MRPilot: A Mixed-Reality System for Responsive Navigation of General Procedural Tasks

Hongliang Yang
Shenzhen University

Jin Zhou
Shenzhen University

Pengfei Xu*
Shenzhen University

Hongbo Fu
HKUST

Hui Huang
Shenzhen University

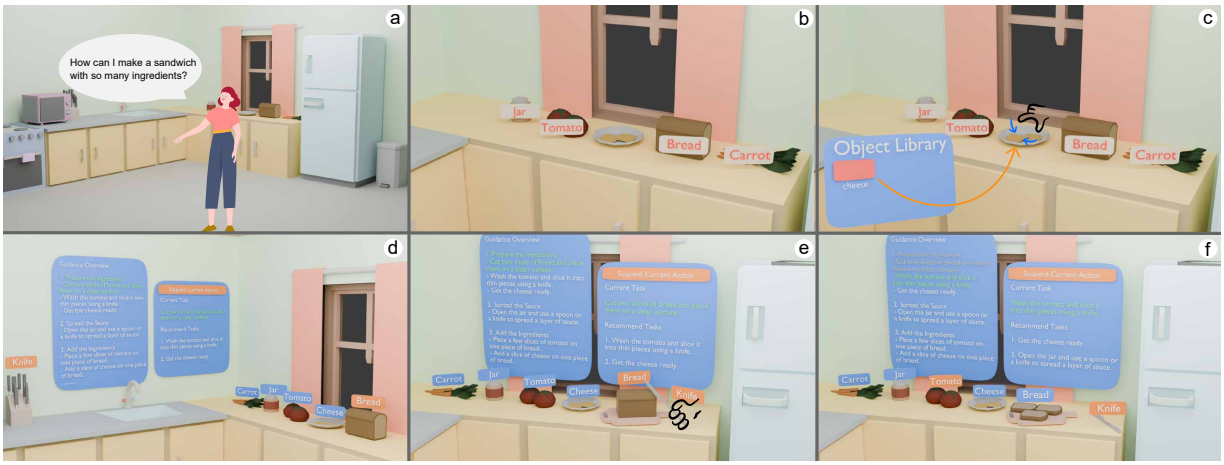

Figure 1: Overview of *MRPilot*: In automatic anchoring mode, labels displaying object names will be overlaid on the recognized objects (b). Once the user confirms the anchor, the object anchor is overlaid on the corresponding object (c). The users can also use hand gestures to manually anchor objects that cannot be automatically recognized left in the hand panel (c). *MRPilot* can automatically detect and track physical objects during the whole task progress (d). It provides users with responsive guidance by monitoring their actions, detecting progress (e), and automatically advancing instructions (f).

Abstract

People often need guidance to complete tasks with specific requirements or sophisticated steps, such as preparing a meal or assembling furniture. Traditional guidance often relies on unstructured paper instructions that require people to switch between reading instructions and performing actions, resulting in an unsmooth user experience. Recent Mixed Reality (MR) systems alleviate this problem by giving spatialized navigation but demand an authoring step and, therefore, cannot be easily adapted to general tasks. We propose *MRPilot*, an MR system empowered by Large Language Models (LLMs) and Computer Vision techniques, offering responsive navigation for general tasks without pre-authoring. *MRPilot* consists of three modules: a **Navigation Builder Module** using LLMs to generate structured instructions, an **Object Anchor Module** exploiting Computer Vision techniques to anchor physical objects with virtual proxies, and an **Action Recommendation Module** giving responsive navigation according to users' interactions with physical objects. *MRPilot* bridges the gap between virtual instructions and physical interactions for general tasks, providing contextual and responsive navigation. We conducted a user study to compare *MRPilot* with a baseline MR system that also exploited LLMs. The results confirmed the effectiveness of *MRPilot*.

Index Terms: Mix Reality, Task Guidance System, Context-aware Interaction, Large Language Models.

*Corresponding author, e-mail: xupengfei.cg@gmail.com

1 Introduction

Activities such as cooking, assembling furniture, and personal care often require interacting with various objects that involve complex steps due to their functional diversity and intricate design. In real-world environments, people frequently encounter challenges due to the complexity of these tasks, making clear and structured instructions crucial for their successful completion. For example, assembling furniture may involve numerous components that need to be aligned and connected in a precise order, where any misstep could compromise the stability of the entire structure. Therefore, step-by-step navigation is crucial, offering users the support they require to confidently and efficiently navigate complex tasks in daily life.

However, following such instructions can be challenging, as users often need to switch constantly between reading instructions and performing actions. For instance, when cooking spaghetti, users must juggle multiple tasks, such as chopping onions, boiling pasta, and adding ingredients according to the recipe. This process demands constant attention-alternating between reading the recipe and carrying out actions such as measuring spices or stirring the sauce. This frequent back-and-forth increases cognitive load [13], especially when users need to follow the task sequence accurately and measure ingredients precisely. As a result, users often have to recheck their recipes frequently, slowing down the process and increasing the likelihood of mistakes, such as overcooking.

To address this challenge, recent advances in Mixed Reality (MR) technology offer solutions by providing pre-authored immersive navigation within MR experiences in specific domains such as cooking [32], dancing [42], assembling furniture [37, 38], and operating machines [23]. However, current MR solutions typically rely on fixed, pre-defined guidelines [5, 36, 49, 14], making it challenging to adapt to dynamic and customized tasks, thereby limiting their flexibility and generalizability in practical scenarios. An effective

navigation system should accommodate users' customized needs, supporting a variety of general tasks without pre-authoring.

Another area for improvement with existing MR task instruction systems is the need for more responsive navigation for users' actions. Current systems [5, 32] offer virtual step-by-step instructions within a convenient range but require manual switching between each step, which can interrupt the user's workflow. Responsive navigation, by contrast, automatically adjusts to user actions and provides real-time feedback, reducing the need for manual input to progress through instructions. Drawing inspiration from advancements in Virtual Reality (VR), which involve tracking user interactions with virtual objects [51], we aim to incorporate similar user-tracking mechanisms into MR systems to enable responsive navigation. By doing so, we can reduce the need for manual switching and enhance the system's adaptability to user actions.

This paper introduces *MRPilot*, an intelligent MR system designed to provide responsive navigation to assist users in accomplishing general procedural tasks without pre-authoring. That is, the system requires no manual per-task setup, such as authoring predefined navigation or tailoring it to the environment in advance. Our solution is motivated by the need for flexible, real-world navigation that dynamically adapts to users' environments, actions, and needs. Unlike previous systems that are confined to task-specific use cases, *MRPilot* leverages advanced Large Language Models (LLMs) and Computer Vision techniques to understand users' requirements and physical environments. Our system generates organized, step-by-step procedures and anchors task-related physical objects in the user's real-world environment. Additionally, during the task consumption process, our system adaptively recommends possible next steps to users as they progress through each step. This significantly reduces the cognitive burden on users and enhances the accuracy with which tasks are completed. We develop a prototype system on a Head-Mounted Display (HMD) MR device (*Meta Quest 3*). In this prototype system, we leverage the power of LLMs in our **Navigation Builder Module** to build a structured step-by-step guide. Our designed **Object Anchor Module** utilizes the built-in depth sensors of the HMD and an open vocabulary object detection technique, *YOLO-world* [6], to automatically anchor task-related interactive objects, thereby enhancing the connection between instructional content and the physical environment. Moreover, we develop a **Step Recommendation Module** that alleviates the need for manual step switching, significantly reducing the user's cognitive load and improving the overall user experience.

We conducted a comprehensive user study involving 21 university students to assess the usability and performance of the proposed *MRPilot* system in comparison to a baseline system that also integrates LLMs. The study allowed participants to interact with both systems through designated and user-customized tasks detailed in Table 1. Extensive evaluations between *MRPilot* and the baseline demonstrated the superior performance of *MRPilot* in task completion and adaptive guidance.

In this paper, we made the following contributions:

- A flexible system without pre-authoring supporting general procedural tasks, leveraging LLMs and computer vision techniques to generate organized step-by-step navigation.
- A responsive navigation mechanism that adapts to users' actions and their environments, enhancing workflow performance during task progression.
- An MR user interface for visualizing structured instructions and current step-related visual cues.
- A user study evaluating the performance of our system compared to a baseline MR system.

2 Related Work

2.1 Navigation Systems for Procedure Tasks

In recent years, significant research [43, 14, 7, 25, 29, 31, 47, 21, 54, 49, 36] has been conducted on guide and navigation systems designed to assist users in completing procedural tasks across various contexts in the HCI community. A common approach in this domain involves utilizing specialized devices or sensors, such as specially designed utility or wearable technology, to monitor user actions and environmental factors. Panavi [43] introduced a sensor-equipped frying pan that delivered real-time, context-aware data such as temperature. Similarly, MimiCook [32] integrated a depth camera, a projector, and a scaling device to provide step-by-step navigation on a kitchen counter, adapting instructions to a user's progress during cooking. PrISM-Observer [2] employed wrist-worn devices to track hand movements and auditory cues, preventing user errors during daily activities. This system predicted when the user might forget or incorrectly perform a step, intervening with reminders before errors occur or notifications if a step was missed. These sensor-driving guidance systems ensure accurate and real-time feedback but are often limited by the requirement for additional hardware, which can be task-specific and may not generalize well across different scenarios. XaiR [35] leverages action logging and video summarization techniques to record expert demonstrations, processing these multimodal data through multimodal large language models (MLLMs) to enhance guidance accuracy. While this method eliminates specialized hardware requirements by utilizing common video recording devices, its reliance on pre-configured expert workflows introduces certain operational constraints. For instance, XaiR assumes environmental objects (e.g., cooking ingredients) should closely correspond to those documented in pre-recorded guidance sequences, which may potentially limit adaptability when users operate in varied environmental conditions.

Another line of research focuses on tracking task progress for better navigation. For instance, live video streams combined with neural networks have been used to support procedural tasks such as industrial [38] and origami [36] assembly. Similarly, procedural guides have been automatically extracted from teaching videos [49], providing step-by-step instructions. However, these approaches still rely on a manual authoring process, which can limit their scalability and adaptability. Systems like AMMA [51] and TutoriVR [41] offered immersive environments where users could practice and complete cooking and sketching with adaptive guidance. These systems effectively provided real-time feedback and error management, allowing users to correct mistakes in a controlled virtual setting. However, their applications were often constrained by the challenge of translating virtual instructions into real-world scenarios, where environmental variability could affect the systems' effectiveness. While some works [4, 55, 24, 11] attempted to address this issue, they often required additional sensors and are confined to specific domains.

Additionally, advancements in LLMs [8, 3] and Computer Vision techniques have contributed to the development of navigation systems that enhance task assistance [40, 5, 9, 48, 18, 44]. PaperToPlace [5] transformed traditional paper-based instructions into step-by-step virtual labels contextually linked to physical objects using BERT [8]. Flaivor [9] used vision LLMs to generate cooking recipes from a photo of food. By integrating LLMs and Computer Vision techniques, these systems aimed to seamlessly bridge the gap between the digital and physical realms. However, these systems lack support for responsive guidance. Users often need to manually click virtual buttons to navigate between steps, which disrupts their workflow and increases cognitive load. In contrast, our system, *MRPilot*, addresses this issue by using virtual object anchors to track users' interactions with real-world objects, automatically recommending and transitioning to the next step. This approach enables seamless task switching and minimizes interrup-

tions.

2.2 Interaction with Real World Objects in AR/MR Environments

In the field of AR/MR, interaction with real-world objects is a critical component for creating immersive and intuitive experiences. Several systems have been developed to explore object anchoring [10, 50, 1, 28], augmentation [20, 19], and interaction techniques [52, 53, 26, 12, 17, 16] within AR/MR environments, each aiming to enhance user engagement and improve task performance.

Prior works have explored innovative approaches to object anchoring and physical interaction in AR environments. ProObjAR [53] enabled designers to prototype spatial interactions with smart objects using AR-HMDs and AR markers for tracking. InfoLED [50] used high-frequency flickering in LED indicator lights for device positioning and communication, while LightAnchors [1] similarly leveraged point lights to anchor digital information without modifying the physical environment. These frameworks relied on predefined or static markers, limiting their adaptability in dynamic scenarios. Meanwhile, systems like Teachable Reality [26], UbiEdge [12], and Ubi-TOUCH [17] focused on enhancing physical interaction through object pre-registration and haptic feedback. These systems enabled tangible interactions with virtual content by leveraging everyday objects, although they faced scalability challenges across different environments.

Object detection methods [45, 6] play a crucial role in identifying and interacting with physical objects within AR/MR environments. XR-Objects [10] proposed a novel paradigm for integrating physical objects as interactive entities in Extended Reality (XR). By combining object segmentation and classification with Multimodal Large Language Models (MLLMs), XR-Objects allowed users to interact with physical objects as if they were digital, offering a seamless blend of the physical and digital worlds. This system opened new possibilities for creating contextually relevant interactions but primarily focused on augmenting individual objects with MLLMs, without establishing strong inter-object relationships.

In summary, while these systems offered various techniques for enhancing interaction with real-world objects in AR/MR environments, they often faced challenges related to integration with dynamic procedure navigation. Our work, *MRPilot*, introduces an object anchoring method that automatically detects and links real-world objects to procedural steps. Our system aims to reduce the cognitive load on users by eliminating the need for manual step-switching and enhancing the overall interaction experience within immersive MR environments.

3 Design Rationale

3.1 User Scenario

Consider the following scenarios: *1) Emily has a minor cut on her arm and begins cleaning and dressing the wound, but skips sterilizing her hands, risking infection. 2) Tom, a novice cook, attempts a complex sandwich recipe. He is overwhelmed by the numerous ingredients and steps, constantly referring to the recipe and following it rigidly, which causes delays and stress. His fear of mistakes discourages him from improving his cooking skills. 3) Alex, another novice cook, attempts to make spaghetti but lacks the key ingredients. He struggles with substitutions and making technique adjustments, which leads to frustration and delays.*

In each case, a context-aware assistant providing timely guidance could help users avoid mistakes and reduce frustration. Our MR-HMD system is designed with these principles: *1) Emily receives clear, step-by-step instructions to prevent skipping important steps. 2) Tom benefits from adaptive navigation that streamlines tasks and reduces overwhelm. 3) Alex is supported in ingredient substitutions and recipe adaptations for efficient meal preparation.*

3.2 Design Consideration

Adaptive Task Flow vs. Static Instruction Sequences. Prior task guidance systems [5, 11, 32, 23] often present static, pre-defined instruction sequences that do not account for the user's progress or environment. With such systems, users are required to manually navigate between steps or stages, which has been shown to cause inefficiencies during workflow deviations [13]. For example, users often pause to correct mistakes or adjust environmental variations, which increases cognitive burden [39].

In contrast, *MRPilot* adopts an adaptive task flow approach that dynamically responds to the user's actions and environment. By leveraging real-time sensor data and object detection, the system automatically updates and presents the next appropriate step based on the user's progress. This design aims to reduce manual intervention and streamline processes, potentially alleviating cognitive demands associated with task switching. Additionally, *MRPilot* explores idle time management by detecting periods of user inactivity. When no actions are detected for a predefined duration, the system can suspend the current task and suggest alternative interim tasks. This feature seeks to minimize unproductive waiting periods while maintaining workflow continuity.

World-Anchored Navigation vs. Headset-Centered Display. In MR environments, the choice between world-anchored navigation (placing digital content within the physical space) and headset-centered displays (attaching UI elements to the user's field of view) may affect immersion and task execution [39]. Headset-centered displays, though accessible, can detract from the sense of presence by imposing a digital interface on top of the real-world view, thus breaking the spatial coherence between the task and its navigation.

To maintain spatial consistency and minimize cognitive disruption, *MRPilot* employs world-anchored navigation, where instructions and digital cues are overlaid onto the physical objects relevant to the task. This allows users to naturally interact with their environment while receiving real-time navigation in context, fostering a deeper connection between digital and physical elements. By aligning the interface with the physical task space, the system seeks to leverage users' spatial awareness for navigating both the task and instructional content, which could potentially address orientation challenges and support task execution.

4 *MRPilot* System

We develop *MRPilot*, an MR system that provides users with guidance to assist them in accomplishing procedural tasks. Based on the previous discussion, we identified the following features in our system: 1) enabling users to receive procedure instructions for general tasks generated by *MRPilot* according to their needs, 2) scanning the environment to provide context-aware navigation and anchor spatial tags over physical objects, and 3) providing a responsive task consumption interface that delivers real-time guidance for task completion. In this section, we discuss the design of *MRPilot* and present our user interface.

4.1 Problem Formulation

Our work primarily focuses on guiding users through their customized requirements for general procedural tasks. These tasks involve a complex sequence of human-object interactions, which can be transformed into a structured, step-by-step process.

We define a structured task as comprising several **major subtasks**, each of which consists of multiple **actions**. Each action is classified into one of four possible statuses: 1) **Not started**: the user has not yet begun this action, 2) **In progress**: the user is actively working on this action, 3) **Suspended**: the action has been temporarily paused while the user focuses on another action, and 4) **Completed**: the user has finished this action.

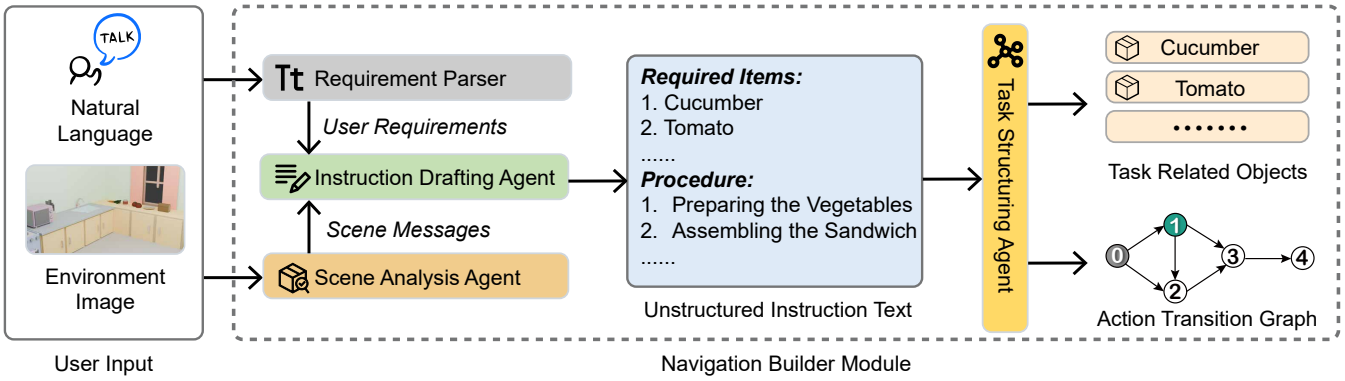

Figure 2: Task preparation of *MRPilot*. A user starts by providing textual instructions via speech or text, followed by an image capture of the environment. These inputs are processed by the **Navigation Builder Module**, which identifies task-relevant objects and generates an action transition graph.

4.2 System Overview

The *MRPilot* system operates through the following steps:

1) Providing textual requirements: The user provides a task description along with an image of the environment to our **Navigation Builder Module** powered by ChatGPT. The navigation builder then returns a set of step-by-step structured instructions for performing the task. The user can either accept, discard, or modify these instructions (see Figure 2).

2) Scanning environment and anchoring physical objects: After the user accepts the instructions, they move through the physical environment to scan relevant objects and their locations, anchoring spatial tags that include the object names within the scene. A user-friendly manual labeling method is also provided to allow users to anchor objects that cannot be automatically detected by **Object Anchor Module** (see Figure 1 (b, c)).

3) Adaptive task completion: *MRPilot* utilizes spatial anchors and step-by-step instructions to offer responsive navigation, assisting the user throughout the task completion process (see Figure 3).

4.3 Navigation Builder Module

The **Navigation Builder Module** is designed to translate users' customized task requirements into a structured set of step-by-step instructions based on the user's intended task and scene information. This process is accomplished by a set of specialized LLM Agents utilizing the ChatGPT API, each responsible for distinct stages of task analysis and instruction generation. The system leverages the following three LLM agents to fulfill this function.

4.3.1 Scene Analysis Agent

The **Scene Analysis Agent** is responsible for analyzing images captured by the user's HMD device in conjunction with the task instructions provided by the user. This agent identifies and filters relevant objects within the user's environment that are necessary for completing the task at hand. By combining visual data with user-provided directives, the agent ensures that only task-relevant items from the physical environment are highlighted for further guidance.

4.3.2 Instruction Drafting Agent

Once the **Scene Analysis Agent** has identified and filtered the necessary objects from the environment, the **Instruction Drafting Agent** takes over. This agent generates an unstructured task instruction document in plain text based on the user's task requirements and the contextual information from the scene analysis. The unstructured navigation draft provides a natural language description of the task steps and objects involved, laying the foundation for the next phase of instruction processing.

4.3.3 Task Structuring Agent

The final step in the process is handled by the **Task Structuring Agent**, which converts the unstructured task description into a structured set of step-by-step instructions. This agent carefully analyzes and breaks down the unstructured document provided by the **Instruction Drafting Agent**, identifying individual steps and the corresponding objects involved at each stage of the task. The result is a detailed, structured set of instructions that guides the user through task completion in an organized manner, following our predefined task structure in Section 4.1. This ensures that all steps are clearly defined and logically sequenced.

Together, these three agents form the backbone of the Navigation Builder Module, providing the capability to dynamically interpret user needs and translate them into actionable, structured task instructions tailored to the specific environment and needs.

4.4 Object Anchor Module

MRPilot uses the Meta Quest 3 MR-HMD as its front-end platform. To anchor physical objects in the environment that are relevant to the task guidance generated by the **Navigation Builder Module**, we capture the RGB pass-through image stream from the Quest 3. When the user enters automatic object anchoring mode, *MRPilot* streams these images from the HMD's field of view (FOV) to a dedicated server. The RGB image stream is processed using an open-vocabulary object detection algorithm [6] to extract the semantic meaning and screen-space locations of the relevant objects identified by the **Navigation Builder Module**. We then utilize the Quest 3's built-in depth sensor to perform ray casting, converting the screen-space coordinates into world-space locations to overlay virtual objects onto the corresponding real-world task-relevant objects, as shown in Figure 1 (b), (c), and (d). These object anchors are then used to highlight the physical objects during the task flow.

However, due to the limitations of [45, 6], it is not always possible to detect the semantic information of all task-relevant objects. To address this, we provide a convenient and user-friendly manual anchoring interface, allowing users to manually anchor objects that cannot be automatically detected (Figure 1 (c)).

4.5 Action Recommendation Module

Once all task-related objects are properly anchored, the **Action Recommendation Module** suggests the next steps in the user's task completion process based on the current task status. We use a task recommendation agent powered by LLMs, followed by a task filtering algorithm. The recommendation agent is activated when an action transitions to the "In progress" status mentioned in section 4.1. The status of the user's task is sent to the recommendation agent,

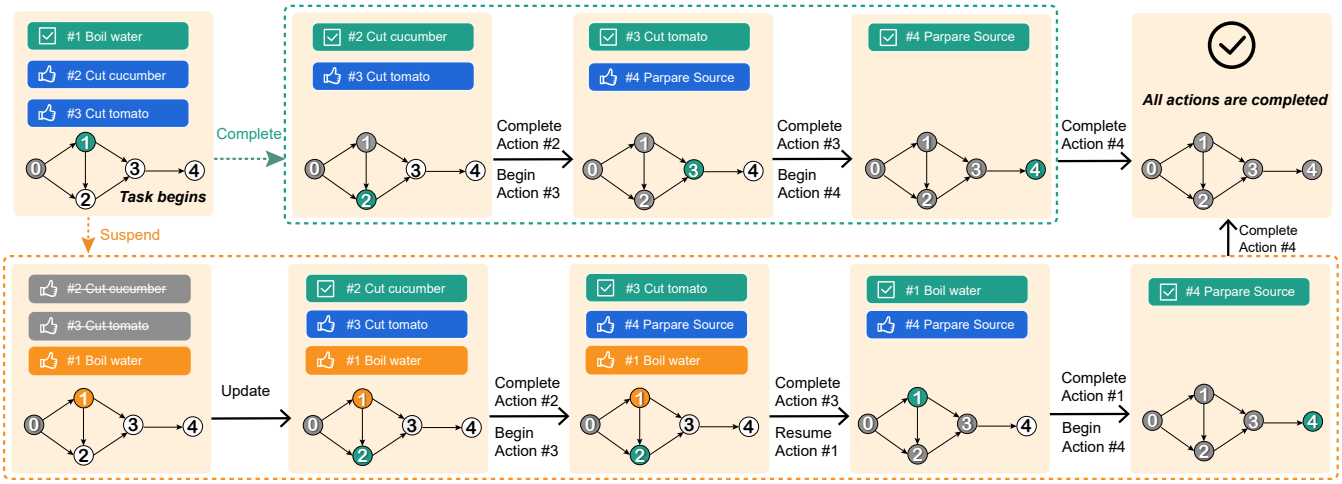

Figure 3: A toy example to show how the **Action Recommendation Module** works. The user can adaptively select the recommended actions, and the **Action Recommendation Module** will suggest the following steps according to the user's selection. Two choices result in two different sequences of actions, and both sequences lead to the accomplishment of the task.

where it is processed using the Chain of Thought (CoT) [46] reasoning principle. From the original response text, we extract parsable recommended steps and then apply our task filtering algorithm, as illustrated in Algorithm 1, to filter out recommended steps with obvious errors. This filtering mechanism aims to reduce redundant suggestions and mitigate hallucinations in the actions generated by the LLM. By removing inconsistent steps such as completed and ongoing actions, Algorithm 1 helps improve the quality and relevance of the recommended task sequence. Figure 3 illustrates how the **Action Recommendation Module** suggests the next steps with a toy task. In this example, the users want to make a meal that requires several actions, including boiling water, cutting cucumbers, cutting tomatoes, and preparing sauce. The users can adaptively select the recommended actions, and the **Action Recommendation Module** will suggest steps according to the user's selection. For example, when the users enter the action of "boil water", they can either wait until the hot water is ready or suspend the "boil water" action and continue with other actions. Two choices result in two different sequences of actions, and both sequences lead to the accomplishment of the task.

Algorithm 1 Action Recommendation Filtering Algorithm

Input: List of original recommend actions T, Current action C, Recommendation action count limit L
Output: List of recommended actions R
Initialize an empty list R and a counter $rCount \leftarrow 0$
if a suspended action exists and C is not suspended **then**
    Add the suspended action to R
    Increment $rCount$
end if
for each action in T **do**
    Extract the status of the action
    if the action is neither Completed, InProgress, nor Suspended **then**
        Add the action to R
        Increment $rCount$
    end if
    if $rCount \geq L$ **then**
        break
    end if
end for
if R is empty **then**
    Add the first NotStarted action from the list of all tasks to R
end if
return R

4.6 MR User Interface

We present an MR user interface that integrates all the functions discussed above, along with additional features such as step visualization and natural interaction. The process of using *MRPilot* consists of three main steps: 1) the task preparation mode, where user requirements and contextual environments are gathered; 2) the object anchoring mode, where virtual tags are anchored to physical objects; and 3) the task consumption mode, where users follow the generated instructions to finish the task. The interface panel is displayed in front of the user's view, allowing easy access to all the information and functions provided by *MRPilot*, as well as facilitating seamless switching between them.

As shown in Figure 4 (a), the user enters task preparation mode by providing a requirement description using a voice command by clicking the *Tell Us With Voice* button. The user then speaks to the system to specify his/her goal. After confirming the request, the user enters the scene capture panel to take an image of the physical environment (Figure 4 (b)). This allows the system to identify relevant objects using gesture recognition and then send this information to the corresponding agents. Once the task navigation drafts are returned, the user can review the generated context and either *accept* the navigation or *discard* it to request a new one by clicking the corresponding button (Figure 4 (c)).

Upon accepting the generated instructions, the user can begin marking physical objects by clicking the *Start Tracking* button, which activates object anchoring mode (Figure 4 (d)). The user can then confirm the anchor in world space using hand gestures. A 3D virtual glow is overlaid on each corresponding physical object, and the user can adjust its position, rotation, and scale through the built-in freehand interactions. Additionally, we provide a manual anchoring interface that allows the user to summon object anchors by simply pinching the highlighted real-world objects on the hand panel in case automatic anchoring fails to capture all objects.

After completing object anchoring mode, the user enters task consumption mode. The user can easily monitor the overall task completion status via the task overview panel (Figure 5 (a)). Besides, the user can seamlessly switch between actions recommended by the **Action Recommendation Module** by interacting with action-related objects or directly clicking the recommended step text. During task consumption mode, the visual effects of object anchors dynamically adjust based on the current action step,

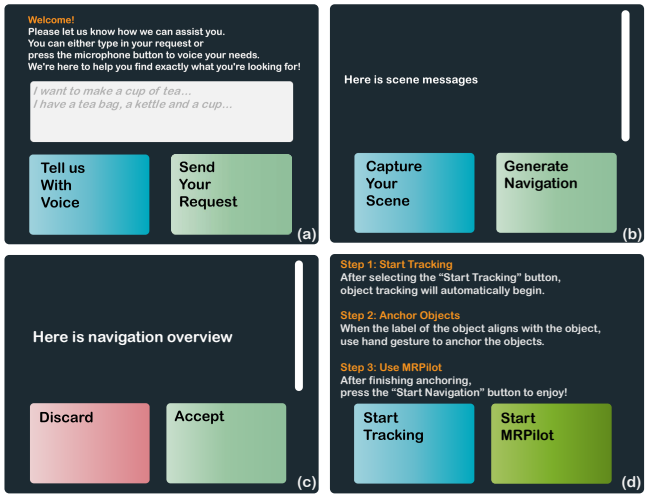

Figure 4: Preparation user interface. (a) The user provides task instructions via voice or text to initiate task preparation mode. Note that this panel is also used in *Baseline* system. (b) The user captures the scene for object recognition using the *"Capture Your Scene"* button. (c) This panel displays the generated instruction draft, where the user can either accept or discard it. (d) The user enters Object Anchoring mode by clicking *"Start Tracking"*, allowing *MRPilot* to anchor virtual visual cues above physical objects in the environment. After all objects are anchored, the user can click the *"Start MRPilot"* button to begin task navigation.

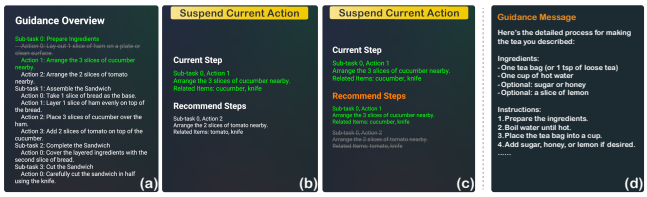

Figure 5: Task consumption mode user interface. (a) Guidance overview panel. (b) The task navigation panel shows the current step and recommended steps. (c) Upon selecting a recommended step, other recommended steps will be discarded. (d) Feedback from the LLM displays in *Baseline*, where users can also regenerate the result using the interface in Figure 4 (a).

changing the glow color accordingly. Anchors related to completed steps that are no longer needed for future actions will be automatically removed. Additionally, we implement an anchor navigation feature: when an object anchor related to the current task is outside the user's field of view, a screen-space indicator provides spatial cues to navigate the user toward the anchor's location.

We have also implemented an action suspension feature, allowing the user to suspend a long-running action (e.g., boiling water) by pressing the *Suspend Current Action* button (Figure 5 (b, c)) and switch to other actions for efficiency. When the user suspends an action, the recommendation module adjusts and recommends steps that do not depend on the suspended action. This enables the user to seamlessly continue progressing on other independent actions, maintaining workflow continuity and overall efficiency.

5 Implementation Details of *MRPilot*

We implemented *MRPilot* using the Meta Quest 3, which offers built-in SLAM tracking and a depth sensor to support MR experiences. The *MRPilot* interface was developed on a local PC (Intel Core i9-14900K CPU, 128GB RAM) using Unity 2022.3.45f1. During the scene environment and object anchoring modes, we used a resolution of 1024×1024 for the RGB image stream. The scene environment image was then processed by our **Scene Analysis Agent** (as described in section 4), which utilizes a multi-modal LLM, specifically *gpt-4o-mini* for the recommendation module and *gpt-4o* for other modules. This model was also employed in the development of other agents within the system.

We further evaluate the latency of our four agents by issuing 50 API requests per agent. On average, a single user request requires approximately 19 seconds to generate a complete response. All agents in our system incorporate a retry mechanism against request failures. Further details are provided in the supplemental materials.

For the object anchoring mode, images were processed locally using a PC equipped with a single NVIDIA RTX 4090 GPU. We leveraged the *Meta All-in-One SDK* [22] to enable seamless interaction between hand gestures, virtual objects, and the user interface. For object detection, we used a detection model [6] pre-trained on a variety of datasets [33, 15, 30, 34]. Object anchors were created in world space by converting screen-space 2D detection bounding boxes into 3D world-space positions, utilizing the depth texture provided by the Quest 3's depth sensor.

Table 1: Task descriptions using *MRPilot* or *Baseline*.

Task No.	Description
T1	The user is asked to brew a cup of tea to get familiar with the MR headset and the systems.
T2	The user is guided by *MRPilot* or *Baseline* to make a sandwich with randomized, precisely quantified ingredients under a variety of sandwich ingredients. A prototypical task involves making a sandwich containing exactly two ham slices, three cucumber slices, and four tomato slices; the sandwich must be fully heated.
T3	The user is guided by *MRPilot* or *Baseline* to change the dressing of a wound wrapped with gauze.
T4	The user is guided by *MRPilot* or *Baseline* to assemble a toy desk using interlocking wooden components.
T5	The user is guided by *MRPilot* or *Baseline* to perform a small chemistry experiment producing oxygen gas from hydrogen peroxide.

6 User Study

We conducted a user study to qualitatively evaluate the usability of our system. The evaluation involved all steps in our system, as well as a comparative study between *MRPilot* and a baseline system (*Baseline*) that also employed LLMs. During the evaluation, participants were asked to use *MRPilot* and *Baseline* for four tasks (T2-T5) that could be easily conducted in a typical office. Each study contained one session where each participant needed to complete two tasks. We used T1 as the training task, through which participants could get familiar with *MRPilot* and *Baseline*. T2-T5 were used for formal evaluations, each of which could be completed in around 10 minutes. Table 1 provides details of task requirements.

6.1 System Configuration

The study was conducted in an indoor environment. We developed a baseline MR system (*Baseline*), where users could interact with ChatGPT via voice commands and view instructions displayed on a virtual panel inside MR (see Figure 5 (d)). Rather than asking participants to use a tablet-based ChatGPT application, placing the ChatGPT application within the MR environment and rendering instructions on a virtual panel in front of the user helped to minimize the impact of confounding factors caused by discomfort from the

Table 2: Session descriptions using *MRPilot* or *Baseline*.

Session No.	*MRPilot*	*Baseline*
1	T2	T3
2	T3	T2
3	T4	T5
4	T5	T4

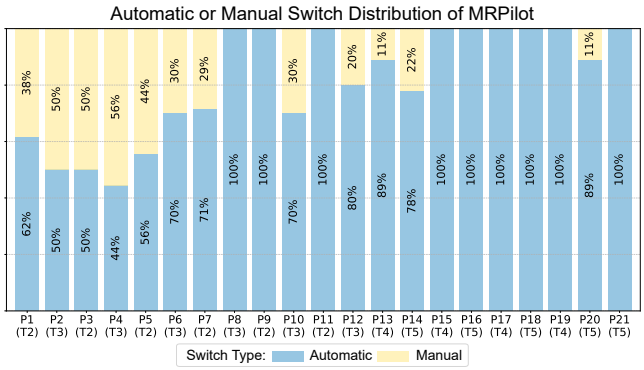

Figure 6: The distribution of automatic and manual switch proportions across individual participants. Percentages within each bar segment represent the relative frequency of each switch type reported by participants. Tasks completed by each participant using *MRPilot* are shown in parentheses. Detailed task assignments are provided in the supplemental material.

headset. We designed four experimental sessions, each comprising two task combinations, with counterbalancing applied to minimize prior learning effects and task familiarity. Before the session began, T1 was used to help participants learn and get familiar with both *MRPilot* and *Baseline*. During the session, participants were instructed to complete 1 session shown in Table 2. Note that we used T2 to test the generalizability of *MRPilot*, as it allowed users to customize their own sandwiches freely. This setting also enabled us to evaluate how well *MRPilot* could perform when assembling sandwiches with randomly selected ingredients, simulating more diverse and unpredictable user preferences.

6.2 Participants and Procedure

We invited 21 university students (aged between 22 and 25) as participants. 13 participants had no prior experience with HMD but were familiar with AR applications on tablets or mobile phones. 8 participants have used head-mounted VR/AR devices before. None of them had previously used our system or had any prior knowledge about it. The entire study lasted about one hour, and each participant was compensated with two delicious sandwiches.

Upon arrival, the participants were given a brief introduction to the study and asked to sign a consent form if they felt comfortable proceeding. We then explained the basic interaction methods with the Meta Quest device, such as how to click buttons on virtual panels using ray interaction or hand-poke interaction, and how to use hand gestures in Quest, since most participants had no experience in using HMD. The participants were introduced to the entire workflows of *MRPilot* and *Baseline*, along with the functions of the UI and hand panel, through a warm-up task T1 (making a cup of tea). Among the participants, 1 had prior experience using Meta Quest, though none had development experience with it. The participants were given sufficient time to familiarize themselves with *MRPilot* and *Baseline* before starting the official study.

To mitigate the potential confounding effects of user fatigue and learning bias towards *MRPilot* in later tasks, we strategically grouped the tasks into counterbalanced blocks during participant assignment. This grouping approach ensured approximately equivalent total execution durations across all experimental conditions, as shown in Table 2 and Figure 8. The grouping detail can be found in the supplemental material. After each task, participants were asked to fill out a System Usability Scale (SUS) questionnaire on a 5-point scale (1 indicates strongly disagree and 5 indicates strongly agree), as well as a NASA-TLX [27] questionnaire on a 5-point scale (the lower, the better) for both *MRPilot* and *Baseline*. In addition, a 15-minute interview was conducted with each participant.

6.3 Result and Analysis

We assessed user workload and system usability using adapted versions of the NASA-TLX and SUS, and applied the Wilcoxon Signed-Rank test for analysis. We also provide further NASA-TLX and SUS data analysis in the supplemental material.

Compared to *Baseline*, *MRPilot* received a more favorable feedback on overall SUS score (*MRPilot*: $M = 75.48, SD = 11.03$, *Baseline*: $M = 49.64, SD = 21.73$; $Z = -3.28, p < .01, d = 1.015$), especially in usability and intuitiveness, as illustrated in Figure 7. Most participants found *MRPilot* easy to use (Q3) (*MRPilot*: $M = 3.86, SD = 0.73$; *Baseline*: $M = 2.67, SD = 1.11$; $Z = -3.07, p < .01$), with instructions and general task procedures that were clear and supportive (Q2) (*MRPilot*: $M = 3.91, SD = 0.70$; *Baseline*: $M = 3.14, SD = 1.20$; $Z = -2.59, p < .01$). The system was also seen as highly intuitive (Q8), requiring minimal effort to operate. Users specifically noted that the visual highlighting of object anchors helped them concentrate on the current task and associated items, thereby enhancing the system's intuitiveness (P3, P7, P17). The system's consistency (Q6) facilitated rapid learning (Q7), reducing the training time and increasing user satisfaction.

In contrast, the participants rated the baseline system mostly neutral or negative in terms of simplicity (Q2) and ease of use (Q3). This was largely because *Baseline* struggled to generate satisfactory results when users provided overly simple instructions (P2, P6, P7, P11, P14, P20). Feedback regarding intuitiveness (Q8) was mixed, with some participants expressing dissatisfaction due to the unstable outcomes of the text-based navigation. As for functional integration (Q5) and consistency (Q6), several participants criticized *Baseline* for its lack of well-integrated features, which caused them to feel lost in lengthy text instructions (P2-5, P7, P11, P17, P21).

A concern shared by both *MRPilot* and *Baseline* was related to user confidence (Q9). While *MRPilot* inspired more confidence, there were still neutral or negative responses. Over half of the participants (P1-P3, P5-P7, P13-P15) indicated that the low resolution of the Quest 3 pass-through made them more cautious during precise operations, such as tearing off tape or cutting ingredients. This also contributed to a reduced willingness to use the MR instruction system (Q1), as users felt less confident in its effectiveness.

Compared to *Baseline*, *MRPilot* outperformed in multiple dimensions of the NASA-TLX scores (Figure 7 (a)). The data suggests a lower perceived workload when using *MRPilot* for task navigation. Specifically, *MRPilot* excelled in lowering users' frustration levels (*MRPilot*: $M = 1.52, SD = 0.60$; *Baseline*: $M = 2.48, SD = 1.21$; $Z = -2.81, p < .01$), effort (*MRPilot*: $M = 1.95, SD = 0.81$; *Baseline*: $M = 3.05, SD = 1.16$; $Z = -2.91, p < .01$), time pressure (*MRPilot*: $M = 1.48, SD = 0.75$; *Baseline*: $M = 2.67, SD = 1.11$; $Z = -2.97, p < .01$), and physical demands (*MRPilot*: $M = 1.67, SD = 0.91$; *Baseline*: $M = 2.52, SD = 1.21$; $Z = -2.90, p < .01$). when compared to *Baseline*. The participants generally found *MRPilot* more intuitive, requiring less mental effort. Moreover, *MRPilot* achieved higher performance scores, with the participants reporting that tasks were completed more smoothly and efficiently because of the structured navigation (P1-P3, P5-P9, P16, P18, P20).

To demonstrate the effectiveness of the **Step Recommendation Module**, we analyzed the proportion of automatic switching based on object anchors and manual switching using the virtual panel dur-

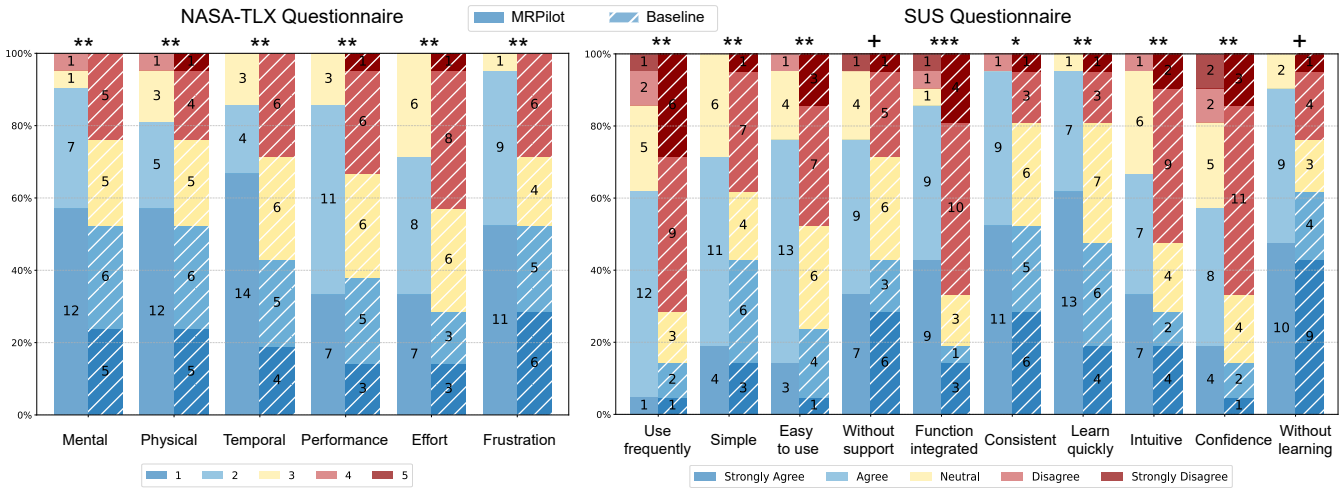

Figure 7: The NASA-TLX (left) and SUS (right) distribution for *MRPilot* (without hatching) and *Baseline* (with hatching). The numbers within each bar segment represent the number of participants who selected the corresponding response option. Statistical significance is indicated above each bar segment ($+:\ .050 < p < .100$, $*:\ p < .050$, $**:\ p < .010$, $***:\ p < .001$). A more comprehensive statistical analysis and boxplot figures are provided in the supplemental material.

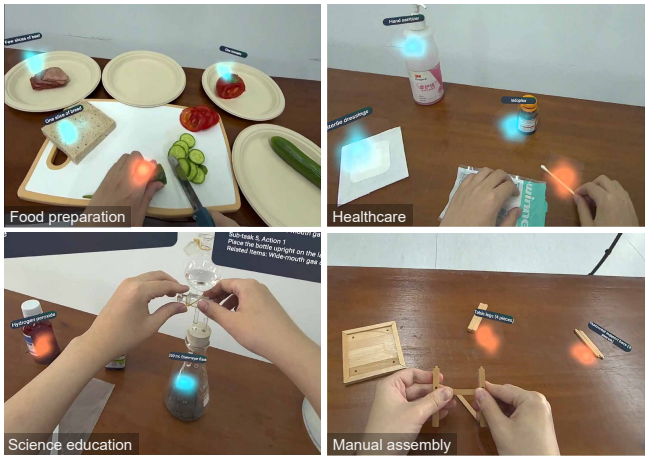

Figure 8: Four example task scenarios of *MRPilot*. These diverse scenarios span domains such as food preparation, healthcare, science education, and manual assembly. During each task, users receive real-time, responsive navigation through *MRPilot* to assist them in completing the procedures accurately and efficiently.

ing users' interactions with *MRPilot*. As shown in Figure 6, the results indicate that for almost all participants, the proportion of using automatic switching based on object anchors exceeded 50%. Additionally, we observed that in tasks generated by the **Navigation Builder Module**, which occasionally included confirmation steps in specific tasks (e.g., confirming whether all ingredients were prepared in the sandwich preparation task or checking that hands had been sanitized in the healthcare task). In this case, users tended to manually switch the action to save time. Besides, since the sandwich preparation task involves a relatively high degree of freedom, users also tended to freely explore feasible steps rather than strictly follow the system-provided sequence. Notably, participants P13-P21 exhibited relatively lower manual switching rates, as their assigned tasks (chemistry experiment and toy desk assembly) contained fewer confirmation steps and required adherence to a fixed sequence compared to other groups. This observation indicates that while automatic switching is predominant, manual switching remains a necessary feature for specific use cases.

To analyze the qualitative feedback gathered from the study and interviews, we conducted a thematic analysis. First, two authors independently read and coded the transcribed data line by line to identify meaningful segments. These initial codes were discussed collaboratively to develop a shared codebook. The coded data were then systematically organized into candidate themes by grouping related codes. To enhance the credibility of the analysis, the themes were repeatedly reviewed against the raw data, and discrepancies were resolved through meetings among the authors. This process ensured that the final themes accurately captured the main patterns in the participants' narratives. As a result, four main themes emerged, which are detailed below.

Theme 1: Overall User Experience (all participants). The participants felt that *MRPilot* was overall easy to learn and/or use, friendly to HMD beginners, and provided great responsive navigation during general task guidance. Several participants mentioned that *MRPilot* made it significantly easier to locate objects related to actions. For instance, P2 noted that he could *"find objects immediately among a number of ingredients."* P5 remarked, *"Using* MRPilot *is simple and intuitive,"* while P7 described it as *"a novel experience to complete tasks in a mixed reality setting. The interaction and audio feedback are clear, which enhances my confidence during task completion."* At the same time, the participants also suggested potential improvements, such as *"offering more flexibility in completing actions, not limiting the interaction to clicking panels or interacting with all objects relevant to the actions"*, as highlighted by P5 and P14.

Theme 2: Differences with Traditional Task Completion (P1, P7, P8, P17). Several participants mentioned the differences between immersive experience and traditional task completion. P1 stated that "MRPilot *helped me carefully complete the process of changing wound dressings, with the structured, step-by-step guidance preventing me from getting lost in lengthy plain text instructions."* In particular, P7 highlighted the advantage of *MRPilot* in recommending recipes based on available ingredients, utensils, and personal taste preferences: "MRPilot *greatly reduced the time spent browsing online and cross-referencing existing ingredients to determine the feasibility of a recipe. Compared to the baseline system,* MRPilot *allowed me to quickly receive accurate guidance without needing repetitive modifications."* Additionally, P8 noted that "MRPilot *excelled in meeting highly customized requirements. Even for me, with limited cooking experience, it enabled me to*

efficiently complete a customized recipe without the hassle of repeatedly searching and comparing different options." Furthermore, P17 emphasized *MRPilot*'s capability in handling unfamiliar objects during tasks: *"When I lacked knowledge about certain items,* MRPilot *helped me effectively differentiate between potentially related objects, ensuring I selected appropriate items confidently."* This aspect made *MRPilot* especially beneficial for users with specific preferences or less familiarity with task-related objects.

Theme 3: Learning Curve (P2, P4, P5, P18, P21). Some participants (P2, P4, P5, P18, P21) noted that their ability to use *MRPilot* improved significantly as they became more familiar with it. This was particularly true for actions such as making the required hand gestures to capture the scene environment and manually anchoring objects. Initially, most users, especially those without prior experience with Quest 3, required instructions from the HMD to understand these interactions. However, after this initial phase, they were able to explore *MRPilot* independently. Participants P2, P4, P5, and P18 also mentioned that using *MRPilot* became very easy once they had learned all of its features. This feedback suggests that there is a slight learning curve, especially for participants with limited knowledge of VR/MR devices.

Theme 4: Limitation and Challenges (P3-P5, P14). The participants also mentioned certain limitations and challenges. These include limited performance in action switching, low resolution of the pass-through display, and the heavy weight of the HMD. Some limitations were inherited from the Meta Quest 3. For instance, as P3 said, *"I cannot perform very fine operations using MR devices because I cannot see the real world clearly."* Similarly, P5 noted, *"I feel unconfident using a knife because the low resolution of the pass-through display makes me afraid of cutting my fingers."* In particular, regarding the action switching limitation, P4, who is very familiar with AI assistant system, mentioned, *"I feel that after the baseline guide is generated, I can operate more freely rather than strictly following predefined steps, which feels better, even though I may need to repeatedly modify my requirements to generate this guide."* P4 also noted that while the structured steps and distributed instructions reduced the burden of reading large amounts of text, they also introduced some unnecessary confirmation steps. For example, when using *MRPilot*, P4 said, *"It makes you confirm whether everything is ready, and I need to provide feedback to the system that I have indeed completed this step, which adds an extra burden."* P14 also expressed that *MRPilot*'s verification steps generated in the process is redundant: *"The system forces me to double-check each preparation step and formally acknowledge completion through explicit notifications, creating additional workflow interruptions."* In the future, we aim to improve action switching by introducing additional criteria for determining when an action is complete. This will involve leveraging machine learning and computer vision techniques to better understand users' actions through the pass-through camera of the HMD.

7 Limitations and Future Work

Task Status Tracking and Object Position Tracking. Although we employ task status tracking methods based on item usage relationships and manual task state switching, this approach has its limitations. For example, it is not always necessary to use all the associated items to complete a given action. While users can manually select the recommended action step to proceed directly to the next one, this still introduces additional confirmation steps, which may slow the workflow. A potential solution to this issue is the use of visual understanding or modeling to estimate the execution time for each action, thereby facilitating seamless task transitions.

We implement object anchoring by overlaying virtual labels on physical items. This allows users to interact with the actual objects while *MRPilot* utilizes Quest 3's built-in interaction recognition. This system monitors user actions and ensures that virtual labels accurately follow the physical items. However, the effectiveness of this tracking is susceptible to limitations such as occlusion, lighting conditions, and the specific way users grasp the objects, resulting in less-than-perfect tracking performance. Adopting an object tracking approach based on predefined object models, such as that employed by *Apple Vision Pro*, could potentially alleviate some of these challenges and improve robustness.

Software and Hardware Constraints. One of the major software constraints is the latency performance of structured task generation. As mentioned in section 5 and supplemental material, generating a complete task takes about 20 seconds. If the network connection is poor and the request fails, it will take even longer for the user to receive a response. In addition to hardware limitations, most users in our study reported discomfort while wearing the Meta Quest 3 due to its weight.

Challenges for Real-World Deployment. Several practical and technological challenges remain for deploying *MRPilot* in real-world environments. While our user studies confirm its feasibility in some procedure tasks, but the generalization to all scenarios still need further investigation. In some cases, factors such as very low lighting conditions and occlusions can introduce performance constraints and limit the system's generalizability. Another current challenge in using LLMs for navigation generation is handling inaccurate instructions caused by hallucinations or limited contextual awareness. At present, Algorithm 1 employs a filtering mechanism to discard clearly unreasonable instructions in most cases but cannot guarantee complete correctness, which is an inherent limitation of LLMs. Additionally, while *MRPilot* effectively highlights task-relevant objects, it does not yet fully exploit MR interaction affordances such as action trajectories and dynamic motion cues. Enhancing the system with these richer spatial visualizations could further improve the task performance of users.

8 Conclusion

In this paper, we introduced *MRPilot*, a novel MR system designed to provide responsive navigation for general procedural tasks pre-authoring. By leveraging LLMs and computer vision techniques, *MRPilot* effectively generates structured, step-by-step instructions tailored to users' specific tasks and physical environments. The system's key innovation lies in its ability to offer real-time, adaptive navigation that responds to users' actions, significantly reducing cognitive load and enhancing workflow efficiency.

Our user study demonstrated that *MRPilot* outperforms the baseline MR system by providing more flexible and dynamic task support. Unlike previous systems that rely on fixed, pre-authored guidance, *MRPilot* is capable of adapting to general tasks, offering contextual feedback and object anchoring that closely integrates virtual instructions with the user's physical environment. The combination of responsive navigation and general task support not only improves the overall user experience but also empowers users to navigate complex tasks more confidently and accurately. We showcase various application examples, which include food preparation, healthcare, science education, and manual assembly. Finally, we discuss the limitations of the current version of *MRPilot* and outline future research directions. We believe that our work represents a significant advancement in the development of intelligent MR systems, demonstrating the potential of MR technologies to enhance task performance in everyday scenarios.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback, and the user study participants for their time. We especially thank Zhida Sun for his valuable guidance in statical analysis. This work was partially supported by grants from NSFC (62472287) and Guangdong Basic and Applied Basic Research Foundation (2023A1515011297).

REFERENCES

[1] K. Ahuja, S. Pareddy, R. Xiao, M. Goel, and C. Harrison. Lightanchors: Appropriating point lights for spatially-anchored augmented reality interfaces. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, p. 189–196. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3332165.3347884 3

[2] R. Arakawa, H. Yakura, and M. Goel. Prism-observer: Intervention agent to help users perform everyday procedures sensed using a smartwatch, 2024. 2

[3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. teusz Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020. 2

[4] Y. Cao, X. Qian, T. Wang, R. Lee, K. Huo, and K. Ramani. An exploratory study of augmented reality presence for tutoring machine tasks. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, p. 1–13. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3313831.3376688 2

[5] C. Chen, C. Nguyen, J. Hoffswell, J. Healey, T. Bui, and N. Weibel. PaperToPlace: Transforming Instruction Documents into Spatialized and Context-Aware Mixed Reality Experiences. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–21. ACM, San Francisco CA USA, Oct. 2023. doi: 10.1145/3586183.3606832 1, 2, 3

[6] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 16901–16911, 2024. 2, 3, 4, 6

[7] S. Chidambaram, H. Huang, F. He, X. Qian, A. M. Villanueva, T. S. Redick, W. Stuerzlinger, and K. Ramani. ProcessAR: An augmented reality-based tool to create in-situ procedural 2D/3D AR Instructions. In *Designing Interactive Systems Conference 2021*, pp. 234–249. ACM, Virtual Event USA, June 2021. doi: 10.1145/3461778.3462126 2

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019. 2

[9] Devpost. Spatial agents, 2023. Accessed: 2024-09-20. 2

[10] M. D. Dogan, E. J. Gonzalez, A. Colaco, K. Ahuja, R. Du, J. Lee, M. Gonzalez-Franco, and D. Kim. Augmented Object Intelligence: Making the Analog World Interactable with XR-Objects, Apr. 2024. arXiv:2404.13274 [cs]. 3

[11] R. Hamada, J. Okabe, I. Ide, S. Satoh, S. Sakai, and H. Tanaka. Cooking navi: assistant for daily cooking in kitchen. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, p. 371–374. Association for Computing Machinery, New York, NY, USA, 2005. doi: 10.1145/1101149.1101228 2, 3

[12] F. He, X. Hu, J. Shi, X. Qian, T. Wang, and K. Ramani. Ubi edge: Authoring edge-based opportunistic tangible user interfaces in augmented reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3544548.3580704 3

[13] N. Hollender, C. Hofmann, M. Deneke, and B. Schmitz. Integrating cognitive load theory and concepts of human–computer interaction. *Computers in human behavior*, 26(6):1278–1288, 2010. 1, 3

[14] G. Huang, X. Qian, T. Wang, F. Patel, M. Sreeram, Y. Cao, K. Ramani, and A. J. Quinn. Adaptutar: An adaptive tutoring system for machine tasks in augmented reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3411764.3445283 1, 2

[15] D. A. Hudson and C. D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6693–6702, 2019. 6

[16] M. Z. Iqbal and A. G. Campbell. Investigating challenges and opportunities of the touchless hand interaction and machine learning agents to support kinesthetic learning in augmented reality. In *Companion Proceedings of the 26th International Conference on Intelligent User Interfaces*, IUI '21 Companion, p. 99–101. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3397482.3450713 3

[17] R. Jain, J. Shi, R. Duan, Z. Zhu, X. Qian, and K. Ramani. Ubi-touch: Ubiquitous tangible object utilization through consistent hand-object interaction in augmented reality. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3586183.3606793 3

[18] A. Khurana, H. Subramonyam, and P. K. Chilana. Why and when llm-based assistants can go wrong: Investigating the effectiveness of prompt-based interactions for software help-seeking. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, IUI '24, p. 288–303. Association for Computing Machinery, New York, NY, USA, 2024. doi: 10.1145/3640543.3645200 2

[19] J. Lee, D. P. Sarda, E. Lee, A. Lee, J. Wang, A. Rodriguez, and J. E. Froehlich. Towards real-time computer vision and augmented reality to support low vision sports: A demonstration of artennis. In *Adjunct Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23 Adjunct. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3586182.3615815 3

[20] J. Lee, A. D. Tjahjadi, J. Kim, J. Yu, M. Park, J. E. Froehlich, Y. Tian, and Y. Zhao. Cookar: Affordance augmentations in wearable ar to support kitchen tool interactions for people with low vision. In *Proceedings of the 2024 ACM Symposium on User Interface Software and Technology*, 2024. 3

[21] Z. Liu, Z. Zhu, E. Jiang, F. Huang, A. M. Villanueva, X. Qian, T. Wang, and K. Ramani. Instrumentar: Auto-generation of augmented reality tutorials for operating digital instruments through recording embodied demonstration. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3544548.3581442 2

[22] Meta. Meta xr all-in-one sdk. https://assetstore.unity.com/packages/tools/integration/meta-xr-all-in-one-sdk-269657, 2024. Accessed: 2024-10-06. 6

[23] Microsoft. Microsoft dynamic 365 remote assist, 2022. Accessed: 2024-09-19. 1, 3

[24] K. Miyawaki and M. Sano. A virtual agent for a cooking navigation system using augmented reality. In *Intelligent Virtual Agents*, pp. 97–103. Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-85483-8_10 2

[25] M. Moghaddam, N. C. Wilson, A. S. Modestino, K. Jona, and S. C. Marsella. Exploring augmented reality for worker assistance versus training. *Advanced Engineering Informatics*, 50:101410, 2021. 2

[26] K. Monteiro, R. Vatsal, N. Chulpongsatorn, A. Parnami, and R. Suzuki. Teachable reality: Prototyping tangible augmented reality with everyday objects by leveraging interactive machine teaching. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3544548.3581449 3

[27] NASA. Task load index (tlx) v. 1.0 manual. Technical report, NASA Ames Research Center, Moffett Field, CA, 1986. 7

[28] R. Nguyen. Integrating in-hand physical objects in mixed reality interactions. In *Companion Proceedings of the 27th International Conference on Intelligent User Interfaces*, IUI '22 Companion, p. 129–133. Association for Computing Machinery, New York, NY, USA, 2022. doi: 10.1145/3490100.3516476 3

[29] K.-B. Park, S. H. Choi, M. Kim, and J. Y. Lee. Deep learning-based mobile augmented reality for task assistance using 3d spatial mapping and snapshot-based rgb-d data. *Computers & Industrial Engineering*,

146:106585, 2020. 2

[30] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *International Journal of Computer Vision*, 123:74 – 93, 2015. 6

[31] S. Rajaram and M. Nebeling. Paper Trail: An Immersive Authoring System for Augmented Reality Instructional Experiences. In *CHI Conference on Human Factors in Computing Systems*, pp. 1–16. ACM, New Orleans LA USA, Apr. 2022. doi: 10.1145/3491102.3517486 2

[32] A. Sato, K. Watanabe, and J. Rekimoto. Mimicook: a cooking assistant system with situated guidance. In *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction*, TEI '14, p. 121–124. Association for Computing Machinery, New York, NY, USA, 2014. doi: 10.1145/2540930.2540952 1, 2, 3

[33] S. Shao, Z. Li, T. Zhang, C. Peng, G. Yu, X. Zhang, J. Li, and J. Sun. Objects365: A large-scale, high-quality dataset for object detection. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8429–8438, 2019. 6

[34] P. Sharma, N. Ding, S. Goodman, and R. Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Annual Meeting of the Association for Computational Linguistics*, 2018. 6

[35] S. Srinidhi, E. Lu, and A. Rowe. Xair: An xr platform that integrates large language models with the physical world. In *2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 759–767. IEEE, 2024. 2

[36] A. Stanescu, P. Mohr, M. Kozinski, S. Mori, D. Schmalstieg, and D. Kalkofen. State-aware configuration detection for augmented reality step-by-step tutorials. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 157–166. IEEE, 2023. 1, 2

[37] A. Stanescu, P. Mohr, D. Schmalstieg, and D. Kalkofen. Model-free authoring by demonstration of assembly instructions in augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 28(11):3821–3831, 2022. 1

[38] A. Stanescu, P. Mohr, F. Thaler, M. Kozinski, L. R. Skreinig, D. Schmalstieg, and D. Kalkofen. Error management for augmented reality assembly instructions. In *2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 690–699. IEEE, 2024. 1, 2

[39] A. Tang, C. Owen, F. Biocca, and W. Mou. Comparative effectiveness of augmented reality in object assembly. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 73–80, 2003. 3

[40] Y. Tang, C.-M. Chang, and X. Yang. Pdfchatannotator: A human-llm collaborative multi-modal data annotation tool for pdf-format catalogs. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, IUI '24, p. 419–430. Association for Computing Machinery, New York, NY, USA, 2024. doi: 10.1145/3640543.3645174 2

[41] B. Thoravi Kumaravel, C. Nguyen, S. DiVerdi, and B. Hartmann. Tutorivr: A video-based tutorial system for design applications in virtual reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290605.3300514 2

[42] A. Treffer, A. Clark, and S. Lukosch. Teaching dance with mixed reality mirrors. In *2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 971–980. IEEE, 2024. 1

[43] D. Uriu, M. Namai, S. Tokuhisa, R. Kashiwagi, M. Inami, and N. Okude. panavi: recipe medium with a sensors-embedded pan for domestic users to master professional culinary arts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, p. 129–138. Association for Computing Machinery, New York, NY, USA, 2012. doi: 10.1145/2207676.2207695 2

[44] M. Verghese, B. Chen, H. Eghbalzadeh, T. Nagarajan, and R. Desai. User-in-the-loop evaluation of multimodal llms for activity assistance. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1144–1154. IEEE, 2025. 2

[45] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding. Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*, 2024. 3, 4

[46] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22. Curran Associates Inc., Red Hook, NY, USA, 2024. 5

[47] G. Wu, J. Qian, S. Castelo Quispe, S. Chen, J. Rulff, and C. Silva. ARTiST: Automated Text Simplification for Task Guidance in Augmented Reality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pp. 1–24. ACM, Honolulu HI USA, May 2024. doi: 10.1145/3613904.3642772 2

[48] X. T. Xu, J. Yin, C. Gu, J. Mar, S. Zhang, J. L. E, and S. P. Dow. Jamplate: Exploring llm-enhanced templates for idea reflection. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, IUI '24, p. 907–921. Association for Computing Machinery, New York, NY, USA, 2024. doi: 10.1145/3640543.3645196 2

[49] M. Yamaguchi, S. Mori, P. Mohr, M. Tatzgern, A. Stanescu, H. Saito, and D. Kalkofen. Video-annotated augmented reality assembly tutorials. In *Proceedings of the 33rd annual ACM symposium on user interface software and technology*, pp. 1010–1022, 2020. 1, 2

[50] J. J. Yang and J. A. Landay. Infoled: Augmenting led indicator lights for device positioning and communication. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, p. 175–187. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3332165.3347954 3

[51] J. J. Yang, L. Qiu, E. A. Corona-Moreno, L. Shi, H. Bui, M. S. Lam, and J. A. Landay. AMMA: Adaptive Multimodal Assistants Through Automated State Tracking and User Model-Directed Guidance Planning. In *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pp. 892–902. IEEE, Orlando, FL, USA, Mar. 2024. doi: 10.1109/VR58804.2024.00108 2

[52] H. Ye and H. Fu. Progesar: Mobile ar prototyping for proxemic and gestural interactions with real-world iot enhanced spaces. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22. Association for Computing Machinery, New York, NY, USA, 2022. doi: 10.1145/3491102.3517689 3

[53] H. Ye, J. Leng, C. Xiao, L. Wang, and H. Fu. Proobjar: Prototyping spatially-aware interactions of smart objects with ar-hmd. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3544548.3580750 3

[54] X. Yu, B. Lee, and M. Sedlmair. Design space of visual feedforward and corrective feedback in xr-based motion guidance systems. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24. Association for Computing Machinery, New York, NY, USA, 2024. doi: 10.1145/3613904.3642143 2

[55] J. Zauner, M. Haller, A. Brandl, and W. Hartman. Authoring of a mixed reality assembly instructor for hierarchical structures. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.*, pp. 237–246, 2003. doi: 10.1109/ISMAR.2003.1240707 2