Autonomous Workflow for Multimodal Fine-Grained Training Assistants Towards Mixed Reality

Jiahuan Pei[1], Irene Viola[1], Haochen Huang[1], Junxiao Wang[3], Moonisa Ahsan[1], Fanghua Ye[4], Yiming Jiang[5], Yao Sai[5], Di Wang[3], Zhumin Chen[5], Pengjie Ren[*,5], and Pablo Cesar[1,2]

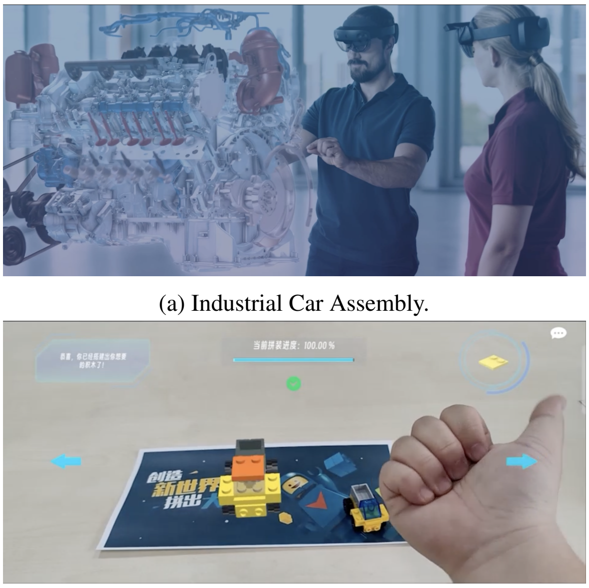
[1] Centrum Wiskunde & Informatica, Amsterdam, The Netherlands, {j.pei,irene,p.s.cesar}@cwi.nl
[2]TU Delft, Delft, The Netherlands
[3]King Abdullah University of Science and Technology, Thuwal, Saudi Arabia
[4]University College London, London, United Kingdom, fanghua.ye.19@ucl.ac.uk
[5] Shandong University, Qingdao, China, jay.ren@outlook.com

Abstract

Autonomous artificial intelligence (AI) agents have emerged as promising protocols for automatically understanding the language-based environment, particularly with the exponential development of large language models (LLMs). However, a fine-grained, comprehensive understanding of multimodal environments remains under-explored. This work designs an autonomous workflow tailored for integrating AI agents seamlessly into mixed reality (MR) applications for fine-grained training. We present a demonstration of a multimodal fine-grained training assistant for LEGO brick assembly in a pilot MR environment. Specifically, we design a cerebral language agent that integrates LLMs with memory, planning, and interaction with MR tools and a vision-language agent, enabling agents to decide their actions based on past experiences. Furthermore, we introduce LEGO-MRTA, a multimodal fine-grained assembly dialogue dataset synthesized automatically in the workflow served by a commercial LLM. This dataset comprises multimodal instruction manuals, conversations, MR responses, and vision question answering. Last, we present several prevailing open-resource LLMs as benchmarks, assessing their performance with and without fine-tuning on the proposed dataset. We anticipate that the broader impact of this workflow will advance the development of smarter assistants for seamless user interaction in MR environments, fostering research in both AI and HCI communities.

1 Introduction

The advent of “Industry 4.0”, centered on the concept of smart manufacturing, presents a landscape with both opportunities and challenges for enhancing production efficiency (Goel and Gupta, 2020; Bécue et al., 2021; Jan et al., 2023). Training assistance for automating and accelerating industrial assembly is in huge demand across various manufacturing applications, such as furniture manufacturing (You et al., 2022), industrial product assembly (Funk et al., 2017), and car assembly (Bellalouna et al., 2020).



(a) Industrial Car Assembly.

(b) LEGO Brick Assembly. We illustrate several use cases in the demo of BrickDream.[1]

Figure 1: Examples of fine-grained assembly in MR systems.

Mixed reality (MR), encompassing both virtual reality (VR) and augmented reality (AR), spans a spectrum from fully real environments to “matrix-like” virtual environments, showing promise for industrial manufacturing assembly tasks (Gavish et al., 2015; Stender et al., 2021; Butaslac et al., 2022). These multimodal, interactive, user-centric environments provide a solution for trainees who experience significant cognitive workload for training (Hou and Wang, 2013; Botto et al., 2020; Dalim et al., 2020). However, the assistance of a senior

[*] Corresponding author.

[1] https://www.youtube.com/watch?v=KkZKL3aKMJs

Findings of the Association for Computational Linguistics: ACL 2024, pages 4051–4066
August 11-16, 2024 ©2024 Association for Computational Linguistics

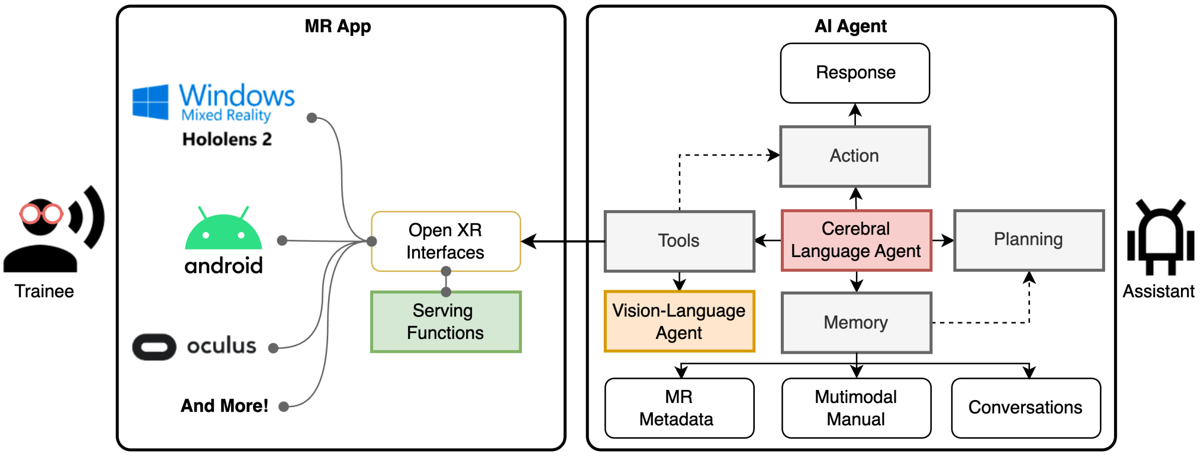


Figure 2: The proposed autonomous workflow, involving an AI agent interacting with an MR application. The AI agent comprises a core cerebral language agent, which interacts with a vision-language agent to interpret the multimodal context into metadata, which can be utilized by the cerebral language agent iteratively. The MR application interacts with AI agents by serving functions as external tools.

person as a trainer is required, either in person or remotely (Fidalgo et al., 2023).

To advance intelligent virtual assistants, traditional work leverages natural language processing (NLP) techniques (Li and Yang, 2021; Li et al., 2021, 2022; Colabianchi et al., 2023) and reinforcement learning (Sloan et al., 2022) to promote human-machine interactions. Large language models (LLMs), as the new era of prevalent NLP techniques, have been observed to elicit diverse interaction patterns across tasks, demonstrating their versatility and feasibility (Mahmood et al., 2023). However, (i) tailoring assistant services by grounding interactions; and (ii) understanding users' situated multimodal contexts remain challenging and under-explored (Dong et al., 2023).

To this end, we introduce an autonomous workflow (see Figure 2) tailored for integrating AI agents into MR applications for fine-grained training. We present a demonstration of a multimodal fine-grained training assistant within a toy MR application for LEGO brick assembly. Specifically, we design a cerebral language agent that integrates the LLM with memory, planning, and interaction with MR serving functional tools and a vision-language agent, enabling agents to decide their actions based on experiences. Then, we introduce LEGO-MRTA, a multimodal fine-grained assembly dataset synthesized automatically by a commercial LLM. This dataset comprises 65 multimodal instruction manuals, 1,423 conversations with vision-language pairs, serving usages of 18 functional tools in an MR environment. Additionally, several prevailing open-resource LLMs are presented as benchmarks, assessing their performance with and without fine-tuning on the proposed dataset. Furthermore, we anticipate that the broader impact of this workflow will advance the development of smarter assistants for seamless user interaction in MR environments, fostering research in both AI and HCI communities.

We summarize our contributions as follows:

- We design a workflow, which integrates autonomous AI agents for fine-grained assembly assistance in an MR demonstration.
- We create a multimodal manual-grounded fine-grained assembly conversation dataset in the MR context.
- We assess several open-resource LLMs as benchmarks, evaluating their performance with and without fine-tuning on the proposed dataset.

2 Related Work

We summarize previous research concerning multimodal datasets and virtual dialogue assistants within the realm of MR.

2.1 Multimodal Datasets towards MR

Traditional multimodal datasets focus on the interactions with sensor data (Patrik et al., 2018) between human-human or human-robot, and only a few of them provide small-scale task-oriented dialogues, such as OFAI-MMTD (Schreitter and Krenn, 2016) and Kontogiorgos et al. (2018), and

	Domain	#Conv.	#Utt.	#Token	#AvgUtt.	#AvgToken
MDC	Minecraft Building	509	15,926	113,116	30.7	7.9 (Architect) / 2.9 (Builder)
CerealBar	Instruction Following	1,202	23,979	3,641	19.9	14.0 (Instructor) / 8.5 (Follower)
CVDN	Navigation	2,050	12,361	2,223	6.0	33.5 (Navigators) / 48.1 (Oracles)
TEACH	Household	3,215	45,000	3,429	13.7	5.7 (Commander) / 3.8 (Follower)
LEGO-MRTA	LEGO Assembly	1,423	35,131	7,173	24.8	26.6 (Trainer) / 12.7 (Trainee)

Table 1: Comparison of dialogue datasets towards MR.

Chinese Whispers (Dimosthenis et al., 2020). ScanScribe (Zhu et al., 2023b) releases a 3D scene-text pairs dataset for 3D vision and text alignment learning. HoloAssistant (Wang et al., 2023c) provides a dataset containing 350 unique instructor-performer pairs with AR metadata to perceive, reason, and interact in the physical world. However, the conversations are not publicly available.

Recent studies have concentrated on multimodal datasets with conversations. MDC (Narayan-Chen et al., 2019) presents a collection of 509 human-human conversations in the Minecraft VR games. CerealBar (Suhr et al., 2019) creates 1,202 human-to-human conversations that map user instructions to system actions in a situated VR game environment. CVDN (Thomason et al., 2020) collects 2,050 human-robot conversations on Amazon Mechanical Turk for improving parsing and perception for natural language commands. Teach (Padmakumar et al., 2022) builds over 3,000 human–human, interactive dialogues to complete household tasks in the simulation.

Different from those aforementioned datasets, LEGO-MRTA gathers 1,423 synthetic natural conversations between trainers and trainees. Unlike robotic commands, the length of utterances is relatively longer. Furthermore, these conversations are generated by grounding both on an instruction manual and responses from an MR, ensuring that the simulated conversations closely resemble natural human language. We compare the statistics of the above datasets in Table 1.

2.2 Virtual Dialogue Assistants for MR

Conventional efforts focus on creating virtual assistants for human-machine interactions using NLP techniques (Li and Yang, 2021; Li et al., 2021, 2022; Colabianchi et al., 2023) and reinforcement learning (Sloan et al., 2022). LLMs, representing the forefront of contemporary NLP techniques, hold tremendous promise for advancing towards the next generation of intelligent assistants (Naveed et al., 2023). The recent remarkable achievements of LLMs have spurred a growing interest in utilizing them to address a great variety of complex tasks (Zhang et al., 2023), with particular attention being drawn to LLM-augmented autonomous agents (Yao et al., 2022; Huang et al., 2022; Shinn et al., 2023; Madaan et al., 2023).

Autonomous agents expand the capabilities of LLMs into sequential action execution, demonstrating their proficiency in interacting with environments and addressing complex tasks through data collection (Wang et al., 2023b; Liu et al., 2023). A crucial aspect of this advancement relies on the capacity of LLMs to generate and interpret images, enabling them to access visual content and provide inputs, thereby integrating with MR environments (Oyanagi et al., 2023; Wei et al., 2024). Regarding skill training, autonomous agents and LLMs can create immersive learning experiences that blend virtual and physical environments. For instance, students can utilize them to explore workflows and concepts in a more interactive and engaging manner (Gong et al., 2023; Li et al., 2024). In the context of MR serving as a sandbox (Li et al., 2023b) for LLMs and autonomous agents, the relationship is mutually beneficial. MR offers a secure, adaptable, and regulated setting for training models (Naihin et al., 2023). LLM-powered autonomous agents together with MR hold the potential to revolutionize our interaction with the digital world (Xu et al., 2023).

The convergence of LLMs, autonomous agents, and MR presents both excitement and challenges. As MR training experiences become more realistic and personalized, they demand larger amounts of data, encompassing detailed information about trainees' behaviors, preferences, and interactions. Ensuring the availability and reusability of this data poses a significant challenge. Overall, our workflow's ultimate goal is to enhance MR training experiences by facilitating more natural language interactions, generating precise 3D models of real-world objects (Li et al., 2023a), and fostering dynamic and interactive experiences. While challenges remain (Xi et al., 2023; Ayache et al., 2023), the potential of this powerful technological fusion offers numerous exciting possibilities that could revolutionize personalization in virtual experiences. This entails the development of dedicated workflows and datasets.

3 Fine-Grained Training Workflow

In this section, we describe the proposed workflow (See Figure 2) that advances AI agents towards MR guided fine-grained training.

3.1 Definition of Fine-Grained Training

In the context of fine-grained training, we anticipate the ability to (i) accurately follow professional training instructions documented in an instruction manual; and (ii) be sensitive to detailed visual information, ultimately for complex industrial assembly tasks, as illustrated in Figure 1 (a).

We define the following two roles during a training session:

- **User:** A human trainee who aims to acquire expertise and will work on fine-grained assembly tasks through interaction with the MR environment.
- **Assistant:** A virtual AI agent who will be able to assist the trainees in training and respond to their inquiries. It offers support with (i) a conversation agent that replies to trainees' requests and provides guidance grounded in the instruction manual; (ii) an interface for users to interact with the MR environment; and (iii) a vision-language agent that understands and transmits users' visual context to language.

3.2 Autonomous AI Agent

We design the autonomous AI agent with a chain of two agents, namely (i) a cerebral language agent that serves to reply to trainees' requests, provide guidance, interact with MR and the vision-language agent; and (ii) a vision-language agent that understands and transmits users' visual context to language, which is then utilized by the cerebral language agent for planning.

3.2.1 Cerebral Language Agent

Inspired by the concept of LLM-powered autonomous agents (Wang et al., 2023b), we develop a cerebral language agent that incorporates an LLM with *memory*, *planning*, and functional *tools* that can interact with MR application, thereby enabling agents to make decisions regarding their *actions* based on past experiences. It can handle multimodal inputs, such as instruction manuals, historical conversations, and metadata within MR environments, and subsequently generate actions (i.e., responses or API calls for the MR application). The scope of responsibility of the agent is defined in a system prompt (See P2, Table 5, Appendix A) Notably, it is able to alleviate the challenges (See §1): (i) it tailors assistant services by seamlessly interaction with MR applications to discover the business needs gradually; (ii) it interacts with a vision-language agent (See §3.2.2), which facilitates the capability of understanding the multimodal context in MR environments.

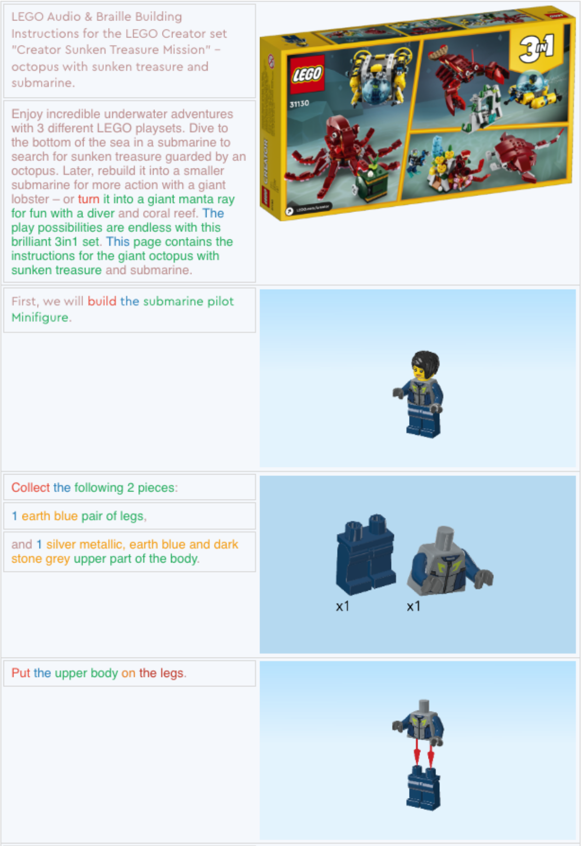
3.2.2 Vision-Language Agent

The vision-language agent's mission is to bridge the gap between understanding visual context and language, enabling effective utilization by the cerebral language agent (See §3.2.1) to conduct comprehensive planning for global optimization. Its core is the vision-language model (VLM), which is a task-driven large model that transmits vision input into language output needed by specific tasks.

In the context of LEGO assembly training, we observe two distinct patterns in LEGO instruction manuals (See an example in Figure 3) and define the following two tasks: **(T1) Object detection.** Given an image or a sequence of images as input, the objective is to predict the position of an object requested in a query and generate output in the format of "<Object> <Xleft> <Ytop> <Xright> <Ybottom>". For example, during assembly step 2, the AI trainer might direct the trainee, "Please gather the earth blue pair of legs and the silver metallic upper part of the body." In response, the trainee may ask, "Is this the one?" The vision-language agent is tasked with recognizing the object the trainee is referring to. **(T2) Assembly state detection.** Given an image or a sequence of images as input, the objective is to identify if the current assembly state matches the reference state provided in the instruction manual. For example, during assembly step 3, the vision-language agent is responsible for assisting the user's request, such as "Am I assembling them correctly?"

3.3 Pilot MR Application Design

We design an MR application as a pilot to show intuitive demonstration. First, we utilized a commercial LLM to generate candidate user requirements using the prompt (See P1, Table 5, Appendix A)



LEGO Audio & Braille Building Instructions for the LEGO Creator set "Creator Sunken Treasure Mission" – octopus with sunken treasure and submarine.

Enjoy incredible underwater adventures with 3 different LEGO playsets. Dive to the bottom of the sea in a submarine to search for sunken treasure guarded by an octopus. Later, rebuild it into a smaller submarine for more action with a giant lobster – or turn it into a giant manta ray for fun with a diver and coral reef. The play possibilities are endless with this brilliant 3in1 set. This page contains the instructions for the giant octopus with sunken treasure and submarine.

First, we will build the submarine pilot Minifigure.

Collect the following 2 pieces:

1 earth blue pair of legs,

and 1 silver metallic, earth blue and dark stone grey upper part of the body.

Put the upper body on the legs.

Figure 3: An example of LEGO instruction manual. It consists of a summary section at the beginning followed by three sequential instruction steps. Each step includes textual instructions paired with corresponding images to guide the assembly process.

as input. Then, we brainstormed and discussed the generated user requirements within a group of researchers and developers and finalized 7 user requirements (See Table 6, Appendix A) and 18 serving functional tools (See Table 2). We develop a standard application programming interfaces (APIs) to enable seamless interactions between functions in the MR application and AI agents.

4 Dataset Creation

In this section, we introduce how to use the proposed workflow (See §3) to create a multimodal dialogue dataset in the MR environment.

4.1 Instruction Manual Crawling

We crawled 65 multimodal instruction manuals for fine-grained training from the LEGO official website[2]. A manual provides illustrated images and textual instructions on how to use, operate, assemble, and install a LEGO brick set. The key sections of an instruction manual include: (i) a summary that describes the general information, such as topics and candidate parts for assembly. It is followed by (ii) a sequence of multimodal step instructions. Each step contains a set of textual instructions and an illustration by image. Key functional phrases such as theme entities are highlighted in textual instructions. Here we show an example of a LEGO instruction manual in Figure 3.

4.2 Tool Response Generation

First, we use the crawled instruction manuals and the well-designed prompt template to produce prompts as the LLM input to generate user functional requirements and decide the serving functional tools. Then, we randomly choose up to 6 tools for each conversation session and record the simulated responses generated from templates.

4.3 VLM-Based QA Construction

First, we use the step instruction to construct a query, containing a special token ("[detection]") for the object detection task and a single instruction in a step. Second, we employ a query and the aligned image as inputs for MiniGPT-v2 (Chen et al., 2023; Zhu et al., 2023a), generating inference output as an answer of the query in the format of "<Object> <Xleft> <Ytop> <Xright> <Ybottom>". Last, we iterate through all instruction steps in a conversation session, repeating the above two steps to construct vision question answering (VQA) pairs.

4.4 Multimodal Context-Aware Conversation Generation

We generate conversations grounded on both the instruction manual and simulated tool responses using a commercial LLM. First, we reconstruct full instruction manuals with a summary and 10 step instructions because the average number of steps per manual is 215.3, which is quite long. This may limit the input tokens of an LLM and potentially distract the LLM with less grounding capability. Second, we instantiate the designed prompt template (See P3, Table 5, Appendix A) with the chunked instruction manuals. Last, we utilize a commercial LLM as the core of the proposed workflow to generate the conversations. Specifically, the system prompt informs the language agent about its responsibilities. The query prompt is used for each round of requests to generate a conversation.

[2] https://legoaudioinstructions.com/instructions

Tool Name	Description
StartAssemble	Initiate the assembly process.
NextStep	Move to the next assembly step.
FrontStep	Go back to the previous assembly step.
Explode	Trigger an explosion for detailed viewing.
Recover	Restore the initial state of AR objects after explosion.
FinishedVideo	End the assembly process and show a video of the assembled LEGO bricks.
ReShow	Repeat the current assembly step.
Enlarge	Enlarge or zoom out the current object.
Shrink	Shrink or zoom in the current object.
GoToStep	Go to the given assembly step number.
Rotate	Rotate the current object to a direction (“Up”, “Down”, “Left”, “Right”, “None”).
ShowPieces	Show all candidate LEGO pieces to be assembled.
HighlightCorrectComponents	Highlight correct attachment points and components.
GetCurrentStep	Get the number of the current step.
GetRemainingStep	Get the number of the remaining steps.
CheckStepStatusVR	Check whether the current step in Unity is accomplished correctly or not.
APICallObjectRecognitionAR	Call the VLM agent to identify LEGO pieces based on the provided video streaming data from AR glasses and highlight the recognized pieces in the AR environment.
APICallCheckStepStatusAR	Call the VLM agent to determine whether the current assembly step is completed correctly or not, using the provided video streaming data from AR glasses as input.

Table 2: Descriptions of serving tools in the pilot extended reality (XR) application.

LEGO-MRTA Instruction Manual	
#Manual	65
#InstructionStep	13,994
#Token	8,676
#Theme Entity	2,412
#AvgInstructionStep	215.3
#AvgConversation	21.9
Modalities	Text, Image

Table 3: Statistics of instruction manuals in the LEGO-MRTA dataset.

The historical rounds of requests are tracked by memory.

4.5 Dataset Statistics

We report the statistics of instruction manuals (See Table 3) and conversations (See Table 1).

We obtain 65 instruction manuals as grounding to lead a commercial LLM to generate 1,423 human-human natural conversations between trainers and trainees. Each instruction manual can make 21.9 conversations on average. Theoretically, the amount of conversations can be enlarged by multiple times of requests. However, we focus on showcasing how to create meaningful datasets automatically. We construct 26,405 context-response pairs from generated conversations and VQA pairs as data samples. The average length is 107 tokens for the context and 145 tokens for the response utterance. We utilize 21.10k samples for fine-tuning open-resource LLMs to enhance the instruction-following capability and evaluate their performance on 5.25k test samples [3]. Compared with existing datasets, LEGO-MRTA ensures that the simulated conversations closely resemble natural human language due to the design of the simulation method.

5 Experimental Setup

5.1 LLM Benchmarks

We consider several prevailing 7B open-source decoder-only LLMs as benchmarks, considering privacy concerns associated with fine-grained training in manufacturing.

- **BLOOM** (Le Scao et al., 2022) is pretrained on the multilingual ROOTS corpus, offering multilingual capabilities for various NLP tasks.
- **Falcon-instruct** (Almazrouei et al., 2023) is pretrained on a large corpus of RefinedWeb data and fine-tuned on mixed chat and instruct datasets.
- **Llama2-Chat** (Touvron et al., 2023) is a pretrained and fine-tuned generative text model optimized specifically for dialogue tasks, ensuring high-quality conversational responses.
- **Vicuna1.5** (Zheng et al., 2023) is a chat assistant derived by fine-tuning Llama 2 on user-shared conversations collected from ShareGPT.
- **OpenChat3.5** (Wang et al., 2023a) is a chat model fine-tuned with the C-RLFT strategy on mixed-quality data, achieving performance com-

[3] https://huggingface.co/datasets/voxreality/vox_arta_lego_v2

Model	BLEU-4		ROUGE-1		ROUGE-2		ROUGE-L		ToolACC (%)		ThemeACC (%)	
PEFT (LoRA)	/wo	/w	/wo	/w	/wo	/w	/wo	/w	/wo	/w	/wo	/w
BLOOM	2.88	54.07	20.49	61.91	6.50	49.52	3.78	58.63	49.62	77.86	26.30	64.61
Falcon	5.38	10.30	8.68	11.33	4.25	7.41	5.11	10.20	22.79	17.65	12.66	10.81
Llama2-Chat	10.23	30.53	18.59	40.65	7.41	25.48	10.59	32.91	21.37	55.73	47.20	55.51
Vicuna1.5	14.11	54.71	29.30	62.64	14.21	50.47	15.48	59.36	**52.67**	78.12	**69.69**	66.79
OpenChat3.5	22.00	6.94	29.70	34.51	15.69	23.69	22.50	11.36	51.97	74.02	58.19	**81.90**
XVERSE	22.42	53.55	28.45	61.54	14.31	49.77	22.39	58.03	49.62	**83.97**	57.53	71.10
BlueLM	22.72	55.69	30.40	63.52	14.98	51.58	23.76	60.35	48.15	82.22	47.51	68.08
Qwen	24.82	**59.78**	31.44	**66.95**	17.69	**55.95**	25.66	**64.26**	45.71	77.14	54.96	71.17
Mistral	**25.87**	54.17	**33.32**	62.07	**17.99**	49.40	**26.32**	58.62	49.62	78.20	54.80	66.65

Table 4: Benchmarking LLMs on our LEGO-MRTA dataset, without (/wo) and with (/w) parameter-efficient fine-tuning (PEFT) using low rank adaptation (LoRA). The bold font indicates the highest score in each column. The underline indicates the performance decreases after fine-tuning.

parable to larger models like ChatGPT.

- **XVERSE**[4] is a versatile model supporting 8k context length, ideal for longer multi-round dialogues, knowledge question-answering, and summarization tasks, trained on a diverse dataset of 2.6 trillion tokens.
- **BlueLM-Chat**[5] is a large-scale language model optimized for chat tasks, offering improved context understanding.
- **Qwen-Chat** (Bai et al., 2023) is a chat model that fine-tunes the pretrained Qwen model using human alignment techniques.
- **Mistral-Instruct** (Jiang et al., 2023) is a fine-tuned version of the Mistral-7B-v0.1, specifically tailored for instruction-based tasks using publicly available conversation datasets.

5.2 Evaluation Metrics

We evaluate the performance in terms of both overlap (BLUE-n, ROUGE-n) and informativeness (ToolACC, ThemeACC):

- **BLUE-n** measures precision, which measures the ratio of n-grams in the generated responses that match those in the reference responses. We consider $n = 4$.
- **ROUGE-n** measures recall, which calculates the ratio of n-grams in the reference responses that are captured by the generated responses. Here we consider $n = 1, 2, L$ and L denotes the number of longest common subsequences.
- **ToolACC** is defined as the ratio of correctly mentioned entities by the generated responses, compared to the reference response, from a list of serving tools.
- **ThemeACC** is defined as the ratio of correctly mentioned entities compared to the reference response, from a list of theme entities obtained from the instruction manual.

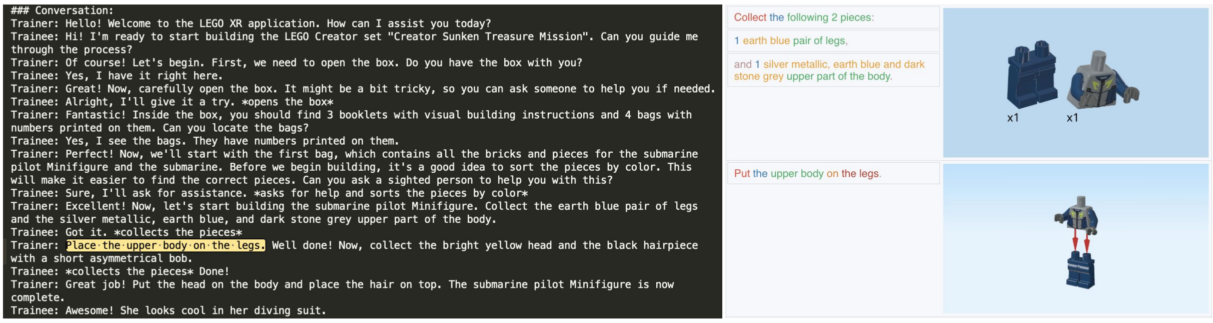
5.3 Implementation Details

The implementation of the workflow is based on LangChain.[6] The model "gpt-3.5-turbo-16k-0613" is used as the commercial LLM for generating data, e.g., conversations, user requirements, and serving functions. The MiniGPT4-v2[7] is used as the VLM to detect the object, followed by simple rules to generate vision-language pairs.

We employ LoRA to conduct parameter-efficient fine-tuning of 7B open-source LLMs on the proposed dataset using the framework proposed by Hiyouga (2023). Specifically, the maximum sequence length is set to 1024 and the learning rate is 5e-05. The model is trained for 3 epochs with a per-device batch size of 4, and accumulated gradients every 4 steps. A cosine learning rate scheduler is employed, with a maximum gradient norm of 1.0. We log results every 5 steps and save model checkpoints every 100 steps. Warm-up steps are set to 0. LoRA is used with a rank of 8 and a dropout rate of 0.1 for regularization. All experiments are run on NVIDIA A100 SXM4 40GB GPUs.

[4] https://github.com/xverse-ai/XVERSE-7B/blob/main/README_EN.md

[5] https://huggingface.co/vivo-ai/BlueLM-7B-Chat



```
### Conversation:
Trainer: Hello! Welcome to the LEGO XR application. How can I assist you today?
Trainee: Hi! I'm ready to start building the LEGO Creator set "Creator Sunken Treasure Mission". Can you guide me
through the process?
Trainer: Of course! Let's begin. First, we need to open the box. Do you have the box with you?
Trainee: Yes, I have it right here.
Trainer: Great! Now, carefully open the box. It might be a bit tricky, so you can ask someone to help you if needed.
Trainee: Alright, I'll give it a try. *opens the box*
Trainer: Fantastic! Inside the box, you should find 3 booklets with visual building instructions and 4 bags with
numbers printed on them. Can you locate the bags?
Trainee: Yes, I see the bags. They have numbers printed on them.
Trainer: Perfect! Now, we'll start with the first bag, which contains all the bricks and pieces for the submarine
pilot Minifigure and the submarine. Before we begin building, it's a good idea to sort the pieces by color. This
will make it easier to find the correct pieces. Can you ask a sighted person to help you with this?
Trainee: Sure, I'll ask for assistance. *asks for help and sorts the pieces by color*
Trainer: Excellent! Now, let's start building the submarine pilot Minifigure. Collect the earth blue pair of legs
and the silver metallic, earth blue, and dark stone grey upper part of the body.
Trainee: Got it. *collects the pieces*
Trainer: Place·the·upper·body·on·the·legs. Well done! Now, collect the bright yellow head and the black hairpiece
with a short asymmetrical bob.
Trainee: *collects the pieces* Done!
Trainer: Great job! Put the head on the body and place the hair on top. The submarine pilot Minifigure is now
complete.
Trainee: Awesome! She looks cool in her diving suit.
```

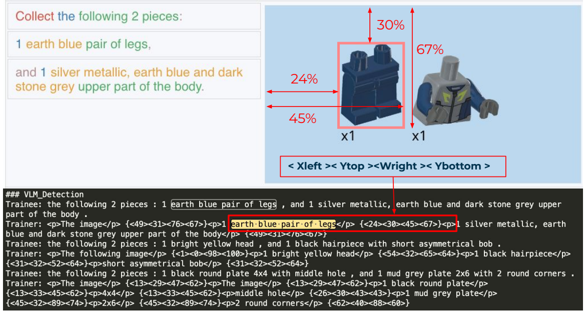
Collect the following 2 pieces:

1 earth blue pair of legs,

and 1 silver metallic, earth blue and dark stone grey upper part of the body.

Put the upper body on the legs.

Figure 4: An example of the generated conversation (left) and the grounding step instructions (right).



Collect the following 2 pieces:

1 earth blue pair of legs,

and 1 silver metallic, earth blue and dark stone grey upper part of the body.

```
### VLM_Detection
Trainee: the following 2 pieces : 1 earth blue pair of legs , and 1 silver metallic, earth blue and dark stone grey upper
part of the body .
Trainer: <p>The image</p> {<49><31><76><67>}<p>1 earth·blue·pair·of·legs</p> {<24><30><45><67>}<p>1 silver metallic, earth
blue and dark stone grey upper part of the body</p> {<49><31><76><67>}
Trainee: the following 2 pieces : 1 bright yellow head , and 1 black hairpiece with short asymmetrical bob .
Trainer: <p>The following image</p> {<1><0><98><100>}<p>1 bright yellow head</p> {<54><32><65><64>}<p>1 black hairpiece</p>
{<31><32><52><64>}<p>short asymmetrical bob</p> {<31><32><52><64>}
Trainee: the following 2 pieces : 1 black round plate 4x4 with middle hole , and 1 mud grey plate 2x6 with 2 round corners .
Trainer: <p>The image</p> {<13><29><47><62>}<p>The image</p> {<13><29><47><62>}<p>1 black round plate</p>
{<13><33><45><62>}<p>4x4</p> {<13><33><45><62>}<p>middle hole</p> {<26><30><43><43>}<p>1 mud grey plate</p>
{<45><32><89><74>}<p>2x6</p> {<45><32><89><74>}<p>2 round corners</p> {<62><40><88><60>}
```

Figure 5: An example of vision-language pair (lower) and the grounding step instructions (upper).

6 Outcomes

6.1 Evaluation on Benchmark LLMs

Table 4 shows the performance on 9 prevailing open-source LLMs, without and with fine-tuning on the LEGO-MRTA dataset.

First, after fine-tuning, the performance of all models gets dramatically improved in terms of all metrics, except for the results that are underlined. This demonstrates the feasibility and effectiveness of tailoring LLMs for fine-grained training in MR environments. In addition, this shows the proposed LEGO-MRTA dataset contains distinct characteristics that have not been captured by existing publicly available datasets.

Second, there exists a trade-off between overlap and informativeness evaluation. For example, for metrics concerning overlap, Qwen consistently achieves the highest scores; while its informativeness is inferior to best models, i.e., ToolACC is 6.83% lower than that of XVERSE, ThemeACC is 10.73% lower than that of OpenChat3.5.

Third, the choice of backbone LLMs inherently impacts the performance of both overlap and informativeness. We compute the standard deviation for each metric over all models to gauge the variability in performance scores: BLEU-4 (19.60), ROUGE-1 (17.79), ROUGE-2 (16.02), ROUGE-L (20.64), ToolACC (19.85), and ThemeACC (19.19).

6.2 Case Study

Figure 4 shows an example to intuitively verify the feasibility of generating conversation based on an instruction manual. As highlighted in the trainer's utterance, "Place the upper body on the legs", this accurately conveys the instruction from the manual in a human-like manner. The generated conversation is feasible at instruction-following capability.

Figure 5 illustrates an example demonstrating the feasibility of constructing queries based on instructions from a manual to accurately request positions in a multimodal context. We transferred the generated position and highlighted a frame with markers. We observed that the prediction was relatively accurate. Additionally, another aspect we observe that needs improvement in the future is the redundant output from the VLM.

7 Discussion of Broader Impact

The research presented in this paper offers a fully new environment to advance how workers are trained and get help by using MR technologies. By integrating AI assistants into MR environments, workers can tackle complex tasks more effectively. This innovation not only enhances worker productivity but also reduces training costs for companies, as it eliminates the need for expert instructors to be physically present for employee training sessions.

8 Conclusion

In this work, we introduce an autonomous workflow to develop smarter multimodal fine-grained training assistants in MR environments. Specifically, we have designed a cerebral language agent that integrates LLM with memory, planning, and interaction with MR tools, along with a vision-language agent. This integration enables agents to make decisions based on past experiences, thereby

[6] https://python.langchain.com/docs/get_started/introduction

[7] https://github.com/Vision-CAIR/MiniGPT-4

addressing the challenge of tailoring assistant services through grounded interactions. We have designed a vision-language agent to better understand users' situated multimodal contexts. Notably, we have created a dataset for fine-grained training in MR. We have compared the performance of open-resource LLMs before and after fine-tuning using this dataset. We aim to facilitate the development of smarter assistants for seamless user interaction, fostering research in AI and HCI communities.

Reproducibility

We release resources including the source code and dataset at https://github.com/Jiahuan-Pei/AutonomousDialogAgent4AugmentedReality.

Limitations

The generation of user requirements and the dataset relies solely on the simulation process. This workflow serves as a fast solution to verify the concept of an LLM agent aiding in a specific use case, such as a LEGO assembly assistant. However, we acknowledge that the study of user requirements are valuable and needed to build up user-centric AI agents and MR applications. Besides, the demonstration codes do not optimize LLM and VLM simultaneously, potentially leading to suboptimal outcomes. We have only assessed LLMs as benchmarks. However, we have not conducted separate assessments of the influence on the vision-language agent and user experience in MR. We plan to explore these aspects in future work.

Ethics Statement

We realize that there are risks in developing a large language model for users, so it is necessary to pay attention to the ethical issues. Therefore, we use the open-resourced LLMs as benchmarks and consider user-centric points: A user will first be provided with an explanation of what will be happening during their MR training experience. Users will then be provided with relevant consent forms to sign, and after signing they will be fitted with AR glasses and the training scenario will begin. After launching the application, the user will be greeted by the virtual assistant and prompted to confirm they would like to begin training. After confirming, the user will then be asked by the virtual assistant which difficulty level they would like to be trained on.

Acknowledgements

This work was supported through the Horizon Europe research and innovation programme, under grant agreement No 101070521 (VOXReality). This work used the Dutch national e-infrastructure with the support of the SURF cooperative grant (No. EINF-7705). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Directorate-General for Communications Networks, Content and Technology. Neither the European Union nor the granting authority can be held responsible for them.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Julia Ayache, Marta Bieńkiewicz, Kathleen Richardson, and Benoit Bardy. 2023. extended reality of socio-motor interactions: Current trends and ethical considerations for mixed reality environments design. In *ICMI*, pages 154–158.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Adrien Bécue, Isabel Praça, and João Gama. 2021. Artificial intelligence, cyber-threats and industry 4.0: Challenges and opportunities. *Artificial Intelligence Review*, 54(5):3849–3886.

Fahmi Bellalouna, Mika Luimula, Panagiotis Markopoulos, Evangelos Markopoulos, and Franco Zipperling. 2020. Fiaar: an augmented reality firetruck equipment assembly and configuration assistant technology. In *CogInfoCom*, pages 000237–000244. IEEE.

Carola Botto, Alberto Cannavò, Daniele Cappuccio, Giada Morat, Amir Nematollahi Sarvestani, Paolo Ricci, Valentina Demarchi, and Alessandra Saturnino. 2020. Augmented reality for the manufacturing industry: the case of an assembly assistant. In *VRW*, pages 299–304. IEEE.

Isidro M Butaslac, Yuichiro Fujimoto, Taishi Sawabe, Masayuki Kanbara, and Hirokazu Kato. 2022. Systematic review of augmented reality training systems. *IEEE Transactions on Visualization and Computer Graphics*.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechu Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*.

Silvia Colabianchi, Andrea Tedeschi, and Francesco Costantino. 2023. Human-technology integration with industrial conversational agents: A conceptual architecture and a taxonomy for manufacturing. *Journal of Industrial Information Integration*, 35:100510.

Che Samihah Che Dalim, Mohd Shahrizal Sunar, Arindam Dey, and Mark Billinghurst. 2020. Using augmented reality with speech input for non-native children's language learning. *International Journal of Human-Computer Studies*, 134:44–64.

Kontogiorgos Dimosthenis, Sibirtseva Elena, and Gustafson Joakim. 2020. Chinese Whispers: A Multimodal Dataset for Embodied Language Grounding. In *LREC*.

Xin Luna Dong, Seungwhan Moon, Yifan Ethan Xu, Kshitiz Malik, and Zhou Yu. 2023. Towards next-generation intelligent assistants leveraging llm techniques. In *SIGKDD*, pages 5792–5793.

Catarina G Fidalgo, Yukang Yan, Hyunsung Cho, Maurício Sousa, David Lindlbauer, and Joaquim Jorge. 2023. A survey on remote assistance and training in mixed reality environments. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2291–2303.

Markus Funk, Andreas Bächler, Liane Bächler, Thomas Kosch, Thomas Heidenreich, and Albrecht Schmidt. 2017. Working with augmented reality? a long-term analysis of in-situ instructions at the assembly workplace. In *PETRA*, pages 222–229.

Nirit Gavish, Teresa Gutiérrez, Sabine Webel, Jorge Rodríguez, Matteo Peveri, Uli Bockholt, and Franco Tecchia. 2015. Evaluating virtual reality and augmented reality training for industrial maintenance and assembly tasks. *Interactive Learning Environments*, 23(6):778–798.

Ruchi Goel and Pooja Gupta. 2020. Robotics and industry 4.0. *A Roadmap to Industry 4.0: Smart Production, Sharp Business and Sustainable Development*, pages 157–169.

Ran Gong, Qiuyuan Huang, Xiaojian Ma, Hoi Vo, Zane Durante, Yusuke Noda, Zilong Zheng, Song-Chun Zhu, Demetri Terzopoulos, Li Fei-Fei, et al. 2023. Mindagent: Emergent gaming interaction. *arXiv preprint arXiv:2309.09971*.

Hiyouga. 2023. Llama factory. https://github.com/hiyouga/LLaMA-Factory.

Lei Hou and Xiangyu Wang. 2013. A study on the benefits of augmented reality in retaining working memory in assembly tasks: A focus on differences in gender. *Automation in Construction*, 32:38–45.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *ICML*, pages 9118–9147. PMLR.

Zohaib Jan, Farhad Ahamed, Wolfgang Mayer, Niki Patel, Georg Grossmann, Markus Stumptner, and Ana Kuusk. 2023. Artificial intelligence for industry 4.0: Systematic review of applications, challenges, and opportunities. *Expert Systems with Applications*, 216:119456.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Dimosthenis Kontogiorgos, Vanya Avramova, Simon Alexanderson, Patrik Jonell, Catharine Oertel, Jonas Beskow, Gabriel Skantze, and Joakim Gustafson. 2018. A multimodal corpus for mutual gaze and joint attention in multiparty situated interaction. In *LREC*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv.2211.05100*.

Bai Li, Xinyuan Li, Yaodong Cui, Xuepeng Bian, Siyu Teng, Siji Ma, Lili Fan, Yonglin Tian, Fei-Yue Wang, et al. 2024. Integrating large language models and metaverse in autonomous racing: An education-oriented perspective. *IEEE Transactions on Intelligent Vehicles*.

Chen Li, Andreas Kornmaaler Hansen, Dimitrios Chrysostomou, Simon Bøgh, and Ole Madsen. 2022. Bringing a natural language-enabled virtual assistant to industrial mobile robots for learning, training and assistance of manufacturing tasks. In *SII*, pages 238–243. IEEE.

Chen Li, Jinha Park, Hahyeon Kim, and Dimitrios Chrysostomou. 2021. How can i help you? an intelligent virtual assistant for industrial robots. In *HRI*, pages 220–224.

Chen Li and Hong Ji Yang. 2021. Bot-x: An ai-based virtual assistant for intelligent manufacturing. *Multiagent and Grid Systems*, 17(1):1–14.

Mingsheng Li, Xin Chen, Chi Zhang, Sijin Chen, Hongyuan Zhu, Fukun Yin, Gang Yu, and Tao Chen. 2023a. M3dbench: Let's instruct large models with multi-modal 3d prompts. *arXiv preprint arXiv:2312.10763*.

Yuan Li, Yixuan Zhang, and Lichao Sun. 2023b. Metaagents: Simulating interactions of human behaviors for llm-based task-oriented coordination via collaborative generative agents. *arXiv preprint arXiv:2310.06500*.

Zhiwei Liu, Weiran Yao, Jianguo Zhang, Le Xue, Shelby Heinecke, Rithesh Murthy, Yihao Feng, Zeyuan Chen, Juan Carlos Niebles, Devansh Arpit, et al. 2023. Bolaa: Benchmarking and orchestrating llm-augmented autonomous agents. *arXiv preprint arXiv:2308.05960*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.

Amama Mahmood, Junxiang Wang, Bingsheng Yao, Dakuo Wang, and Chien-Ming Huang. 2023. Llm-powered conversational voice assistants: Interaction patterns, opportunities, challenges, and design guidelines. *arXiv preprint arXiv:2309.13879*.

Silen Naihin, David Atkinson, Marc Green, Merwane Hamadi, Craig Swift, Douglas Schonholtz, Adam Tauman Kalai, and David Bau. 2023. Testing language model agents safely in the wild. *arXiv preprint arXiv:2311.10538*.

Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. 2019. Collaborative dialogue in minecraft. In *ACL*, pages 5405–5415.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.

Akimi Oyanagi, Kazuma Aoyama, Kenichiro Ito, Tomohiro Amemiya, and Michitaka Hirose. 2023. Virtual reality training system using an autonomy agent for learning hospitality skills of a retail store. In *HCI*, pages 483–492. Springer.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. 2022. TEACh: Task-driven Embodied Agents that Chat. In *AAAI*, volume 36, pages 2017–2025.

Jonell Patrik, Bystedt Mattias, Fallgren Per, Kontogiorgos Dimosthenis, Lopes José, Malisz Zofia, Mascarenhas Samuel, Oertel Catharine, Eran Raveh, and Shore Todd. 2018. ARMI: An Architecture for Recording Multimodal Interactions. In *LREC*.

Stephanie Schreitter and Brigitte Krenn. 2016. The ofai multi-modal task description corpus. In *LREC*, pages 1408–1414.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *NeuralIPS*.

Hannah Sloan, Richard Zhao, Faisal Aqlan, Hui Yang, and Rui Zhu. 2022. Adaptive virtual assistant for virtual reality-based remote learning. In *ASEE Annual Conference & Exposition*.

Birga Stender, Johannes Paehr, and Thomas N Jambor. 2021. Using ar/vr for technical subjects in vocational training–of substancial benefit or just another technical gimmick? In *EDUCON*, pages 557–561. IEEE.

Alane Suhr, Claudia Yan, Jack Schluger, Stanley Yu, Hadi Khader, Marwa Mouallem, Iris Zhang, and Yoav Artzi. 2019. Executing instructions in situated collaborative interactions. In *EMNLP-IJCNLP*, pages 2119–2130.

Jesse Thomason, Aishwarya Padmakumar, Jivko Sinapov, Nick Walker, Yuqian Jiang, Harel Yedidsion, Justin Hart, Peter Stone, and Raymond Mooney. 2020. Jointly improving parsing and perception for natural language commands through human-robot dialog. *Journal of Artificial Intelligence Research*, 67:327–374.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023a. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2023b. A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432*.

Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, et al. 2023c. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world. In *ICCV*, pages 20270–20281.

Yuxi Wei, Zi Wang, Yifan Lu, Chenxin Xu, Changxing Liu, Hao Zhao, Siheng Chen, and Yanfeng Wang. 2024. Editable scene simulation for autonomous driving via collaborative llm-agents. *arXiv preprint arXiv:2402.05746*.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

Fengli Xu, Jun Zhang, Chen Gao, Jie Feng, and Yong Li. 2023. Urban generative intelligence (ugi): A foundational platform for agents in embodied city environment. *arXiv preprint arXiv:2312.11813*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Yingchao You, Ze Ji, Xintong Yang, and Ying Liu. 2022. From human-human collaboration to human-robot collaboration: automated generation of assembly task knowledge model. In *ICAC*, pages 1–6. IEEE.

Jianguo Zhang, Kun Qian, Zhiwei Liu, Shelby Heinecke, Rui Meng, Ye Liu, Zhou Yu, Silvio Savarese, and Caiming Xiong. 2023. Dialogstudio: Towards richest and most diverse unified dataset collection for conversational ai. *arXiv preprint arXiv:2307.10172*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023a. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 2023b. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *ICCV*.

A Appendix

A.1 Prompt Templates

A.2 Generated User Requirements

A.3 Qualitative Analysis of the Dataset

The collective results as seen in the figure 6 illustrates that our simulated model has practical assistance capabilities as summarized below:

Realistic simulation. LEGO is a well-known block building concept. The dataset simulates various real-world scenarios encountered during LEGO assembly tasks. By replicating factors such as piece variability, environmental conditions, and assembly constraints, the dataset provides a realistic training environment for machine learning models. This realism enhances the model's ability to generalize to unseen situations, ensuring reliable performance in diverse assembly settings.

Diversity in task difficulty. From simple structures to intricate designs, the dataset exposes the model to diverse assembly scenarios, enabling it to learn robust representations of LEGO building principles. This diversity fosters adaptability in the model, empowering it to tackle simple to difficult or probably novel assembly tasks with confidence and efficiency.

Transfer learning to other tasks. The dataset is structured to facilitate transfer learning, allowing knowledge and representations learned from one assembly task to be applied to related tasks or domains. By leveraging pre-trained models or features learned from similar assembly tasks, machine learning models can bootstrap their learning process on new assembly tasks. This transfer learning capability accelerates model adaptation to new environments and tasks, reducing the need for extensive retraining and improving overall training efficiency.

A.3.1 Instruction Tokens (Figure 6a)

Analyzing the provided instruction tokens, we can derive several factors that contribute to the usability and effectiveness of our simulated dataset for XR training and assembly training:

- **Clear instructional guidance.** Tokens like "Put," "Find," "Collect," and "Open" provide clear and concise instructions for performing various assembly tasks. These instructions guide users by the assembly process step-by-step, ensuring clarity and direction in the training environment.
- **Spatial orientation and manipulation.** Tokens such as "front," "right," "left," "horizontally," and "vertically" offer spatial orientation cues, helping users understand the spatial relationships between LEGO components and how to manipulate them during assembly. This spatial awareness enhances users' ability to accurately position and align LEGO pieces.
- **Feedback and assistance.** Tokens like "help" and "response" indicate provisions for feedback and assistance within the training environment. Offering assistance and feedback helps users troubleshoot issues, learn from mistakes, and improve their assembly skills over time, enhancing the learning experience.
- **Multimodal learning**: The inclusion of tokens like "Audio Instructions" suggests the incorporation of multimodal learning techniques within the training environment. Integrating audio instructions alongside visual cues enhances usability by catering to different learning styles and preferences, making the training experience more accessible and engaging for users.
- **Adaptive learning.** Tokens such as "current" and "previous" imply a dynamic learning environment where users can track their progress and revisit previous steps if needed. Adaptive learning features enhance usability by allowing users

(P1) Prompt template for user requirement generation

[*Task description*]
You are an AI agent who acts as a Unity developer for AR applications. Your role is to analyze users' functional needs based on the manuals and then develop the corresponding functions in an AR training system. Note that is not for visually impaired users, but for trainees who are visually healthy and able to wear HoloLen2 AR glasses.
Here are samples of manuals:
[*Manuals*]

(P2) Prompt template for conversation generation

1. System prompt
[*Task description*]
Brief version: The task is to generate multiple turns of conversations and called tools between the trainer (assistant) and trainee (user) grounded on the task-specific guidelines and tools in LEGO XR application.
Full version: The trainer aims to teach the trainee how to accomplish the assembly task based on the task-specific guidelines, supported by an XR application. Specifically, the trainee is wearing AR glasses to see both VR environment and real world. The trainee knows nothing about the guidelines before trainer's guidance. For each step, the trainee must ask at least one deep-dive question, or request a troublesome issue if he or she cannot follow the guide, or call tools from XR application and learn how to use those tools; the trainer must answer the question, assist the trainee, show them the responses to the execution of the tools. At the end of a conversation, first, the trainer must ask if the trainee has accomplished the task and the trainee must tell if the trainee can accomplish the task; second, the trainer must ask how is user experiences, and the trainee provide feedback on the user experience. You must add a section title to separate which key point in the guideline in the generated conversation and generate until the final step of the guidelines.
[*Tool description as shown in Table 2*]

2. Query prompt
[*Task description (Brief version)*]
[*Summary and step instructions in a manual*]
Imagine some trainee's utterances have the intent of using the tools with the following responses:
[*Tool responses*]

LEGO Assembly Assistant prompt (P3)

You are a helpful AI assistant who aims to train the user how to assemble a LEGO car in XR immersive system.
Extended Reality (XR) directs to the assortment of Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR).
Please make sure you complete the objective above with the following rules:
(1) The user is a trainee who is wearing HoloLen 2 glasses and is able to see XR environments in real-time.
(2) You are able to call Unity functions in the LEGO AR application.
(3) You are able to obtain HoloLens 2 Sensor Streaming data.
(4) Alert if the user asks you something outside of the LEGO assembly task but do not give overconfident answers.
Your task is to answer the user's questions and assist the user in understanding how to complete the LEGO assembly task in XR.

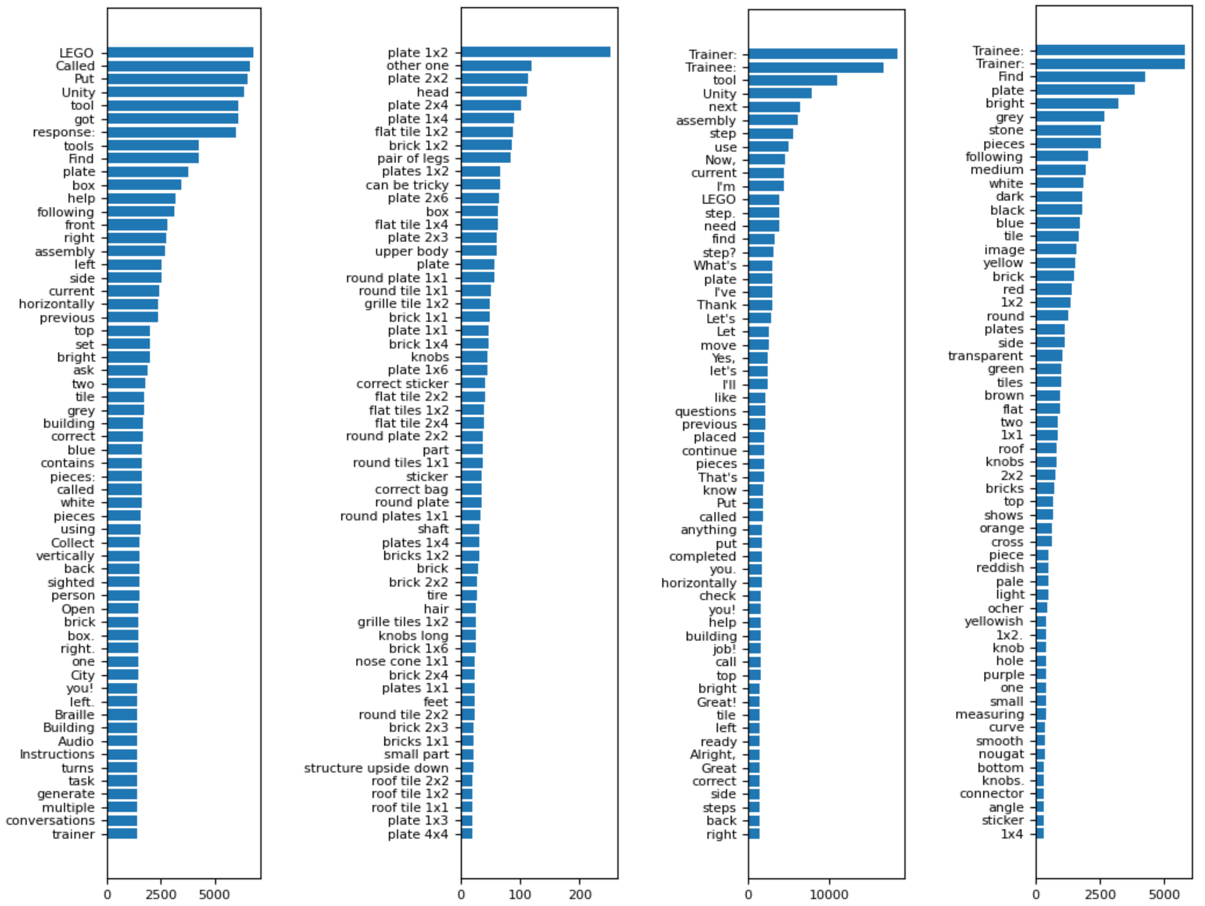
Table 5: Prompt templates used in this work.

User requirement	Description
3D Model Interaction	Create 3D models of the LEGO pieces and the Monster Truck assembly. Trainees can interact with these 3D models using hand gestures and voice commands, making it easier to understand the assembly process.
Step-by-Step Guidance	Display step-by-step instructions directly in the trainees' field of view. This can include both visual instructions and written or spoken guidance.
Real-Time Feedback	Provide real-time feedback to trainees as they assemble the LEGO set. Use AR to highlight the correct attachment points and components, and indicate when they've completed a step correctly.
Object Recognition	Implement object recognition so that HoloLens 2 can identify LEGO pieces and highlight them when trainees look at them. This can help trainees quickly find the right pieces.
Progress Tracking	Keep track of trainees' progress and provide them with an overview of the steps they have completed and those remaining. This can help them stay organized and motivated.
Troubleshooting Assistance	Include a troubleshooting mode that guides trainees through common problems and solutions they might encounter during the assembly.
Data Logging	Collect data on trainees' performance and interaction with the AR training system to analyze their progress and make improvements to the training process.

Table 6: User requirements of the XR training system.

to learn at their own pace, review concepts as needed, and progress through the training material in a structured manner.

- **Interactive learning environment.**: The presence of tokens like "conversations" and "trainer" indicates an interactive learning environment



(a) Instruction tokens. (b) Theme entities. (c) Conversation tokens. (d) Vision-language tokens.

Figure 6: Distribution of top 60 frequent tokens in the above four parts: (a) instructions, (b) entities in the manual, (c) conversations, and (d) vision-language pairs. The x-axis denotes frequency and the y-axis denotes tokens in four parts of the LEGO-MRTA dataset.

where users can engage in dialogue and receive guidance from trainers or virtual assistants. Interactivity enhances usability by promoting engagement, collaboration, and active participation in the learning process, leading to more effective skill acquisition and retention.

The instruction tokens in our simulated dataset indicate clear guidance, and spatial orientation cues within an interactive learning environment.

A.3.2 Theme Entities (Figure 6b)

Based on the theme entities provided, we analyze their relevance to the learning process:

- **Part identification.** Tokens such as "plate 1x2," "plate 2x4," and "brick 1x1" provide specific identifiers for different LEGO parts commonly used in assembly tasks. By including a variety of part identifiers, the dataset facilitates part recognition and identification, enabling the model to learn the characteristics and properties of each component.
- **Spatial orientation and configuration.** Tokens like "head," "upper body," and "feet" suggest the inclusion of assembly instructions related to spatial orientation and configuration of LEGO structures. Understanding the spatial arrangement of components is essential for accurate assembly, and these tokens help the model grasp the hierarchical structure of assemblies and the placement of parts within them.
- **Assembly techniques.** Tokens such as "can be tricky" and "structure upside down" hint at the inclusion of assembly techniques and strategies within the dataset. Learning various assembly techniques is crucial for efficiently building complex structures, and these tokens provide guidance on overcoming challenges and optimizing assembly processes.
- **Component variations.** Tokens like "round plate 1x1," "flat tile 2x4," and "grille tile 1x2" introduce variations of standard LEGO compo-

nents, reflecting the diversity of parts encountered in real-world assembly scenarios. By including a range of component variations, the dataset contains different types of parts and adapts to varying assembly requirements.

- **Accessory identification.** Tokens such as “pair of legs,” “tire,” and “hair” denote accessory pieces commonly used in LEGO constructions, adding realism and complexity to assembly tasks. Recognizing and incorporating accessory pieces is essential for creating realistic and detailed models, and these tokens help the model understand the role of accessories in assembly.
- **Quality control and correctness.** Tokens like “correct sticker” and “correct bag” emphasize the importance of quality control and correctness in assembly tasks. Ensuring that the correct parts are used in the right context is essential for achieving accurate and high-quality assemblies, and these tokens highlight the need for attention to detail and accuracy in the assembly process.
- **Structural components.** Tokens such as “shaft,” “structure upside down,” and “roof tile” suggest the inclusion of structural components and building techniques within the dataset. Understanding the role of structural components and mastering advanced building techniques is critical for creating stable and aesthetically pleasing assemblies, and these tokens provide guidance on constructing sturdy and well-balanced structures.

The theme entities included in our simulated dataset provide a realistic representation of the assembly tasks by encompassing part identification, spatial orientation, assembly techniques, component variations, accessory recognition, quality control, and structural components, the dataset contains the knowledge and skills necessary to effectively assemble LEGO structures in MR.

A.3.3 Conversation Tokens (Figure 6c)

We can infer several aspects that contribute to the usability and effectiveness of our simulated dataset for conversations during assembly training:

- **Role identification.** The presence of “Trainer” and “Trainee” tokens indicates a clear distinction between the roles of the instructor guiding the training session and the learner receiving instructions. This role identification fosters clarity and structure in the conversation, ensuring effective communication between trainer and trainee.
- **Instructional guidance.** Tokens such as “step,” “plate,” “use,” and “find” suggest the provision of instructional guidance within the conversation. The trainer entity likely provides step-by-step instructions and prompts to the trainee, guiding them through the assembly process and facilitating learning in a structured manner.
- **Interactive dialogue.** The conversation tokens include interactive dialogue cues such as “Let’s,” “Yes, let’s,” and “Thank you!” These cues foster engagement and collaboration between the trainer and trainee entities, creating a supportive and interactive learning atmosphere conducive to effective learning and skill development.
- **Feedback and encouragement.** Tokens like “Great!” and “Alright” suggest the inclusion of positive feedback and encouragement within the conversation. Positive reinforcement enhances motivation and engagement, encouraging active participation and fostering a positive learning experience for the trainee.
- **Error handling and assistance.** The presence of tokens like “check,” “help,” and “completed” indicates provisions for error handling and assistance within the conversation. The trainer entity likely offers guidance and support to the trainee in identifying and correcting errors, ensuring a constructive learning process and facilitating skill development.
- **Spatial orientation and task management.** Tokens such as “right,” “left,” “back,” and “steps” provide spatial orientation cues and references to assembly tasks. This spatial orientation facilitates effective communication of assembly instructions and task management between the trainer and trainee entities, ensuring accurate placement and alignment of LEGO components during assembly.

The conversation tokens provide instructional guidance, facilitate interactive dialogue, offer feedback and encouragement, handle errors, and provide spatial orientation cues for task management.

A.3.4 Vision-language Tokens (Figure 6d)

Analyzing the provided tokens from the vision language model, we can identify several factors contributing to its usability and effectiveness:

- **Object recognition.** Tokens such as “plate,” “brick,” “tile,” and “knob” represent common LEGO elements that users encounter during assembly tasks. By including these tokens, the dataset enables the vision language model to recognize and identify various LEGO components accurately, facilitating object recognition and understanding in XR training environments.

- **Color detection.** Tokens like “bright,” “grey,” “white,” and “blue” provide color descriptors for different LEGO pieces. Incorporating color information allows the vision language model to detect and differentiate between LEGO components based on their color, enhancing the model’s ability to interpret and analyze assembly scenes accurately.
- **Shape recognition.** Tokens such as “round,” “flat,” “roof,” and “connector” describe the shapes and configurations of LEGO elements. By including shape descriptors, the dataset enables the vision language model to recognize and classify different types of LEGO pieces based on their shapes, facilitating shape recognition and classification in XR training environments.
- **Size specification.** Tokens like “1x2,” “2x2,” and “1x1” specify the sizes and dimensions of LEGO elements. Incorporating size information allows the vision language model to understand the scale and proportions of LEGO components within assembly scenes, aiding in size estimation and spatial reasoning during XR training tasks.
- **Material and texture.** Tokens such as “smooth,” “nougat,” and “transparent” describe the materials and textures of LEGO elements. Including material and texture descriptors enables the vision language model to identify and distinguish between different surface finishes and textures, enhancing its ability to recognize and characterize LEGO components accurately.
- **Part relationships.** Tokens like “side,” “top,” and “bottom” provide spatial relationship cues between LEGO elements. By including part relationship descriptors, the dataset enables the vision language model to understand the spatial arrangement and orientation of LEGO components within assembly scenes, facilitating the interpretation of complex assembly structures and configurations.
- **Visual context understanding.** Tokens such as “image” and “shows” suggest the inclusion of visual context information within the dataset. Providing visual context cues enables the vision language model to interpret and analyze assembly scenes holistically, incorporating visual information to enhance its understanding of the surrounding environment and improve object recognition accuracy.

Our simulated dataset successfully provides object recognition, color detection, shape recognition, size specification, material and texture characterization, part relationships, and visual context understanding. Altogether, these tokens contribute to the usability and effectiveness of the training environment by providing clear guidance, realistic representation of components and challenges, interactive dialogue, and enhanced vision understanding. These elements collectively enhance the learning experience and skill development in XR assembly tasks.

A.3.5 Called Tools (Figure 7)

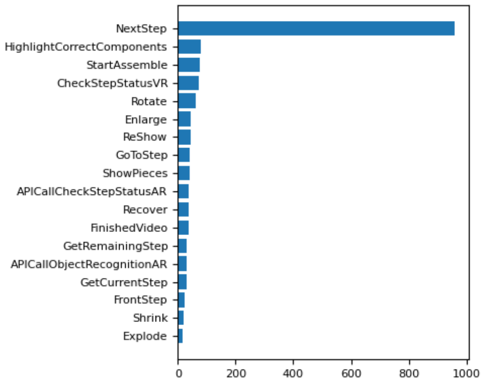


Figure 7: Distribution of called tools in conversations.

As shown in Figure 7, we plot the distribution of the number of tools invoked in the generated conversations. The most frequently called and essential functional tools are those related to process control: “NextStep” (57.02%), “StartAssemble” (4.58%), “CheckStepStatusVR” (4.28%), “GoToStep” (2.44%), “GetRemainingStep” (1.90%), “GetCurrentStep” (1.78%), “1.31%”. This indicates that users prioritize adherence to the assembly procedure during the fine-grained assembly task. Functional tools related to user interactions are also significant, for example, “HighlightCorrectComponents” (4.64%).

A.4 Engineering Details in the Workflow

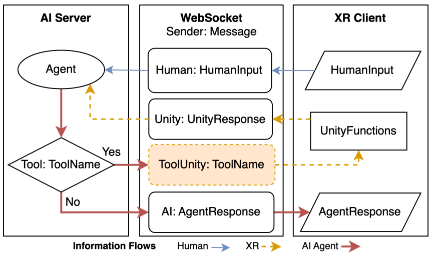


Figure 8: Information flow.