# Explainable XR: Understanding User Behaviors of XR Environments using LLM-assisted Analytics Framework

Yoonsang Kim, Zainab Aamir, Mithilesh Singh, Saeed Boorboor, Klaus Mueller, and Arie E. Kaufman, *Fellow, IEEE*


Fig. 1: Explainable XR provides a streamlined pipeline to record, visualize, and analyze users of an immersive session, facilitating researchers from various domains of expertise, to readily comprehend and study them. Our structured user data recording format captures users' actions (e.g., GazeAt, Move Hand, Select) and their contextual reasoning. Through our data analyzer and LLM-generated insights, we present base analytics interpretation of the users' action data on top of our visual analytics interface, and assist researchers to approach the analysis from multifaceted perspectives.

**Abstract**—We present *Explainable XR*, an end-to-end framework for analyzing user behavior in diverse eXtended Reality (XR) environments by leveraging Large Language Models (LLMs) for data interpretation assistance. Existing XR user analytics frameworks face challenges in handling cross-virtuality – AR, VR, MR – transitions, multi-user collaborative application scenarios, and the complexity of multimodal data. Explainable XR addresses these challenges by providing a virtuality-agnostic solution for the collection, analysis, and visualization of immersive sessions. We propose three main components in our framework: (1) A novel user data recording schema, called User Action Descriptor (UAD), that can capture the users' multimodal actions, along with their intents and the contexts; (2) a platform-agnostic XR session recorder, and (3) a visual analytics interface that offers LLM-assisted insights tailored to the analysts' perspectives, facilitating the exploration and analysis of the recorded XR session data. We demonstrate the versatility of Explainable XR by demonstrating five use-case scenarios, in both individual and collaborative XR applications across virtualities. Our technical evaluation and user studies show that Explainable XR provides a highly usable analytics solution for understanding user actions and delivering multifaceted, actionable insights into user behaviors in immersive environments.

**Index Terms**—Extended Reality, Cross Reality, Multimodal Data Collection, User Behavior, Visual Analytics, Personalized Assistive Techniques, Large Language Models

## 1 Introduction

The recent advancements in eXtended Reality (XR) technologies – Augmented Reality (AR), Virtual Reality (VR), and Mixed Reality (MR) – have significantly enhanced user experiences. It has transformed the way we retrieve information, interact with data, and collaborate with peer users. As a result, immersive technologies and their applications are being widely adopted in various multidisciplinary domains such as education, healthcare, and industry [4, 5, 25, 37, 54, 55, 84]. Compared to traditional interaction methods, immersive interaction techniques have uncovered diverse and complex patterns of user behaviors and responses, highlighting the unique ways users engage with each other and the environment [28, 29, 41, 55, 67, 73, 80]. Recent studies on analyzing user actions within XR environments [11, 31, 35, 58] have made significant contributions in capturing, visualizing, and interpreting user behavior data.

In addition to enabling rich, immersive experiences, XR technologies also offer the capability to transition seamlessly between different virtualities within a single application. This flexibility has spurred a growing interest amongst researchers, leading to a number of studies on cross-virtuality experiences and multi-user collaboration [23, 39, 44, 62, 74, 78]. As user roles, interaction patterns, and behaviors in XR environments become increasingly complex and diverse, there is a pressing need for a unified standard that accommodates various types of immersive experiences and enables consistent, systematic evaluation, analytics, and visualization across them. Moreover, a single session of an XR application can generate a large volume of multimodal data, including spatial, temporal, visual, and audio. As more sessions are integrated,

*All authors are with Center for Visual Computing at Stony Brook University, New York. E-mail: {yoonsakim, zaamir, mkssingh, sboorboor, mueller, ari}@cs.stonybrook.edu.*

*Received 18 September 2024; revised 13 January 2025; accepted 13 January 2025. Date of publication 10 March 2025; date of current version 31 March 2025. This article has supplementary downloadable material available at https://doi.org/10.1109/TVCG.2025.3549537 provided by the authors Digital Object Identifier no. 10.1109/TVCG.2025.3549537*

the volume and complexity of the data increases, posing additional challenges in deriving insightful visualizations and interpretations due to data overload.

To address these challenges, we present ***Explainable XR*** (in short, *EXR*), an end-to-end user behavior analytics framework for the collection, analysis, and visualization of XR user(s) in a spectrum of XR environments (Fig. 1). Its main features are (i) User Action Descriptor (UAD) - a user action-centric standardized structured schema for data recording; (ii) an easy-to-use, Unity-based platform-agnostic XR session recorder; (iii) a web-based visual analytics interface presenting multimodal data – spatial, temporal, visual, and audio – of user interactions of XR sessions in a single view; and (iv) leveraging Large Language Models (LLMs) to facilitate data analysis. We list our main contributions as:

- A publicly available [1], end-to-end XR user action analytics framework that supports a wide range of XR environments (AR, VR, MR), including cross-virtuality applications, multi-user scenarios, and various immersive platforms.
- A standardized user action-centric data schema that facilitates ease of use and scalability while capturing detailed information about user actions and the context surrounding each action.
- A Unity-based plugin that allows for easy customization and adaptation to specific user requirements.
- Leveraging Large Language Models (LLMs) to synthesize multimodal data and generate user-input tailored summaries and analytical insights, facilitating XR session analysis and interpretation.
- A web-based analytics interface for visualizing the recorded data in a unified view, combined with generated insights for a more comprehensive analysis.
- The demonstration of versatility and usefulness of Explainable XR with five diverse XR scenarios ranging from single/multi-user, synchronous/asynchronous, to individualistic/collaborative applications.

## 2 Related Work

In this section, we review relevant works and highlight the unique contributions of EXR in comparison to them. Tab. 1 supplements this discussion by illustrating the gaps.

**Recording of XR User Behavior Data** Capturing various data streams in an XR session, including 6DoF (Degrees-of-Freedom) transformations, actions, gestures, and physiological data such as gaze can provide valuable insights into understanding user intentions and behavior patterns [17, 56, 59, 64, 66, 70, 88]. The recording of such data provides the means for a deeper analysis of users and the context behind every XR interaction.

Several toolkits and frameworks for immersive sessions have been developed to derive insights from these various user data streams. One of the early toolkits to capture user actions in AR and MR is MRAT [58]. MRAT offers user information logging per task-basis. Given a set of tasks for user performance testing, it tracks the user's task status, spatiotemporal data, interaction type, target virtual object, gaze, screenshot, gestures, and voice commands, which are then used to visualize and analyze the task performance of a user. UXF [7] proposes a Unity-based framework that simplifies VR experiment development and data collection for behavioral research. Other works also place emphasis on human behavioral experiments in Virtual Reality [6, 26, 77]. Frameworks such as Cognitive3D [14] advance a step further by providing support for cross-virtuality (AR, VR, MR) recording and playback. Furthermore, PLUME [35] and other recent studies [19] incorporate the ability to record users' physiological signals such as eye tracking and heart rates as well. A set of research concentrates on collecting data from a first-person perspective [15, 52, 79]. Some works focus specifically on the multi-user aspect of XR, recording multiple users' actions [9, 58, 71].

The above works and EXR record similar XR multimodal data such as visual, gaze, and audio, as well as user actions, context, spatial and temporal data. Our work, however, takes a more structured, user action-centric approach. We consider user actions as a trigger for an information update in an XR environment, and we link all tracked XR data to a user's action. This logging structure allows our EXR to capture user behavior and relevant context, while filtering out non-essential session data. Additionally, this user action-centric approach allows for scalability in multi-user scenarios.

[1] GitHub Page: https://github.com/yoonsang0910/ExplainableXR

Table 1: Comparison of XR user analytics systems. Symbols (◗, ●) denote partial and full functionality support of a system, respectively. Action Context refers to the built-in capability to save the context behind users' actions. Cross-user indicates support for single and multi-user sessions. Task-agnosticism highlights the systems adaptability to diverse tasks, rather than being tailored to a specific task. Novel functionalities exclusive to EXR are omitted.

| System | Virtuality | Action Context | Cross-user | Task-agnostic |
|---|---|---|---|---|
| ARGUS [11] | AR | ● | | |
| PLUME [35] | AR, MR, VR | | ◗ | ◗ |
| MIRIA [9] | AR, MR | | ● | ◗ |
| MRAT [58] | AR, MR | ◗ | ● | ◗ |
| ReLive [31] | AR, MR, VR | ◗ | ● | ◗ |
| **EXR (Ours)** | AR, MR, VR | ● | ● | ● |

**Visualization of XR User Behavior Data** Effective visualization of an XR session is crucial for visual interpretation and the comprehension of user behaviors. There have been two main approaches: In-situ and ex-situ. The former, often used with immersive analytics, focuses on first-person views to trace users' reasoning processes [9,49,58]. Ex-situ visualizations are typically paired with 2D visual analytics interfaces, and use third-person views to provide a broader understanding of users' actions and the contexts behind their actions [8, 14, 31]. Some studies take advantage of both and combine immersive in-situ and non-immersive ex-situ analysis [34, 35].

ARGUS from Castelo et al. [11] provides an analytics interface for visualizing AI model outputs of an AR session. They offer both real-time (online) and retrospective (offline) tracking of the AR user's scene and interactions, along with debugging functionalities. The work focuses on the scenario where a user with an AR Head-mounted Display (HMD) performs physical tasks through an AI-guided system, analyzing spatiotemporal properties of user actions as well as the AI model's outputs. Also, to visualize the physical environment (action context) at the moment of a user action, they use sparse point clouds on top of screenshot captures, similar to Yu et al. [86]. ReLive [31] proposes a visual analytics interface to provide a holistic visualization of individual user actions and a synchronized, aggregated view of multiple users. Their data logging toolkit captures and stores the action context of users – 3D scene for virtual applications (VR) and accept real-world scan for physical applications (AR, MR). Also, they maintain screenshots for action events.

Existing systems often use visual elements such as graphs, plots, glyphs, or attention maps (2D or 3D) to enhance the understanding of user's actions by visualizing observable data. EXR extends this approach by not only tracking and visualizing user behaviors but also providing insights into the potential reasoning behind each action, powered by LLM. This provides analysts with a deeper understanding into the motivations and contexts behind XR user interactions. By integrating spatial, temporal, and interaction data, EXR allows for a comprehensive analysis of XR session dynamics.

**AI-assisted Analytics and Visualization** The integration of Artificial Intelligence (AI) through Large Language Models (LLMs) has opened new avenues for data analysis and visualization. This has enabled advanced pattern recognition, predictive analytics, automated insight generation, and context-aware visualization, enhancing the data analytics experience [3, 63, 69, 81].

Recent studies have explored various AI-assisted methods to enhance data analytics. InsightPilot [50] proposes a streamlined data exploration process that generates data insights reducing the effort needed to understand the data [13, 57, 89]. LIDA [20] proposes an infographics genera-

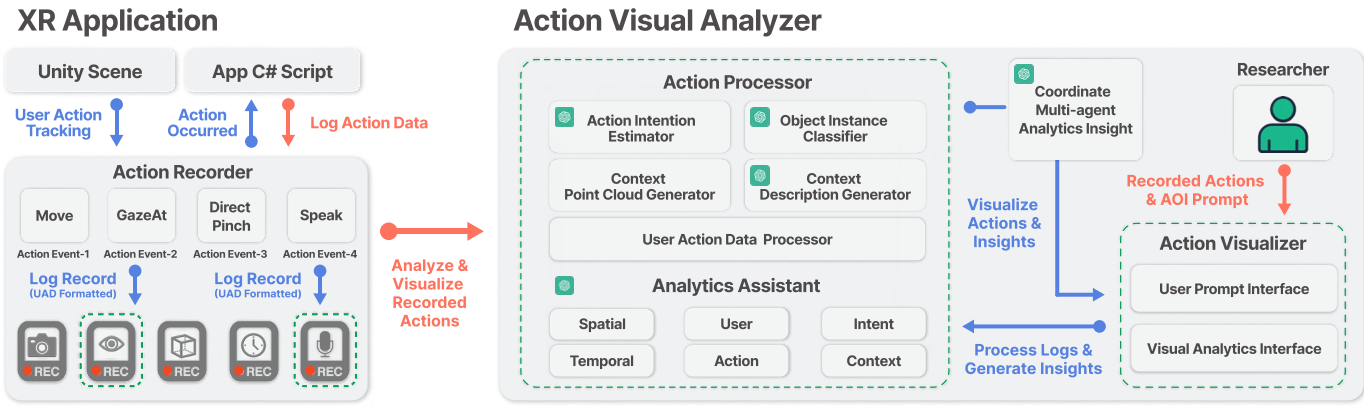


Fig. 2: Explainable XR Pipeline Overview: **The blue arrows** denote the internal calls and flows of Explainable XR, and **red arrows** denote the inputs of the researcher in our framework. The pipeline initiates by recording the multimodal interactions of the subjects in XR sessions, and importing it into our Action Visual Analyzer via User Prompt Interface. The researcher can perform analytical tasks in our Visual Analytics Interface and optionally utilize our analytics insights during the analysis.

tion pipeline using LLM to suggest visualizations. Other works have also utilized AI for generating visualizations [16, 18, 21]. While Shen et al. [68] applies task decomposition with multiple LLM agents for automatic data storytelling generation. Although AI-powered data analysis and visualization offers powerful capabilities, it present challenges on reliability and trustworthiness of the outputs, leading to research on the transparency and explanability of AI generated output [48].

Building on these developments, EXR incorporates AI/LLM-assisted analytics optimized specifically to our user action-centric data structure. We maintain the human-in-the-loop approach to ensure reliable data exploration and decision making. In EXR, LLM is used to understand large, multivariate datasets, emphasizing key information and patterns. The use of LLM is underscored as EXR manages complex visual analytics involving multiple users across diverse XR configurations and virtualities.

**Task-agnostic User Analytics Framework for XR** Numerous studies have focused on designing analytical frameworks for XR sessions [8, 11, 14, 31, 35, 58]. These works commonly provide XR session recorder and a viewer, but most are tailored for specific tasks such as measuring user performance or testing [7, 31, 58], or an AI-guidance system [11, 12], although they can be customized to a degree. PLUME [35] proposes a more open-ended framework that is not bound to a specific task. It utilizes low-level compile time code modifications to track log event raises for logging XR session data. Additionally, PLUME provides both in-situ and ex-situ visualization of the XR session. However, it lacks scalability for multi-user scenarios and practical AR/MR applications involving physical scenes with no prior context information (pre-scanned scene).

As compared to other works, we extend the use-case of our framework beyond user performance measurement by introducing a versatile action-centric data structure. This supports diverse tasks such as spatial, temporal, topic, and action intention analyses, positioning EXR as a general-purpose analytics framework for XR sessions.

## 3 Explainable XR: Design and Implementation

Analyzing behavioral patterns is critical for understanding users and task efficiency in XR environments. EXR is designed to facilitate domain experts from diverse backgrounds interested in studying human subjects in XR settings, to be able to readily collect, visualize, and analyze the behavioral patterns of the subjects of the immersive sessions. As an end-to-end framework, it provides base template designs for tasks spanning from recording to analysis, with the ability to customize any functionality. It is designed on top of Unity3D [76], a widely used engine for XR application development. Fig. 2 illustrates the pipeline of our framework. In designing our framework, we abstract the convoluted inner workings of data processing and the LLM logistics.

As illustrated in Tab. 1, EXR is designed as an "all-in-one package" framework that bridges the gap between existing XR user analytics systems. It is general-purpose (not confined to a specific task), operates seamlessly across virtualities, supports both single and multi-user sessions, inherently embeds contextual information (environment) with every user action, and offers intelligent assistive techniques to enhance the analytics experience.

The process begins with the Action Recorder, which can log the actions of the users such as 'Move,' 'Grab Object,' or 'GazeAt.' Subsequently, each log is grouped by user and action, and structured using our User Action Descriptor (UAD) schema. This descriptor can be used across XR virtualities and configurations. Next, the structured data is loaded using a User Prompt Interface, along with an optional Analysis-of-Interest (AoI) prompt. Providing an AoI allows EXR to generate user-interest tailored insights using an LLM. Then, through our Visual Analytics Interface (VAI), they can visually analyze user task performance and patterns. The components of VAI are interconnected, allowing for a unified analysis where selections from one component update the others. Henceforth, in this paper, we interchangeably use the terms user behavior, action, and interaction of XR. Moreover, we represent the user using the XR application and being recorded, as a ***Subject***, and the user of EXR analytics framework as ***Analyst***. The term user in naming the EXR components refers to the subjects.

### 3.1 User Action Descriptor

UAD is an action-centric structure that preserves various aspects of a subject interacting with an individual or collaborative XR application session. As discussed in Sec. 2, existing works loosely define event-trigger conditions for data logging. Any occurrence of an event in the session, including events that are irrelevant to the subject's view or action in the XR session space, can be recorded. For example, a moving cube in a VR session can be tracked and stored throughout, even when it is out of the user's viewing frustum. This can result in data overload, for both processing and analysis. The information presented in XR applications is typically egocentric. That is to say, the presented visualizations, flow of information, and interactions initiate from a subject's point-of-view (PoV) when the subject performs an action with the XR environment. This is also true for third-person AR applications, as they rely on the user-induced in-application camera (virtual camera) position and orientation. Let's consider a practical AR application scenario:

"*Jake entered the kitchen wearing an AR HMD to cook a dish with the help of an AI-guided AR application.*"

Loading the recipe data was triggered by Jake's entrance (**User Action1**) into the kitchen, and Jake selected (**User Action2**) one of the options from the cooking recipe on the AR User Interface, which gray-highlighted the interface button. Also, Jake tapped (**User Action3**) on the button of the interface for the selection. This indicates that a subject's action was the source of all information in XR. In other words, a subject's action triggers an update in an XR environment.

Table 2: User Action Descriptor: This structure organizes all immersive session data centered around a subject's actions, enabling EXR to analyze multivariate connections between actions and their surrounding contexts. It is adaptable for any immersive application or task across virtualities.

| Field | Description | Data Type | Example Value |
|---|---|---|---|
| Name | The name of the action | String | "Navigate", "GazeAt", "Touch" |
| Type | The type of the action | Enum (Type of Action) | Discrete, Continuous |
| Intent | The intent behind the action | String | "Load immersive plots" |
| User | The user identity of the action invocation | String | "User1" |
| Location | The 6DoF locations of the action invocation | List<Transform> | [(Pos(0,0,0), Rot(0,5,5)),..] |
| TriggerSource | The medium on which the action is triggered | Enum (InputAction Device) | XRHMD, XRController, Audio |
| StartTime | The start time of the action event | TimeStamp (Ymd:HMS:f) | 240801:092855:031 |
| Duration | The lengths of the action event | TimeDelta (Ymd:HMS:f) | 000000:000135:328 |
| Referent | The target object of the action | Bytes (GLB or PNG) | GameObject.glb, Screenshot.png |
| ReferentType | The reality in which the target object exists | Enum (Type of Reality) | Physical, Virtual |
| ReferentLocation | The 6DoF locations of the target object | List<Transform> | [(Pos(10,5,4), Rot(0,-5,5)),..] |
| Context | The context behind the action | Bytes (GLB) | PointCloud.glb |
| ContextType | The reality in which the context exists | Enum (Type of Reality) | Physical, Virtual |

Drawing from that, we have designed the UAD to consolidate the collected session data based on the subject's actions, gestures, or behavior. These include examples such as navigating, touching an AR-projected object, pinching a virtual cube, or gazing at another subject. The UAD inherits its base concept from the Kipling method, 5W1H – When, Where, Who, What, Why, and How – to describe the context and the information of an incident [10, 33, 85]. Below, we outline the association of the Kipling method (in bold) with the schema of each UAD field (in italics). The complete list of the UAD fields is provided in Tab. 2.

**When:** analyzes the temporal property of an action. We record the occurrence of an action with *StartTime* and *Duration*.

**Where:** analyzes the spatial property of an action. We track the 6DoF of the invoked action and the 6DoF of the action's target, stored as *Location* and *ReferentLocation*, respectively.

**Who:** traces the source of an action, stored as the *User*. This information is especially useful for analyzing collaboration in multi-user XR scenarios. We enforce subject anonymity and assign *User* with a numerical identifier for separating unique subjects.

**What:** defines the action within the XR space, categorized by its *Name* and *Type*. The *Type* can be discrete ('Button Press', or 'Pinch') or continuous ('Move Object' or 'GazeAt'). Unlike discrete, continuous actions involve a sequence of spatial movements within an action.

**Why:** identifies the *Intent* of an action, allowing a single action to have more than one usage in an application. For instance, a single 'pinch' action can be used for "grasping an object" or "initiating a teleportation", as defined by the application developer. Moreover, *Intent* describes the specific consequences of an action as well. Thus, we leverage this to track the reasoning behind the subject's action, in the later stage of Action Intention Analysis and inference.

**How:** specifies the method used to trigger an action, recorded in *TriggerSource* by tracking the sensor or module recognizing the action. For seamless use across XR environments and platforms, we utilize Unity's Input System [75], inheriting all trackable sensors and modules such as 'HandheldARInputDevice', 'XRHMD', and 'XRController'.

**And More:** identifies additional visual cues of the action to provide additional context for an action. In the UAD, the *Referent* denotes the interactable target of an action. We store the *Referent* using GLB, a platform-agnostic format, to support the use of UAD across all XR environments. It can store the name, geometry, and material of any virtual entity in Unity. To even support the storage of a physical entity in the *Referent* field, we classify the physical referent via our post-hoc processing module, Action Referent Classifier, and store it.

We also maintain the circumstantial context of an action. The *Context* field stores the semi-dense 3D point cloud of the subject's XR scene, at the time of an action, allowing EXR to associate each action to the interaction space/scene directly. We choose point cloud over other 3D representations, such as Neural radiance fields [53] or Gaussian splats [40], for its compatibility and portability [45, 74]. Moreover, it is more robust than a surface mesh in maintaining the spatial information. The *Context* is stored as a GLB as well and supports recording of both physical and virtual scenes. The point cloud reconstruction is done in the post-hoc processing module, Context Point Cloud Generator. This eliminates the need to upload prior context (VR), a scanned environment (AR/MR), or a digital twin (AR/MR) of the subject's interaction space. Furthermore, since we bind the context to every action of a subject, EXR can even capture and visualize the consecutive movements of an object as long as an action was involved (e.g., Gazed At, Grabbed). To illustrate a usecase, consider the following example:

"*At the beginning* ***(When)*** *of a Mixed Reality user study session, Bob* ***(Who)*** *pressed* ***(What)*** *a UI button* ***(Referent)*** *that is anchored near the starting position* ***(Where)****, with his hand* ***(How)****, to visualize immersive analytics data* ***(Why & Context)****.*"

### 3.2 User Action Recorder

Action Recorder offers an easy-to-use plug-in for logging a subject's multimodal XR session data, simplifying the complex logistics of data recording. Optimized and tailored for the UAD format, it supports two recording methods to capture a subject's actions: Template-based logging and Direct logging.

**Template-based Logging.** To provide a starting point for developers and investigators to readily use the UAD format of EXR, we have designed a Unity editor-based GUI called Action Template Logging Editor that generates a base template C# script for logging from input actions. As shown in Fig. 3, our visual editor can accept any number of actions and their associated intents and sources. The list of trackable devices (*Trigger source*) is device-agnostic, as the recorder module can be deployed on any Unity-developed XR device. Also, the *Trigger source* can be customized on top of the existing *Trigger sources*. Once the developer completes the configuration of the actions on the visual editor, our framework automatically generates a script with the basic structure for action event listening, condition statements, and template codes for logging. The code generator assigns each action intent a separate function that is named after the user-specified names of the action, intent, and the *Trigger source*, in the visual editor.

**Direct Logging.** Direct Logging is a more advanced approach to recording subjects' session information. Using this approach, based on the analyst's needs, they have the freedom to insert a logging function directly into their scripts. Fig. 4 shows an example of the logging function and its detailed arguments. This approach requires manual initialization of the Logger and data storage at the completion of a session.

Both Template-based and Direct Logging, invoke the *Log* function shown in Fig. 4, to record an action. Once the *Log* function is invoked from the application C# script, the defined parameters are processed to follow the UAD format. For the *Referent* field, a Unity object is converted to GLB and a visual capture of a physical object is stored.

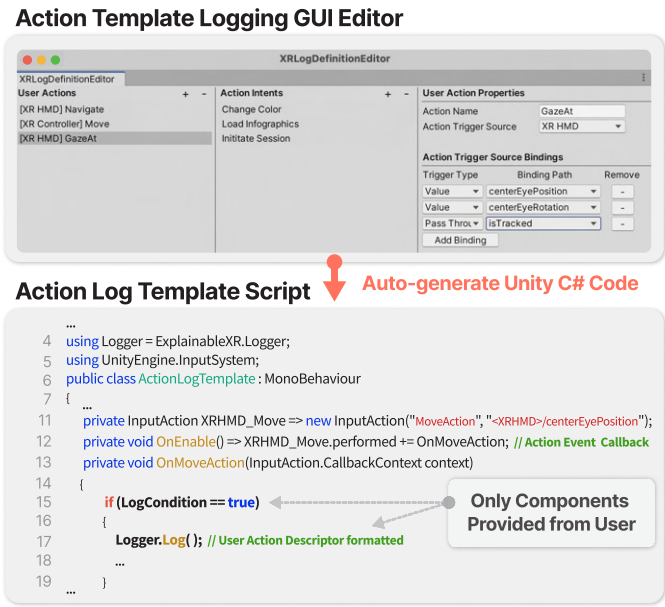


Fig. 3: Action Template Logging Editor and its auto-generated code: Our visual editor streamlines the process of action logging by generating a Unity C# base template code. The user can record subjects' immersive session data with a simple modification of the conditional statement (*LogCondition*) and the log (*Logger.Log*) function arguments. The code is generated with a button press in the visual editor, from the user.

Logger.Log(**Who**: "*User1*", **Where**: *XRHMD.transform*, **When**: *Time.Now*, **What**: "MoveAction", **Why**: "*Navigating Scene*", **How**: "*XRHMD*", **Referent**: *targetObject*, **Context**: *snapshot.png*

Fig. 4: Structure of Logging Function: It conforms to the User Action Descriptor format and internally stores the action data in JSON format, upon invocation.

For the *Context* field, the visual (RGB) and depth (Depth) images are captured using the XR device, and the camera parameters (intrinsic, extrinsic) are stored. These are subsequently used in the post-hoc processing phase to reconstruct the 3D point cloud of the subject's action context as illustrated in Fig. 5. Snapshot captures help minimize the overhead induced by an application in logging information such as physical referent detection, classification, audio processing, and Action Context Image Analysis.

### 3.3 User Action Visual Analyzer

As illustrated in Fig. 2, User Action Visual Analyzer consists of three main components: Action Processor, Analytics Assistant, and Action Visualizer. In this section, we describe how the recorded action data is processed and prepared for visual analysis, followed by an explanation of the techniques used to provide the multifaceted analyses.

#### 3.3.1 Action Processor

The Action Processor is the core preparation stage for the visual analysis of the session data. Given the recorded data, it concatenates the action data of each subject into a single file, generates semi-dense point clouds from the context data, and performs various LLM inferences to facilitate analysis. The inference includes Action Context Description Generation, Action Intention Estimation, and Action Referent Classification.

**Action Context Point Cloud Generation.** Point clouds are generated to visualize the contextual depiction of the subject's surroundings at the point of an action. It provides an enhanced background reasoning of the action. As shown in Fig. 5, we generate a semi-dense point cloud from logged images from the logical in-application camera for VR applications and logged RGB-D images from the physical device camera for AR/MR applications.

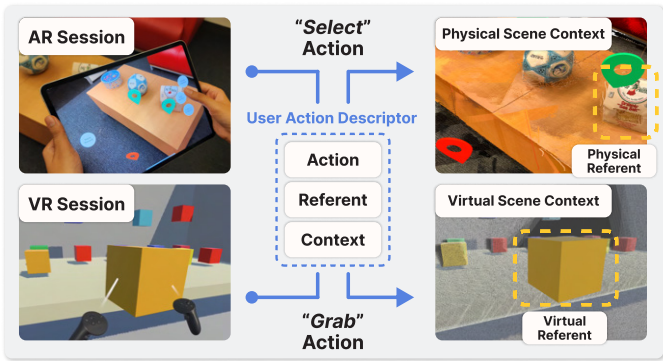


Fig. 5: Virtuality-agnostic Session Reconstruction: UAD binds each action of a subject with a referent and a scene context. The referent that is a physical entity is inferred through Action Referent Classifier, and the virtual entity is logged through gameobject storage. The context point cloud can be generated using the snapshots from a Unity in-application camera, or through a physical XR device camera. The former is mostly used for VR, and the latter, for AR/MR.

**Action Context Description Generation.** To further assist data exploration and analysis, a textual description of the context information is generated. The natural language descriptions such as caption or annotation, along with the visualization, can present multimodal insights to the users [72, 81, 87]. To this end, we leverage a multimodal LLM by querying the subject actions, intents, and referents, to describe a snapshot of the subject action. The textual description or annotation is then appended to the UAD *Context* field.

**Action Intention Estimation.** As shown by the 'Speak' action in Fig. 3, not all action intentions can be determined during application development. The intention of a verbal discussion ('Speak' action), can only be interpreted within the full context upon the completion of a discussion. We classify such cases as 'Post Defined' intentions. To address these, we employ a multimodal LLM agent tasked with deducing the plausible intention of an action based on the subject's context, action, visual snapshot, and transcribed verbal communication. The significance of intention deduction for verbal interactions is particularly critical in multi-user collaborative scenarios. Verbal expressions serve as explicit indicators of a subject's reasoning [58]. When combined with other actions, they provide valuable insights into subjects' revealed or hidden interests and behaviors.

**Action Referent Classification.** In AR/MR, a subject's interaction is not limited to the virtual realm as it involves augmenting information on physical objects. In practice, XR applications cannot exhaustively recognize interactable elements in the physical world. This can limit scalability and limit accuracy to VR applications. Thus, for AR/MR settings, we utilize the logged snapshot and infer the referent of the action in the post-processing by classifying the physical object using the LLM agent. The deduced object class along with its confidence score, is then added to the *Referent* field.

#### 3.3.2 Analytics Assistant

An increased amount of presented information can negatively impact the user's ability to absorb the information due to information overload [2, 51]. To address this, EXR offers two solutions: Analytics Insight and Analysis-of-Interest Marker.

**Analytics Insight.** To facilitate efficient data exploration and analysis, a curated list of LLM-generated insights (up to 10) are presented to the analyst, which serves as entry points for analyzing extensive datasets. These insights provide a concise recapitulation of the recorded XR session, offering analysts a quick overview of key points and patterns. To customize the generated insights, we have incorporated an AoI feature, where the analyst can specify their analytical focus. The AoI is prompted at the beginning of the Action visualizer pipeline via 'User Prompt Interface' shown in Fig. 2.

The extracted insights cover six distinct analytical aspects: space, time, action, intent of action, context of action, and user (interaction/collaboration analysis). This multifaceted approach enables EXR to guide

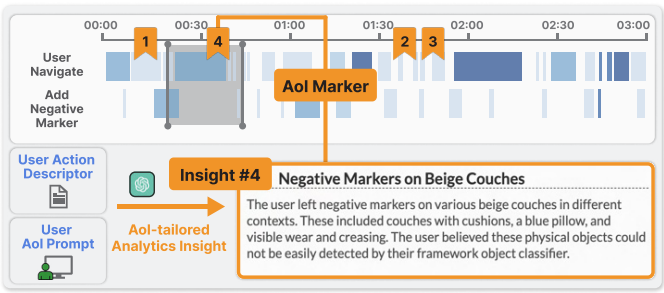


Fig. 6: LLM-Analytics Assistance: Given the prompt for the direction of the analysis from the user, we generate analysis direction-tailored Analytics Insights using multimodal LLM agents. The Analysis-of-Interest Markers are associated to every insight, to pin-point the data locations of the referred insights and user's key analytics interests.

users through a wide range of topics and tasks, from broad queries similar to "*Insights on the discussed topics in user collaboration*" to more specific inquiries such as "*Insights on the time spent object with Gaze action*".

During the insight generation process, LLM plays a critical role in synthesizing multimodal data streams such as gaze, gestures, and spatial interactions, captured in the UAD format. By contextually binding these diverse inputs, LLM generates cohesive insights that reflect the context-awareness of user behaviors in XR environments. This enhances the *'explain'* ability of data analytics, enabling the identification of implicit data patterns across modalities that may be difficult to derive through unguided data exploration and analysis. The example use-cases of our LLM-assisted insights, along with AoI prompts, are illustrated in Fig. 8. For details on the techniques we applied to the LLM-generated results, refer to the Supplementary Materials.

**AoI Marker.** As shown in Fig. 6, the AoI Marker is an analytics guidance module that visually highlights potential interests derived from the Analytics Insight module. Each highlighted insight points to the source action from which the reasoning is derived and its timestamp marked on the Temporal Viewer. This visual guide enhances the user's ability to navigate and focus on pertinent data points while maintaining the original information.

#### 3.3.3 Action Visualizer

The visual analytics of subject behaviors starts from importing the recorded action log to the User Prompt Interface shown in Fig. 2. The primary goal of the Visual Analytics Interface is to provide multifaceted analytical views of the UAD, processed by the Analytics Insight and AoI Marker modules. We now describe each individual component of the Action Visualizer.

**Spatial Viewer.** The Spatial Viewer visualizes the subject locations, contexts, and the referents at the point of an action, as shown in Fig. 7-B. A Trace Map displays the locations and frequency of each action for every user. It is specifically useful to infer spatial patterns of actions such as trajectory and locations of collaboration. Each action is mapped with a different color shade assigned to a user. As shown in Fig. 7, each trace point is a 3D point that represents an action overlaid with other spatial viewer components. The Spatial Viewer supports the simultaneous visualization of multiple action instances. In other words, if a range of time steps is selected in the Temporal Viewer, meaning more action instances, a more holistic spatial context of the subjects' actions can be visualized. Leveraging this technique, we visualize the full coverage of the subject's visited locations across any virtuality of XR, with only UAD-formatted recordings.

To reduce visual clutter in large collaborative multi-user scenarios, across time and space – synchronous/asynchronous and co-located/remote, a spatial data filter is provided that allows analysts to filter users, contexts, referents, or actions.

**Temporal Viewer.** The Temporal Viewer (Fig. 7-D) visualizes the occurrences, duration, and the frequency of actions for all subjects. Each action is a horizontal bar, where the width indicates its start and end times and color-coded using a blue sequential colormap indicating the frequency of an action, where dark indicates higher action frequency. The viewer is in chronological order of the actions of the subjects. AoI Markers are placed above the Temporal Viewer time axis as shown in Fig. 6. To examine actions in varying time-granularity, the analysts can also adjust the sampling interval. To support interactivity across the analytic viewer components, selecting a time range in the Temporal Viewer updates the Spatial Viewer, Insight Viewer, and plots, aligning all information with temporal contexts.

**Data Manager.** The Data Manager consists of Data Filter, Data Viewer, and Plot Viewer. The Data Filter consists of the list of all the actions present in the data, and provides a filter for each action. The Data Viewer presents the raw UAD information of the selected actions, including the *User*, *Intent*, and other information (Fig. 7-A). It is useful to view the 'as-is' data of actions. The Plot Viewer has two data visualization (Fig. 7-C) that can further assist the user in comprehending the subjects' behavior information.

**Insight viewer.** Insight Viewer visualizes the Analytics Insights. It provides a filter for the insights that enable analysts to consolidate the insights most relevant to their interests. When a user selects an insight box, the AoI Markers associated with that insight are highlighted. Insights are structured hierarchically for improved readability, with each insight recapitulated by a short summary ("*Negative Markers on Beige Couches*" in Fig. 6) followed by detailed content (the text below the summary headings in Fig. 6). This summary is referred to as the 'Title' of an Insight.

## 4 Case Study and Evaluation

We design five prototype applications (A1-A5) to demonstrate the capabilities of EXR. As shown in Fig. 8, the applications encompass various aspects of XR, including selection, interaction, and collaboration across virtualities. We summarize the apps are explained below:

- **A1** is a multi-user VR HMD-based application (Meta Quest Pro). The subjects are instructed to freely navigate the virtual scene and interact with the virtual objects in the scene. The application tracks their movements, gaze, and gestures.
- **A2** is an MR application using an Apple Vision Pro. The subject is given a task to select nodes of a 3D graph visualization as efficiently as possible. The selection task comprises two parts, Ray-based selection and Gaze-based selection.
- **A3** An AR application that runs on an iPad Pro. The subject is asked to scan the surroundings of their physical scene and place two types of markers. A red marker on a flat surface, and a green marker on any physical object of their preference other than the flat surface.
- **A4** We designed an AR analytics application that run on iPad Pro, for co-located subjects. The subjects are informed to openly analyze the provided data, and was encouraged to collaborate and share opinions. Both verbal and non-verbal collaboration. The application included basic tools for the analytics such as 3D barchart visualization, bar value show, and a marker for storing an interest point in the data visualization.
- **A5** This is an AR Maintenance/Inspection application that is run on an iPad Pro. The subjects of the application is instructed to anchor an AR marker on the physical locations they find interesting, and record a voice memo of the reasoning behind their placements.

Note that the recordings of our prototype application sessions were conducted prior to the user evaluation with a different set of user groups. Refer to Supplementary Materials for further details on our prototype applications.

### 4.1 User Evaluation

We conduct user study to evaluate EXR. Participants were asked to complete tasks using our Visual Analytics Interface, which presents the pre-recorded data of our prototype XR sessions. Below, we describe the study participants, procedure, and tasks. We term the users of the evaluation study, '**participants**'.

**Study Design.** We recruited 14 participants (P1-P14; 8 males, 6 females) aged 22-36 years ($\mu$=27.2, $\sigma$=3.8), from diverse fields with expertise in HCI, Visualization, XR, Computer Vision, and Systems.

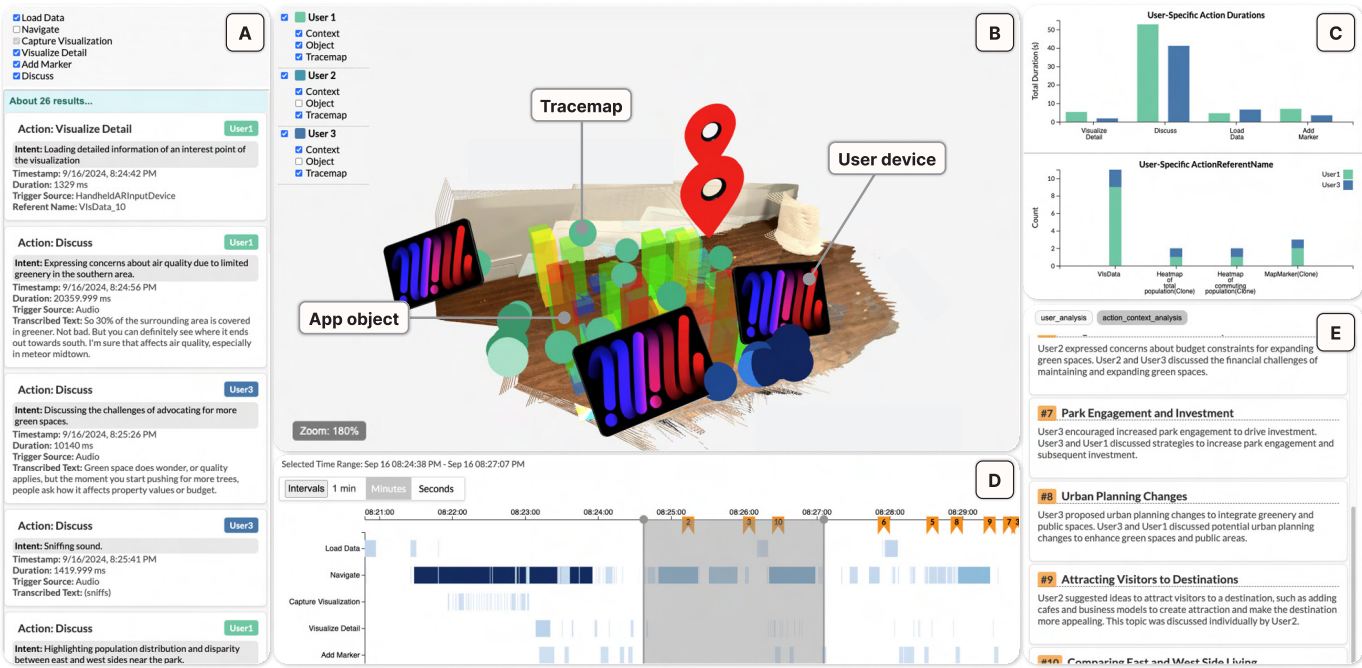


Fig. 7: Visual Analytics Interface: (A) The Data Manager includes Data Filter that filters the visualized action across the viewers, and Data Viewer which provides a way to examine the raw recorded action data of the selected time range. (B) Spatial Viewer visualizes the spatial properties of the subjects' actions, referent, the context point cloud, and action traces. The Trace Map can visualize the spatial patterns of actions (e.g., trajectory, locations of collaboration). (C) Plot Viewer visualizes the statistics of the subjects' interacted objects (Referent), and the duration of each action. (D) Temporal Viewer is the main control point to the information shown across the viewers. Users can select the sampling bin size of the time steps, or the time range, and interact with the AoI Marker to locate the interest points of their AoI. (E) Insight Viewer shows the LLM-generated insights facilitating the users' analytics and reasoning. The 'User device' in this example is an iPad used from A4.

The participants represent the broader user base of data analyst, referred as 'Researcher' in Fig. 2. Thus, we target the study to the participants with prior data analysis experience. The study involves participants using our Visual Analytics Interface (VAI), which displays pre-recorded data of XR sessions. For a balanced user experience assessment of VAI, we conduct the study on the participants across varying familiarity of a visual analytics interface ($\mu$=3.1, $\sigma$=1.2; 0=unfamiliar, 5=familiar). Before the session, participants were briefed on the functionalities of the analytics interface with a demo, and given 15 minutes to familiarize themselves. Each participant completed tasks using 5 interfaces, one for each XR prototype application, ensuring all tasks were completed by all participants. We collect their reasoning processes and usage patterns for each interface component through a task-by-task questionnaire. The session concluded with a semi-structured interview on the usability and reliability of our framework, followed by a 5-point Likert scale questionnaire for an overall assessment of EXR.

**Tasks.** Participants were given four tasks. For each task, we load the corresponding XR prototype application and configure the AoI prompt to align with the task's analysis goal (Fig. 8). Following are the tasks: **T1**, participants are instructed to use the interface to analyze action patterns of the subjects recorded in A1 and A2. **T2** focuses on analyzing the context behind the subjects' actions of our prototype app, A3. Participants are asked to describe the background scene at the time of the actions. For the third task, **T3**, participants are asked to summarize the actions of the subjects of A4, and report the gists of subjects' collaboration if it exists. We ask about verbal and non-verbal collaboration separately. Finally, in **T4**, participants are asked to derive the action intentions of the subjects in A5, to the best of their ability.

## 4.2 Task Results

We present the key findings from our study that demonstrate how EXR enables comprehensive analysis of user actions and intentions in XR environments based on the data collected from the participants.

**Action Pattern Finding.** All participants agreed ($\mu$**=4.1**) on the overall usefulness of our interface for pattern analysis (**T1**). One of the participants indicated the usability of Insight Viewer for pattern finding, "*The Insight viewer was incredibly helpful in judging the task performance time.*' (P4). Another participant was pleased to share their findings on the analysis of error rate for each metaphor, "*Gaze-based had more errors than the ray-based which seemed interesting to me*" (P11). We observe that the Plot Viewer and the Spatial Viewer, were the most commonly used to complete the task. Participants were able to pin-point the exact number and duration of interactions along with the subjects' action patterns "*Subject2 interacted with 262 cubes and 27 spheres through gaze and controller, which is more than Subject1*" (P14).

**Action Context Understanding.** The participants were overall satisfied with the the ability of the interface in visualizing the spatial context (**T2**), while displaying all the information on the subject's actions ($\mu$**=4.3**). A participant said "*I used the spatial viewer to visually get the hang of what happened and used the Plot Viewer to get the Referent Object name.*" (P2), indicating the joint use of the Spatial Viewer and Plot Viewer. Another participant used the combination of Spatial, Temporal, and Data Viewer. "*The user placed a marker at timestamp 8/20/2024, 4:48:26PM, near the Bean bag.*" (P1). Throughout the task, participants relied on the class labels of physical action referents, highlighting the importance of our post-hoc Action Referent Classifier spatial context reasoning. One user also emphasized the usefulness of spatial view filters: "*Context, object, and tracemap to see the exact way user mapped markers*" (P10).

**Action Summarization.** Participants correctly identified and summarized (**T3**) the collaboration between the two XR subjects, noting "*Users discussed planting trees, this was a collaborative discussion. They also discussed how wealthier people might populate places near the workplace*" (P2). One participant, in addition to successfully completing the task, also made an insightful observation " *Voice memo action is for marking warnings ,if you select the whole time, and see what the user is saying, it says..*" (P8). The study indicated that for this task, participants often selected the whole time range of a session using the Temporal and then use the Data Viewer to review the transcribed audio logs "*I selected the whole time range and then the Data Filter to focus on tasks and then the Data Viewer showed how many results there were*" (P7). A few participants (4) relied on the Insight Viewer

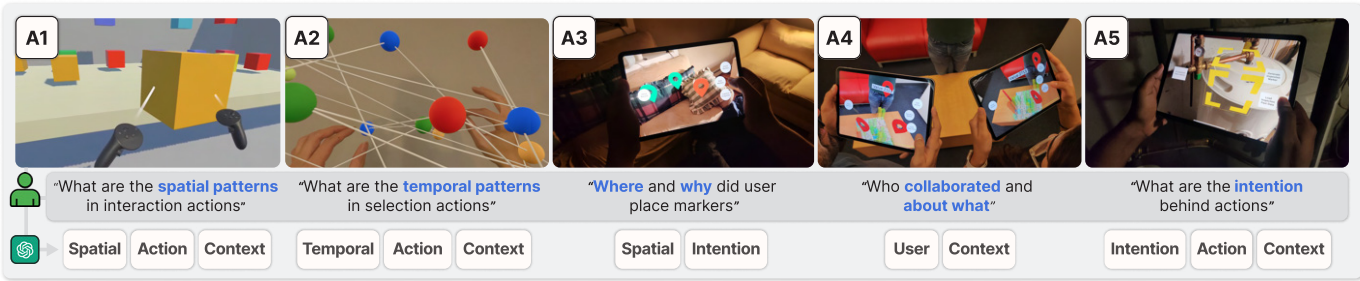


Fig. 8: General-purpose Analytics Framework: EXR can be utilized for diverse analytics tasks such as pattern finding, contextual visualization, and intention grasping, of the users of an XR session. It can also provide tailored key insights based on the analyst's AoI, after assigning the appropriate LLM agent of the AoI prompt. The multifaceted insights are generated by six LLM agents specialized in Spatial, Temporal, User, Action, Context, and Intent analyses. EXR is comprehensively assessed with five prototype apps with disparate theme: (A1) VR Game, (A2) MR Selection Techniques, (A3) AR 3D Scene Reconstruction, (A4) AR Collaborative analytics, (A5) AR Maintenance/Inspection.

for a summary of the session "*Insight Viewer: I used it to look at the collaboration and the summarization of what each user contributed to it.*"(P12), participants agreed on the utility of the interface for data summarized ($\mu$=**4.4**).

**Action Intention Inference.** The participants actively utilized the Insight Viewer to infer action intentions (**T4**). "*I used insight viewer to get hints of the session. It was telling me the user tended to inspect the findings and helped figure out the user's intention.*" (P11). Howevers, participants often chose to not solely rely on the Insight Viewer, a participant opted to use the Spatial and Data Viewer to infer the intention behind the subject's action on anchoring AR Sticky note (P9). Participants seem to verify the results of the Insight Viewer "*I like the summaries in the insight viewer...Then I checked this through other components and the summary is correct.*"(P5). Participants agreed on the usefulness of EXR for inferring intentions behind actions ($\mu$=**4.3**).

### 4.3 User Feedback

**Usability.** Participants rated our Visual Analytics Interface highly for ease of use ($\mu$=**4.5**), low learning curve ($\mu$=**4.5**), and overall usefulness ($\mu$=**4.6**) across the tasks. However, one participant expressed the difficulty in using the AoI Marker "*Maybe the marker should be sorted in chronological order*" (P13). We link each AoI Marker to one or more Analytics insights based on its relevancy to the insight. And the order is sorted based on the timestamp of the first Marker of every Insight. The participants reported that they all jointly utilized all the components of our analytics interface, and found them useful.

**Reliability.** We interviewed the participants on the output quality and usefulness of our LLM-generated Analytics Insights. All participants found the insights helpful for data interpretation and as a foundation for building base insights ($\mu$=**4.2**) "*Yes. I used it to get an idea of the discussions between users*" (P1), "*Overall, it is a good addition to the interface for analysis*" (P2), "*I like the summaries in the insight viewer, it helped me learn the users' actions and interactions between users.*" (P5). The majority did not identify any invalidity on the output of Insight viewer. One participant verified the insights using other viewers during the sessions. "*Then I checked this through other components and the summary is correct. I even didn't realize these summaries in the insight viewer were generated by AI, and I thought they were generated by human experts to help finish these tasks*" (P5). However, a failure case was noted by a participant on its accuracy for analyzing temporal patterns. "*I feel some of the maths are wrong*" (P13). Our analysis showed occasional inconsistencies in the agent's math computations, such as for the "Average task completion time" task. Overall, participants evaluated the usefulness of the Analytics insights highly, and indicated a strong likelihood of using LLM-assisted analytics insights again for data analysis ($\mu$=**4.0**).

### 4.4 System Evaluation

We evaluate the performance of the base setup of EXR, Action Recorder, *Log*, by measuring its overhead across various XR platforms and devices. To accuractely represent the overhead, all functions were converted to synchronous calls. As Tab. 3 indicates, the base Log function, which excludes *Context* and *Referent* data, has a negligible impact on XR applications (<**0.14 ms**). Even with Physical Context capture (Device camera snapshot) on an iPad Pro, application performance remains above **28.20ms** (∼**35 FPS**). For Virtual Context capture (In-application Unity camera snapshot), latency is maintained at **32.61ms** (∼**30.67 FPS**). Combining this with *Referent* storage increases latency to 101.44ms due to the complexity of storing material and geometric properties of a gameobject in GLB format. However, in practical scenarios, we implement asynchronous multi-threading, reducing perceived latency to ∼**1.13ms** and minimizing the impact on XR user experience. The overhead of context snapshot capture arises from GPU readback, texture encoding, and storage. Capturing an image at 1920x1080 resolution on an iPad introduces 143.94 ms of latency. Thus, we downsample the resolution by 75%, to 480x270, reducing the performance impact to 28.20ms (Log+PC). Downsampling is applied to both Physical and Virtual Scene snapshot capture of the *Context*.

We compare the performance of a recent work, ReLive, which is comparable to EXR in functionality – tracking user behavior data, action context (screenshots), action target referents, and supporting cross-platform environments. Under identical testing conditions (iPad Pro; averaged over 100 calls), EXR and ReLive reap similar overall performance. EXR outperforms ReLive by ∼**0.08ms** for base Log calls and achieves a negligible improvement of ∼**1.66ms** for Virtual Context capture (480x720). However, for logging with *Referent*, ReLive completes the task in 1.01ms, while EXR takes 101.44ms (or 1.13ms asynchronously), due to the GLB format conversion overhead compared to the OBJ format of ReLive.

We evaluate the usefulness of LLM-generated insights within VAI and justify our choice of multi-agent approach over a single-agent approach by comparing the output quality of the two methods. The multi-agent approach decomposes the XR interaction pattern analyses into smaller sub-tasks (Spatial, Temporal, Contextual, etc.), assigns each to a specialized agent, coordinates, and ensembles to derive the best result. In contrast, the single-agent approach performs all analyses at once without task distribution. To assess the output quality, we apply the concept of self-evaluating agent [20, 27, 38, 47]. We develop the evaluation metrics inspired by the SEVQ metrics of LIDA [20]. Our five criteria are : (**C1**) Relevance to the analysis goal, (**C2**) Compliance with the analyst's AoI prompt, (**C3**) Title representation – *"How well does the Title represent the insight?"*, (**C4**) Alignment with subject actions – *"How well does the insight represent the subject's actions?"*, and (**C5**) Overall diversity of insights – *"How many unique aspects are covered in an insight?"*.

Our results (Tab. 4) show that our framework generates insights well-aligned with the analyst's analytics needs ($\mu$=**9.04** out of 10; Mean of C1-C4) and provides multifaceted perspectives ($\mu$=**8.90** out of 10; C5). In all criteria, the multi-agent approach outperforms the single-agent approach, except in Criteria-3 (C3). Our evaluation reveals that single-agent tends to produce a more generic descriptions of insights, leading to higher score in generalizability (C3). However, EXR prioritizes constructive and specific insights, making abstract descriptions less ideal for our framework. A qualitative comparison underscores this difference: the single-agent produces outputs such as *("Inspection*

*Log Entries", "User1 left inspection logs for various objects")*, while multi-agent offers more detailed insights such as *("Interaction with Objects and Inspection Logs", "User1 interacted with objects including QR codes,..")*. Underlined terms highlight the generic nature of single-agent outputs.

## 5 Limitation and Discussion

**Reliance on Verbal Input for User Behavior Analysis.** EXR leverages multimodal data streams and embodied interactions to comprehensively analyze user behaviors. The UAD captures both predefined actions (e.g., Pinch, GazeAt) and Post Defined actions (e.g., Speak) in a structured format. LLMs, then, contextually bridge the stream of data, and identify patterns. Most importantly, EXR can generate meaningful insights even without Post Defined actions such as verbal input. For instance, in A1, EXR successfully analyzed user attention and spatial interactions using only gaze, gestures, and spatiotemporal data. Reliance on verbal input becomes a limitation only when the AoI of an analyst is narrowly centered on the verbal channel, such as analyzing "*Topics of discussion between users.*" In such cases, while our visual analytics interface remains functional, LLM-assisted insights will rely solely on non-verbal data such as interacted referents, spatiotemporal patterns, and visual feeds, potentially lacking crucial contextual cues. To address this, we plan to integrate additional modalities – physiological signals (e.g., EEG, heart rate) – to enhance the robustness of EXR in communication-agnostic scenarios.

**Privacy Measures on XR Device Camera Captures.** In recent XR platforms, such as Vision OS and Meta Horizon OS, third-party apps are restricted from directly accessing device camera sensors by the OS [1, 32, 42, 43, 65]. Since we rely on camera snapshots to reconstruct 3D context point clouds, we resort to an in-application (Unity) camera snapshot to store the captures in these cases. To further address this, we plan to extend our framework to store permitted 3D scene meshes provided by the OS or middleware (e.g., ARKit).

**Data Recording Overhead.** We identify two sources of potential performance degradation in our Action Recorder. First, exporting a referent object involves an asynchronous GLB conversion that incurs a latency of 101.44ms. To mitigate this, we plan to perform the conversion at the end of the XR session. Alternatively, we could include an option to store referents in OBJ format, which requires less conversion overhead than GLB. While the OBJ format lacks support for PBR materials, animations, and complex hierarchical structures, the addition of an option could be beneficial for applications with performance priority. The second is due to the context snapshot capture. While we reduce the overhead by lowering the capture resolution, this degrades the quality of context point clouds. Thus, we plan to introduce asynchronous readback to preserve point cloud quality while balancing performance.

**Offline Analytics Interface.** While EXR can capture and analyze diverse XR environments, it is currently limited to offline, retrospective visualization of sessions. As shown in Fig. 2, our framework involves multiple processing steps including physical entity classification, context point cloud generation, and analytics insight generation, making real-time visualization challenging. In our following version of EXR, we plan to support an online visualizer that can optionally stream the session data without requiring post-processing steps.

**Room for Human Error in Analysis Interest Prompting.** As demonstrated through the diverse tasks of our prototype applications and evaluation, EXR can perform as a general-purpose analytics framework. Analysts can input any AoI prompt in a plain English and visualize tailored analytics insights, which help them establish base insights and preliminary hypotheses. However, we observed that our Analytics Assistant (Insight generator) fails to pin-point the useful patterns when an ambiguous AoI prompt such as "*Tell me user actions*", is given from the analyst. It simply returned the list of referents with varying names: "*Interaction with Cube1, Cube45, Cube4*". A prompt that is more task-specific such as "*Users' interacted objects (properties of the objects; e.g., color, shape) across actions and their patterns*" can output more meaningful results. We do not expect the domain researchers to prompt engineer the AoI. For our future work, we plan to integrate an agent that guides the researcher to input a structured prompt for a descriptive AoI, mitigating the possibilities of vaguely defined outputs.

Table 3: Average overhead comparison of Log function in Action Recorder across devices and configurations. The Quest and Vision Pro do not support capture through physical device camera. Measurements, recorded in milliseconds, represent averages across 100 calls from 10 independent application runs. (Log: Base logging without referent or context storage; PC: Physical Context snapshot; VC: Virtual Context snapshot; R: Referent object save)

| Device | Log | Log+PC | Log+VC | Log+R |
|---|---|---|---|---|
| iPad Pro | 0.08 | 28.20 | 32.61 | 101.44 |
| Meta Quest Pro | 0.13 | - | 35.04 | 82.54 |
| Apple Vision Pro | 0.14 | - | 23.64 | 72.31 |

Table 4: Average scores per criterion for different Analytics insight generation methods, measured on a scale of 0-10. Each criterion score represents an average across 10 independent runs. (C1: Relevance to type of analysis; C2: Compliance to user's analysis-of-interest; C3: Alignment of insight to the Title; C4: Alignment of insight to action; C5: Overall diversity of insights)

| Method | C1 | C2 | C3 | C4 | C5 |
|---|---|---|---|---|---|
| Single-agent | 8.20 | 7.64 | 9.38 | 8.72 | 8.00 |
| Multi-agent **(Ours)** | 8.73 | 8.78 | 9.29 | 9.35 | 8.90 |

**Incorrectness in Data Extrapolation.** We leverage LLM to deduce the intention of subjects' actions, and provide useful insights beyond the original recorded data: "*This repetitive action suggests a detailed examination of the environment*" (LLM output), or to infer the collaboration between users: "*This topic was a major focus, with User2 and User3 collaborating closely on it*" (LLM output). However, we identified that these extrapolation can suggest insights that are inaccurately derived, due to the hallucination of the LLM [22, 46, 82, 83]. As one of the solutions, we expect to enhance the correctness of the output by adopting the concept of multi-agent debate and self-correction [24, 30, 36] for each agent. In addition, we plan to guide the agents to output the confidence scores of each analysis so that the researchers can assess the credibility of the insights, themselves. We also believe that this problem will be further mitigated with the advancement of reasoning ability of LLM [60, 61].

## 6 Data Privacy and Ethics

This research was conducted under IRB of the Office of Research Compliance at Stony Brook University (1173920_*MODCR*005). All subjects provided informed consent prior to participation. To ensure privacy, data was anonymized with aliases in the *User* field of the UAD, and audio recordings were transcribed to text. For our analysis, only textual interaction data was used. Captured snapshots were processed to exclude any identifiable features such as faces and name tags. No collected XR session data will be publicly released, beyond what is shared in this manuscript, to further protect subject privacy.

## 7 Conclusion

In this paper, we presented Explainable XR, an end-to-end framework capable of recording, processing, and visualizing user behaviors across virtualities in diverse XR settings. Central to our approach is the User Action Descriptor, a novel format designed to integrate multimodal behavior data and action context while ensuring consistency across virtualities and multi-user scenarios. We showcased the applicability of UAD in capturing immersive sessions and showcased the usability of our LLM-assisted visual interface, which enriches the analytics experience with intelligent insights and tailored visual guidance. Through five practical XR applications, we presented a comprehensive evaluation of EXR. We envision EXR as a tool for understanding XR user experiences, enabling researchers from various disciplines to adopt XR into their studies.

## ACKNOWLEDGMENTS

We express our gratitude to Hyunji Yoon for the art work, and Matthew Castellana for video editing and narration. Also, we thank Muhammad Farrukh, Sumeer Ahmad, the anonymous reviewers, and the study participants for their valuable feedback and contributions that made this work possible. This research was supported in part by NSF award IIS2107224 and ONR award N000142312124.

## REFERENCES

[1] Apple. Developer. https://developer.apple.com/videos/play/wwdc2024/10139/, 2024. Sep. 14. 2024. 9

[2] M. Arnold, M. Goldschmitt, and T. Rigotti. Dealing with information overload: a comprehensive review. *Frontiers in Psychology*, 14:1122200, 2023. 5

[3] D. Bohus, S. Andrist, N. Saw, A. Paradiso, I. Chakraborty, and M. Rad. Sigma: An open-source interactive system for mixed-reality task assistance research–extended abstract. In *Prof. of VRW*, pp. 889–890, 2024. 2

[4] S. Boorboor, M. S. Castellana, Y. Kim, C. Zhu-Tian, J. Beyer, H. Pfister, and A. E. Kaufman. VoxAR: adaptive visualization of volume rendered objects in optical see-through augmented reality. *IEEE TVCG*, 30(10):6801–6812, 2023. 1

[5] S. Boorboor, Y. Kim, P. Hu, J. M. Moses, B. A. Colle, and A. E. Kaufman. Submerse: Visualizing storm surge flooding simulations in immersive display ecologies. *IEEE TVCG*, 30(09):6365–6377, 2024. 1

[6] K. Brandstätter and A. Steed. Dialogues for one: Single-user content creation using immersive record and replay. In *Proc. of VRST*, pp. 1–11, 2023. 2

[7] J. Brookes, M. Warburton, M. Alghadier, M. Mon-Williams, and F. Mushtaq. Studying human behaviour with virtual reality: The unity experiment framework. *bioRxiv*, 2018. 2, 3

[8] F. Brudy, S. Suwanwatcharachat, W. Zhang, S. Houben, and N. Marquardt. Eagleview: A video analysis tool for visualising and querying spatial interactions of people and devices. In *Proc. of ISS*, pp. 61–72, 2018. 2, 3

[9] W. Büschel, A. Lehmann, and R. Dachselt. Miria: A mixed reality toolkit for the in-situ visualization and analysis of spatio-temporal interaction data. In *Proc. of CHI*, pp. 1–15, 2021. 2

[10] Y. Cao, Y. Lan, F. Zhai, and P. Li. 5w1h extraction with large language models. *arXiv preprint arXiv:2405.16150*, 2024. 4

[11] S. Castelo, J. Rulff, E. McGowan, B. Steers, G. Wu, S. Chen, I. Roman, R. Lopez, E. Brewer, C. Zhao, et al. ARGUS: Visualization of AI-assisted task guidance in AR. *IEEE TVCG*, 2023. 1, 2, 3

[12] S. Castelo, J. Rulff, P. Solunke, E. McGowan, G. Wu, I. Roman, R. Lopez, B. Steers, Q. Sun, J. Bello, et al. Hubar: A visual analytics tool to explore human behavior based on fnirs in ar guidance systems. *IEEE TVCG*, 2024. 3

[13] K. Choe, C. Lee, S. Lee, J. Song, A. Cho, N. W. Kim, and J. Seo. Enhancing data literacy on-demand: Llms as guides for novices in chart interpretation. *IEEE TVCG*, 2024. 2

[14] Cognitive3D. Cognitive3d. https://cognitive3d.com/product/objectives/. Mar. 26. 2024. 2, 3

[15] R. Cools, X. Zhang, and A. L. Simeone. Crest: Design and evaluation of the cross-reality study tool. In *Proc. of MUM*, pp. 409–419, 2023. 2

[16] W. Cui, X. Zhang, Y. Wang, H. Huang, B. Chen, L. Fang, H. Zhang, J. Lou, and D. Zhang. Text-to-viz: Automatic generation of infographics from proportion-related natural language statements. *IEEE TVCG*, 26(01):906–916, 2020. 3

[17] B. David-John, C. Peacock, T. Zhang, T. S. Murdison, H. Benko, and T. R. Jonker. Towards gaze-based prediction of the intent to interact in virtual reality. In *Proc. of ETRA*, pp. 1–7, 2021. 2

[18] F. De La Torre, C. M. Fang, H. Huang, A. Banburski-Fahey, J. Amores Fernandez, and J. Lanier. Llmr: Real-time prompting of interactive worlds using large language models. In *Proc. of CHI*, pp. 1–22, 2024. 3

[19] J. Deuchler, W. Hettmann, D. Hepperle, and M. Wölfel. Streamlining physiological observations in immersive virtual reality studies with the virtual reality scientific toolkit. In *Proc. of VRW*, pp. 485–488, 2023. 2

[20] V. Dibia. Lida: A tool for automatic generation of grammar-agnostic visualizations and infographics using large language models. *arXiv preprint arXiv:2303.02927*, 2023. 2, 8

[21] M. D. Dogan, E. J. Gonzalez, K. Ahuja, R. Du, A. Colaço, J. Lee, M. Gonzalez-Franco, and D. Kim. Augmented object intelligence with xr-objects. In *Proc. of UIST*, pp. 1–15, 2024. 3

[22] H. Duan, Y. Yang, and K. Y. Tam. Do llms know about hallucination? an empirical investigation of llm's hidden states. *arXiv preprint arXiv:2402.09733*, 2024. 9

[23] D. Enriquez, W. Tong, C. North, H. Qu, and Y. Yang. Evaluating layout dimensionalities in pc+ vr asymmetric collaborative decision making. In *Proc. of ISS*, pp. 112–132, 2024. 1

[24] P. Feldman, J. R. Foulds, and S. Pan. Trapping llm hallucinations using tagged context prompts. *arXiv preprint arXiv:2306.06085*, 2023. 9

[25] D. Gasques, J. G. Johnson, T. Sharkey, Y. Feng, R. Wang, Z. R. Xu, E. Zavala, Y. Zhang, W. Xie, X. Zhang, et al. Artemis: A collaborative mixed-reality system for immersive surgical telementoring. In *Proc. of CHI*, pp. 1–14, 2021. 1

[26] G. Gorisse, O. Christmann, and C. Dubosc. Rec: A unity tool to replay, export and capture tracked movements for 3d and virtual reality applications. In *Proc. of AVI*, pp. 1–3, 2022. 2

[27] Z. Guo, R. Jin, C. Liu, Y. Huang, D. Shi, L. Yu, Y. Liu, J. Li, B. Xiong, D. Xiong, et al. Evaluating large language models: A comprehensive survey. *arXiv preprint arXiv:2310.19736*, 2023. 8

[28] P. Hu, S. Boorboor, S. Jadhav, J. Marino, S. Mirhosseini, and A. E. Kaufman. Spatial perception in immersive visualization: A study and findings. In *Proc. of ISMAR-Adjunct*, pp. 369–372, 2022. 1

[29] P. Hu, Q. Sun, P. Didyk, L.-Y. Wei, and A. E. Kaufman. Reducing simulator sickness with perceptual camera control. *ACM ToG*, 38(6):1–12, 2019. 1

[30] H. Huang, Z. Lin, Z. Wang, X. Chen, K. Ding, and J. Zhao. Towards llm-powered verilog rtl assistant: Self-verification and self-correction. *arXiv preprint arXiv:2406.00115*, 2024. 9

[31] S. Hubenschmid, J. Wieland, D. I. Fink, A. Batch, J. Zagermann, N. Elmqvist, and H. Reiterer. Relive: Bridging in-situ and ex-situ visual analytics for analyzing mixed reality user studies. In *Proc. of CHI*, pp. 1–20, 2022. 1, 2, 3

[32] S. Jana, D. Molnar, A. Moshchuk, A. Dunn, B. Livshits, H. J. Wang, and E. Ofek. Enabling {Fine-Grained} permissions for augmented reality applications with recognizers. In *Proc. of USENIX Security*, pp. 415–430, 2013. 9

[33] S. Jang, E.-J. Ko, and W. Woo. Unified user-centric context: Who, where, when, what, how and why. *Proc. of ubiPCMM*, 149, 01 2005. 4

[34] P. Jansen, J. Britten, A. Häusele, T. Segschneider, M. Colley, and E. Rukzio. Autovis: Enabling mixed-immersive analysis of automotive user interface interaction studies. In *Proc. of CHI*, pp. 1–23, 2023. 2

[35] C. Javerliat, S. Villenave, P. Raimbaud, and G. Lavoué. Plume: Record, replay, analyze and share user behavior in 6dof xr experiences. *IEEE TVCG*, 2024. 1, 2, 3

[36] Z. Ji, T. Yu, Y. Xu, N. Lee, E. Ishii, and P. Fung. Towards mitigating hallucination in large language models via self-reflection. *arXiv preprint arXiv:2310.06271*, 2023. 9

[37] Q. Jin, Y. Liu, S. Yarosh, B. Han, and F. Qian. How will vr enter university classrooms? multi-stakeholders investigation of vr in higher education. In *Proc. of CHI*, pp. 1–17, 2022. 1

[38] S. Kadavath, T. Conerly, A. Askell, T. Henighan, D. Drain, E. Perez, N. Schiefer, Z. Dodds, N. DasSarma, E. Tran-Johnson, S. Johnston, S. El-Showk, A. Jones, N. Elhage, T. Hume, A. Chen, Y. Bai, S. Bowman, S. Fort, and J. Kaplan. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 07 2022. 8

[39] S. Kasahara, V. Heun, A. S. Lee, and H. Ishii. Second surface: multi-user spatial collaboration system based on augmented reality. In *Proc. of SIGGRAPH Asia*, pp. 1–4, 2012. 1

[40] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM ToG*, 42(4):139–1, 2023. 4

[41] A. Khurana, M. Glueck, and P. K. Chilana. Do I Just Tap My Headset? How Novice Users Discover Gestural Interactions with Consumer Augmented Reality Applications. *Proc. of IMWUT*, 7(4):1–28, 2024. 1

[42] Y. Kim, S. Boorboor, A. Rahmati, and A. Kaufman. Design of privacy preservation system in augmented reality. In *Proc. of VizSec*, 2021. 9

[43] Y. Kim, S. Goutam, A. Rahmati, and A. Kaufman. Erebus: Access control for augmented reality systems. In *Proc. of USENIX Security*, pp. 929–946, 2023. 9

[44] B. Lee, X. Hu, M. Cordeil, A. Prouzeau, B. Jenny, and T. Dwyer. Shared surfaces and spaces: Collaborative data visualisation in a co-located immersive environment. *IEEE TVCG*, 27(2):1171–1181, 2020. 1

[45] Y. Lee, B. Yoo, and S.-H. Lee. Sharing ambient objects using real-time point cloud streaming in web-based xr remote collaboration. In *Proc. of*

*Web3D*, pp. 1–9, 2021. 4

[46] Y. Li, Y. Du, K. Zhou, J. Wang, W. X. Zhao, and J.-R. Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023. 9

[47] S. Lin, J. Hilton, and O. Evans. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*, 2022. 8

[48] S. Lubos, T. N. T. Tran, A. Felfernig, S. Polat Erdeniz, and V.-M. Le. Llm-generated explanations for recommender systems. In *Proc. of UMAP Adjunct*, pp. 276–285, 2024. 3

[49] W. Luo, Z. Yu, R. Rzayev, M. Satkowski, S. Gumhold, M. McGinity, and R. Dachselt. Pearl: Physical environment based augmented reality lenses for in-situ human movement analysis. In *Proc. of CHI*, pp. 1–15, 2023. 2

[50] P. Ma, R. Ding, S. Wang, S. Han, and D. Zhang. Insightpilot: An llm-empowered automated data exploration system. In *Proc. of EMNLP*, pp. 346–352, 2023. 2

[51] M. N. Mahdi, A. R. Ahmad, R. Ismail, M. A. Subhi, M. M. Abdulrazzaq, and Q. S. Qassim. Information overload: the effects of large amounts of information. In *Proc. of IT-ELA*, pp. 154–159, 2020. 5

[52] E. S. Martinez, A. A. Malik, and R. P. McMahan. Clovr: Collecting and logging openvr data from steamvr applications. In *Proc. of VRW*, pp. 485–492, 2024. 2

[53] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Proc. of ECCV*, 65(1):99–106, 2021. 4

[54] S. Mirhosseini, P. Ghahremani, S. Ojal, J. Marino, and A. Kaufman. Exploration of large omnidirectional images in immersive environments. In *Proc. of VR*, pp. 413–422, 2019. 1

[55] S. Mirhosseini, I. Gutenko, S. Ojal, J. Marino, and A. Kaufman. Immersive virtual colonoscopy. *IEEE TVCG*, 25(5):2011–2021, 2019. 1

[56] V. Nair, W. Guo, J. Mattern, R. Wang, J. F. O'Brien, L. Rosenberg, and D. Song. Unique Identification of 50,000+ Virtual Reality Users from Head & Hand Motion Data. In *Proc. of USENIX Security*, pp. 895–910, 2023. 2

[57] D. Nam, A. Macvean, V. Hellendoorn, B. Vasilescu, and B. Myers. Using an llm to help with code understanding. In *Proc. of ICSE*, pp. 1–13, 2024. 2

[58] M. Nebeling, M. Speicher, X. Wang, S. Rajaram, B. D. Hall, Z. Xie, A. R. Raistrick, M. Aebersold, E. G. Happ, J. Wang, et al. Mrat: The mixed reality analytics toolkit. In *Proc. of CHI*, pp. 1–12, 2020. 1, 2, 3, 5

[59] N. Numan and A. Steed. Exploring user behaviour in asymmetric collaborative mixed reality. In *Proc. of VRST*, pp. 1–11, 2022. 2

[60] OpenAI. Learning to reason with llms. https://openai.com/index/learning-to-reason-with-llms/, 2024. Sep. 14. 2024. 9

[61] OpenAI. Openai o1 system card. https://assets.ctfassets.net/kftzwdyauwt9/67qJD51Aur3eIc96iOfeOP/71551c3d223cd97e591aa89567306912/o1_system_card.pdf, 2024. Sep. 14. 2024. 9

[62] T. Piumsomboon, Y. Lee, G. Lee, and M. Billinghurst. Covar: a collaborative virtual and augmented reality system for remote collaboration. In *Proc. of SIGGRAPH Asia*, pp. 1–2. 2017. 1

[63] H. Qu, Y. Cai, and J. Liu. Llms are good action recognizers. In *Proc. of CVPR*, pp. 18395–18406, 2024. 2

[64] F. Robert, H.-Y. Wu, L. Sassatelli, S. Ramanoel, A. Gros, and M. Winckler. An integrated framework for understanding multimodal embodied experiences in interactive virtual reality. In *Proc. of IMX*, pp. 14–26, 2023. 2

[65] F. Roesner, D. Molnar, A. Moshchuk, T. Kohno, and H. J. Wang. World-driven access control for continuous sensing. In *Proc. of CCS*, pp. 1169–1181, 2014. 9

[66] D. Romero, R. J. Patel, A. Markopolou, and S. Elmalaki. GaitGuard: Towards Private Gait in Mixed Reality. *arXiv preprint arXiv:2312.04470*, 2023. 2

[67] D. Saffo, S. Di Bartolomeo, C. Yildirim, and C. Dunne. Remote and collaborative virtual reality experiments via social vr platforms. In *Proc. of CHI*, pp. 1–15, 2021. 1

[68] L. Shen, H. Li, Y. Wang, and H. Qu. From data to story: Towards automatic animated data video creation with llm-based multi-agent systems. *IEEE TVCG*, 2024. 3

[69] L. Shen, E. Shen, Y. Luo, X. Yang, X. Hu, X. Zhang, Z. Tai, and J. Wang. Towards natural language interfaces for data visualization: A survey. *IEEE TVCG*, 29(6):3121–3144, 2022. 2

[70] C. Slocum, Y. Zhang, N. Abu-Ghazaleh, and J. Chen. Going through the motions:ar/vr keylogging from user head motions. In *Proc. of USENIX Security*, pp. 159–174, 2023. 2

[71] A. Steed, L. Izzouzi, K. Brandstätter, S. Friston, B. Congdon, O. Olkkonen, D. Giunchi, N. Numan, and D. Swapp. Ubiq-exp: A toolkit to build and run remote and distributed mixed reality experiments. *Frontiers in VR*, 3:912078, 2022. 2

[72] N. Sultanum, M. Brudno, D. Wigdor, and F. Chevalier. More text please! understanding and supporting the use of visualization for clinical text overview. In *Proc. of CHI*, pp. 1–13, 2018. 5

[73] Q. Sun, A. Patney, L.-Y. Wei, O. Shapira, J. Lu, P. Asente, S. Zhu, M. McGuire, D. Luebke, and A. Kaufman. Towards virtual reality infinite walking: dynamic saccadic redirection. *ACM ToG*, 37(4):1–13, 2018. 1

[74] H. Tian, G. A. Lee, H. Bai, and M. Billinghurst. Using virtual replicas to improve mixed reality remote collaboration. *IEEE TVCG*, 29(5):2785–2795, 2023. 1, 4

[75] Unity. Input system. https://docs.unity3d.com/Packages/com.unity.inputsystem@1.10/manual/index.html, 2024. Aug. 27. 2024. 4

[76] Unity. Unity engine. https://unity.com/products/unity-engine, 2024. Aug 31. 2024. 3

[77] S. Villenave, J. Cabezas, P. Baert, F. Dupont, and G. Lavoué. Xrecho: A unity plug-in to record and visualize user behavior during xr sessions. In *Proc. of MMSys*, pp. 341–346, 2022. 2

[78] C. Y. Wang, D. Saffo, B. Moriarty, and B. Maclntyre. Collabxr: Bridging realities in collaborative workspaces with dynamic plugin and collaborative tools integration. In *IEEE VRW*, pp. 454–457, 2024. 1

[79] E. Wen, T. I. Kaluarachchi, S. Siriwardhana, V. Tang, M. Billinghurst, R. W. Lindeman, R. Yao, J. Lin, and S. Nanayakkara. Vrhook: A data collection tool for vr motion sickness research. In *Proc. of UIST*, pp. 1–9, 2022. 2

[80] D. Wolf, J. J. Dudley, and P. O. Kristensson. Performance envelopes of in-air direct and smartwatch indirect control for head-mounted augmented reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 347–354, 2018. 1

[81] A. Wu, Y. Wang, X. Shu, D. Moritz, W. Cui, H. Zhang, D. Zhang, and H. Qu. Ai4vis: Survey on artificial intelligence approaches for data visualization. *IEEE TVCG*, 28(12):5049–5070, 2021. 2, 5

[82] Z. Xu, S. Jain, and M. Kankanhalli. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*, 2024. 9

[83] J.-Y. Yao, K.-P. Ning, Z.-H. Liu, M.-N. Ning, and L. Yuan. Llm lies: Hallucinations are not bugs, but features as adversarial examples. *arXiv preprint arXiv:2310.01469*, 2023. 9

[84] K. Yu, U. Eck, F. Pankratz, M. Lazarovici, D. Wilhelm, and N. Navab. Duplicated reality for co-located augmented reality collaboration. *IEEE TVCG*, 28(5):2190–2200, 2022. 1

[85] Y. Yu and Y. Bi. A study on "5w1h" user analysis on interaction design of interface. In *Proc. of CAIDCD*, vol. 1, pp. 329–332, 2010. 4

[86] Z. Yu, D. Zeidler, V. Victor, and M. Mcginity. Dynascape: Immersive authoring of real-world dynamic scenes with spatially tracked rgb-d videos. In *Proc. of VRST*, pp. 1–12, 2023. 2

[87] P. Zhang, C. Li, and C. Wang. Viscode: Embedding information in visualization images using encoder-decoder network. *IEEE TVCG*, 27(2):326–336, 2020. 5

[88] Y. Zhang, C. Slocum, J. Chen, and N. Abu-Ghazaleh. It's all in your head (set): Side-channel attacks on ar/vr systems. In *Proc. of USENIX Security*, pp. 3979–3996, 2023. 2

[89] Y. Zhao, Y. Zhang, Y. Zhang, X. Zhao, J. Wang, Z. Shao, C. Turkay, and S. Chen. Leva: Using large language models to enhance visual analytics. *IEEE TVCG*, 2024. 2