Integrating Large Language Models into Extended Reality: A Literature Review

Timo Haubner
Technical University of Munich
Munich, Germany
timo.haubner@tum.de

***Abstract*—Both Extended Reality (XR) and Large Language Models (LLMs) have seen significant advancements in recent years and are now widely adopted across various domains. Integrating LLMs into XR leverages their capabilities in natural language processing and reasoning to enable more intuitive interaction, enhance user immersion, and improve accessibility. To provide an overview of this emerging field, this review analyzes a systematic selection of 36 studies published since 2023 in ACM and IEEE venues. These works are categorized into three thematic clusters: LLMs for Generative Content Creation and Manipulation, Embodied LLM Agents, and LLM-powered Assistive Systems. Within each cluster, key approaches, technologies, and results are examined and compared. Building on this analysis, the review identifies cross-cutting challenges related to latency, spatial understanding, evaluation practices, and concerns related to privacy and ethics. For each of these areas, specific directions for future research are proposed.**

I. Introduction

Extended Reality (XR) has seen rapid advancements in recent years, driven by progress in hardware capabilities and content development. However, most systems still face challenges in achieving natural and intuitive interaction between the user and digital environments, limiting both immersion and engagement. The rise of Large Language Models (LLMs) in the last years offers a promising solution to bridge this gap. With their powerful natural language processing and reasoning skills, these models enable more flexible, human-like communication and interaction. From conversations with lifelike avatars to the creation of whole scenes using only words, the integration of LLMs has the potential to reshape the way users interact with digital content in XR.

As in many previous studies, Extended Reality is used here as an umbrella term for Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality (MR) [1]. This review adopts a holistic approach and therefore often refers to XR as a whole. However, it is sometimes necessary to differentiate between the individual subcategories. Virtual Reality describes a technique in which the user is fully immersed in a synthetic, computer-generated environment. In this case, the physical world is entirely replaced by a virtual environment that can replicate the properties of the real world or present fictional scenarios [2]. This is typically achieved through the use of virtual reality headsets, a type of head-mounted display [3]. Common applications include gaming (e.g., VRChat [4]), simulation (e.g., realistic pilot training scenarios), and engineering (e.g., molecular visualization) [5].

In Augmented Reality, the user's physical environment is not replaced but rather enhanced with digitally created virtual elements [2]. By overlaying images, sounds, or graphics onto the real world, a new interactive environment is created. Various devices such as smartphones or AR glasses can be used for this purpose. There are widespread use cases for Augmented Reality such as education, shopping, interior design, and gaming (e.g., Pokémon Go) [5].

For the last subcategory, Mixed Reality, there are multiple ways to define the concept in literature with no unified clear meaning. Originally, the term was used to describe the entire continuum between virtuality and reality [2], but it has also been used simply as a synonym for AR [6]. Therefore, this paper will refrain from using the term where possible.

Large Language Models are pre-trained neural networks designed for natural language processing (NLP) tasks. They are based on the transformer architecture [7] and operate at large scale, utilizing tens to hundreds of billions of parameters. For this, LLMs are trained on vast amounts of text, usually scraped from the internet. By doing so they gain the ability to understand, generate and interact with natural language. With their understanding of language, LLMs also inherit strong skills in reasoning and learning, making them widely applicable [8].

Since 2022, the research on LLMs has seen significant advances, partially driven by the launch of the powerful LLM-based ChatGPT, which has drawn significant academic and public attention [9]. While general-purpose chatbots like ChatGPT, Claude, or DeepSeek remain the most visible examples, LLMs are now employed across a wide range of domains, including code generation for software development, legal document drafting, medical question answering, and language translation [10].

The usage of LLMs in Extended Reality is a novel and rapidly evolving field with a wide range of applications and emerging research directions. However, at the moment achieving a comprehensive overview or comparing different approaches remains challenging as most literature reviews either focus more broadly on Artificial Intelligence in XR or analyze LLMs and XR as separate topics (section II). Therefore, this review aims to provide a holistic perspective on the integration of Large Language Models into Extended Reality, the technologies employed in this context and the associated challenges. For this purpose, the existing literature

was analyzed, and 36 relevant studies were identified, as described in section III. These papers were then categorized in three separate clusters: LLMs for Generative Content Creation and Manipulation (section IV-A), Embodied LLM Agents (section IV-B) and LLM-powered Assistive Systems (section IV-C). The respective sections summarize and compare the various approaches and studies. Finally, the discussion in section V summarizes the findings from the review, highlights challenges and proposes future research directions.

In summary, the main contributions of this review are:

- The structured categorization of existing work into three topic clusters: LLMs for Generative Content Creation and Manipulation, Embodied LLM Agents and LLM-powered Assistive Systems.
- A comparative analysis of applications, technical implementations and evaluation approaches within each cluster.
- The identification of benefits using LLMs in XR, as well as common challenges and limitations currently faced by research in this area.
- The derivation of further research directions considered essential for the successful integration of LLMs into XR systems.

II. Related Work

Numerous studies explore the development of Extended Reality in general. In [11], advances in mixed reality research are analyzed, while Wang et al. [12] provide an overview of recent developments in multimodal interaction with XR headsets. Both works also touch upon the use of language models in this context. One area that overlaps with LLM integration in XR is the application of LLMs to 3D tasks and spatial understanding. Ma et al. [13] and Zha et al. [14] offer comprehensive surveys on spatial reasoning with LLMs. While LLMs are employed in XR for similar purposes, their application in this domain extends beyond spatial reasoning alone.

Literature reviews that focus explicitly on the intersection of LLMs and XR remain rare. Several broader surveys examine the use of Artificial Intelligence in XR [15, 16], covering techniques such as Generative Adversarial Networks (GANs) and reinforcement learning, without a specific focus on language models. Other works explore domain-specific applications of LLMs in XR. For instance, Brito et al. [17] evaluate their use in digital humans for VR applications. So far, the survey by Tang et al. [18] appears to be the only comprehensive review with a similar objective to this work. However, the papers included in their review span from 2021 to July 2024. As noted by them, the field is rapidly evolving, making it important to continuously track new developments. With two-thirds of the analyzed papers published since the third quarter of 2024 (see Fig. 1), the present review therefore provides a valuable and timely addition. Furthermore, in contrast to many prior studies, this work provides an in-depth analysis of technical architectures and implementation strategies for integrating LLMs into XR systems.

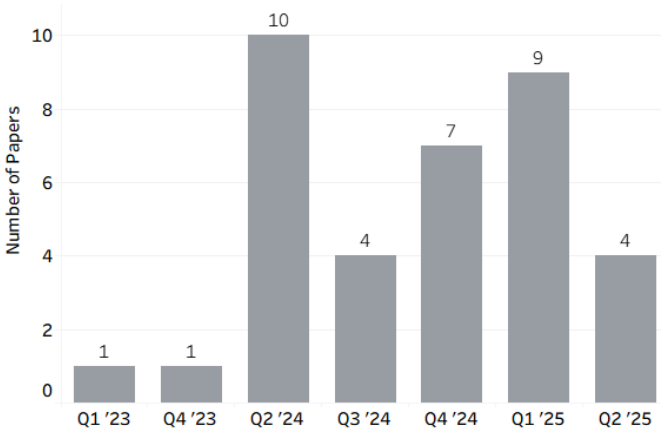

Fig. 1. Number of reviewed papers per quarter of publication.

III. Methodology

The analysis of a base paper [19] served as a starting point for the literature review. Furthermore, studies published from 2023 onward were considered, since the release of ChatGPT in late 2022 has triggered a rapid surge in research activity in this area. Prior to 2023, only a limited number of works addressed the integration of LLMs into XR. To identify relevant publications, searches were conducted using Google Scholar, IEEE Xplore, and the ACM Digital Library. The review is restricted to papers published at different ACM and IEEE venues, including IEEE VR, ACM CHI, ACM UIST, IEEE VRW, IEEE ISMAR, IEEE AIVR, ACM SIGGRAPH and ACM IUI (ranked in descending order according to the number of included papers). Consequently, all analyzed studies were published and written in English.

The main query "("Large Language Models" OR LLMs) AND ("Extended Reality" OR XR OR "Virtual Reality" OR VR OR "Augmented Reality" OR AR OR "Mixed Reality" OR MR)" was applied to filter the literature along with various combinations of these terms. Another inclusion criteria required that papers explicitly address the direct use of LLMs within or for XR applications. Studies such as those merely using LLMs to analyze data collected from XR environments were excluded. In addition, papers must specifically relate to the integration of LLMs into Extended Reality, and not the other way around. For instance, a study addressing the visualization of Large Language Models using XR would therefore not be included. To further refine the selection, backward and forward searches were performed. In total, 36 papers were included in the final analysis.

IV. Clustering and Thematic Analysis

In the following sections, each thematic cluster is defined, and the associated studies are analyzed and compared. The analysis covers their objectives, areas of application, technical implementation, as well as evaluation methods and results. Common approaches as well as content-related similarities and differences between the studies are highlighted. An overview of the included papers per cluster is provided in Table I.

TABLE I
REFERENCES PER THEMATIC CLUSTER

Cluster	Count	Paper
Content Creation	10	[19, 20, 21, 22, 23, 24, 25, 26, 27, 28]
Embodied Agents	11	[4, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38]
Assistive Systems	21	[19, 22, 23, 26, 27, 35, 39, 40, 41, 42, 43] [44, 45, 46, 47, 48, 49, 50, 51, 52, 53]

A. LLMs for Generative Content Creation and Manipulation

A key area of application for LLMs in Extended Reality lies in the procedural generation and manipulation of content. Creating XR experiences is typically labor-intensive and requires domain-specific expertise [25]; the integration of generative AI helps to reduce both workload and entry barriers. Among other things, LLMs can be used to generate and manipulate 3D objects [22, 23, 24, 19], scenes [24, 25, 27, 19, 28], code [24, 25, 26, 19], animations [21, 25, 19] and even sounds [20]. This content can be generated not only for XR environments but also within them: "real-time" content generation enables users to spontaneously manipulate the virtual world in real time, directly from within the XR environment [24, 26, 19]. Therefore, LLMs can not only reduce the effort required to create assets, but also fundamentally change the process of creation and how users experience it. Notably, this cluster only includes papers with a clear focus on generating and manipulating content. The mere use of the generative properties of LLMs, for example to create avatar responses or supporting overlays in AR, is not included.

1) Direct Application of NLP Capabilities: The most obvious approach for utilizing LLMs in content generation is by directly using the capabilities in language comprehension and generation. Mann et al. [28] present a pipeline for the automated creation of interactive AR tours. In this approach, GPT-3 is used to predict the location and time of historical events by analyzing archival texts, benefiting from the LLM's language processing abilities. Based on the extracted data a full scene is assembled using geocoding and environmental scanning.

Another way to utilize the NLP skills in a straightforward manner is through programming. Since the datasets LLMs are trained on also include substantial amounts of source code, they demonstrate strong skills in handling programming tasks [10]. The DreamCodeVR framework [26] allows users without proficiency in programming to manipulate the behavior of objects within a scene of the Unity Engine in real-time. Instructions are given by voice input, which are transcribed and forwarded to GPT-4 together with a prompt. The code from the response is extracted, compiled using the Roslyn Compiler [54] and integrated into the VR scene. This allows actions such as controlling the movement of an object or changing its color. However, the possibilities are limited to what can be achieved through a single script, meaning the commands only relate to a single object rather than the entire scene and only primitives. This incremental integration of code into an active scene is also applied in other, more complex frameworks [24, 27, 19], typically based on Unity and Roslyn.

2) Combination with Additional Models and Assets: In many applications the pure generation of text is not sufficient. Due to their inherent limitation to text, LLMs must be combined with external frameworks or datasets to produce more complex assets like 3D models or sounds. In [21] animations of human bodies can be generated based on natural language input. The user provides a story, which is interpreted by an LLM. It segments the narrative into appropriate scenes and generates prompts for each, which are then passed on to SayMotion, a generative model creating animations from textual input. Yin et al. [25] adopt a similar approach to generate and populate realistic scenes based on user prompts. By leveraging LLM-generated descriptions, their framework creates backgrounds using the 2D image generator Skybox AI and generates 3D models through the diffusion-based model Shap-E. In [19] a refined approach is proposed to incorporate open-source 3D assets into a generated scene. A textual description is used to generate a target image using Dall-E, while simultaneously downloading a collection of relevant model screenshots from the internet. The contrastive neuronal network CLIP is then employed to select the 3D model that most closely resembles the target image. Multiple of these approaches are combined in [22, 23]: Initially a repository of pre-existing models is searched for matches; if no fitting model is found, a new one can be generated using Shap-E.

Beyond visual content, auditory components can also be generatively produced using similar techniques. The framework proposed in [20] enables users to create contextually appropriate sounds in real time within an augmented reality environment. To achieve this, interactions and contextual information are converted into a textual description and passed to GPT-4, which orchestrates the sound acquisition process. The LLM can retrieve sounds by their names from local and online databases or generate custom sounds by prompting the Audio Diffusion Model AudioLDM.

3) Modular Pipelines: For more complex tasks, such as the creation of entire scenes, a single LLM is often not sufficient. Therefore, the studies use pipelines that involve multiple autonomous LLM agents, which, as described earlier, are combined with other frameworks [24, 25, 27, 19]. The variety of necessary tasks such as modeling, lighting, scene analysis, and scripting is distributed across different, individually contextualized LLM instances [19]. In [25], an LLM agent first generates a description of the desired scene based on a user prompt. Then, different agents sequentially generate a skybox and 3D models, determine the size and position of objects, and finally, add source code and animations.

The authors of [24] propose the use of an agent to break down complex tasks provided by users into smaller sub-tasks. These are then passed on to another LLM agent that generates the appropriate code. This makes the framework robust to tasks of varying length and complexity. In [19], this approach is extended with additional modules: The Scene Analyzer filters the scene for all relevant objects, and the Skill Library provides the

relevant skills for the current task to the Builder generating the code. Both agents address limited context windows of LLMs — a detailed description of the entire scene or all possible skills would exceed the model's effective processing capacity. Additionally, the Inspector module checks the generated code and, in case of errors, suggests a fix for regeneration, resulting in an iterative improvement of the source code. Due to their probabilistic nature, LLMs are prone to errors and therefore benefit from this approach, which is why it is applied in other studies such as [24] or [25].

The effectiveness of such modularized pipelines was evaluated in multiple reliability studies. In [19] it is shown that the framework, through the use of the specialized modules, was able to reduce the code generation error rate by up to a factor of four compared to baseline GPT-4, while also demonstrating greater robustness above tasks of varying complexity and length. The ablation study in [25] shows a similar result, demonstrating a significantly reduced error rate through the division of tasks across multiple modules and iterative testing. However, it should be noted that the evaluated error rate in this context only measures the frequency of code errors and does not provide any assessment of the quality of the generated results.

4) Spatial Understanding: To manipulate existing scenes, frameworks must process spatial information. An understanding of the positions of objects and their relationships to one another is essential for handling user prompts in a meaningful and consistent manner [24]. Since LLMs only operate on text and therefore lack any inherent spatial understanding, this poses a challenge. The most straightforward approach is to rely on existing textual descriptions of scenes [24, 27, 19]. For example, the scene graph of the Unity engine provides a hierarchical listing of all objects in a scene and their relationships. While this represents an effective solution, it also comes with several limitations. The frameworks depend on meaningfully named objects and cannot derive any meta-information from the graph, such as the objects' functions or their affordances [19]. Most importantly, this approach only works as long as such a textual scene description is available.

However, once the physical environment is integrated, this is no longer the case. A proposed solution [22, 19] involves the use of vision-language models (VLMs), which process not only text but also visual inputs. An example of how VLMs can be integrated into AR systems is demonstrated by Behravan et al. [23]. Their framework Matrix proposes a designer assistant that provides users with real-time suggestions for objects to be placed in the scene that contextually align with the physical environment. To achieve this, the AR device captures an image of the scene, which is then used by the VLM LLaVA to generate recommendations for objects along with proposed placement positions. These suggestions account for visual similarity, scene context, and object functionality. Both, a technical evaluation and a user study confirmed the VLM's high level of contextual understanding, demonstrating its capability to semantically interpret the physical environment and thereby enable more natural interactions.

5) User Expectations and Interaction Behavior: While many studies, due to the early stage of development, primarily focus on technical implementation, the works of Chen et al. [27] and Manesh et al. [24] explore how users interact with the described generative frameworks. In [27], a Wizard-of-Oz-based elicitation study ($N = 22$) was conducted to evaluate users' expectations and approaches when verbally prompting VR scenes. The study found that users expect the LLM agent to have a spatial and functional understanding of objects in the scene. In particular, the agent should understand the user's position and role within the scene when the user is situated in VR, in order to fulfill embodied prompts. This emphasizes the relevance of the topic discussed in the previous paragraph. Participants also expected the agent to remember previous states and prompts; however, this demand is technically challenging due to the limited context window. For example, in [19], the memory of the LLMs is constrained, preventing changes to previously generated content.

The exploratory user study ($N = 12$) in [24] evaluated common interaction patterns and interaction barriers when completing tasks such as scene editing in XR using a custom-developed LLM agent. In general, the study found that the system helped participants quickly adopt effective interaction strategies, resulting in faster and higher-quality task completion. It again highlights the importance of spatial reasoning and advocates for the increased use of multimodal inputs, such as speech, while warning about the potential amplification of LLM-specific issues like hallucinations in XR.

B. Embodied LLM Agents

The second cluster focuses on the implementation of LLMs in non-player characters (NPCs), meaning computer-controlled avatars which users can interact with. Such avatars are frequently deployed in XR environments but, in their current form, remain limited and face several challenges [32]. The vast majority of NPCs rely on scripted rules and behaviors. Dialogues are typically restricted to selecting from pre-defined response options, which results in conversations lacking diversity and quickly becoming predictable and repetitive. NPCs are often not able to process the user's context and behavior, which negatively affects immersion and user engagement — aspects that are crucial in XR [31]. The current approach is also highly dependent on manual labor, making it time-consuming and poorly scalable [30]. Integrating LLMs offers a promising solution to these problems by enabling the creation of human-like, intelligent NPCs. Their NLP capabilities allow for natural language driven dialog, making communication more intuitive and versatile. Moreover, contextual reactions [4], scene understanding [29] and dynamic animations [31] can further enhance the agents' believability. It is worth noting that this cluster is explicitly limited to papers whose focus is on LLMs with physical presence. If communication with an LLM agent takes place without the agent in attendance it is not considered as embodied (e.g., [42]).

1) Prompt Construction and Context Handling: The most evident advantage is that communication with an NPC no

longer requires a user interface or predefined commands. Instead, in all of the reviewed studies, at some point in the pipeline the user input is passed in text form to an LLM, which then generates a response. For the LLM to provide convincing answers, it is essential that it has access to knowledge of the current conversational state. Since LLMs inherently lack memory, this contextual information must be included alongside the user input within the given prompt. In [33], for instance, the most recent action cues and messages are tracked and incorporated into a context prompt for GPT-4 Turbo. Similar strategies can be found in other studies, such as [36] and [37].

While this approach may be sufficient for small-scale scenarios with short dialogues and limited interactions, it does not scale effectively to larger applications. The underlying problem is the limited context window of LLMs, as mentioned in section IV-A because only an input of limited length can be processed at once, it eventually becomes impossible to pass knowledge of all preceding events to the LLM within a prompt. Accordingly, during the user evaluation in [33], agents demonstrated difficulties in tracking changes in object states, as the LLM only had access to newly provided meta-information. As an implication of these findings, the integration of a memory retrieval architecture was proposed.

Such a system was developed by Wan et al. [4]. The social NPCs tested in the multiplayer game VRChat rely on an external database that stores relevant information, called observations. The base observations are defined when creating the NPC and include static attributes such as name, age or personality traits. These are combined with context observations, which function as dynamic memories. With every user input, a retrieval function calculates recency and relevance scores for all observations, based on which the most significant observations are selected. These observations are then passed to GPT-4 as contextual input to process the user's message and generate a response. Subsequently, new context observations are created based on the LLM's response stored back in the database. The optimal combination of observations was determined during an evaluation: using three base observations and five context observations produced the most plausible and coherent replies.

2) Free-Form Conversation: To what extent free-form conversations offer advantages over conventional dialogue options was evaluated by Christiansen et al. [30]. In their user study $(N = 22)$, participants interacted with LLM-based agents in a murder mystery game, once using free-form speech input via voice recognition and once using traditional dialogue options. Voice input conveyed a subjectively higher sense of immersion but could feel overwhelming. As a solution, the authors proposed a hybrid approach in which players still communicate freely by voice but receive suggested inputs from the framework.

However, communication via speech is preferred in most of the reviewed studies, with text input only rarely supported (e.g., [34], [35]). Within these pipelines, user input is typically converted to text through speech-to-text (STT), and the LLM's output is converted back to speech via text-to-speech (TTS). Commonly used plugins include WhisperAI [30, 31, 38] for STT, as well as Meta's Voice SDK [29, 30, 36, 37] for both STT and TTS. A modular open-source pipeline for integrating such STT and TTS capabilities into LLM-based agents was implemented by Buldu et al. [32]. Of particular interest is its streaming functionality, which allows incremental delivery of the response even before the full answer is completed by the LLM, thereby reducing latency. Whether embodied LLM agents are generally preferable to purely voice-based assistants with only symbolic presence was explored in [36].

Participants in this user study $(N = 25)$ found conversations with humanlike embodied NPCs more pleasant and natural and were able to recall more information from these interactions. The development of realistic avatars is therefore recommended to further enhance immersion and communication effectiveness. These findings are supported by a comparative study with a larger sample size $(N = 210)$ in [34]. In that study as well, embodied agents were rated as more engaging and interesting, although they offered no advantage in terms of learning outcomes compared to non-embodied agents, which were perceived as more focused and to the point. A further difference emerged depending on the personality traits of the NPCs: agents high in extraversion and agreeableness were rated as more engaging, open, and sympathetic than their less sociable, low-trait counterparts.

3) Avatar Animation: A crucial aspect in creating realistic NPCs is the use of lifelike animations and movements. The lack of convincing body language, facial expressions, eye contact, or hand gestures quickly reveals a computer-controlled avatar as non-human. At minimum, basic facial animations are implemented in many of the reviewed papers. For example, in [37], lip movements are synchronized with the output sound using Oculus Lipsync for Unity, while the Meta Avatars SDK provides appropriate facial expressions and eye movements. In [4], the avatars select from a pool of pre-generated expressions and gestures, based on the LLM's response and its associated context. In [36], a separate pretrained model is used to analyze the conversation flow and select one of 15 predefined animations accordingly. The framework described in [33] refines these approaches: the LLM used for the NPC can select actions from a function list with corresponding parameters depending on the context. The Unity Engine then validates these actions and either executes the corresponding animation or sends an error message back to the LLM. After a successful execution, the NPC's status is updated through a success message. To implement these animations, tools such as the Realistic Eye Movement package for facial expressions, Oculus Lipsync for mouth movements, and inverse kinematics for skeletal animations are used, all coordinated through Unity's Animator system.

Gunawardhana et al.[31] take the interactivity of animations even further. They employ a neural network architecture known as a Conditional Variational Autoencoder (CVAE), trained on recorded conversations between real humans. During communication with the NPC, users' head movements and

gaze directions in VR are captured and fed into the model, which generates appropriate head movements, postures, and gaze responses of the avatar in real time. At the same time, lip movements are synchronized to the agent's speech output, and matching hand gestures are generated through a separate model. This allows the artificial avatar to partially mirror the user's behavior, as happens in natural conversations. Here as well, inverse kinematics were used to ensure correct skeletal movements. In a comparative user study ($N = 15$), this approach significantly increased participants' attention and engagement compared to conventional animation techniques.

4) Scene Awareness and Interaction: Actions and animations should not be limited merely to facial expressions and lip movements. For smart NPCs to be perceived as a natural part of the XR experience rather than as foreign entities, it is essential that they understand and interact with their environment [38]. In [30], the inability to reference objects was identified as a limitation affecting the perceived presence of NPCs. Users in [36] referenced their surroundings only minimally during conversations with an agent in augmented reality, suggesting that these agents appeared out of place. It has therefore been proposed to incorporate explicit indicators of an NPC's understanding of the physical world.

As previously discussed in section IV-A, there are several approaches for enabling spatial understanding through LLMs. In [33], the text-based approach is used within a VR scene. Therefore a spot List, an object List, and a character List store relevant information about important locations, objects, and characters. Each entry includes details on position, orientation, and metadata, such as what a character is holding in their hand. From these lists, a structured prompt is generated before every interaction, enabling the LLM to gain an understanding of the scene. This contextual knowledge can be incorporated into the NPC's responses, facilitates navigation within the scene and also allows for actions involving objects from the environment. In the background, a Unity plugin dynamically updates the stored information to ensure coherent interactions. A user study ($N = 14$) confirmed that the NPCs were generally able to successfully perform spatial interactions and conversations. However, limitations of such a text-based approach were also noted, including the so-called glass-eye effect, where the NPC might see objects through walls.

As a potential solution, the use of vision-based models has once again been suggested. Lataifeh et al. [29] employ this type of technology for their assistant avatars in AR. The headset's camera provides real-time images of the physical environment, which are used to generate meshes for all surfaces — necessary for collision avoidance and navigation during movement. At the same time, the camera images are fed into a convolutional neural network (CNN), which detects and classifies objects and attaches 2D coordinates that are then mapped into the 3D space. Additionally, the images are sent to GPT-4o, which enriches the scene understanding beyond mere object recognition. By combining this information, the NPC can move within the physical environment, look at and point to objects, and provide contextually appropriate cues.

5) Use in Education and Training: It is worth highlighting the frequent use of embodied LLM agents in education and training [35, 36, 37, 38]. NPCs can serve as virtual teachers [35], enable customized realistic training scenarios [37], or recreate historical characters [38]. In [35], a digital twin teacher is designed to relieve human teachers and allow them to focus on answering the most complex questions. When students ask questions, the digital teacher searches for relevant subject matter context in a vector database using embeddings, and then uses this context to generate its responses.

Li et al. [37] propose using LLM-powered NPCs not as teachers, but as realistic conversational partners in job-related training scenarios. This approach is intended to give people with autism the opportunity to practice behaviors for potentially stressful situations in a safe environment. The system enables flexible design of role-play scenarios, which can be adapted by job coaches through simple prompts, while the language-based interaction with the LLMs creates an authentic conversational atmosphere. An initial exploratory user study indicates positive effects, though its findings are limited by a sample size of only three participants. Expanding on the possibilities of LLM-based teachers, Zhu et al. [38] propose a VR-based system in which LLM-driven agents embody historical figures. Through adaptive role-switching, the agent can take on the character most appropriate to the student's question. GPT-4o is used to select the most suitable role based on the user input and the descriptions of all available roles before generating the actual response. Accordingly, the tone, voice, and appearance are dynamically adjusted. In a user study ($N = 84$), this combination of roles was shown to increase perceived expertise and presence, contributing to deeper learning. However, too frequent role changes risk disrupting the consistency of the learning experience.

C. LLM-powered Assistive Systems

The previous clusters, with their focus on embodiment and generative content creation, can be defined relatively clearly. This is not the case for the current cluster: the term LLM-powered Assistive Systems encompasses a broad range of different applications. One growing concern is the presence of accessibility barriers in Extended Reality, which can lead to the exclusion of people with disabilities [55]. Assistive systems can help address these challenges, for instance by generating auditory scene descriptions [44] or supporting voice-driven coding [26]. Challenging interactions in XR, such as text input [43], can be made more efficient through LLMs, thereby improving usability. Intelligent assistants can also support users in real-world tasks via AR — for instance, by retrieving information about physical objects [39], suggesting dialogue options to conversation partners [46], or supporting surgeons during complex operations [53]. What these studies have in common is that they directly change the way users interact with XR, usually with a focus on usability or user experience. Certain papers from other clusters can also be categorized as assistive systems [22, 23, 26, 27, 35, 19]. For example, one suggested use case of the generative framework proposed

in [19] involves color-marking objects to make scenes more accessible to users with color vision impairments.

1) Enhancing Input Modalities: One area where LLM assistance shows potential, is in enhancing input modalities within XR environments. Conventional input methods often prove inefficient and inaccurate. One potential solution is shown in the previously mentioned [26]. The framework enables more efficient programming by using high-level instructions via speech instead of explicit code. Without such an approach, programming in VR remains cumbersome, particularly due to generally inefficient text input in immersive environments. Chen et al. [43] aim to improve this aspect by introducing three assistive methods: Simplified Spelling, which allows users to omit letters that are then completed by the LLM at the sentence level; Content Prediction, which dynamically suggests words based on the previous two sentences; and Keyword-to-Sentence Generation, which expands individual keywords into full sentences. In a user study ($N = 22$), these methods reduced manual keystrokes by 16% to 50% and improved typing speed by 21% to 76%, depending on the method. Simplified Spelling yielded the smallest gains but was primarily used in scenarios requiring exact wording. Overall, the task load was significantly reduced, especially the physical load (measured using NASA-TLX).

However, in time-critical scenarios, such as taking notes during a live lecture in AR, these improvements might not suffice. For such use cases, [45] proposes a combination of eye-tracking and LLM-generated suggestions. The lecturer's speech is captured through the AR headset, transcribed via STT, and parsed into keywords using GPT-4. For in-context notes, users can directly select from these keywords. For beyond-context notes, the system allows users to derive additional keywords or choose from predefined terms. Using gaze selection, users choose the most relevant words, which are then used to generate multiple LLM-based sentence suggestions that can be stored as notes. Two user studies showed reduced distraction and improved note accuracy over manual text input or fully automated note-taking. Another time-critical scenario is addressed by Javaheri et al. [53], who propose an AR system for use during pancreatic surgery. The system overlays patient-specific 3D models onto the abdominal area and simultaneously provides CT scans and relevant medical history to the surgeons. Due to spatial constraints, stress, occupied hands, and strict hygiene requirements, physical input methods are not feasible in the operating room. Here, an LLM-powered speech interface enables sterile, hands-free interaction with the system.

2) Adaptive User Interfaces: An essential aspect of XR systems is the user interface (UI). Unlike static UIs on screen-based devices, interfaces in XR can be embedded into the user's physical environment [40]. For example, [39] demonstrates this by equipping physical objects with seamless, context-based menus. However, adapting the UI to dynamic surroundings poses significant challenges. Study [48] explores LLM-based chatbots in XR vehicle environments, noting that traditional linear layouts struggle with space constraints and movement limitations. A proposed multi-window UI was shown to significantly lower task load and completion time while improving usability. One challenge in AR lies in the user interfaces being on top of important objects or people. In [45], UI elements are deliberately positioned at the periphery to avoid obstructing a visible speaker. Li et al. [40] propose an adaptive UI system powered by LLM reasoning, which dynamically adjusts its layout based on contextual factors in shared spaces. Using RGB-D camera input, the system constructs a 3D map of the environment and employs GPT-4 Vision to evaluate potential UI placements in terms of functionality, aesthetics, and social acceptance. An optimization module then determines the ideal arrangement based on these scores and the spatial layout. Built on the AUIT toolkit, the system adds two novel cost functions targeting overlay minimization and interaction suitability. In a classroom scenario, it was rated by 12 participants as more socially appropriate and functionally effective than two non-adaptive baseline UIs.

In [50], the adaptation of the user interface is proposed not only to the environment but also to the user's cognitive state. Wen et al. developed an assistive system that supports pilots in the cockpit through visual, auditory, and text-based cues. The system uses real-time data from functional near-infrared spectroscopy (fNIRS) to assess the pilot's cognitive workload. Based on this input, the PHI-3 language model applies chain-of-thought reasoning to generate adaptive instructions tailored in both modality and level of detail. In underload scenarios, the system provides more information to stimulate the user; during overload, it prioritizes essential visual cues to help the pilot stay calm and focused. A user study ($N = 8$) conducted in a VR simulation of the pre-flight phase showed an increased occurrence of optimal working memory states and a trend toward faster task completion. However, no significant improvements were observed in perception or attention.

3) Context-sensitive Support: Many assistive systems support users through spatial and situational understanding. This can take place in fully virtual VR environments. For instance, in [52], players of an escape room game receive help from a voice assistant based on which objects they have already discovered in the scene. Many studies in this area rely on Multimodal Large Language Models (MLLMs) for spatial understanding [41, 39, 23, 44]. These models can process not only text but also other input modalities, such as the vision-language models discussed earlier. Oliveira et al. [44], for example, employ such models to improve the accessibility of VR environments for individuals with visual impairments. Based on 360-degree images and spoken user queries transcribed via STT, the MLLM automatically generates scene descriptions. Additionally, the system uses Retrieval-Augmented Generation (RAG) to incorporate contextual information from external databases, such as background on artworks in virtual museums, into the model's responses.

Spatial understanding becomes especially critical in AR, where the physical world is integrated. Dogan et al. [39] propose the concept of Augmented Object Intelligence (AOI) for mobile devices. In this approach, physical objects in AR

can be interacted with as if they were digital entities, using object-based context menus. Real-time object recognition and classification are handled by Google's MediaPipe framework, which uses a CNN to assign each object a 2D bounding box. These detections are then projected into 3D space using depth maps and raycasting. Each recognized object is assigned a dedicated MLLM instance that processes visual input from the bounding box and external web data, while also maintaining an object-specific interaction history. Users can query contextual information, compare multiple objects, or anchor digital widgets like timers or notes via touch or voice input. A user study ($N = 8$) showed that the system enabled 24% faster task completion and yielded higher comfort and satisfaction compared to a standard MLLM chatbot. A similar approach is extended in [41] with a split architecture, allowing the framework to run on lightweight mobile AR headsets. The headset captures camera images, voice input, and depth data, generates 3D mesh representations of the environment, and renders AR content. More resource-intensive tasks such as STT processing and local (i.e., non-cloud-based) MLLM inference are offloaded to a more powerful server backend.

Yang et al. [46] move beyond object interaction and introduce a framework that proactively supports users during social interactions. It leverages camera, microphone, and motion sensor data from an AR headset to capture both verbal and nonverbal cues – such as gestures, facial expressions, and contextual factors like location and formality level. For dynamic persona adaptation, the system automatically extracts and stores information about the user and conversation partners from previous interactions, reusing it as context in future dialogues. Initially, the system draws on a social factor–based cache to provide rapid suggestions in familiar situations, which are refined by an LLM if necessary. When no suitable suggestion is found, LLM-based generation is triggered even before the interlocutor has finished speaking, working from partial utterances to minimize latency and avoid disrupting the flow of conversation. The final output is displayed on the AR headset in a structured format, for instance bullet points and example phrases. Evaluation using three public datasets and a user study ($N = 20$) demonstrated a 38% increase in engagement compared to baselines and a 95% willingness among participants to adopt the system in daily life.

4) Applications in Education and Culture: Just as section IV-B, this cluster includes several papers that explore applications of LLM-based systems in educational and cultural contexts. In [49], a virtual classroom scenario is explored in which an LLM-based assistant supports students, however, without a fully embodied agent as seen in [35]. To ask a question, students raise their hand and use voice input, which is transcribed and processed by GPT-3.5 Turbo. The assistant's response appears on a digital board next to a static teaching figure, allowing the scripted lesson to continue uninterrupted. In a user study ($N = 52$), this setup led to significantly higher student engagement and lower cognitive load compared to a version without the LLM assistant, though quiz performance was not significantly improved.

Cheng et al. [42] propose a framework that moves learning entirely out of the classroom and into the real world through AR-based activities on the topic of environmental protection. In one such activity, students write letters to a fictional character, to which GPT-4 responds with personalized replies and reflective follow-up questions. In a user study ($N = 50$), primary school children wrote more meaningful letters and showed greater willingness to continue the activity when interacting with the LLM-based character compared to a static control version.

Virtual museums also present strong use cases for LLM-driven assistance. In [51], users can explore virtual paintings in a physical environment using AR. Through voice interaction, they can ask GPT-3.5 questions about the artworks, such as details about the artist or historical context. The system extracts relevant information by converting the spoken input into a semantic embedding and comparing it with precomputed embeddings of stored artwork descriptions. A user study ($N = 39$) found that participants generally perceived the LLM-based system as user-friendly and trustworthy, although differences to the control group were not statistically significant. A more advanced system for virtual museum tour guides is proposed by Wang et al. [47]. Here, user speech input is contextually analyzed and transformed into multimodal feedback such as text panels, spoken explanations, avatars, or directional cues. The system follows a two-stage architecture employing several specialized LLM instances, similar to [24] and [19]: First, a Classifier determines the type of user intent, and a Compiler identifies the necessary information. Then, an Explorer provides content explanations, a Navigator handles wayfinding, and an Identifier enables personalized interactions.

V. Discussion & Future Directions

A. Key Insights from the Literature

1) Advantages of LLMs in XR: The literature highlights a wide range of potentials for integrating LLMs into Extended Reality. A central advantage lies in the use of natural language as an intuitive interface for user interaction. By that, scene manipulation and content creation can be performed directly via speech or text input [22, 23, 24, 25]. Tasks that previously required specialized expertise, such as programming [26, 19] or animation [21], can now be carried out without technical knowledge. The intuitive nature of language input allows users to control generative systems in real-time directly from within XR environments [24, 26, 19]. Natural language also serves as an efficient and accessible communication method for assistive systems, particularly through voice input [44, 51, 53]. Its greatest value, however, may be in interactions with AI-controlled avatars. Thanks to the NLP capabilities of LLMs, predefined responses are no longer necessary, enabling open-ended, dynamic conversations [4, 31, 33]. This form of interaction more closely mirrors human dialogue, enhancing engagement and immersion, both critical aspects of XR [30].

Another major advantage of LLMs in XR is their ability to respond contextually and adapt their behavior to the current situation. Unlike traditional systems limited to fixed input

sets, LLMs can dynamically respond to a wide variety of contextual cues. This improves the believability of virtual agents, who can now draw on previous user interactions to generate coherent responses [4, 36, 37]. Spatial data enables NPCs to develop an awareness of their environment, allowing them to reference and interact with nearby objects [29, 33]. Many assistive systems also benefit from dynamic adaptation. In AR, for example, systems can provide context-aware overlays and guidance in real time based on spatial understanding [41, 39, 23, 53]. This contextual responsiveness extends to social dynamics [40, 45, 46], user state [50], and prior knowledge [52], allowing support to be personalized accordingly.

Moreover, the generative capabilities of LLMs enable notable efficiency gains in both content creation for XR and within XR environments themselves. Particularly in content production, they help make development workflows more accessible and significantly faster. Labor-intensive tasks like level design, modeling, and animation can be greatly accelerated [25, 19, 28]. Similar benefits apply to NPC development, where LLMs enhance realism while eliminating the need for manual dialogue scripting [30]. Many of the assistive applications are designed to enhance the efficiency of existing XR systems, for instance by improving input methods or optimizing user interfaces [40, 43, 45, 48]. In educational contexts, LLMs support more efficient learning by enabling immersive, personalized scenarios that boost both performance and motivation [42, 37, 38], while virtual tutors help reduce the workload of real instructors [35, 49].

2) Implementation Strategies: Across the reviewed studies, several recurring implementation patterns emerge. Many frameworks employ modular pipelines, combining multiple LLM instances [4, 24, 25, 27, 46, 47, 19]. These agents are assigned distinct subtasks and are individually prompted for context-specific responses. Such systems are often coupled with iterative feedback loops and specialized validation modules [24, 25, 19]. These architectures are designed to mitigate typical LLM limitations such as hallucinations or limited context windows. Especially for complex tasks, this structured approach can significantly reduce error rates [25, 19].

A common strategy is the integration of LLMs with complementary technologies to extend their capabilities and meet the demands of XR systems. Since LLMs fundamentally generate only text, their standalone use in XR is limited. To enable gestures, facial expressions, and avatar movements, LLMs are used to drive existing animation systems or select appropriate actions [4, 31, 33, 36, 37]. Memory or knowledge integration is typically handled via databases and retrieval systems that supply relevant information in context [4, 35, 44, 46, 51]. In content creation, LLMs are combined with other generative AI tools, such as diffusion models, or used to select from existing assets [20, 21, 22, 23, 25, 19]. These approaches effectively bypass the inherent limitation of LLMs to text.

Multimodal input processing is also common; very few studies rely solely on textual interaction. Most frameworks support speech input via STT systems (e.g., [30, 32, 38]), which aligns naturally with conversational interaction and contributes to user immersion. Visual input such as camera feeds and depth data is also frequently incorporated, particularly in AR applications where scene understanding is essential. To handle such sensory inputs, many systems employ multimodal LLMs, particularly VLMs [41, 39, 29, 40, 44, 46, 47]. Additionally, some systems process other modalities such as gaze tracking [31, 45], head movement [31, 46], or physiological signals like brain activity [50], further broadening the interaction spectrum of LLM-driven XR applications.

B. Challenges and Limitations

1) Latency and "Real-Time": Latency emerges as the most consistent challenge across all clusters and studies. In XR applications, it plays a particularly critical role, as users can perceive even minor delays in system feedback [26]. This becomes first problematic in generative frameworks designed for "real-time" use [22, 26, 27, 19], where completion times for content generation can range from several seconds to minutes. One study on the use of LLMs in gaming [56] argues that systems like [19] and similar frameworks are unsuitable for such contexts due to their lack of responsiveness.

Likewise, the effectiveness of assistive systems can be severely constrained by latency. In [43], each interaction with the LLM is associated with a delay of about 1.5 to 1.8 seconds. When this response time is not factored into the measurement of typing speed, the observed gains from assistance methods can increase by up to twofold. In [45], the average delay for generating relevant keywords during dynamic note-taking exceeds four seconds. The issue becomes even more critical in high-stakes scenarios such as the pilot support system proposed in [50], where delayed assistance could lead to panic.

Embodied agents, in particular, suffer from latency due to their focus on immersion [4, 30]. When a virtual avatar takes several seconds to respond, it becomes immediately distinguishable from a real human. Unlike LLM-based systems, traditional NPCs with scripted responses do not exhibit this issue, which may hinder the broader adoption of LLMs in this area. Overall, latency can largely be attributed to the response time of language models, network delays, and other computationally intensive steps such as speech-to-text transcription [30, 26, 33, 43]. This results in a fundamental trade-off between latency and response quality: more thoroughly reasoned answers or deeper contextual understanding tend to come at the cost of longer waiting times.

Many of the papers analyzed would benefit from reduced LLM response time. One proposed solution is the use of locally deployed LLMs [33, 43], which eliminate network delays and reduce dependence on cloud services. However, locally deployed LLMs also come with certain drawbacks: without access to powerful cloud servers, their performance is limited, and the required hardware is often expensive. On mobile XR devices, this is difficult to implement, especially in scaling applications where, for example, multiple embodied LLM agents are intended to be used [57]. To address these limitations, task-specific model compression could offer gains

in speed and memory efficiency. Techniques such as pruning and distillation can remove redundant components or yield smaller models with comparable performance [58]. Given that the LLMs currently used are designed for general-purpose tasks and only a fraction of their capabilities is relevant for XR scenarios, such specialization appears logical. Finally, it is worth noting that XR applications are likely to benefit passively from ongoing advancements in language modeling. Given the rapid pace of technical progress and the broad applicability of LLMs, further reductions in latency can be expected in the near future.

2) Spatial Understanding: Since XR environments are inherently based on three-dimensional structures, spatial reasoning is a crucial requirement for the effective integration of LLMs. Very few applications can function entirely without spatial references [43, 45, 28]. For context-aware assistive systems [40], believable NPCs [29], and prompting 3D scenes [27], a solid understanding of objects and their spatial relationships within a scene is essential. The approaches used can be broadly divided into two categories: spatial reasoning based on textual descriptions and spatial reasoning based on the processing of visual information.

The first category is mainly employed in VR applications, where the engine often provides a scene graph with structured textual information about the positions and attributes of all objects within a scene [24, 33, 27, 19]. These setups are relatively easy to implement, as they directly leverage the NLP capabilities of LLMs. However, this approach is limited in terms of accuracy and coherence. Existing frameworks struggle with dynamic states and unknown objects [24, 19]; perspective reasoning is also difficult to achieve using plain object lists: NPCs, for example, sometimes refer to objects they should not be able to see from their position [33].

Physical environments lack structured textual representations, making the processing of visual data necessary for AR environments [41, 39, 29, 40, 23, 44, 46]. Here, sensor inputs such as RGB images or depth maps are interpreted by MLLMs, mainly vision-language models. However, currently the input of these models is usually limited to individual static images [39, 40, 23, 44]. Their use is often restricted to specific high-level tasks: in [40], for instance, VLMs analyze the layout of UI elements when triggered by user interaction; in [23], they generate design suggestions based on camera input. While these models are effective at providing contextual cues or generating scene descriptions [44], their current usage does not enable a genuine, three-dimensional understanding. In most frameworks, core spatial analysis is offloaded to upstream technologies. Object classification and localization are handled by dedicated CNNs, SLAM algorithms, or meshing techniques, which produce 3D data such as point clouds or voxels. These outputs are processed separately from the LLM, which only receives partial 2D representations and does not directly engage with spatial data [41, 39, 40, 23].

Nevertheless, XR applications across all domains could benefit significantly from more robust spatial intelligence — for example, for avatar navigation, scene manipulation, or task planning. To achieve this, MLLMs would need to process not just images but also structured 3D data like depth maps, point clouds, or voxels. This remains challenging, partly because many VLMs are trained exclusively on 2D images [14]. Training LLMs on specialized 3D datasets tailored to XR use cases could unlock new capabilities. An alternative approach is proposed by Avetisyan et al. [59]: instead of converting sensor data into conventional 3D representations, their scene reconstruction framework translates environmental inputs into a structured scripting language, producing a textual description of the scene. This text could then serve as input to an LLM — a hybrid approach that transfers the advantages of text-based processing to real-world environments.

Importantly, spatial reasoning via LLMs is not only relevant to XR but also to adjacent fields such as robotics and autonomous driving. Therefore, similar to the development of local LLMs, continuous progress in this area is to be expected [60]. At the same time, further progress in spatial understanding through LLMs in XR depends heavily on hardware capabilities: robust performance requires devices equipped with suitable sensors and low-latency, unrestricted access to raw sensory data. This remains a limiting factor for current systems [41, 39].

3) Systematic and Meaningful Evaluation: One major challenge in analyzing the current body of work on integrating LLMS into Extended Reality is the lack of comparable and comprehensive evaluations. A recurring issue is the often small sample size in user studies. This can be attributed not only to the high resource demands of XR experiments and the early stage of research in this field, but also to the technical focus of many studies, where user evaluation is frequently treated as secondary. However, larger-scale evaluations would be essential, especially when key findings of a paper rely heavily on user feedback. For instance, in [4], only LLM-based judges and a single human evaluator are used to determine two core parameter values.

A more critical issue, however, is the absence of standardized metrics to assess the use of LLMs and enable cross-study comparisons. Subjective metrics such as NASA-TLX for perceived workload [41, 26, 43, 27, 45, 48, 49, 50] and the System Usability Scale (SUS) for perceived usability [43, 27, 48, 38] are frequently employed — particularly in assistive systems with a user-centered design. On the other hand, technical metrics are used, like error rate, response time, or accuracy ratings for specific tasks such as object recognition [22, 23, 25, 26, 19]. While these metrics can meaningfully assess individual components, they fall short of supporting holistic comparisons. For example, both [25] and [19] use error rate as their primary metric, yet this offers little insight into the quality and prompt alignment of their generated outputs.

Overall, existing metrics fail to adequately capture the complexity of multimodal LLM applications, such as spatial and contextual understanding or dynamic object generation. Consequently, the development of new evaluation methods and metrics is essential for advancing this field in a more systematic and comparable way. These methods must be tai-

lored specifically to the unique challenges of applying LLMs in XR. Tang et al. [61], for example, propose a meaningful set of evaluation dimensions that future metrics should address, including spatial-conceptual awareness, coherence, proactivity, multimodal integration, hallucination, and question-answering accuracy. Going forward, such metrics, or comparable alternatives, should be formally defined and incorporated into upcoming studies.

4) Privacy and Ethical Concerns: Although not addressed in the initial analysis, ethical and privacy concerns represent critical aspects that must now be discussed. One major issue is the potential displacement of designers and developers by generative systems [62]. However, the argument made by [19] appears credible: unlike LLMs in other domains, current XR frameworks remain far from enabling end-to-end development and still rely heavily on human input.

More pressing are the potential violations of data privacy, particularly in AR frameworks used in social settings [45, 46]. Special caution is warranted when these systems involve facial recognition or the extraction of personal information from conversations [46]. Another concern is the widespread use of cloud-based LLMs such as ChatGPT in the studies reviewed. Well-known issues include the risk of leaking sensitive data and the lack of transparency in how data is processed [63]. These risks may be amplified in XR environments: Bozkir et al. [64] warn of novel privacy invasions that may arise from combining LLM-powered NPCs with multimodal sensor data.

Embodied agents introduce additional challenges. Due to biases in their training data, LLMs are prone to producing discriminatory or offensive content [65]. In immersive XR settings, where intelligent agents are rendered with high realism, such utterances could have an intensified emotional impact on users. Furthermore, parasocial relationships between users and LLM chatbots have already been documented [66]. When transferred to virtual avatars, this phenomenon could lead to serious psychological consequences.

Even if some of these concerns ultimately prove less severe, proactive consideration is essential. The rapid pace of technological development in this area must not compromise user safety. A first step toward improving data protection would be the adoption of locally hosted language models, as previously mentioned, to ensure greater control over data storage and processing. AR systems designed for social contexts should require explicit consent from individuals before collecting sensor data or should be configured to capture only non-identifiable information. Finally, realistic embodied LLM agents should be evaluated for both psychological impact and privacy risks. It may become necessary to establish ethical guidelines that limit certain features in order to protect users.

C. Future Directions

Based on the identified challenges and limitations, the following research directions could advance the integration of LLMs into Extended Reality systems:

- *Latency Reduction*: Development of lightweight, locally deployable LLMs tailored for XR using techniques such as pruning or distillation to minimize inference time and enhance responsiveness.
- *Robust Spatial Reasoning*: Training of multimodal LLMs on 3D datasets specifically adapted to XR environments, with the goal of enabling LLMs to directly participate in the processing of 3D data.
- *Evaluation Metrics*: Design and formalization of standardized methods for meaningful evaluation of LLM performance in XR applications, including domain-specific factors like spatial awareness, contextual relevance or coherence.
- *Privacy and Ethics*: Development of systems that minimize data collection and ensure transparency in how user data is handled. Investigation of potential psychological risks associated with deploying LLMs in immersive environments.
- *Hardware-Software Co-Design*: Aligning the sensors used in XR devices with the requirements of multimodal input. Integration of dedicated processing units for local LLM deployment and optimization of hardware pipelines to reduce latency.

VI. Conclusion

This survey provided an overview of recent progress in the integration of Large Language Models into Extended Reality. It systematically reviewed works from ACM and IEEE sources published since 2023 and categorized them into three thematic clusters: LLMs for Generative Content Creation and Manipulation, Embodied LLM Agents, and LLM-powered Assistive Systems. Finally, cross-cutting challenges were identified in the areas of latency, spatial understanding, and evaluation, as well as privacy and ethics. For each of these domains, specific directions for future research were outlined to address existing limitations and enable the widespread use of LLMs in real-world XR applications.

VII. Acknowledgement

For linguistic support, the AI model ChatGPT-4o and the AI-based translation service DeepL were used in the preparation of this work. Their use was limited to translations from pre-written German passages into English and to stylistic improvements. No conceptual or content-related input was generated by AI.

References

[1] M. Vasarainen, S. Paavola, and L. Vetoshkina, "A systematic literature review on extended reality: Virtual, augmented and mixed reality in working life," *International Journal of Virtual Reality*, vol. 21, no. 2, p. 1–28, Oct. 2021. [Online]. Available: https://ijvr.eu/article/view/4620

[2] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino, "Augmented reality: A class of displays on the reality-virtuality continuum," *Telemanipulator and Telepresence Technologies*, vol. 2351, 01 1994.

[3] V. Angelov, E. Petkov, G. Shipkovenski, and T. Kalushkov, "Modern virtual reality headsets," 06 2020, pp. 1–5.

[4] H. Wan, J. Zhang, A. A. Suria, B. Yao, D. Wang, Y. Coady, and M. Prpa, "Building llm-based ai agents in social virtual reality," in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3613905.3651026

[5] S. Nagpal, S. Bansal, M. Kumar, A. Mittal, and K. Saluja, "Augmented reality: A comprehensive review," *Archives of Computational Methods in Engineering*, vol. 30, 10 2022.

[6] R. Skarbez, M. Smith, and M. Whitton, "Revisiting milgram and kishino's reality-virtuality continuum," *Frontiers in Virtual Reality*, vol. 2, 03 2021.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available: https://arxiv.org/abs/1706.03762

[8] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large language models: A survey," 2025. [Online]. Available: https://arxiv.org/abs/2402.06196

[9] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen, "A survey of large language models," 2025. [Online]. Available: https://arxiv.org/abs/2303.18223

[10] J. Kaddour, J. Harris, M. Mozes, H. Bradley, R. Raileanu, and R. McHardy, "Challenges and applications of large language models," 2023. [Online]. Available: https://arxiv.org/abs/2307.10169

[11] A. Aiersilan, "Literature review of mixed reality research," 2023. [Online]. Available: https://arxiv.org/abs/2312.02995

[12] Z. Wang, M. Rao, S. Ye, W. Song, and F. Lu, "Towards spatial computing: recent advances in multimodal natural interaction for xr headsets," 2025. [Online]. Available: https://arxiv.org/abs/2502.07598

[13] X. Ma, Y. Bhalgat, B. Smart, S. Chen, X. Li, J. Ding, J. Gu, D. Z. Chen, S. Peng, J.-W. Bian, P. H. Torr, M. Pollefeys, M. Nießner, I. D. Reid, A. X. Chang, I. Laina, and V. A. Prisacariu, "When llms step into the 3d world: A survey and meta-analysis of 3d tasks via multi-modal large language models," 2024. [Online]. Available: https://arxiv.org/abs/2405.10255

[14] J. Zha, Y. Fan, X. Yang, C. Gao, and X. Chen, "How to enable llm with 3d capacity? a survey of spatial reasoning in llm," 2025. [Online]. Available: https://arxiv.org/abs/2504.05786

[15] T. Hirzle, F. Müller, F. Draxler, M. Schmitz, P. Knierim, and K. Hornbæk, "When xr and ai meet - a scoping review on extended reality and artificial intelligence," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, ser. CHI '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3544548.3581072

[16] F. Rahimi, A. Sadeghi, and S.-M. Choi, "Generative ai meets virtual reality: A comprehensive survey on applications, challenges, and future direction," *IEEE Access*, vol. PP, pp. 1–1, 01 2025.

[17] I. A. Brito, J. S. Dollis, F. B. Färber, P. S. F. B. Ribeiro, R. T. Sousa, and A. R. G. Filho, "Integrating personality into digital humans: A review of llm-driven approaches for virtual reality," 2025. [Online]. Available: https://arxiv.org/abs/2503.16457

[18] Y. Tang, J. Situ, A. Y. Cui, M. Wu, and Y. Huang, "Llm integration in extended reality: A comprehensive review of current trends, challenges, and future perspectives," in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, ser. CHI '25. New York, NY, USA: Association for Computing Machinery, 2025. [Online]. Available: https://doi.org/10.1145/3706598.3714224

[19] F. D. L. Torre, C. M. Fang, H. Huang, A. Banburski-Fahey, J. A. Fernandez, and J. Lanier, "Llmr: Real-time prompting of interactive worlds using large language models," 2024. [Online]. Available: https://arxiv.org/abs/2309.12276

[20] X. Su, J. E. Froehlich, E. Koh, and C. Xiao, "Sonifyar: Context-aware sound generation in augmented reality," in *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3654777.3676406

[21] K. He, A. Lapham, and Z. Li, "Enhancing narratives with saymotion's text-to-3d animation and llms," in *ACM SIGGRAPH 2024 Real-Time Live!*, ser. SIGGRAPH '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3641520.3665309

[22] M. Behravan and D. Gracanin, "From voices to worlds: Developing an ai-powered framework for 3d object generation in augmented reality," 2025. [Online]. Available: https://arxiv.org/abs/2503.16474

[23] M. Behravan, K. Matković, and D. Gracanin, "Generative ai for context-aware 3d object creation using vision-language models in augmented reality," 01 2025, pp. 73–81.

[24] S. Aghel Manesh, T. Zhang, Y. Onishi, K. Hara, S. Bateman, J. Li, and A. Tang, "How people prompt generative ai to create interactive vr scenes," in *Designing Interactive Systems Conference*, ser. DIS '24. ACM, Jul. 2024, p. 2319–2340. [Online]. Available: http://dx.doi.org/10.1145/3643834.3661547

[25] Z. Yin, Y. Wang, T. Papatheodorou, and P. Hui, "Text2vrscene: Exploring the framework of automated

text-driven generation system for vr experience," 03 2024, pp. 701–711.

[26] D. Giunchi, N. Numan, E. Gatti, and A. Steed, "Dreamcodevr: Towards democratizing behavior design in virtual reality with speech-driven programming," 03 2024, pp. 579–589.

[27] J. Chen, J. Grubert, and P. O. Kristensson, "Analyzing multimodal interaction strategies for llm-assisted manipulation of 3d scenes," 2024. [Online]. Available: https://arxiv.org/abs/2410.22177

[28] E. Mann, J. Dortheimer, and A. Sprecher, "Toward a generative pipeline for an ar tour of contested heritage sites," 12 2022.

[29] M. Lataifeh, I. Afyouni, Z. A. S. Shaduly, A. Abdulkarim, and N. Ahmed, "An adaptive multimodal framework for designing intelligent virtual agents in mixed reality," in *Companion Proceedings of the 30th International Conference on Intelligent User Interfaces*, ser. IUI '25 Companion. New York, NY, USA: Association for Computing Machinery, 2025, p. 133–136. [Online]. Available: https://doi.org/10.1145/3708557.3716361

[30] F. R. Christiansen, L. N. Hollensberg, N. B. Jensen, K. Julsgaard, K. N. Jespersen, and I. Nikolov, "Exploring presence in interactions with llm-driven npcs: A comparative study of speech recognition and dialogue options," in *Proceedings of the 30th ACM Symposium on Virtual Reality Software and Technology*, ser. VRST '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3641825.3687716

[31] B. S. Gunawardhana, Y. Zhang, Q. Sun, and Z. Deng, "Toward user-aware interactive virtual agents: Generative multi-modal agent behaviors in vr," in *2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2024, pp. 1068–1077.

[32] K. B. Buldu, S. Özdel, K. H. Carrie Lau, M. Wang, D. Saad, S. Schönborn, A. Boch, E. Kasneci, and E. Bozkir, "Cuify the xr: An open-source package to embed llm-powered conversational agents in xr," in *2025 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR)*. IEEE, Jan. 2025, p. 192–197. [Online]. Available: http://dx.doi.org/10.1109/AIxVR63409.2025.00037

[33] Z. Li, H. Zhang, C. Peng, and R. Peiris, "Exploring large language model-driven agents for environment-aware spatial interactions and conversations in virtual reality role-play scenarios," in *2025 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, 2025, pp. 1–11.

[34] S. Sonlu, B. Bendiksen, F. Durupinar, and U. Güdükbay, "The effects of embodiment and personality expression on learning in llm-based educational agents," 2024. [Online]. Available: https://arxiv.org/abs/2407.10993

[35] M. Fiore, M. Gattullo, and M. Mongiello, "First steps in constructing an ai-powered digital twin teacher: Harnessing large language models in a metaverse classroom," in *2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, 2024, pp. 939–940.

[36] J. Zhu, R. Kumaran, C. Xu, and T. Höllerer, "Free-form conversation with human and symbolic avatars in mixed reality," 10 2023, pp. 751–760.

[37] Z. Li, P. P. Babar, M. Barry, and R. L. Peiris, "Exploring the use of large language model-driven chatbots in virtual reality to train autistic individuals in job communication skills," in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3613905.3651996

[38] Z. Zhu, A. Yu, X. Tong, and P. Hui, "Exploring llm-powered role and action-switching pedagogical agents for history education in virtual reality," 2025. [Online]. Available: https://arxiv.org/abs/2505.02699

[39] 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 *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3654777.3676379

[40] Z. Li, C. Gebhardt, Y. Inglin, N. Steck, P. Streli, and C. Holz, "Situationadapt: Contextual ui optimization in mixed reality with situation awareness via llm reasoning," 2024. [Online]. Available: https://arxiv.org/abs/2409.12836

[41] S. Srinidhi, E. Lu, A. Singh, S. Kartik, A. Lin, T. Laroia, and A. Rowe, "An xr platform that integrates large language models with the physical world," in *Proceedings of the 23rd ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 700–701. [Online]. Available: https://doi.org/10.1145/3715014.3724366

[42] A. Y. Cheng, M. Guo, M. Ran, A. Ranasaria, A. Sharma, A. Xie, K. N. Le, B. Vinaithirthan, S. T. Luan, D. T. H. Wright, A. Cuadra, R. Pea, and J. A. Landay, "Scientific and fantastical: Creating immersive, culturally relevant learning experiences with augmented reality and large language models," in *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, ser. CHI '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3613904.3642041

[43] L. Chen, Y. Cai, R. Wang, S. Ding, Y. Tang, P. Hansen, and L. Sun, "Supporting text entry in virtual reality with large language models," in *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, 2024, pp. 524–534.

[44] E. Oliveira, D. Silva, and A. Filho, "Improving vr accessibility through automatic 360 scene description using

multimodal large language models," 09 2024, pp. 289–293.

[45] H.-R. Tsai, S.-K. Chiu, and B. Wang, "Gazenoter: Co-piloted ar note-taking via gaze selection of llm suggestions to match users' intentions," in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, ser. CHI '25. ACM, Apr. 2025, p. 1–22. [Online]. Available: http://dx.doi.org/10.1145/3706598.3714294

[46] B. Yang, Y. Guo, L. Xu, Z. Yan, H. Chen, G. Xing, and X. Jiang, "Socialmind: Llm-based proactive ar social assistive system with human-like perception for in-situ live interactions," 2024. [Online]. Available: https://arxiv.org/abs/2412.04036

[47] Z. Wang, L.-P. Yuan, L. Wang, B. Jiang, and W. Zeng, "Virtuwander: Enhancing multi-modal interaction for virtual tour guidance through large language models," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, ser. CHI '24. ACM, May 2024, p. 1–20. [Online]. Available: http://dx.doi.org/10.1145/3613904.3642235

[48] D. Kim and J.-W. Jeong, "Designing llm response layouts for xr workspaces in vehicles," 12 2024, pp. 1–2.

[49] K. Tracy and O. Spantidi, "Impact of gpt-driven teaching assistants in vr learning environments," *IEEE Transactions on Learning Technologies*, vol. PP, pp. 1–14, 01 2025.

[50] S. Wen, M. Middleton, S. Ping, N. N. Chawla, G. Wu, B. S. Feest, C. Nadri, Y. Liu, D. Kaber, M. Zahabi, R. P. McMahan, S. Castelo, R. Mckendrick, J. Qian, and C. Silva, "Adaptivecopilot: Design and testing of a neuroadaptive llm cockpit guidance system in both novice and expert pilots," 2025. [Online]. Available: https://arxiv.org/abs/2501.04156

[51] N. Constantinides, A. Constantinides, D. Koukopoulos, C. Fidas, and M. Belk, "Culturai: Exploring mixed reality art exhibitions with large language models for personalized immersive experiences," 06 2024, pp. 102–105.

[52] A. Rychert, M. L. Ganuza, and M. N. Selzer, "Integrating gpt as an assistant for low-cost virtual reality escape-room games," *IEEE Computer Graphics and Applications*, vol. 44, no. 4, pp. 14–25, 2024.

[53] H. Javaheri, O. Ghamarnejad, P. Lukowicz, G. A. Stavrou, and J. Karolus, "Aras: Llm-supported augmented reality assistance system for pancreatic surgery," in *Companion of the 2024 on ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 176–180. [Online]. Available: https://doi.org/10.1145/3675094.3677543

[54] J. Murray, "Introduction to roslyn: The compiler as a service," 2024.

[55] C. Creed, M. Al-Kalbani, A. Theil, S. Sarcar, and I. Williams, "Inclusive augmented and virtual reality: A research agenda," *International Journal of Human–Computer Interaction*, vol. 40, no. 20, pp. 6200–6219, 2024.

[56] R. Gallotta, G. Todd, M. Zammit, S. Earle, A. Liapis, J. Togelius, and G. N. Yannakakis, "Large language models and games: A survey and roadmap," *IEEE Transactions on Games*, p. 1–18, 2024. [Online]. Available: http://dx.doi.org/10.1109/TG.2024.3461510

[57] L. Seymour, B. Kutukcu, and S. Baidya, "Large language models on small resource-constrained systems: Performance characterization, analysis and trade-offs," 2024. [Online]. Available: https://arxiv.org/abs/2412.15352

[58] X. Zhu, J. Li, Y. Liu, C. Ma, and W. Wang, "A survey on model compression for large language models," 2024. [Online]. Available: https://arxiv.org/abs/2308.07633

[59] A. Avetisyan, C. Xie, H. Howard-Jenkins, T.-Y. Yang, S. Aroudj, S. Patra, F. Zhang, D. Frost, L. Holland, C. Orme, J. Engel, E. Miller, R. Newcombe, and V. Balntas, "Scenescript: Reconstructing scenes with an autoregressive structured language model," 2024. [Online]. Available: https://arxiv.org/abs/2403.13064

[60] Z. Liu, S. Zheng, S. Chen, C. Zhao, L. Liang, X. Xue, and Y. Fu, "A neural representation framework with llm-driven spatial reasoning for open-vocabulary 3d visual grounding," 2025. [Online]. Available: https://arxiv.org/abs/2507.06719

[61] Y. Tang, J. Situ, and Y. Huang, "Beyond user experience: Technical and contextual metrics for large language models in extended reality," in *Companion of the 2024 on ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 640–643. [Online]. Available: https://doi.org/10.1145/3675094.3678995

[62] J. Liu, X. Xu, X. Nan, Y. Li, and Y. Tan, ""generate" the future of work through ai: Empirical evidence from online labor markets," 2025. [Online]. Available: https://arxiv.org/abs/2308.05201

[63] B. C. Das, M. H. Amini, and Y. Wu, "Security and privacy challenges of large language models: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2402.00888

[64] E. Bozkir, S. Özdel, K. H. C. Lau, M. Wang, H. Gao, and E. Kasneci, "Embedding large language models into extended reality: Opportunities and challenges for inclusion, engagement, and privacy," in *ACM Conversational User Interfaces 2024*. ACM, Jul. 2024, p. 1–7. [Online]. Available: http://dx.doi.org/10.1145/3640794.3665563

[65] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?" in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 610–623. [Online]. Available: https://doi.org/10.1145/

3442188.3445922

[66] T. Maeda and A. Quan-Haase, "When human-ai interactions become parasocial: Agency and anthropomorphism in affective design," in *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 1068–1077. [Online]. Available: https://doi.org/10.1145/3630106.3658956

[67] A. Fasth, Fast-Berglund, L. Gong, and D. Li, "Testing and validating extended reality (xr) technologies in manufacturing," *Procedia Manufacturing*, vol. 25, pp. 31–38, 01 2018.