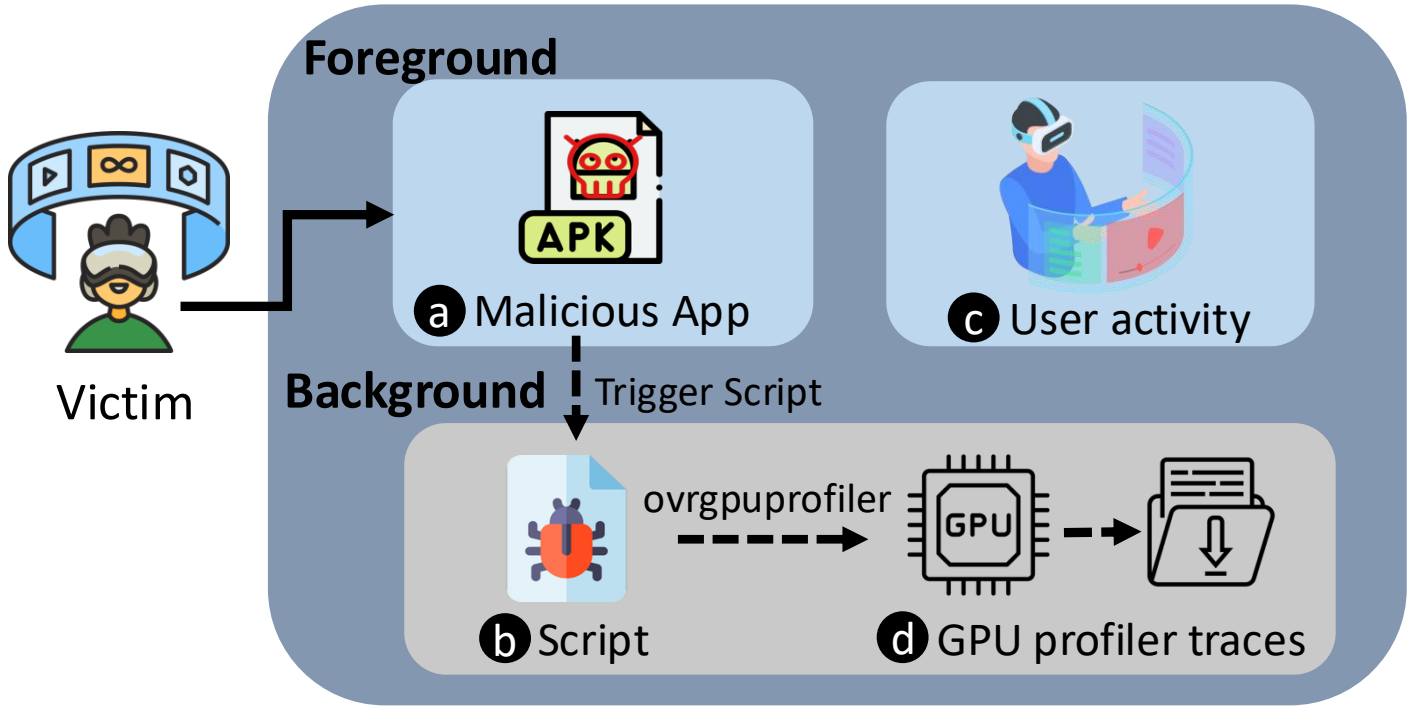


(a) The user installs a normal-looking malicious app that can trigger a script.

(b) Malicious app triggers a script or binary file before it terminates and runs in the background.

(c) Users do normal activities in the foreground, launching AR/VR apps.

Threat Model



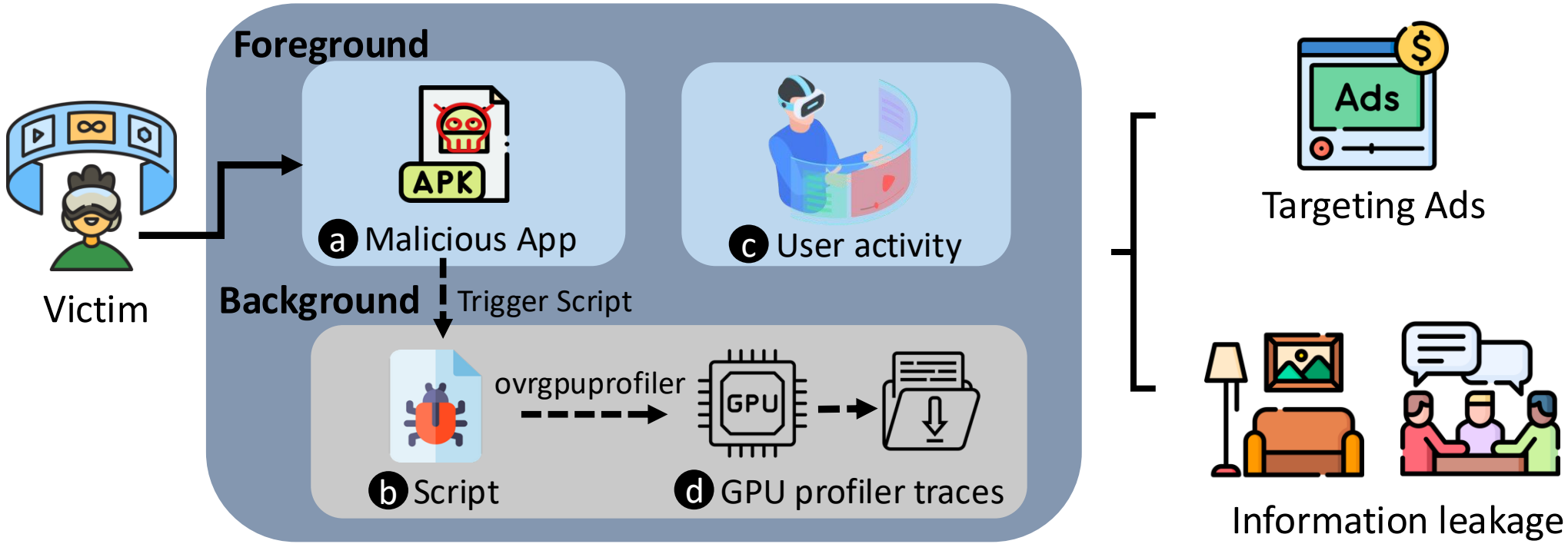
(a) The user installs a normal-looking malicious app that can trigger a script.

(b) Malicious app triggers a script or binary file before it terminates and runs in the background.

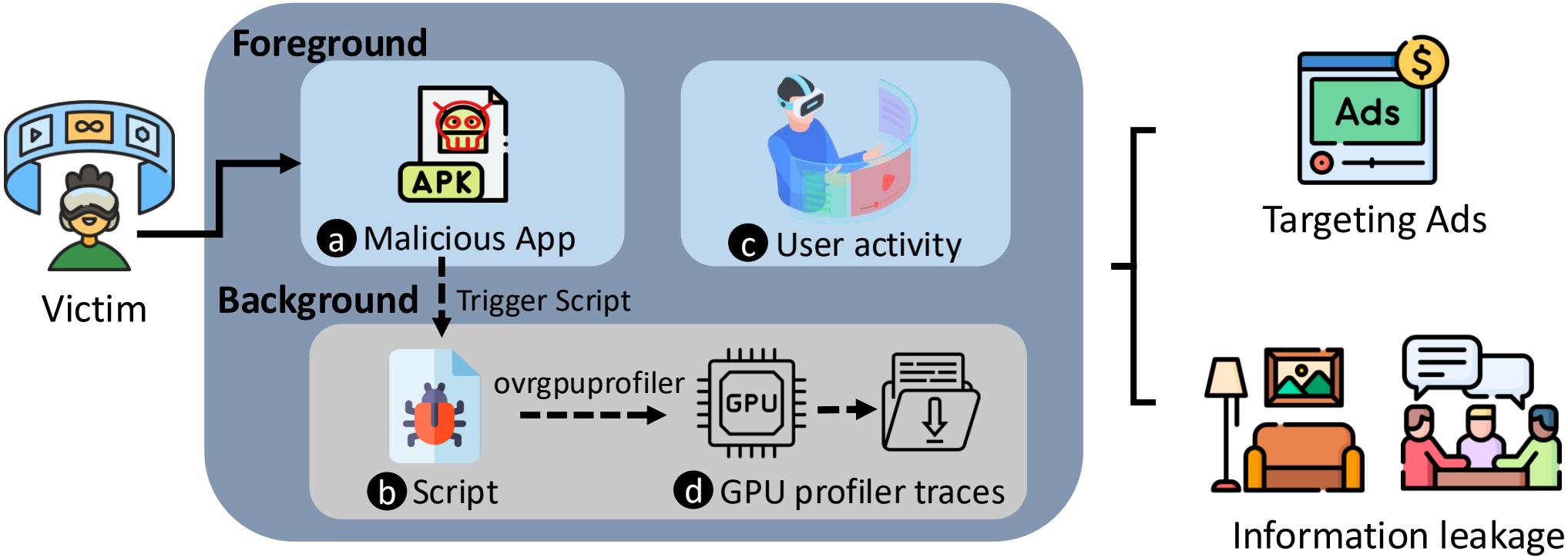
(c) Users do normal activities in the foreground, launching AR/VR apps.

(d) In-built GPU profiler profiles the GPU metrics and saves the traces in the devices.

Threat Model



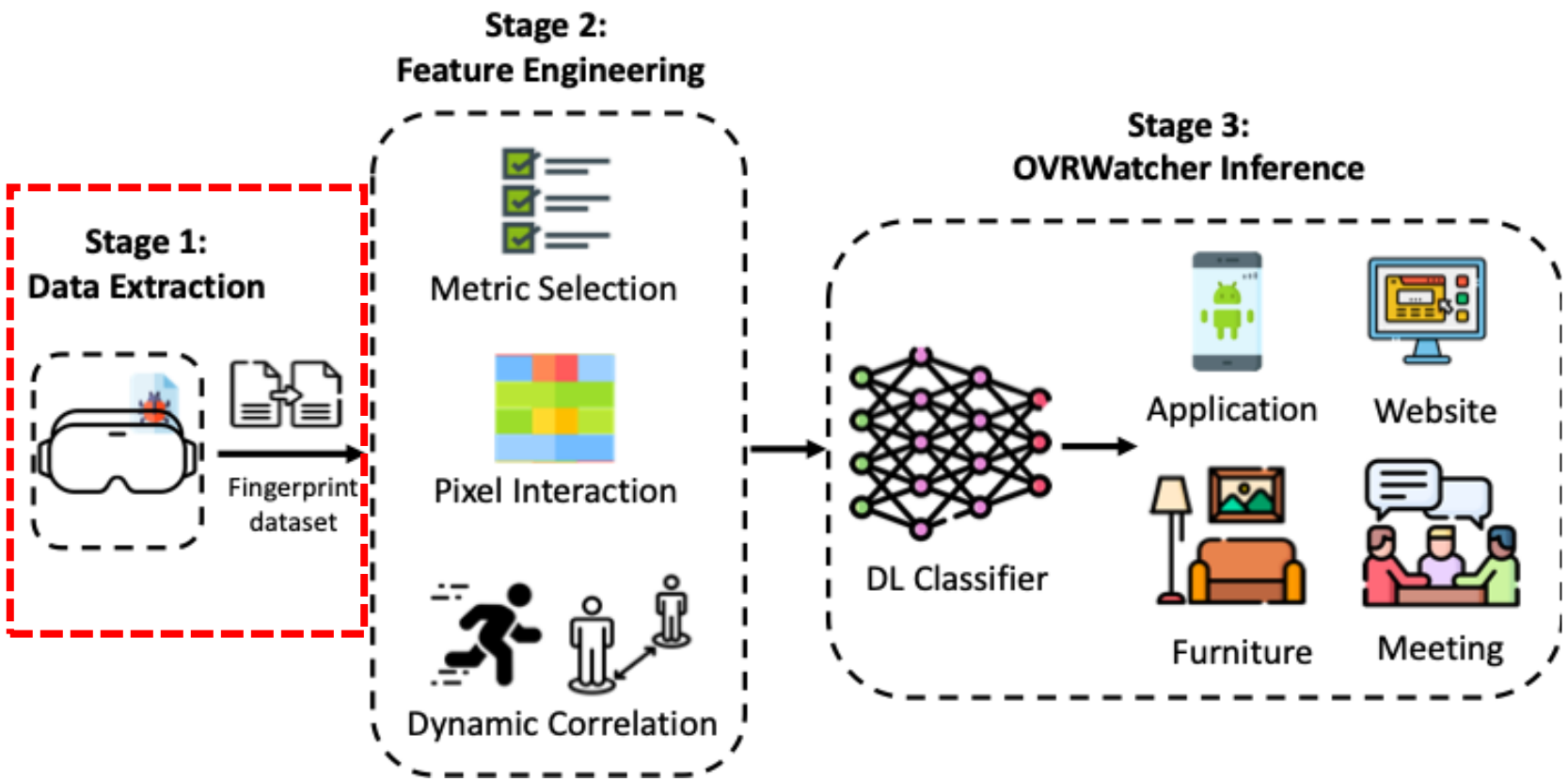
Threat Model



Constrain addressed

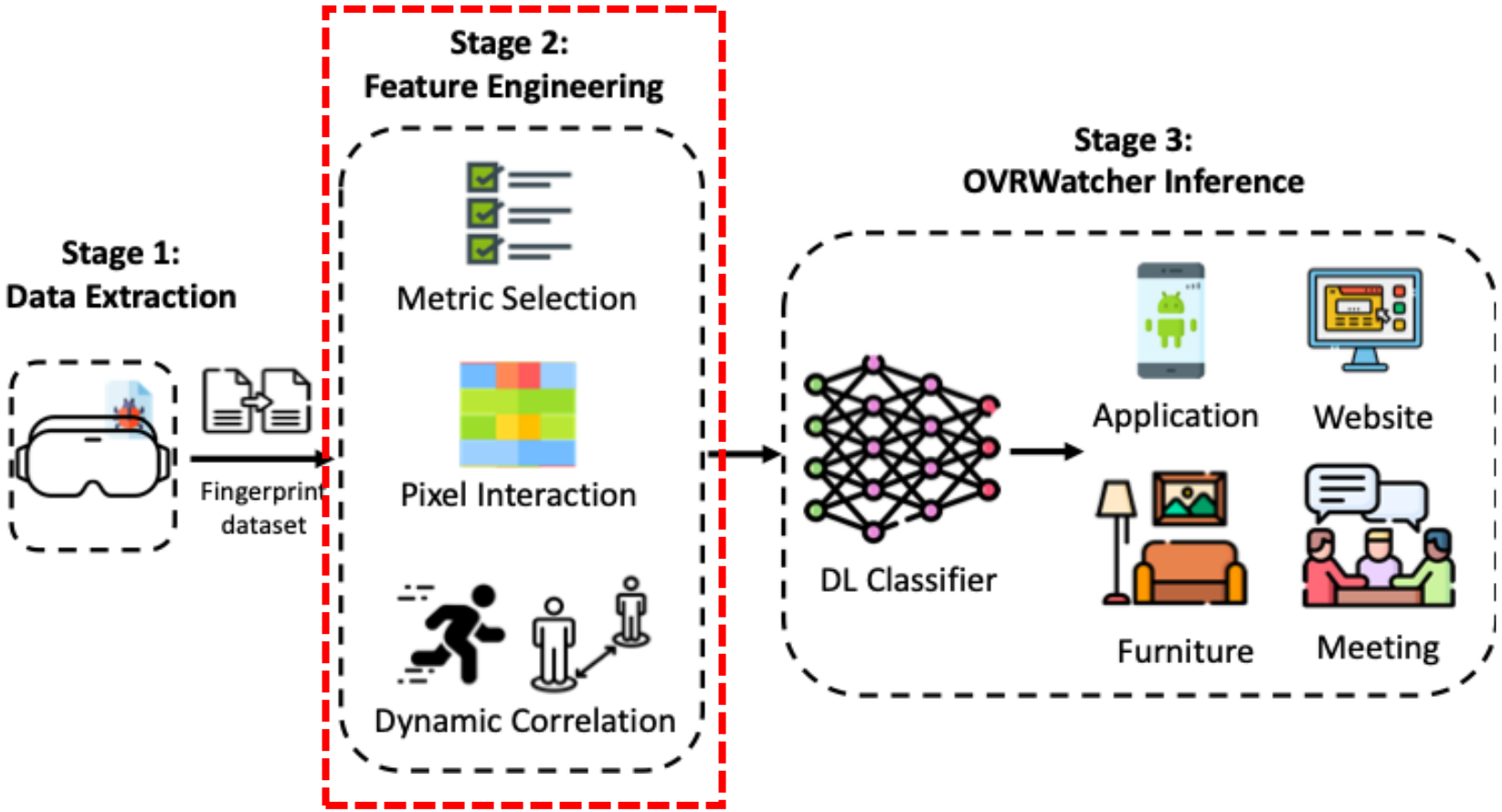
- No concurrent app execution needed
- Selecting only the most informative GPU metrics that capture real-time user interactions
- Works with 1Hz sampling rate (harder to detect & block)

OVRWatcher Pipeline



Stage 1: Collect enough time to create a user activity fingerprint dataset and extract that dataset

OVRWatcher Pipeline

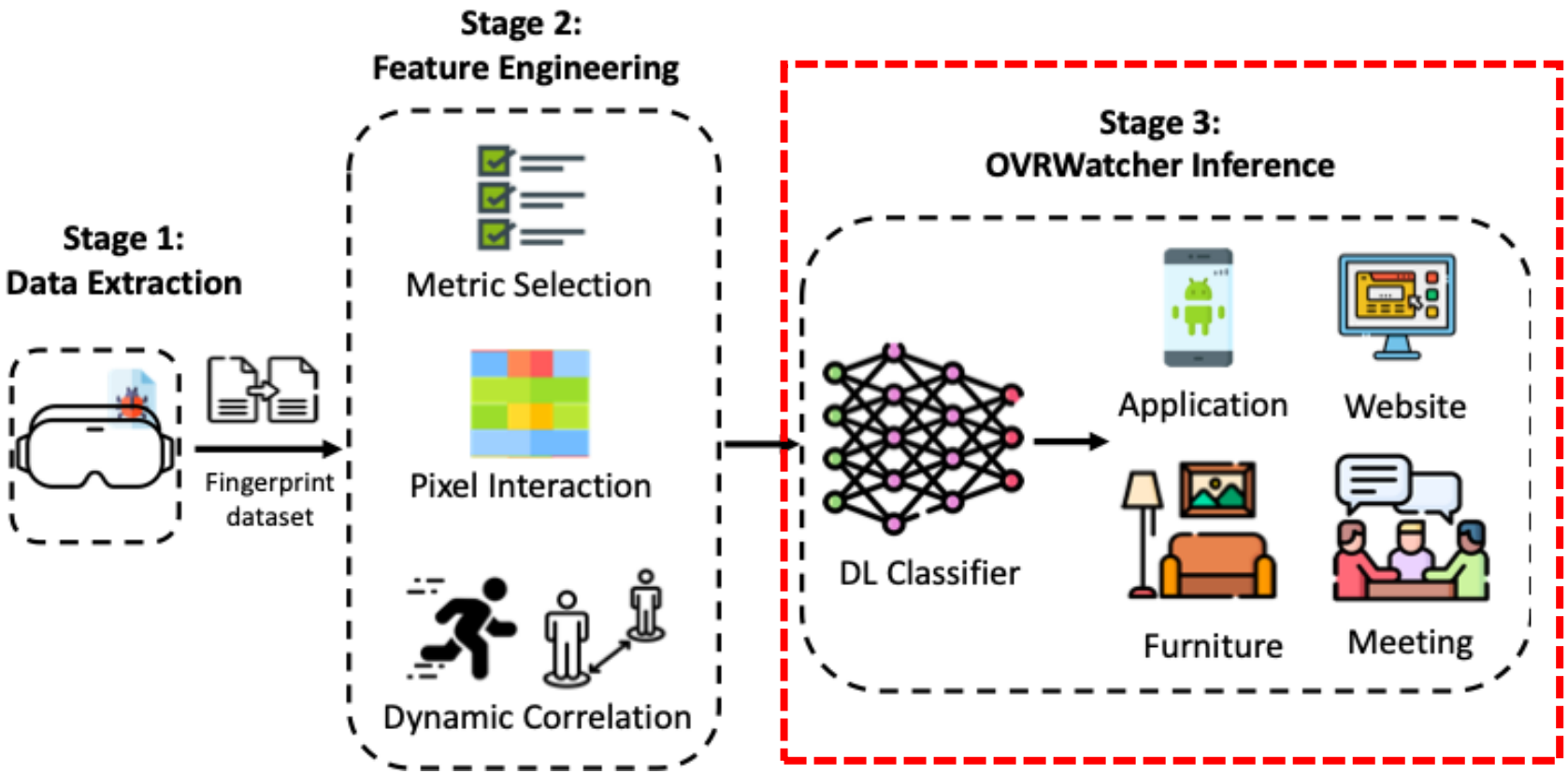


Stage 1: Collect enough time to create a user activity fingerprint dataset and extract that dataset

Stage 2: Reverse engineer the collected fingerprints

- Select the most informative metrics to capture real-time interactions
- Map metric correlation with pixels and movements.

OVRWatcher Pipeline



Stage 1: Collect enough time to create a user activity fingerprint dataset and extract that dataset

Stage 2: Reverse engineer the collected fingerprints

- Select the most informative metrics to capture real-time interactions
- Map metric correlation with pixels and movements.

Stage 3: Apply DL models (CNN, LSTM, Random Forest, and SVM)

- Classify user-rendered apps or websites
- Extract user user-rendered 3D object within the app
- Infer the number of participants in the virtual meeting.

Feature Extraction

- Metric selection

Category	Metric
GPU Utilization	GPU Frequency, GPU Bus Busy, Preemptions / second, Avg Preemption Delay
Stalls	Vertex Fetch Stall, Texture Fetch Stall, Texture L2 Miss, Stalled on System Memory
Memory Access	Vertex Memory Read (Bytes/Second), SP Memory Read (Bytes/Second), Global Memory Load Instructions, Global Buffer Data Read Request BW (Bytes/sec), Global Buffer Data Read BW (Bytes/sec), Global Image Uncompressed Data Read BW (Bytes/sec), Bytes Data Write Requested, Bytes Data Actually Written
Shader/ Instruction	Vertex Instructions / Second, Local Memory Store Instructions, Avg Load-Store Instructions Per Cycle, Avg Bytes / Fragment, L1 Texture ache Miss Per Pixel
Geometry/ Rasterization	Pre-clipped Polygons/Second, Prims Trivially Rejected, Prims Clipped, Average Vertices/Polygon, Average Polygon Area
Texture/Filtering	Nearest Filtered, Anisotropic Filtered, Non-Base Level Textures

30 GPU metrics that impact 3D object rendering

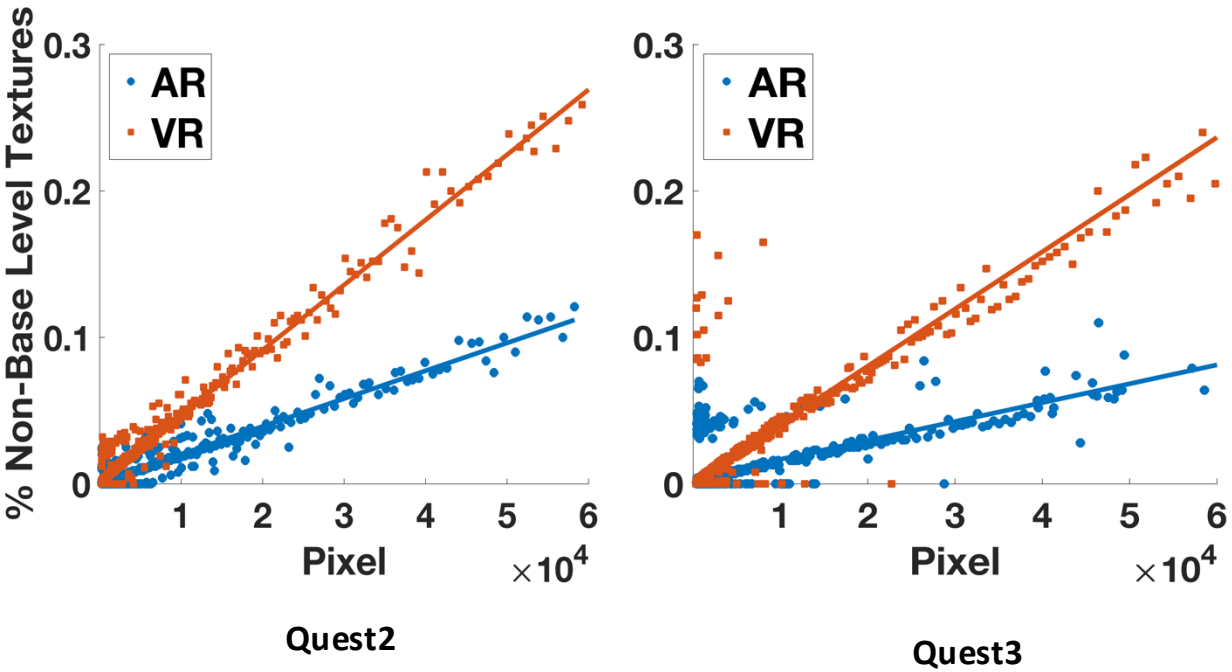
- **Non-Base Level Texture:** How often the GPU uses smaller copies of a texture

  3D object gets close/large
  ➔ GPU samples more small texture copies (stability)
  ➔ % Non-Base Level Textures ↑

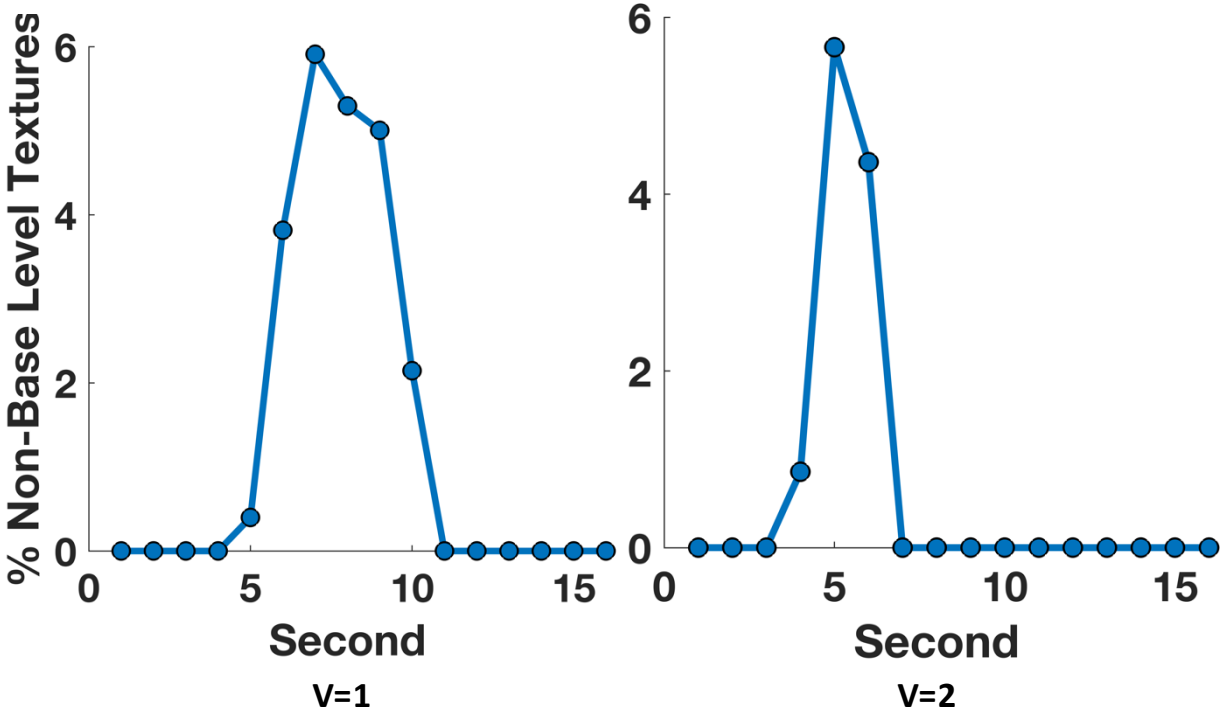
  3D object gets far/small
  ➔ GPU samples fewer smaller texture copies
  ➔ % Non-Base Level Textures ↓

Feature Extraction

- Correlation analysis



Pixel correlation with GPU metric

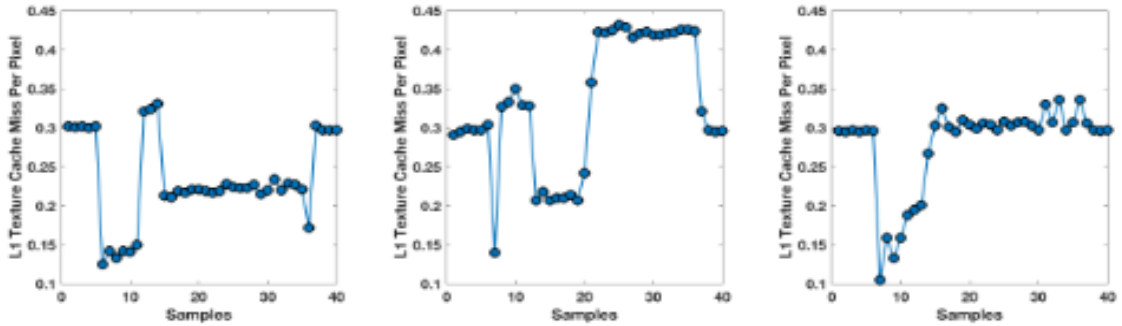


Dynamic correlation with GPU metric

Evaluation with Case Studies

- App Classification

  100 different **standalone apps** and 100 different **websites (WebXR)** with nearly **100%** accuracy

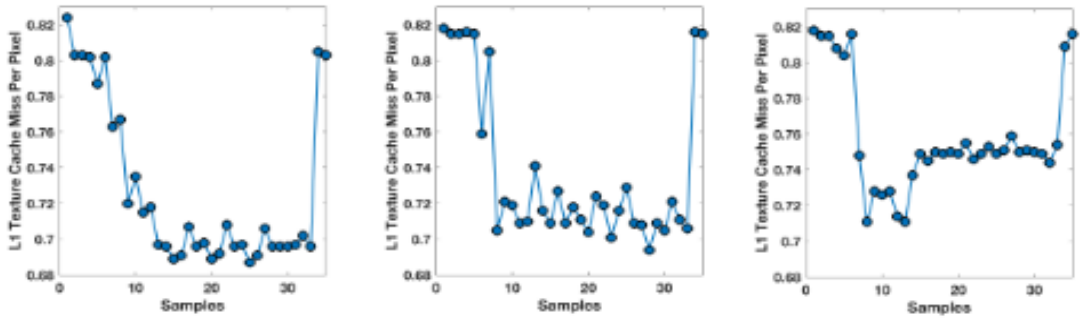


(a) Gorilla Tag (b) Bigscreen Beta (c) Multiverse

Example of fingerprints collected from Meat Quest 2 for AR/VR standalone apps

Model	F1 (%)	Precision (%)	Recall (%)	Accuracy (%)
CNN	98.3	98.5	98.4	99.3
LSTM	98.7	98.9	98.9	98.6
RF	99.4	99.5	99.6	99.5
SVM	85.1	86.9	87.6	86.8

AR/VR Standalone app fingerprint results with a combination of 30 GPU metrics



(a) VARTISTE (b) Moon Rider (c) Towermax Fitness

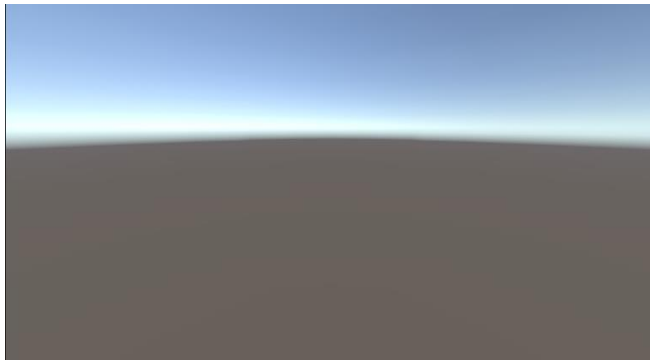
Example fingerprints for WebXR apps

Model	F1 (%)	Precision (%)	Recall (%)	Accuracy (%)
CNN	97.5	98.3	97.6	97.6
LSTM	96.1	96.7	96.4	96.4
RF	99.2	99.4	99.0	99.0
SVM	72.8	74.8	74.5	74.5

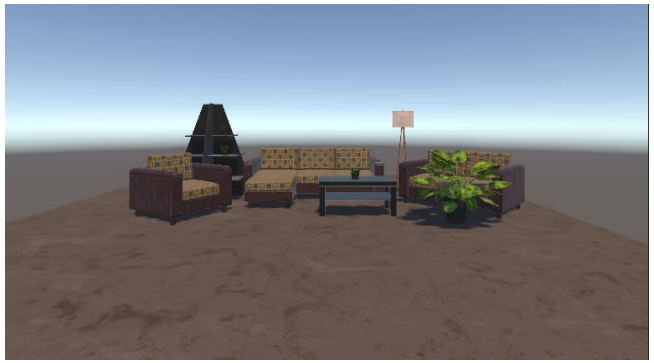
WebXR app fingerprint results with a combination of 30 GPU metrics

Evaluation with Case Studies

- Virtual Object Detection
  35 different objects with **98%** in four different environments

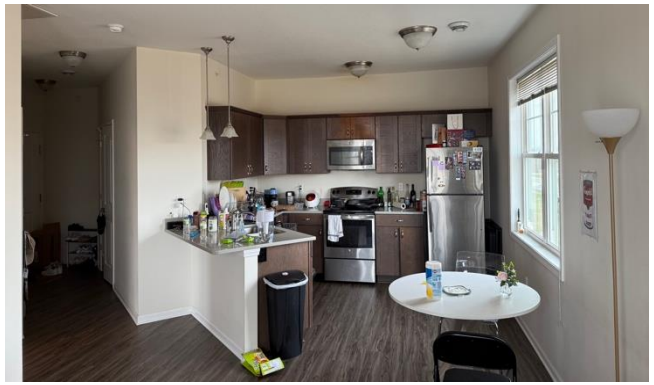


Empty VR Scene



Living Room VR Scene

Office AR Scene

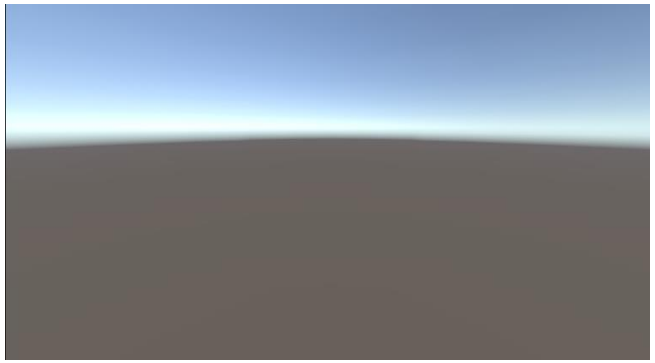


Living Room AR Scene

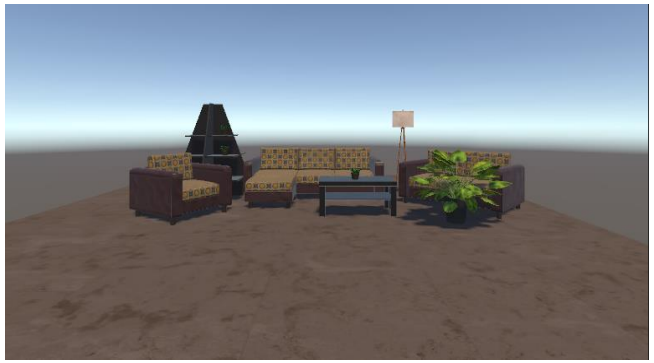
Experiment environments

Evaluation with Case Studies

- Virtual Object Detection
  35 different objects with **98%** in four different environments

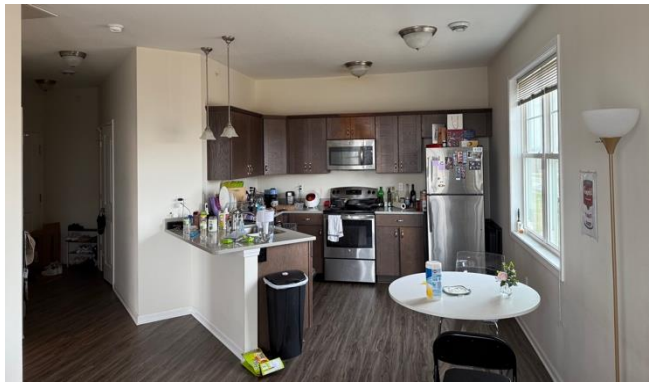


Empty VR Scene



Living Room VR Scene

Office AR Scene



Living Room AR Scene

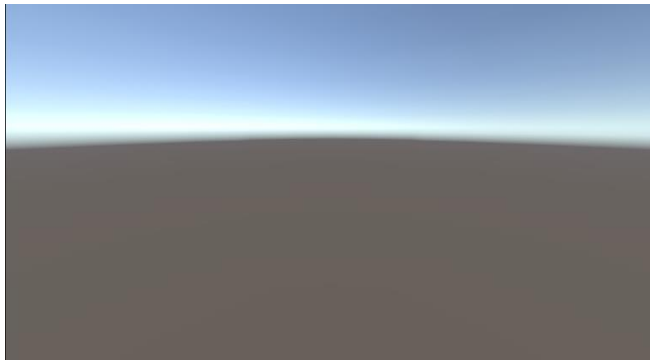
Experiment environments

Example of selected prefabs

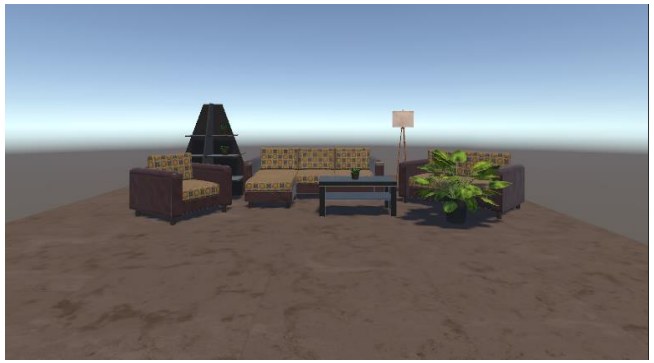
Evaluation with Case Studies

- Virtual Object Detection

  35 different objects with **98%** in four different environments

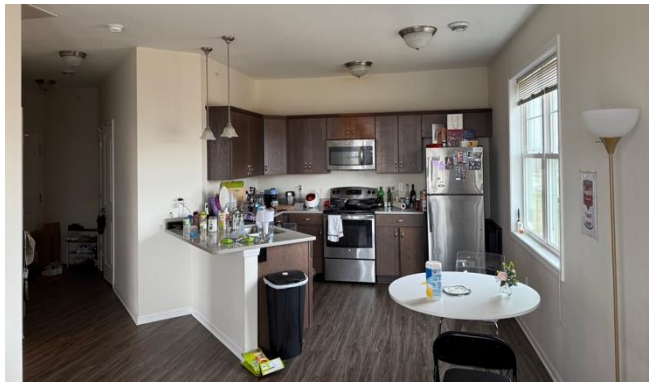


Empty VR Scene



Living Room VR Scene

Office AR Scene



Living Room AR Scene

Experiment environments

Example of selected prefabs

	VR Scene						AR Scene					
	Default			Living room			Office			Living room		
Model	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**
CNN	96.5	96.6	94.3	88.3	90.6	90.2	95.1	94.6	96.7	94.4	94.6	94.5
LSTM	92.6	92.3	94.3	75.8	80.1	80.9	89.8	90.9	91.3	91.4	91.1	93.0
RF	98.1	98.2	98.2	95.6	96.4	96.2	98.1	98.1	98.2	96.6	97.0	97.0
SVM	50.6	50.7	59.6	36.7	39.6	43.3	50.6	50.7	59.6	56.0	57.6	62.6

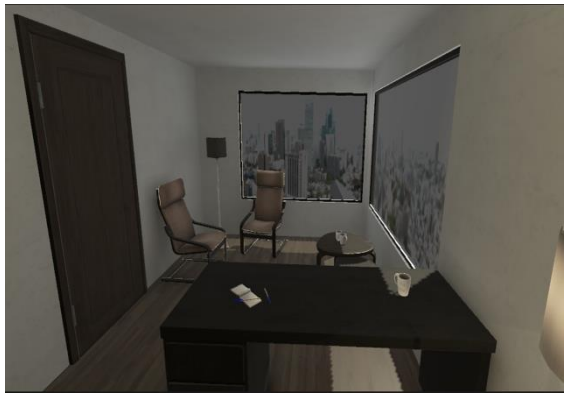
Result of VR Object classification

Evaluation with Case Studies

- Meeting room inference

  100% with 20 metrics individually utilizing the Random Forest (RF) model

Office AR Scene



Office VR Scene

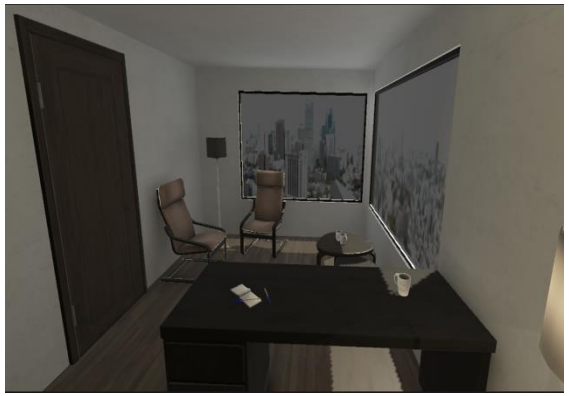
Experiment environments

Evaluation with Case Studies

- Meeting room inference

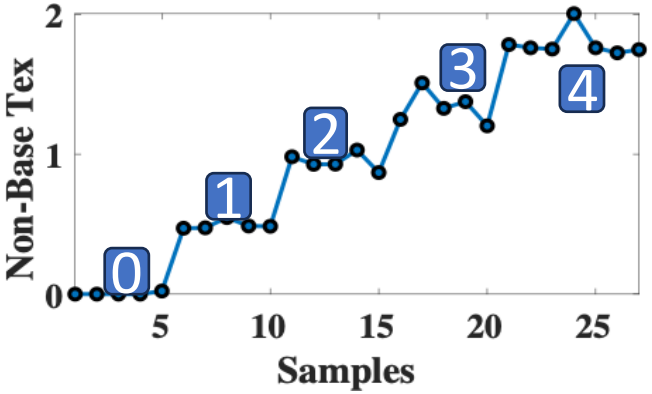
  100% with 20 metrics individually utilizing the Random Forest (RF) model

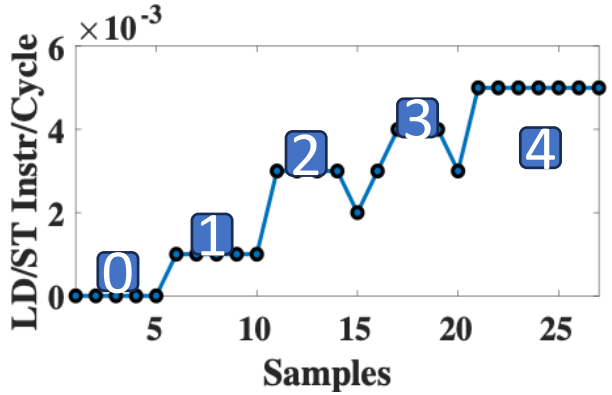
Office AR Scene



Office VR Scene

Experiment environments

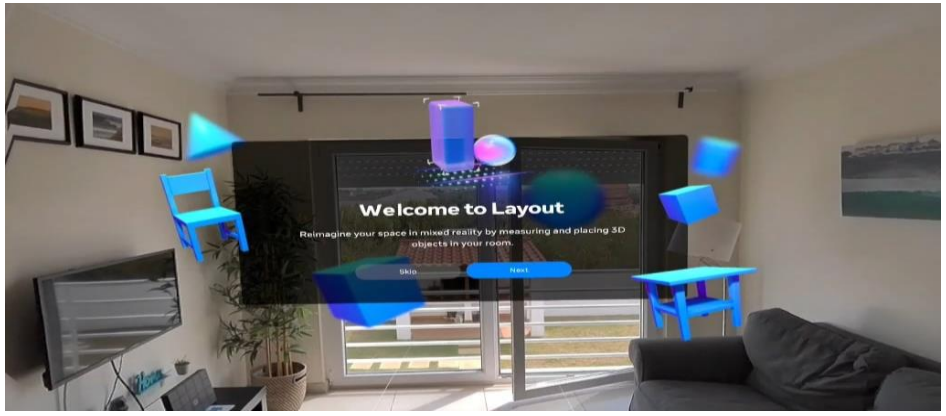


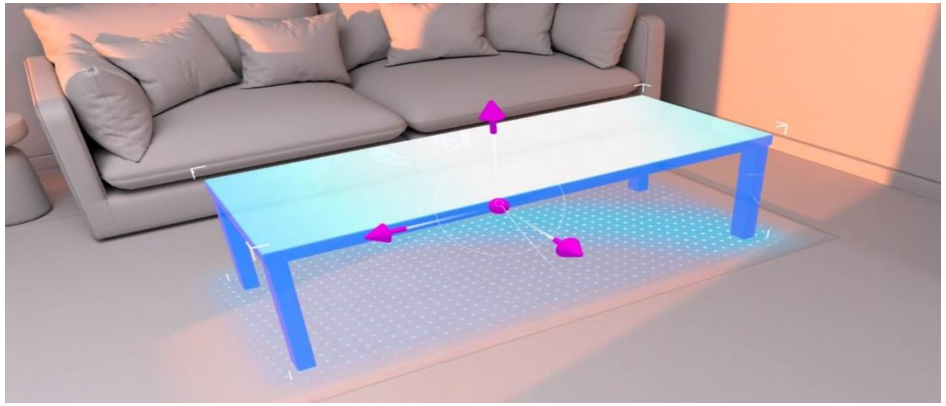


Example of a fingerprint dataset of meeting participant inferences in the default VR scene setup.

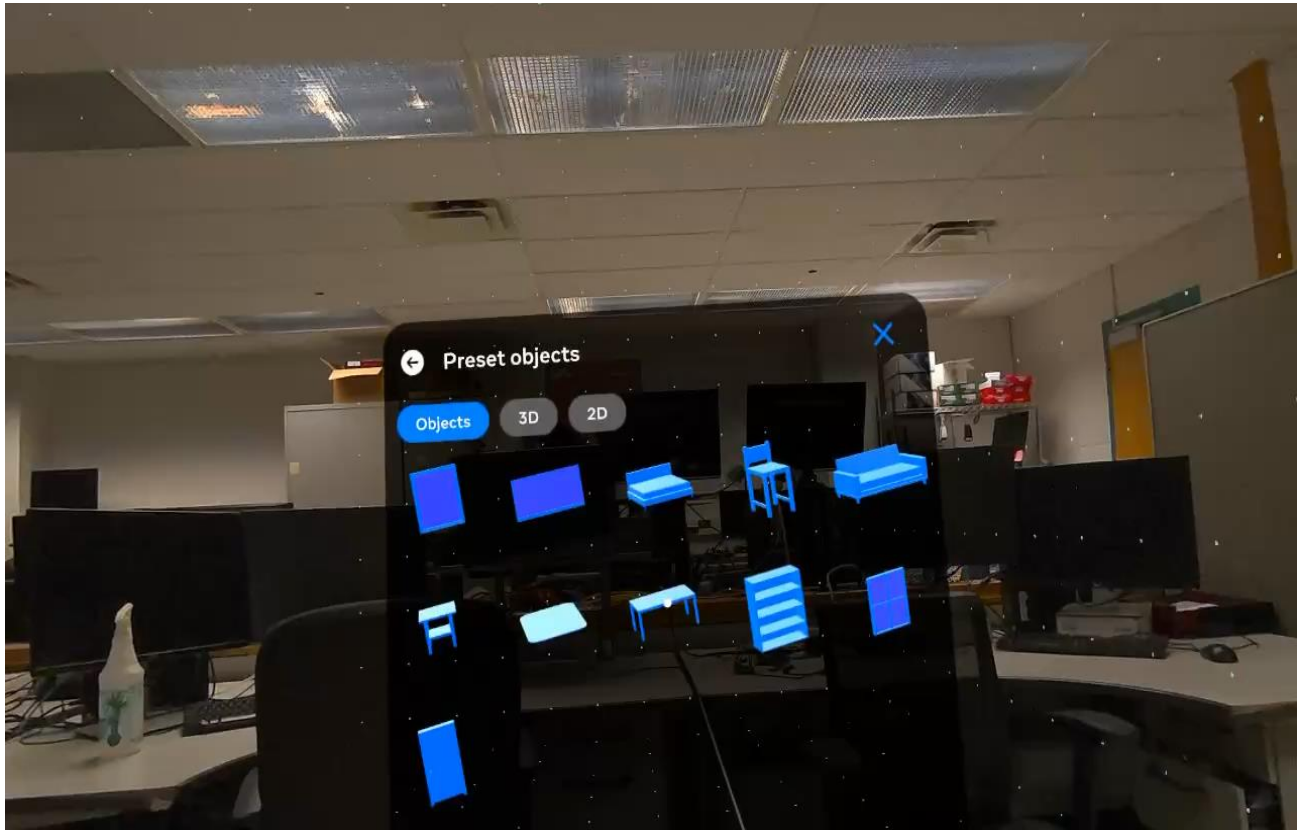
Evaluation with Case Studies

- Open World User Interaction
  - Test with Meta's built-in App "Layout"
  - 5 Users for user-study (IRB approval through ISU)





Example of Meta's Layout App



User Interaction Example Video

Evaluation with Case Studies

1. **Static Scenario:**
   a) Participants hold a Quest controller button to generate one of the five 3D furniture items (chair, sofa, table, desk, and bed) into the user's view for 5 seconds
   b) Make the 3D object disappear for another 5 seconds by pressing the button on the controller.
   c) Render-remove cycle takes 10 seconds and is repeated 10 times for each furniture item.
   d) Total of 100 seconds of GPU profiling traces

2. **Dynamic Scenario:**
   a) Participant opens the in-scene panel in the Layout app
   b) Selects a furniture item
   c) Drag it into their field of view for 10 seconds.
   d) After 10 seconds, the participant drags the 3D object out of their view.
   e) This scenario takes around 40 seconds and is repeated 5 times for each 3D object.

Scenario	Model	Test (%)	5-Fold (%)
Static Interaction	RF	83	86 ± 6
	XGB	82	88 ± 6
	SVM	30	38 ± 4
Dynamic Interaction	RF	81	77 ± 5
	XGB	67	66 ± 9
	SVM	78	80 ± 10

Accuracy of RF, XGB and SVM in the static and dynamic user-study scenarios on Meta Quest 3S

Evaluation with Case Studies

1. **Static Scenario:**
   a) Participants hold a Quest controller button to generate one of the five 3D furniture items (chair, sofa, table, desk, and bed) into the user's view for 5 seconds
   b) Make the 3D object disappear for another 5 seconds by pressing the button on the controller.
   c) Render-remove cycle takes 10 seconds and is repeated 10 times for each furniture item.
   d) Total of 100 seconds of GPU profiling traces

2. **Dynamic Scenario:**
   a) Participant opens the in-scene panel in the Layout app
   b) Selects a furniture item
   c) Drag it into their field of view for 10 seconds.
   d) After 10 seconds, the participant drags the 3D object out of their view.
   e) This scenario takes around 40 seconds and is repeated 5 times for each 3D object.

Scenario	Model	Test (%)	5-Fold (%)
Static Interaction	RF	83	86 ± 6
	XGB	82	88 ± 6
	SVM	30	38 ± 4
Dynamic Interaction	RF	81	77 ± 5
	XGB	67	66 ± 9
	SVM	78	80 ± 10

Accuracy of RF, XGB and SVM in the static and dynamic user-study scenarios on Meta Quest 3S

Countermeasure

- **Restrict GPU Profiler Access**
  - GPU profiler APIs are behind developer mode and app store approval, so normal apps cannot sample metrics
- **Dynamic Detection**
  - Detect repetitive profiler polling patterns at runtime and warn the user or throttle access with minimal overhead
- **Noise Injection**
  - Add a controlled dummy GPU workload or random rendering to blur fingerprints

Limitations

- **Applicability to Diverse Devices**
  - Different XR platforms expose different profiler interfaces and metrics
- **Generalizability**
  - Multi-app usage and dynamic scenes can reduce accuracy

Conclusion

- Multi-app Built-in GPU profilers create a measurable privacy leakage surface in XR
- Distinct GPU fingerprints enable inference of standalone apps, WebXR scenes, objects, and meeting participation size
- Attacks can remain effective even at a **low 1 Hz sampling** rate
- Practical defenses need platform changes: restrict profiler access, detect abnormal polling, and consider noise injection with performance tradeoffs

Achievement

- **Security impact highlights**
  - Received **Meta Bounty Award** by Meta's Security Team

NDSS SYMPOSIUM/2026

THANK YOU

- **Seonghun Son**
  - Ph.D. Candidate at Iowa State University
  - seonghun@iastate.edu

- More details:
  https://arxiv.org/pdf/2509.10703

Seonghun's website