Large Language Model Agent for Embodied Intelligence: A Survey and Perspectives

Chen Gao **Xiaochong Lan** **Jinzhu Mao** **Jun Zhang** **Zhiheng Zheng**
Sibo Li **Jiaao Tang** **Zihan Huang** **Yuwei Du** **Jie Feng** **Yong Li**
Tsinghua University, Beijing, China
{chgao96, liyong07}@tsinghua.edu.cn

Abstract

Embodied intelligence is considered one of the promising approaches toward artificial general intelligence (AGI), focusing on the ability to perceive first-view data from the world and make decisions adaptively based on feedback received. Alan Turing's concept of ”situated AI,” which aims to build embodied intelligences situated in the real world, underscores the necessity of embodiment in AI. However, despite advancements in machine learning and deep learning, achieving true embodied intelligence has remained challenging. Traditional machine learning methods, reliant on offline data, struggle with generalization, indicating a significant gap before reaching genuine embodied intelligence. Recently, large language models (LLMs) have redefined the boundaries of artificial intelligence with their impressive abilities in language-related tasks, reasoning, and decision-making. While LLMs excel in understanding and generating human language and possess rich knowledge and reasoning abilities, they also have notable limitations, such as susceptibility to errors and hallucinations, and challenges in logical reasoning and fine-grained visual understanding. To address these shortcomings, researchers have proposed LLM-empowered agents, which integrate advanced reasoning, learning mechanisms, and interactive capabilities. These agents represent a promising direction for achieving human-like intelligence by combining the strengths of LLMs with additional components to simulate more holistic and dynamic cognitive processes. This paper reviews recent advances in LLM agent-based embodied intelligence, discusses the necessity and potential of LLM agents in this area, and provides resources and future directions for research and development.

1 Introduction

Embodied intelligence is considered one of the possible approaches toward artificial general intelligence (AGI), which focuses on the ability to perceive first-view data from the world and make decisions adaptively based on the feedback received. Alan Turing once mentioned the concept of *situated AI* that “aims at building embodied intelligences situated in the real world”[**?**], considered one of the earliest statements attempting to embody the artificial intelligence models. Although the necessity of embodiment is widely accepted, it has been challenging to achieve for a long time despite the progress of machine learning and deep learning. From the perspective of the learning paradigm, traditional machine learning methods rely on learning from collected offline data, which restrains the ability to generalize, and there is a long way to go before real embodied intelligence.

Recently, large language models have redefined the border of artificial intelligence with their astonishing abilities not in various language-related tasks but in plenty of reasoning and decision-making problems. Overall speaking, large language models LLMs have many strengths. They have mastered the rules of human language, which is one of the most complicated tasks They have rich knowledge, e.g., common sense knowledge, domain knowledge, etc. They have good reasoning ability, such as the chain of thoughts reasoning. But, however, LLMs have many weaknesses They often make mistakes as the corpuses for training are not perfect. There could also be hallucination, which largely affects the real-world deployment of large language models. Moreover, the large language models may be not so good at logical reasoning/fine-grained visual understanding, etc. That is, the capabilities of large language models are limited to patterns and information derived from their training data without genuine understanding or consciousness.

To address these fundamental shortcomings of large language models, inspired by how humankind performs complex tasks, researchers propose various agents empowered by large language models. These LLM-empowered agents, equipped with advanced reasoning, learning mechanisms, and interactive abilities, hold the potential to bridge the gap, offering a more holistic and dynamic approach to simulating human-like intelligence. Specifically, the agents could be a composed solution in which the kernel is a large language model, partly playing the role of the human brain. Since today's large language models still work in a prompt-output manner, researchers add various components.

Since LLM agents' abilities were so attuned to the challenges faced in embodied intelligence, it became an important and promising research direction for building various embodied agents with large language models. In this paper, we systematically review the recent advances of LLM agent-based embodied intelligence, before which we present the background context and carefully discuss why LLM agents are required and essential approaches in this area. Furthermore, we organize relevant benchmarks, datasets and other resources in the related areas, which can not only help readers have a clear view of this area, but also help practitioners to design and implement their own LLM-based embodied agents. Last, since this area is a young and fast-growing research domain, we point out the remaining issues and concerns, which may inspire the following works to address and advance the frontier and border of this area.

The structure of this survey is organized as follows. We first introduce the background of large language models, LLM-agents, embodied intelligence in Section 2. We then discuss the motivations in Section 3. We elaborate on the representative methods of LLM agent-based embodied intelligence in Section 4 by following the taxonomy of embodied intelligence. We discuss the unresolved and open problems in this area, provide ideas of the future directions in Section 5, and conclude this survey in Section 6.

2 Background

2.1 Large language model and LLM agent

Large language models (LLMs) are the most advanced neural networks, typically based on the Transformer model, that have been trained on vast corpora of text data. These models, such as GPT-4, etc., have demonstrated remarkable capabilities in understanding and generating human language. By leveraging extensive datasets and sophisticated training techniques, LLMs can capture complex linguistic patterns and semantic nuances, enabling them to perform a wide range of natural language processing tasks, including translation, summarization, question answering, and text generation. The development of LLMs represents a significant milestone in AI, pushing the boundaries of what machines can achieve in terms of language comprehension and generation.

LLM agents are sophisticated AI systems that integrate large language models with additional components to extend their capabilities beyond pure language processing. While LLMs excel at understanding and generating text, LLM agents leverage these models as core elements within broader frameworks designed for more complex, multi-modal tasks. These agents utilize the linguistic prowess of LLMs for communication and reasoning, while incorporating sensory inputs, memory systems, and decision-making mechanisms to interact with and adapt to their environment. This integration enables LLM agents to perform a wide array of tasks that require not only language understanding but also situational awareness, adaptability, and real-world interaction.

Memory, reflection, and workflows are crucial concepts that underpin the functionality of LLM agents. Memory allows agents to store and retrieve information about past interactions and experiences, enabling them to learn and adapt over time. Reflection involves the agent's ability to analyze its own actions and decisions, fostering continuous improvement and adaptive behavior. Workflows refer to the structured sequences of actions and processes that an agent follows to accomplish specific tasks. These concepts collectively enable LLM agents to operate with a level of coherence and sophistication, mimicking human-like cognitive processes and enhancing their ability to perform complex, dynamic tasks in varied environments.

The key components of LLM agents include environment perception, reasoning, and decision-making, which together form the foundation for their intelligent behavior. Environment perception involves the ability to gather and interpret sensory data from the surroundings, using tools such as computer vision and auditory processing to understand the context. Reasoning is the cognitive process through which agents analyze information, draw inferences, and generate solutions to problems. Decision-making is the final step, where agents choose and execute actions based on their reasoning and perceived environment. These components enable LLM agents to interact effectively with the world, respond to dynamic situations, and achieve their goals with a degree of autonomy and intelligence.

LLM agents possess a range of key abilities that make them versatile and powerful tools in various applications. These abilities include natural language understanding and generation, which allow them to communicate effectively with humans and other systems. They also have strong reasoning and problem-solving skills, enabling them to analyze complex situations and devise appropriate responses. Additionally, LLM agents can learn from their experiences, adapting their behavior over time to improve performance. Their ability to integrate sensory data and make decisions in real-time allows them to operate autonomously in dynamic environments. These combined abilities make LLM agents capable of performing sophisticated tasks, from customer service and virtual assistance to autonomous navigation and advanced research.

2.2 Embodied Intelligence

Embodiment for artificial intelligence refers to the creation of AI systems that are integrated with physical entities, allowing them to perceive, interact with, and respond to their environment in real time. This embodiment represents the most advanced form of AI because it combines cognitive processes with physical actions, enabling a more holistic form of intelligence that can navigate and manipulate the real world. Embodied AI systems, such as robots and autonomous vehicles, leverage sensors, actuators, and advanced algorithms to perform tasks that require a deep understanding of their surroundings. This integration allows them to operate more effectively and adaptively in dynamic and unstructured environments compared to purely computational AI systems, making embodied AI the frontier of artificial intelligence research and application.

Embodied intelligence is crucial because it equips AI systems with key abilities that are essential for real-world applications. These abilities include advanced sensory perception, which allows the AI to gather and interpret data from its environment, and motor skills, which enable it to physically interact with objects and navigate spaces. Additionally, embodied AI can perform complex decision-making and problem-solving tasks in real time, adapting to changes and unexpected events. This capability to learn from direct interaction and experience, much like humans do, allows for more natural and efficient performance of tasks that require both cognitive and physical actions. Embodied intelligence thus opens up new possibilities in fields such as healthcare, manufacturing, autonomous transportation, and service robots, enhancing their effectiveness and usability.

The taxonomy of embodied intelligence encompasses various domains where AI integrates cognitive and physical capabilities. *Embodied visual perception* involves AI systems using cameras and sensors to understand and interpret visual data from their environment, crucial for tasks like object recognition and scene understanding. *Robotics* focuses on designing and controlling robots that can perform physical tasks, from simple pick-and-place operations to complex manipulations. *Navigation* involves autonomous movement through environments, requiring path planning and obstacle avoidance. *Embodied task alignment and allocation* refers to optimizing how tasks are assigned and performed by embodied agents, ensuring efficiency and collaboration. *Multi-agent embodied interactions* cover scenarios where multiple AI agents interact and coordinate with each other and with humans, enhancing collective task performance. *Embodied QA and dialogue* involves systems that can engage in conversations and answer questions based on their understanding of the physical world, improving human-AI interactions in practical settings.

Despite its advancements, embodied intelligence faces unresolved challenges, particularly in areas where traditional machine learning models fall short but large language models (LLMs) show promise. Traditional models often struggle with integrating diverse sensory inputs and making real-time decisions in complex environments. In contrast, LLMs excel at understanding and generating human-like responses, which can enhance the reasoning and interaction capabilities of embodied AI systems. However, combining these capabilities with physical embodiment requires overcoming obstacles such as ensuring accurate and reliable perception, developing sophisticated control mechanisms, and achieving seamless integration of cognitive and motor functions. Furthermore, ensuring safety, ethical behavior, and robust performance in unpredictable and dynamic real-world scenarios remains a significant challenge. Addressing these issues is critical for the advancement and practical deployment of embodied intelligence.

3 Motivation: Why LLM agents are required for embodied intelligence

3.1 Challenges for developing embodied intelligence

Various topics of embodied intelligence has been widely studied in the past years, e.g., robotics [75], autonomous driving [10] and visual-language navigation [83]. With the help of methods such as reinforcement learning methods and BERT based methods, significant progress has been made in the low-level control aspects of embodied intelligence. However, in the real world, they still face significant challenges that have not been resolved: *high-level* planning capabilities in complex tasks, transferability and generalization in out-of-domain tasks and human-machine interaction capabilities in daily life.

3.2 Unique abilities of LLM Agent

As introduced before, LLM powered agents provide us new chances for solving these challenges of embodied intelligence. Due to the powerful language understanding and generation ability, it is much easier for LLM agents to communicate with human effectively via language. The world knowledge and common sense contained in LLM make the LLM agents very capable of understanding the diverse real world, endowing them with extraordinary generalization abilities in massive tasks. Besides, the tool and memory enhanced agents are also able to conduct long-term high-level reasoning, planning and self-update easily. These powerful capabilities make the LLM agents highly suitable for addressing key bottlenecks in embodied intelligence, creating new opportunities for its development.

3.3 LLM Agents as the brain of embodied intelligence

LLM agents can be regarded as the central intelligent brain of embodied system. They first act as the perception unit to handle the multi-modal sensory inputs from environment. Next, they menage and update their memory to learn from the experience and prepare for further processing. Leveraging their powerful reasoning and planning capabilities, LLM agents can break down the goals into small steps and generate low-level control actions to interactive with the environment and solve tasks.

4 Advances of LLM agent-based embodied intelligence

In the following, we elaborate on the recent advances of LLM agent-based embodied intelligence, focusing on the key areas of visual perception, robotics, navigation and search, task assignment, multi-agent systems, and embodied question answering (EmbodiedQA). The typical applications in these areas are illustrated in Figure , and the details are shown in Table 1.

4.1 Visual perception with LLM

The visual perception module is designed to provide rich information for embodied agents to accomplish downstream tasks. It takes the RGB or RGBD video stream from cameras on the agents as indispensable input. RGB videos are streams of 2D images, similar to most videos in daily life. In contrast, RGBD videos add depth information for each pixel in the videos, captured from binocular cameras. Natural language input is often included to enhance the semantic understanding of visual models via LLMs. The output of the module depends on specific tasks, with major output forms including embedding, detected objects, and segmentation results.

Table 1: A list of representative works of large language model agent-based embodied intelligence.

Domain	Advance	Application/Task
Visual perception	R3M [45]	Pioneer of time-aware representative learning for robots
Visual perception	VIP [42]	Time-aware representative learning for Reinforcement learning
Visual perception	Voltron [29]	Capture video semantics at different levels
Visual perception	3D-LLM [20]	Capture 3D features with 2D vision-language models
Visual perception	ShapeLLM [51]	Directly capture 3D features of point clouds
Visual perception	3D-vista [88]	Language-vision alignment in 3D scenarios
Visual perception	3D-Concept graph [18]	Object-base representation in natural languages
Visual perception	Interactron [32]	Adaptive learning for object detection in fixed environment
Visual perception	[7]	Memorizing objects in world coordinates
Visual perception	[59]	Joint optimization of camera adjustment and embodied object detection
Robotics	Code as Policies [37]	Robot-centric formulation of LLM generated programs
Robotics	RT-2 [5]	Vision-language-action models transfer web knowledge to robotic control
Robotics	ManipLLM [36]	Image-text multimodal large language models for robot control
Robotics	FILM [44]	Following embodied instructions in Language with modular methods
Robotics	VoxPoser [24]	A motion planner for open-set of instructions and objects with language models
Robotics	SayCan [1]	Robotic manipulation with world-grounding and task-grounding
Robotics	Exceptional Handling [68]	World-grounding manipulation with exceptional handling of unacceptable results
Robotics	AutoRT [2]	Autonomous robot task generation and execution in real-world
Robotics	TidyBot [70]	A personalized tidying-up robot
Robotics	LLM-Personalize [19]	Aligning task executions with preferences in partially observable scenarios
Navigation and Search	NavGPT [85]	Navigation in indoor environment
Navigation and Search	Truong et al.[86]	Indoor pretraining and outdoor naviagation
Navigation and Search	RILA [78]	Zero-shot semantic audio-visual navigation
Navigation and Search	ESC [86]	Zero-shot search
Navigation and Search	PONI [55]	Search and navigate in unknown environment
Navigation and Search	L3MVN [81]	Search by infer frontier map
Task Assignment	LLM-Planner [63]	Few-shot planning capabilities to specific agents
Task Assignment	TaPA [71]	Alignment method combined with a visual perception model
Task Assignment	Huang *et al* [23]	Enhancing the executability of actions
Task Assignment	GITM [87]	Hierarchical task decomposition and generating action plans
Task Assignment	SMART-LLM [28]	Convert high-level task instructions into multi-robot task plans
Task Assignment	Co-NavGPT [80]	Multi-robot collaborative visual target navigation
Multi-agent Systems	Camel [35]	A communicative agent framework
Multi-agent Systems	SimClass [84]	A multi-agent virtual classroom simulation framework
Multi-agent Systems	S^3 [17]	A social network simulation system
Multi-agent Systems	ChatDev [53]	a chat-powered software development framework
Multi-agent Systems	Du *et al* [13]	Agents debate their responses to enhance mathematical reasoning
Multi-agent Systems	MAD [38]	Addressing the Degeneration-of-Thought (DoT) problem
Multi-agent Systems	Xu *et al* [73]	A tuning-free framework for engaging LLMs in Werewolf games
EmbodiedQA	ScanQA [3]	Static question answering
EmbodiedQA	SQA3D [41]	Static question answering
EmbodiedQA	OpenEQA [26]	Active question answering

A common question is what distinguishes embodied vision tasks from other computer vision tasks. The environment in which the embodied agent operates makes a significant difference. Embodied agents work continuously in specific and real-world environments, while traditional computer vision algorithms typically work on pre-collected datasets. The environments usually cover a certain area and are less variable than datasets. Besides, embodied agents can engage with and explore the environment, meaning they can actively collect data based on specific demands. In summary, we list the particular abilities required for embodied vision below:

- **Ability to capture videos**: The agent takes a video stream as input, and what it sees in the next step is relevant to its current action. Thus, the vision module needs to be aware of the temporal order of frames to assist causal understanding.
- **Ability to understand 3D scenarios**: Embodied agents work in real-world 3D environments, so they need to capture 3D features to better understand spatial relationships in scenarios.
- **Ability to interact with the environment**: With the ability to move and explore, agents can interact with the environment to gather more information. This includes actively adapting, memorizing, and exploring the environment.

4.1.1 Video understanding

Understanding videos requires the model to be aware of the temporal order among frames, which conveys the overall information of a video segment. For example, a video may depict a robot moving

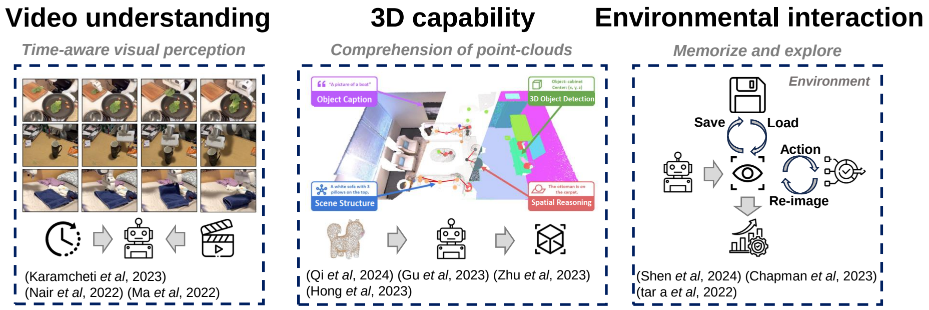
Video understanding
Time-aware visual perception
(Karamcheti et al, 2023)
(Nair et al, 2022) (Ma et al, 2022)
3D capability
Comprehension of point-clouds
"A picture of a boat"
Object Caption
Object: cabinet
Center: (x, y, z)
3D Object Detection
A white sofa with 3 pillows on the top.
Scene Structure
The ottoman is on the carpet.
Spatial Reasoning
(Qi et al, 2024) (Gu et al, 2023) (Zhu et al, 2023)
(Hong et al, 2023)
Environmental interaction
Memorize and explore
Environment
Save
Load
Action
Re-image
(Shen et al, 2024) (Chapman et al, 2023)
(tar a et al, 2022)

Figure 1: Illustration of Embodied Vision Modules

a box from left to right, and the visual perception module needs to capture the continuous location changes of the box. The following works achieve temporal awareness through different mechanisms.

Reusable Representations for Robotic Manipulation (R3M) achieve time awareness through contrastive learning [45]. The model is designed to distinguish the representations of different frames in the embedding space. Representations of frames with smaller time intervals are expected to be closer in the embedding space, while representations of frames further apart in time are repelled. Visual-language alignment is also implemented in the representation space. This attraction or repulsion is achieved via contrastive losses in a CLIP framework, resulting in significant improvements in most embodied tasks. **Value-Implicit Pre-training (VIP)** optimizes the representation of video frames from another perspective [42]. It claims that downstream reinforcement learning (RL) algorithms benefit from representations with these features: a) the start and end frames of an operational video should be close in the embedding space, and b) consecutive frames should be repelled. This temporal relationship is also captured via contrastive learning. **Voltron** makes deeper use of language to enhance the semantic understanding of videos. It employs an auto-encoder framework that takes different time frames and language descriptions as input and output [29]. The training strategy includes two parts: a) language-conditioned reconstruction, using language and masked frames to reconstruct frames, enabling lower-level understandings, and b) visually-grounded language generation, using masked frames to reconstruct full frames and language descriptions, forcing the model to grasp higher-level semantics by summarizing the whole video into captions. This two-level understanding enriches its performance on tasks like imitation learning.

In summary, video-aware modules aim to use a model with a few input frames to encode the entire video. These models capture the relevant temporal order of frames, which is essential for embodied operations. The language modality in these works assists semantic understanding in a relatively straightforward manner.

4.1.2 3D capability

In 3D vision tasks, the basic elements of input shift from 2D pixels to 3D point clouds, which are voxels in real-world space. 3D tasks like grounding and Q/A are usually more difficult than 2D ones, as they require the embodied agent to capture more spatial comprehension and demand higher computational cost. Additionally, 3D datasets and large pre-trained models are less sufficient than 2D ones. Therefore, some 3D models include 2D vision-language models (VLMs) and adapt them to 3D tasks, while others design specific modules to capture 3D scenarios directly.

3D-LLM constructs 3D features from multiple 2D images [20]. Powerful 2D vision models are applied to capture features in 2D images, and then the features of multiple 2D images are converted to the embedding of each 3D voxel via three 2D-3D reconstruction methods. These 3D features are then used to fine-tune the vision-language model for downstream tasks, where it achieves good performance at a relatively low cost. While benefiting from 2D pre-trained models, the 2D-3D reconstruction step may cause information loss. Thus, **ShapeLLM** utilizes a 3D representation module called ReCON++ to directly capture the representation of point clouds, and similarly uses these features to train multimodal LLMs [51]. ReCON++ enables the model to perform multiple tasks like 3D-grounding, 3D-captioning, and scene understanding. **3D-Vista** focuses more on language-vision alignment in 3D scenarios [88]. It applies the representation of point clouds and language descriptions separately, and then uses a masked autoencoder architecture to enable cross-modality understanding. All the above works require an embedding vector for each voxel in the area, which is expensive to compute. **3D concept graph** shifts from voxel-based to object-based representation to reduce complexity [18]. It sets up a graph structure where the nodes are detected objects, and edges are spatial relationships between objects. All nodes and edges are attached with natural language descriptions from VLMs, enabling downstream controls with LLMs. The concept-graph requires minimal training and shows rich information in the graph architecture for control and planning tasks.

These 3D embodied vision models involve deep modality alignment and fusion via vision-language models. On the input side, language assists the model in obtaining semantic comprehension, while on the output side, it can also act as a human-friendly interface to verify the vision understanding.

4.1.3 Environmental interaction

In addition to capturing specific characteristics of the environment, embodied agents can move and explore intelligently according to their demands. This enables active interaction in visual perception. For instance, the agent may move to better detect an object or memorize the location of a previously detected object. Concepts like reinforcement learning and adaptive learning contribute to this aspect by combining visual tasks and robot controls.

Interactron introduces an adaptive learning method for embodied object detection, which contains two modules: the ”detector” and the ”supervisor” [32]. The detector is continuously adapted in both training and testing, while the supervisor generates a supervision signal during the testing stage. This allows the detector to better fit the specific testing environment gradually and increase the success rate. Instead of fine-tuning the model in a specific environment, [7] introduces a memory technique to enhance embodied object detection. The memory contains the location of objects in world coordinates, generated via projection and used to augment input pixels by reverse projection. The memory of world coordinates has proven effective and informative for detection tasks. A language module with a CLIP architecture is also attached to enable open-world detection. The embodied agent in [59] is designed to control the camera to improve detection in the environment. A reinforcement learning algorithm with a student-teacher framework is applied to co-optimize camera adjustment and object detection. Specifically, the model encodes the states, rewards, and actions into tokens and uses a GPT architecture to generate actions. This allows flexible task instructions in the form of natural language, enhancing the model’s overall understanding.

In short, embodied agents benefit from adapting to and exploring environments, which enriches their potential for further cognition and understanding. The language modality in these models brings rich information, and vision-language alignment contributes to overall performance.

4.2 Robotics

4.2.1 Robotic control

We present the LLM agent-based embodied intelligence of robotic control in the following two scenarios.

Motion control. With the aim to accomplish a predefined motion trajectory or behavior, motion control refers to the process of parsing high-level instructions to regulate and manage the movement of robot joints and components in a specific translation and rotation. Since it has been shown that LLMs trained on code completion can generate basic Python programs from docstrings [8], some researchers discovered that these code-writing LLMs can be applied to write robot policy code given by natural language commands. For instance, Liang *et al.* [37] propose Code as Polices (CaP), a robot-centric formulation of language model generated programs (LMPs) that can represent reactive policies like impedance controllers as well as waypoint-based policies including visionbased pick and place and trajectory-based control. After offering prompts to LLMs, which consists of available basic APIs and ”demonstration” examples, the LLMs will generate robot policy code with classic logical structure in iteration. In each iteration, LLMs implements hierarchical function generation by modifying and supplementing errors such as undefined functions in the historical generated code to generate more complex and reliable execution code. Based on this, the robot can understand

and execute corresponding instructions without additional fine-tuning procedure, thus improving the robot's ability to perform complex tasks. On the other hand, as training robots requires large, diversified and high-quality datasets to improve their ability to parse high-level instructions, while in reality data sets are usually limited, many researchers are committed to improving the generalization ability of robots motion control under limited data conditions. The RT-2 model [5], for example, integrates vision-language model (VLM) into end-to-end robot control, in order to boost the robot's capacity for generalization and emergent semantic reasoning. Their aim is to enjoy the advantages of large-scale pre-training on Internet-scale language and visual-language data, while enabling a single end-to-end training model to learn to map robot observations to actions. They thus put forth the vision-language-action (VLA) paradigm. Using images as input, the model initially carries out VLM pre-training, producing token sequences that reflect natural language text. Next, the robot actions are encoded. Each action, including 6-DoF positional and rotational displacement of the robot end-effector, will be converted into a text token to be suitable for VLM model fine-tuning by using the standard VQA format. Most importantly, the key technical detail of RT-2 model is co-fine-tuning robotics data with the original web data instead of naive finetuning on robot data only, which significantly improves robot performance. While Li *et al.* [36] considering that robots can only trained on a limited category within a simulator, they introduce Multimodal Large Language Models (MLLMs) to enhance the stability and generalization of robots control. In particular, they proposed the ManipLLM model, which first aligns the encoded text prompt with the CLIP-encoded RGB image, then input it into LLaMa. This allowed for the realization of multimodal understanding of text and image, and at the end, the robot end-effector pose will be output for real-world operation. The model fine-tuning is mainly divided into three parts: The first is the *Object Category Identification* process, which is used to identify the object category. The second is *Affordance Prior Reasoning* process, which aims to enable the model aware where of the object region can be manipulated, specifically by letting the LLMs do judgment questions to have the ability of determining which pixels can be used to operate the object. Finally is the *Finetuning* and *Mask Language Modeling* processes, which are used to generate the precise robot end-effector pose, including the contact point and gripper direction.

Motion planning. While motion control focus on generating accurate robot end-effector poses, motion planning pay more attention in planning the robot's motion path from the beginning point to the target point and avoids obstacles. For example, Min *et al.* [44] address the issue of embodied instruction following and propose a modular structural model called FILM. FILM will first execute language processing (LP) upon receiving the instruction, parsing the high-level instructions into a structured sequence of subtasks utilizing two BERT modules. Secondly, at every time step, a semantic top-down map of the scene is updated based on predicted depth and instance segmentation processed from the egocentric RGB images. Additionally, the deterministic policy will output navigation if the robot is uncapable of performing actions on the target object, otherwise interaction action. Lastly, at a coarse time scale, the semantic search policy outputs the coarse-grained position information of the search goal. Especially for small and hard-to-find objects, this policy is to use the observed spatial configuration of larger objects to forecast the positions of smaller objects. Huang *et al.* [24] aim to synthesize robot trajectories, a dense sequence of 6-DoF end-effector waypoints given by an open-set of instructions and objects, and therefore propose a model called VoxPoser. By leveraging code-writing as well as affordances and constraints inference capabilities of LLMs, the model incorporates VLM, obtaining the spatial geometric information of the objects, with LLMs, to compose 3D value maps. The value map contains affordance maps and constraint maps. The former descirbes the robot's target manipulation area, and the latter is the area that the robot needs to avoid. According to the two 3D value maps, a traditional path search algorithm is employed for motion planning, and the path trajectory with the highest reward value is output as the robot motion execution trajectory.

4.2.2 Robotic manipulation

LLM agents can also significantly contribute to robotic manipulation by interpreting high-level tasks expressed in natural language and manipulating robots to accomplish them. Despite the rich knowledge, reasoning, and decision-making capabilities of LLMs, they face critical challenges when directly applied to robotic manipulation. Two significant limitations are their lack of word-grounding information and personalized information, both of which are essential for practical real-world tasks. Here, we discuss recent advances that address these issues, enabling the effective use of LLM agents in robotic manipulation tasks.

World-grounding. The majority of real-world robotic manipulation tasks are physically-grounded, however, LLMs lack awareness of the current physical environment in the specific task, which may result in answers that are semantically correct but physically infeasible due to environmental restrictions. For example, when asked to clean a spill with the prompt ”I spilled my drink, can you help?”, an LLM might suggest “you could try using a vacuum cleaner”, which is reasonable but impractical if no vacuum cleaner is available in the environment. This is a major challenge in LLM agent based robotic manipulation. Therefore, it is crucial to incorporate physical world-grounding into the LLM’s guidance to ensure the generated instructions are not only reasonable but also feasible.

One solution, presented by the SayCan framework [1] , involves scoring each possible skill a robot might execute based on its reasonability and feasibility in the environment. In this framework, the robots manipulated by the agent are equipped with some basic skills such as ‘pick up the sponge’. The agent consists of two parts, namely Say and Can, where Can is a learned affordance function that evaluates the probability that the certain skill can be completed in the current circumstance, providing world-grounding, and Say is the LLM which determines whether that skill is useful for completing the high-level task, providing task-grounding. By multiplying these two probabilities, SayCan assesses the likelihood of each skill to execute successfully and contribute to the accomplishment of the task, according to which the agent decides the next guidance for the robot by choosing the most probable skill. While SayCan has demonstrated viability in various real-world tasks, it has some limitations, it cannot handle scenarios where the highest scored skill is implemented but still fails, and it also may scale inadequately to dynamic environments since the scores of the skills need to be renewed whenever the circumstance changes. Wang *et al.* [68] proposes another method that can address these deflects by restricting the output with prompts. In this work, the LLM is provided with a set of executable primitive actions (e.g., “move_to”) as Pythonic import statements which include only movements with no object, ensuring feasibility of the primitive actions regardless of the environment. This encourages LLM to constrain the outputs to these executable pre-defined actions, with the available objects in the environment being the parameters of Pythonic action functions. By this mean ,LLM is supposed to generate an executable action sequence in the format of a Pythonic list, with actions and objects being the function and the parameters respectively. Additionally, to avoid the hallucinations of LLMs, this approach also employs an exception handling module to add new prompts when the LLM’s responses deviate from the expected format, which further ensured the successful execution of the task. World grounding is essential for large-scale real-world robotic manipulation tasks. For instance, AutoRT [2] is a system that utilizes LLM agents to propose instructions for robots in order to collect large-scale diverse data from the robots. A fundamental part in the process is establishing rules for the robots and assessing whether the generated tasks adhere to these rules and are feasible. After deploying the system in a variety of real-world environments, AutoRT successfully achieved its data collection objectives with careful consideration of world grounding.

Personalization. Aligning task execution with individual user preferences is another crucial aspect of robot manipulation, particularly for household robots. For example, given a general task like ”Put the socks on the floor in the right place,” different users may have different preferences for where and how the socks should be placed. For a household robot, only outcomes that match the user’s preferences are acceptable. In these situations, personalization is of great importance. Since LLMs inherently possess only general knowledge, it is necessary to train them to incorporate personalized information.

Tidybot [70] is a personalized tidying robot that can place users’ belongings into appropriate receptacles. It leverages LLMs’ summarization capabilities to extract user preferences from a few given examples, specifically focusing on personalized receptacle selection and primitive action choices. Once a specific object is detected, the LLM determines which primitive actions to apply and which receptacles to use based on these references, enabling the robot to tidy up in a personalized manner. This approach has been effective in experiments conducted in real-world scenarios. However, it requires all receptacles to be known as prior knowledge, which is often not feasible in practical applications. Han *et al.* [19] proposed another robotic agent framework that can perform object rearrangements in multi-room and partially observable household scenarios. This framework comprises three components: a context generator that provides the current household context as a prompt periodically, an LLM planner that perceives tasks and generates a sequence of high-level actions based on context and perception, and a controller that executes these actions. The agent is optimized through imitation learning and self-training to learn personalized user preferences. Specifically, during the imitation learning phase, the LLM agent is initially aligned with user preferences through demonstrations. Following this, it is further optimized through self-training. This method achieved excellent performance in experiments, demonstrating the potential of using LLM-based agent in personalized robotic manipulation tasks.

These advancements demonstrate the potential for LLMs to effectively guide robotic manipulation tasks by addressing the critical issues of world grounding and personalization, paving the way for more practical and meaningful applications in real-world environments.

4.3 Navigation and Search

Navigation and search are critical tasks for embodied intelligence [12, 69, 33, 62]. Navigation refers to guiding a robot to a specified location based on map information or known path data, sometimes following linguistic instructions, while search refers to finding a specific item or target in the absence of environmental or path information. Large language models have good commonsense understanding of the world and basic abilities for reasoning and decision-making, thus researchers have proposed various methods to leverage there powerful capability to improve navigation and search tasks. These methods typically retain traditional sensors and controllers but employ LLMs as the intelligent core of the agents, providing the agents with commonsense knowledge, basic reasoning, and decision-making abilities.

4.3.1 Navigation

Visual language navigation (VLN) refers to the task of guiding an agent through an environment based on visual inputs and language instructions. Traditional VLN approaches often suffer from poor generalization, struggling to handle unknown environments and language commands effectively. Some studies have proposed using topological maps and zero-shot methods to accomplish VLN tasks, but these methods involve implicit decision-making processes that are difficult to interpret and control. NavGPT [85] leverages the reasoning capabilities of large language models to perform VLN tasks, offering high generalization and interpretability. It decomposes the navigation task and employs the LLM's capabilities at various stages, including instruction decomposition, commonsense knowledge integration, landmark identification, navigation progress tracking, and plan adjustment. For each task, it designs a specific module and utilizes a prompt manager to drive LLM-based decisions. It predicts actions in a zero-shot manner, relying on the pre-trained knowledge of LLMs on large corpora rather than learning parameters from VLN datasets. Truong et al.[86] proposed a navigation method that pre-trains in indoor environments and then transfers to outdoor environments. This approach does not explicitly use LLMs but incorporates the pre-training concept which are the core of LLMs. The work employs an end-to-end deep reinforcement learning method to train visual navigation strategies in simulated indoor environments. Additionally, it uses extra environmental context information (such as satellite maps or human-drawn sketches) to guide robots in outdoor navigation. The robots utilize this additional contextual information along with onboard sensors to navigate outdoor environments, avoiding unmarked obstacles on the map. This method enables quadruped robots, trained solely in simulated indoor environments, to successfully navigate several hundred meters in new outdoor environments. RILA [78] employs large language models to assist in zero-shot semantic audio-visual navigation, achieving high-performance navigation in complex environments. Traditional methods often lack spatial reasoning capabilities. To tackle this issue, RILA introduces LLMs as the spatial reasoning core. RILA is composed of three modules: the perception module, the imaginative assistant, and the reflective planner. The perception module processes audio and visual inputs to generate text-based descriptions. The imaginative assistant infers the room layout and provides strategic guidance to enhance spatial understanding. In the reflective planner, the LLM performs zero-shot reasoning and decides the direction of exploration. RILA significantly outperforms previous methods in scenarios involving long distances and intermittent sounds.

4.3.2 Search

Compared to navigation, search emphasizes more on finding **unknown** objects in **unknown** environments. Traditional methods use pre-trained visual encoders (such as ResNet, CLIP, etc.) to encode first-person view images, which are then fed into the navigation agent network. Search strategies are trained through large-scale imitation learning or reinforcement learning. These methods require extensive training data and are limited to specific environments and target categories. Some other works explicitly constructs semantic maps and infers the location of target objects based on semantic information from the training data. These methods also need training tailored to specific environments and target categories. However, using LLMs enables open-world object search without any prior environment search experience or visual environment training.

ESC [86] proposes a zero-shot search approach that leverages pre-trained text-image matching models for open-world semantic scene understanding, obtaining location information for target objects and rooms. Additionally, it combines the commonsense reasoning capabilities of large language models to infer the relationship between target objects and rooms, guiding the exploration strategy. What's more, ESC designs a soft constraint-based exploration strategy, selecting frontier areas most likely to be near the target object for exploration. In summary, ESC utilizes the capabilities of pre-trained models and large language models to achieve effective target search in zero-shot scenarios. PONI [55] designs a method for agents to search and navigate to a specified category of target objects in an unknown environment. It uses a potential function to guide navigation, facilitating the discovery and navigation to target objects. The potential function network derives value functions from a partially filled semantic map to determine where to search for objects. In this method, the design of the potential function can be substituted with LLM-designed ones. PONI effectively achieves target object search and navigation, demonstrating high efficiency and success rates. L3MVN [81] leverages large language models to infer semantic information about frontiers and select long-term goals, enhancing exploration and search efficiency. In this method, a frontier map is generated by combining the explored map and obstacle map. Initially, the maximum contours from the explored map are identified to extract the explored edge. Subsequently, the obstacle map's edge is dilated, and the frontier map is derived as the difference between the explored and obstacle maps. This approach allows LLMs to interpret the semantics of the frontier, facilitating the selection of optimal long-term exploration targets and improving overall navigation and search performance.

4.4 Task Assignment

Task assignment is the process of allocating specific tasks to particular agents, which is crucial for enhancing the ability of agents to handle complex tasks. In multi-agent systems, task assignment optimizes resource utilization and collaboration efficiency by reasonably distributing responsibilities according to the agents' capabilities and environmental demands.

LLMs provide powerful tools for understanding and solving complex tasks by processing and generating natural language, which benefits task assignment in embodied AI. LLMs significantly enhance the understanding of agents by transforming abstract instructions into concrete actions through their exceptional natural language comprehension abilities. By leveraging contextual associations, these models enable agents to better understand complex commands. LLMs can process large-scale data, allowing agents to adapt to environmental changes and optimize strategies. Through adaptive learning and scenario prediction, agents are better equipped to handle unknown environmental variations. In multi-task environments, LLMs achieve effective resource and priority allocation. By implementing multi-objective planning, priority management, and resource scheduling, they ensure that agents can simultaneously handle multiple tasks without interference. Lastly, LLMs enhance the robustness and real-time decision-making capabilities of agents. Through feedback mechanisms and error detection and recovery systems, agents can quickly recover from issues and continue executing tasks. Task assignment typically involves three main processes: planning [63, 71], decomposition [23, 87], and allocation [28, 80].

4.4.1 Planning

Planning involves designing and generating a series of ordered actions for embodied agents to achieve goals in a specific environment. Task planning requires agents to make efficient decisions and adapt to dynamic changes in the environment. The complete stages of task planning include (1) Environment Modeling, where agents first perceive and construct a model of the environment, forming the basis for planning actions; (2) Action Planning, which involves designing a series of specific actions to accomplish each step along the path; (3) Real-time Adjustment, where agents adjust their planning and execution strategies in real-time based on environmental changes or changes in task requirements. The success of task planning depends on accurate perception and modeling of the environment, effective decision-making algorithms, and flexible execution mechanisms.

LLM-Planner [63] provides few-shot planning capabilities to specific agents by leveraging LLM as a few-shot planner through the design of high-level planning, low-level planning, and dynamic replanning algorithms. TaPA [71] proposes a LLM-based alignment method combined with a visual perception model to generate executable plans based on the objects present in a scene. As a novel task planning agent, TaPA successfully integrates the semantic knowledge of LLM with real-world physical scenarios to produce executable action plans with a higher success rate.

4.4.2 Decomposition

Decomposition refers to the process of breaking down a complex overall task into several smaller and more manageable subtasks. This process helps simplify the complexity of the task, making each subtask easier to execute. Task decomposition improves the manageability of the task, enhances the system's robustness and flexibility, and enables agents to better adapt to dynamic and uncertain environments. A complete task decomposition includes four stages: (1) Task Analysis: First, the agent needs to understand and analyze the requirements and goals of the overall task. (2) Subtask Generation: Based on the results of the task analysis, the overall task is broken down into several subtasks. (3) Subtask Relationship Definition: Determine the dependencies and execution order between the subtasks, which is key to ensuring coordinated task execution. (4) Subtask Planning and Allocation: Develop specific execution plans for each subtask and allocate them to the corresponding agents or subsystems. Huang *et al* [23] proposes a method that enhances the executability of actions by taking into account the existing environmental conditions and translating task planning semantics. This suggests that when pre-trained language models are sufficiently large and given appropriate prompts, they can effectively decompose high-level tasks into intermediate plans without any further training. To address task generalization capabilities on the open platform of Minecraft, Zhu *et al* [87] proposed the Ghost in the Minecraft (GITM) framework. This framework utilizes LLMs for hierarchical task decomposition, generating action plans, and executing these plans through textual interaction. Textual knowledge collected from the internet is used to provide the necessary information for the decomposition.

4.4.3 Allocation

Allocation is the process of reasonably and effectively distributing tasks among the agents in an embodied intelligent system, aiming to enable them to collaborate efficiently and complete the overall task. This process aims to optimize system resource utilization, improve task completion efficiency, and ensure the reliability of task execution. Task allocation ensures effective coordination among agents, which is crucial for the smooth completion of complex tasks. When performing task allocation, it is necessary to comprehensively consider various factors such as the capabilities of each agent, the complexity of the tasks, and environmental changes. The complete task allocation process includes the following key stages: (1) Task Description: Clarify the specific requirements and goals of the task to provide clear guidance for task allocation. (2) Resource Assessment: Evaluate the capabilities, resources, and status of each agent to determine the types and scopes of tasks they can undertake. (3) Allocation Strategy: Develop strategies and methods for task allocation to ensure tasks are reasonably assigned based on the characteristics of the agents and the task requirements. (4) Execution Monitoring: Monitor the task execution in real-time and make dynamic adjustments according to the actual situation to cope with environmental changes and uncertainties in task progress. Kannan *et al* [28] introduce a framework called SMART-LLM, which leverages the capabilities of LLMs to convert high-level task instructions into multi-robot task plans. The framework guides the execution of task decomposition, coalition formation, and task allocation stages, all under a few-shot prompting paradigm directed by a programmatic LLMs. Yu *et al* [80] present Co-NavGPT framework, which utilizes LLMs to address the challenge of multi-robot collaborative visual target navigation. Co-NavGPT encodes environmental information as prompts input into LLMs, thereby assigning the most appropriate boundaries to each robot as long-term goals.

4.5 Multi-Agent System

A multi-agent system (MAS) is a system composed of multiple interacting intelligent agents. These agents are autonomous entities capable of making decisions and performing tasks to achieve their individual or collective goals. MASs enhance the capabilities of single LLM agents by leveraging the specialized abilities and collaborations among multiple agents. Through coordinated efforts, these systems can accomplish tasks beyond the reach of individual agents. Each agent in a MAS has distinct capabilities and roles, working together to achieve common objectives. This collaboration, which includes activities like debate and reflection, is particularly effective for tasks requiring deep thought and innovation. Recent advancements have demonstrated the potential of MASs in handling complex real-world scenarios, such as simulating interactive environments [47, 27, 84, 17], role-playing [35, 73, 53], and reasoning [13, 38, 82].

4.5.1 Simulating Interactive Environment

Simulating interactive environments refers to creating a virtual space where multiple agents can interact with each other, mimicking real-world scenarios and behaviors. In these environments, agents simulate human behavior by perceiving the environment, interacting with other agents, and performing specific tasks. LLMs play a crucial role in LLM-based multi-agent systems, providing capabilities such as natural language understanding and generation, multi-agent coordination and cooperation, environmental perception and adaptation, role-playing and personalization, and intelligent decision-making and reasoning. These functions enable agents to achieve more realistic, natural, and efficient interactions within simulated interactive environments, driving innovation and development across various application domains. [47] constructs a small town environment with 25 agents, where the agents engage in daily activities and exhibit a range of believable individual and social behaviors. Zhang *et al* [84] propose SimClass, a multi-agent virtual classroom simulation framework involving user participation. It demonstrates that LLMs can effectively simulate traditional classroom interaction patterns while also exhibiting emergent group behaviors. Gao *et al* [17] propose the S^3 system, a social network simulation system based on LLMs, and finds that agents can produce accurate population-level phenomena when simulating emotions, attitudes, and interaction behaviors.

4.5.2 Role-playing

Role-playing is a technique that involves simulating different roles and interacting within specific scenarios. In this way, multi-agents can be set to different perspectives, practicing and enhancing relevant skills. The application of LLMs in role-playing leverages their powerful natural language processing capabilities, providing realistic dialogue content, understanding context, and adapting to scenarios, thus offering a wide range of applications and advantages. Li *et al* [35] propose a communicative agent framework called Camel, and finds that this framework can guide chat agents to autonomously complete tasks through inception prompting, while generating conversational data to study the cooperative behaviors and capabilities of multi-agent systems. ChatDev [53] is a chat-powered software development framework using large language models to unify communication across design, coding, and testing phases, and finds that linguistic communication enhances system design and debugging while facilitating multi-agent collaboration. Xu *et al* [73] proposesa tuning-free framework for engaging LLMs in communication games, and finds that the approach effectively enables LLMs to play the Werewolf game and exhibit emerging strategic behaviors without parameter tuning.

4.5.3 Reasoning

Reasoning involves analyzing problems, drawing conclusions, making decisions, and generating coherent dialogue through logic and knowledge. Pre-trained on large-scale data, LLMs possess a vast knowledge base, enabling them to extract key information from text, understand its meaning, and handle both structured and unstructured data. LLMs-based MAS can manage multi-step reasoning tasks, breaking down complex problems and solving them incrementally. They can also integrate multi-modal information, such as text, images, and audio, to provide more comprehensive reasoning results. These advantages enable LLMs-MAS to deliver efficient and accurate reasoning capabilities in various complex tasks. [13] presents a ”society of minds” approach where multiple language model instances propose and debate their responses to enhance mathematical reasoning, factual validity, and overall language generation capabilities. Li ang *et al* [38] introduces the Multi-Agent Debate (MAD) framework to address the Degeneration-of-Thought (DoT) problem in large language models by having multiple agents engage in a debate and a judge manage the process, improving reasoning in complex tasks like commonsense machine translation and counterintuitive arithmetic reasoning. CoELA [82] is a framework that uses LLMs for multi-agent cooperation, showing improved performance in planning and communication tasks, and better human-agent interaction.

4.6 EmbodiedQA

Embodied Question Answering (EQA) [43, 66, 40] is a task for embodied agents to answer natural language questions by understanding the environment. Large language models have found extensive applications in various fields, such as conversational systems [25, 39] and machine translation [30, 46]. Compared to these traditional applications, EQA is more challenging because it requires not only language understanding but also the integration of visual information and environmental perception to perform real-time reasoning and decision-making in dynamic and complex environments. This makes EQA a multidisciplinary task, necessitating the integration of knowledge and technologies from computer vision, natural language processing, and robotics. The core of the EQA task is enabling agents to answer real-world questions through visual information and language comprehension [26]. LLMs naturally provide an interface for question-answer interaction, encompassing language comprehension, world knowledge, commonsense understanding, and the perceptual abilities granted by in-context learning, making them well-suited to empower Embodied QA. Various studies have explored the employment of LLMs for the task of EQA in different settings, such as open vocabulary support [43], situational query processing [11], 3D spatial understanding [3], and contextual comprehension [41]. EQA can be categorized into two types: static question answering, which relies on situational perception and memory, and active question answering, which depends on active exploration. In the following, we will introduce these two categories of work separately.

4.6.1 Static Question Answering

Static scene question answering refers to answering questions based on given, fixed scene information [3, 41]. In real-world 3D scenes, correctly understanding and answering questions related to object alignment, orientation, and positioning is crucial. ScanQA [3] introduces a novel 3D question answering (3D-QA) task, where the model receive visual information from an entire 3D scene and answer textual questions about the scene. ScanQA learns descriptors to associate linguistic expressions with the geometric features of 3D scans by integrating 3D object proposals and encoded sentence embeddings. This process enables the model to regress 3D bounding boxes to identify the objects described in the textual questions. Additionally, this work introduces a new ScanQA dataset. ScanQA demonstrates excellent performance on the 3D-QA task. SQA3D [41] focuses on situational question answering within 3D scenes. This work introduces the SQA3D task, which requires intelligent agents to understand contexts within 3D environments and answer related questions. SQA3D demands that agents first comprehend and locate their situation (position, orientation, etc.) within the three-dimensional scene based on textual descriptions, and then respond to questions pertaining to that context. To accomplish this task, agents must accurately understand the scenario and visualize the corresponding egocentric view. This work releases the SQA3D dataset based on ScanNet as a benchmark of SQA3D task.

4.6.2 Active Question Answering

Active question answering is to answer questions about the environment through active exploring. OpenEQA [26] introduces a novel framework for Embodied Question Answering (EQA) tasks, encompassing both situational memory and active exploration scenarios. The OpenEQA benchmark includes over 1600 non-templated questions designed to test various aspects such as attribute recognition, spatial understanding, functional reasoning, and world knowledge. These questions are accompanied by episodic histories, along with corresponding questions and answers. FOr episodic memory-based EQA (EM-EQA), the agent parses a sequence of historical perceptual observations. For active EQA (A-EQA), the agent must explore a scanned real-world environment to gather information necessary to answer questions. This work proposes the LLM-Match metric for scoring answers, which shows strong consistency with human evaluations.

The future development directions of EQA research mainly include the following aspects:

- **Multimodal Integration**: Further exploration of how to effectively integrate visual, linguistic, and other sensor data to enhance environmental perception and question-answering capabilities.

- **Reinforcement Learning**: Enhancing agents' exploration and decision-making abilities in dynamic environments through reinforcement learning, enabling them to better adapt to complex real-world scenarios.
- **Model Optimization**: Developing more efficient and robust foundational models and algorithms tailored to the specific needs of EQA tasks, narrowing the gap between machine and human performance.

5 Generalist Agents and Foundation Model

5.1 General Foundation Model

Generalized foundation model is the core basis for the aforementioned agent's functionality. As a component of embodied intelligence, foundation models can gather information through perception modules during interactions with the real environment, providing embodied agents with the necessary data for decision-making and enhancing their spatial understanding and self-perception [52]. Embodied agents can help eliminate illusions in foundation models and generalize the capabilities of foundation models [74]

5.1.1 Category of General Foundation Model

We classify the generalized foundation models into three categories: vision-centric foundation model, language-centric foundation model and low-level action centric foundation models. In most cases, the former two models are utilized to generate high-level actions and the latter one is utilized to generate low-level actions. Detailed information of these foundation models are introduced as follows.

Vision-Centric Foundation Model. Visual Foundation Models (VFMs) are specialized foundation models designed for image generation tasks. VFMs typically incorporate components of large language models, enabling them to generate images based on textual input prompts. Their paradigm is Visual-Action (VA), which involves making decisions directly based on visual task inputs.

An example of the VA paradigm is [77], which can transform traditional computer vision tasks into video generation tasks. This approach to visual foundation models is conceptually similar to models that use large language models (LLMs) for solving visual question answering. Both can break down high-level goals into specific sub-goals to generate answers for operations-rich tasks, where the state and underlying action spaces consist of pixels rather than text.

Language-Centric Foundation Model. Language Foundation Models (LFMs) Using their knowledge and capabilities to make decisions for embodied tasks. In the field of Natural Language Processing (NLP), large language models (LLMs) such as BERT, GPT-3, GPT-4, and MPT-30B are foundational models, enabling enterprises to build custom-tailored chat or language systems for specific tasks and to understand human language in order to enhance customer engagement.Their paradigm is Visual-Language-Action (VLA), which entails understanding language instructions, perceiving the visual environment, and generating appropriate actions.

Gato [56], an example of the VLA paradigm, presents a model capable of playing Atari games, answering visual questions, and manipulating a robotic arm to stack blocks. It utilizes a unified tokenization scheme that allows training across all multimodal tasks simultaneously. During training, to differentiate between tasks within a modality, specific portions are extracted and prefixed with prompts. These prompts can be acquired through interaction with the environment. Gato represents a significant milestone, demonstrating the potential to build ”multimodal, multi-task generalist embodied agents.”

Low-level Action-Centric Foundation Model. The generality of large models is reflected in the variety of tasks and operational scenarios. The diversity of task scenarios is manifested in the variety of operational environments, objects, and objectives. Foundation models for underlying actions generalize these actions to accommodate different embodied tasks and various application scenarios within the same task.

The article [74] proposes a network model for generalizing the underlying robotic manipulation task scenarios. Robotic manipulation tasks can be decomposed into two parts: a task planning module that provides the trajectory of the object, and a manipulation module that guides the robot to achieve this trajectory. This network consists of a feature extractor and a CVAE, which optimizes both the task planning and manipulation modules simultaneously. The model network takes as inputs the robot manipulator's point cloud, physical properties of objects, target motion, and a mask of the manipulation area, and outputs contact points, contact forces, and related post-contact motions. Additionally, complex algorithms are introduced to optimize the operational effectiveness of different types of robotic hands.

5.1.2 Training Paradigms of the Generalized Foundation Model

The training paradigms for general large models can be divided into two main categories: end-to-end learning and compositional learning.

End-to-end Training. It is relatively straightforward, treating the LLM as a high-level task planner and integrating multi-modal modules directly into the LLM. Because it requires training with data from multiple tasks and modalities, the training cost is relatively high. [5] is a classic example of end-to-end training, dedicated to applying the capabilities of multi-modal LLMs directly in robotic tasks. It introduces collaborative fine-tuning, i.e., jointly fine-tuning on robotic trajectory data and visual language tasks (such as visual question answering). This training scheme enhances the model's generality and emergent capabilities. RT-2 represents the effort to integrate low-level control strategies with high-level task planners to pursue a more comprehensive robotic system. Subsequent models in the RT series, like RT-X, use larger and more diverse datasets, achieving better effects.

Compositional Training. In the compositional training process, the LLM acts as the brain, and other skill libraries act as the cerebellum, with the LLM as a high-level planner only needing to call upon these skill libraries. The calling process can use language or code as an intermediary to exchange multi-modal information. [65] This is an example of training embodied large models in a compositional way. It replaces the LLM's input and output modules with perception modules and action decoders, training only these two components during the pre-training phase; during the training phase, it further freezes the visual encoder in the perception module and adopts reinforcement learning strategies, updating policies based on sampling of collected data. This work presents a very important conclusion that embodied intelligence can continuously learn and improve through environmental interactions without supervision, providing new insights into data collection for embodied intelligence.

5.2 Generalist Embodied Agent

(fengjie, jinzhu) Although general foundation models possess strong general capabilities, most agents built upon them are specifically designed and fine-tuned for particular tasks in order to achieve optimal performance on those specific tasks. Some researchers also try to build generalist agents to solve diverse tasks as much as possible. For example, Huang et al. [22] try to build a embodied generalist agent for 3D in-door world with carefully design instruction tuning dataset and LoRA optimization methods. Besides, Cai [6] explore the potential of building generalist agents in the digital world by learning from diverse videos. Building generalist agents is the essential path to achieving general embodied intelligence. We look forward to more exciting new works in the future.

6 Platform, simulator and benchmark

Since training and testing LLM agent-based embodied intelligence in the real world is extremely costly and time-consuming, e.g., building robots and constructing scenarios, the use of virtual platforms or simulators is the preferred approach. At present, a number of platforms and simulators have emerged to support LLM agent-based embodied intelligence training and evaluation, and benchmarks have also been constructed on the basis of such platforms and simulators. In this section, we will introduce the popular works about platform, simulator and benchmark which are listed in Table 2, and try to categorize them.

Scenario-specific Simulation Platforms. First, we summarize the most common approach, which is scenario-specific simulation platforms. The approach realizes the simulation and testing of a specific scenario embodied intelligence by developing a corresponding simulation program (usually using a 3D video game engine) for the scenario, such as household activities, and providing a pro-

Table 2: Platforms, simulators and benchmarks for LLM agent-based embodied intelligence.

Category	Platform & Simulator	Benchmark
Scenario-specific Simulation Platform	AI2-THOR [31]	ALFRED [60]
	AI2-THOR [31]	TaPA [71]
	ALFWorld [61]	ALFRED [60]
	VirtualHome [49]	Watch-And-Help [50]
	ThreeDWorld [15]	ThreeDWorld Transport Challenge [16]
	iGibson [34]	BEHAVIOR [64]
	Habitat [48]	Habitat [48]
Open World Game	MineCraft	MINEDOCO [14]
	Red Dead Redemption II	CRADLE [67]
Generative Model	UniSim [76]	-

gramming interface to control the behavior of the agents within the program. AI2-THOR [31] is a simulator implemented by Unity 3D game engine, which focuses on near photo-realistic 3D indoor scenarios and allows AI agents to navigate and interact with objects like microwave, bread, toaster and so on, by Python API. Benefiting from the rich scenarios and interaction capabilities provided by AI2-THOR, several benchmarks are designed and published. ALFRED [60] is designed as a benchmark with 25k language directives and the corresponding expert demonstrations of household tasks. AI agents are asked to decompose task steps, navigate, and interact in AI2-THOR to complete the household task based on vision acquired from the simulator and the input natural language directives. To further test the ability of large models to handle complex tasks, in TaPA [71] the researchers construct datasets with longer action sequences for benchmarking compared to ALFRED. ALFWorld [61] aligns the interactive text-based simulation engine with the ALFRED dataset to textualize embodied simulation scenarios, enabling agents training to benefit from abstract textual simulation environments. VirtualHome [49] is also a household simulator that provides interactive objects, vision outputs, and control interfaces. Compared to AI2-THOR, the main difference in VirtualHome is its support for multi-agent simulation. Utilizing this feature, a benchmark named Watch-And-Help [50] is designed to test social intelligence between agents. The benchmark requires an agent to guess the task intent after observing a demonstration of the task, and then help another agent to complete the task faster. ThreeDWorld Transport Challenge [16] designs a series of object transporation tasks in a 3D virtual physics simulator ThreeDWorld [15], requiring the agents to locate the target objects, pick them up, and transport them to specified locations. iGibson [58, 34] is also an indoor simulator like AI2-THOR, which provides more types of virtual sensor signals and more ways to do domain randomization with faster simulation speed compared to AI2-THOR. BEHAVIOR [64] is a benchmark for household tasks that accompanies iGibson like ALFRED. Habitat [48] is a simulator for studying human-robot interaction, which enables human-in-the-loop interaction and provides collaborative human-robot tasks in home environments as benchmark. In summary, scenario-specific simulation platforms are highly adapted to the needs of agents training and testing for embodied intelligence tasks. However, due to the difficulty of developing for scenarios, most of the current simulators and benchmarks are limited to indoor scenarios, with little work focusing on more complex outdoor scenarios.

Open World Games. For embodied intelligence, video games actually provide ready-made simulation environments. Among the many video games out there, there is a category known as open world games that are ideally suited with verifying the level of embodied intelligence of agents. These games usually allow the game character to perform actions at will in the game world, including moving, exploring, fighting, interacting with objects, using props, and changing terrain, etc. One of the most popular open world game is MineCraft. MINEDOJO [14] proposes a framework built on the MineCraft game for embodied agent research, which contains more than 1000 open-ended and language-prompted tasks in the game. The observation space of agents consists of the RGB frame, voxel, inventory, location, etc. The action space consists of moving, attacking, equipping, etc. MINEDOJO draws on the variety of scenarios and actions in the MineCraft game itself to provide a wide range of tasks including, but not limited to, gathering specific resources, finding special terrain, killing enemies, building specific structures, etc. Besides, in CRADLE [67], the researchers make LLM-based agents complete a series of tasks in the game of Red Dead Redemption II by receiving video and audio data as inputs and deciding keyboard and mouse operation as outputs. Overall, open-world games have great potential for use in testing LLM agent-based embodied intelligence, and they exempt researchers the process of targeted development of simulation environments at the cost of differences in the games themselves relative to the real world. This may affect the migration of the embodied agents to real-world applications.

Generative Models. All of the above approaches are based on the basic idea of modeling and simulation, regardless of whether the process of modeling and simulation is done by researchers or game developers. This limits the ability to quickly migrate and test new scenarios. Currently, generative models [57, 9, 72] can generate realistic images and videos with controllability. Through the powerful generative capability of the diffusion model, UniSim [76] collects a large number of multi-dimensional embodied intelligence-related video datasets, and realizes the generation of videos observed by embodied intelligences based on multi-frame histories with textual actions as conditions. From UniSim as an example, we can see that generative models plays an important role in the environment construction and simulation with embodied intelligence. Using generative models, embodied intelligence researches are no longer limited to specific scenarios, which will accelerate the migration of embodied intelligence between different scenarios and facilitate real-world applications.

7 Application

Embodied intelligent agents are systems that integrate physical and cognitive capabilities to interact with the physical world. They appear as robots in various domains, including industrial automation, scientific experimentation, and autonomous driving, while in virtual realms, they are known as digital humans. This paper presents an overview of significant advancements and applications in these areas.

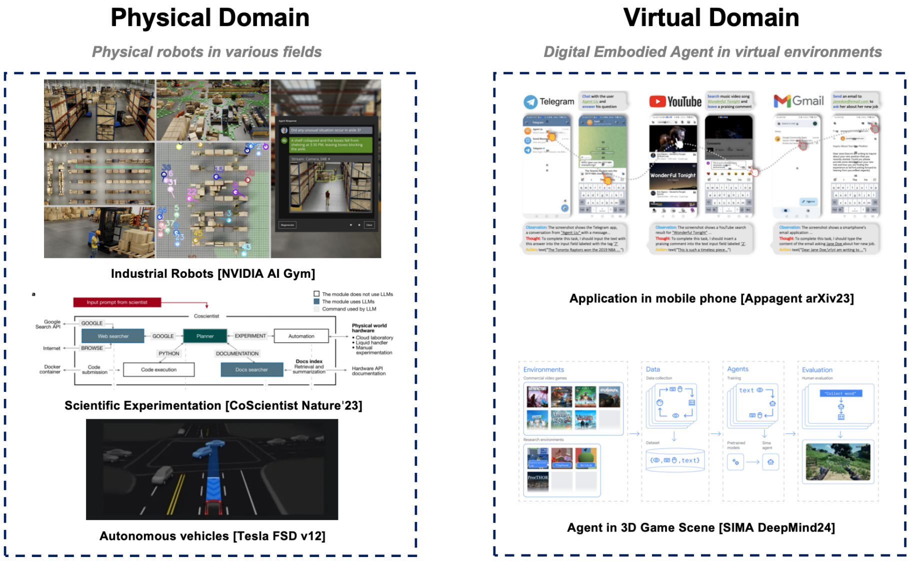

Figure 2: Applications in Physical Domain and Virtual Domain

7.1 Industrial Applications

NVIDIA has developed a platform called AI Gym for industrial robots, facilitating their training and evaluation in complex industrial environments[1]. This platform comprises four main components:

- Omniverse: A precise 3D simulation environment for creating digital twin factories. It allows developers to test and optimize solutions before real-world deployment.

[1] Nvidia AI Gym

- Metropolis: Analyzes video and sensor data streams from the simulation environment, enabling AI systems to interpret and respond to visual information, ensuring efficient monitoring of digital factories.
- Isaac: Focuses on the simulation and training of robotics technology. Every robot in the AI Gym is built using Isaac, which provides advanced robotic simulation tools for constructing robots.
- cuOpt: Optimizes complex logistics and scheduling problems through rapid algorithmic resource allocation and path planning, crucial for applications requiring efficient logistics management.

7.2 Scientific Experimentation

A notable application in scientific experimentation is the CoScientist system, presented in a 2023 Nature article. [4] CoScientist uses large models to autonomously design, plan, and execute complex scientific experiments. Its system architecture includes multiple information-exchanging modules:

- Planner: The central module based on GPT-4, interacts with users via natural language and executes code through Python to address complex problems.
- Modules for Specific Tasks: CoScientist performs tasks such as synthesizing routes, organizing literature, controlling cloud lab instructions, precision sampling, completing complex syntheses, and optimizing results.

CoScientist showcases significant capabilities, including autonomous correction of errors in equipment control codes without human intervention, and optimization of reaction processes to improve yields.

7.3 Autonomous Driving

Autonomous driving vehicles are advanced embodiments of intelligent agents, requiring perception, decision-making, and control signal output. The traditional approach uses manually defined rules and modules for perception, decision-making, and planning. However, this method struggles with complex, unforeseen scenarios. Recent advancements favor end-to-end AI models for seamless integration of these functions.

SenseTime [21] and Tesla: Both have transitioned to end-to-end models, abandoning rule-based systems for comprehensive AI solutions that unify perception, decision-making, and planning.

7.4 Digital Humans

In the virtual domain, digital humans simulate human-like interactions and functionalities. Key developments include:

- AppAgent [79]: AppAgent emulates human actions in operating applications by learning through interaction and observation without relying on backend access. It uses vision and interaction tools to navigate applications and perform tasks through exploratory learning and demonstration observation.
- Gato [56]: Gato can play diverse games and operate robotic arms, demonstrating transfer learning capabilities primarily in grid-like games.
- SIMA [54]: SIMA operates in expansive 3D game environments, following natural language instructions to execute over 600 tasks. It combines game image information with player behavior data to control in-game characters through keyboard and mouse inputs.

Embodied intelligent agents have shown remarkable versatility and potential across various fields. From industrial automation and scientific experimentation to autonomous driving and virtual interactions, these agents are redefining the boundaries of technology and its applications. Future research and development will likely continue to expand their capabilities and integration into more complex and dynamic environments.

8 Open problems and future directions

8.1 Sim2Real Environment: linking general and generative artificial intelligence

The challenge of transferring skills and knowledge learned in simulated environments to real-world applications, known as the Sim2Real problem, remains a significant barrier in achieving embodied intelligence. Simulated environments offer controlled, reproducible conditions ideal for training large language model (LLM) agents, allowing them to practice and refine tasks without the unpredictability of the real world. However, these simulations often fail to capture the full complexity and variability of real-world scenarios, leading to a gap in performance when the agents are deployed outside their training environments. Bridging this gap requires developing more sophisticated and realistic simulations, as well as advanced transfer learning techniques that can enable LLM agents to adapt effectively from simulated to real-world contexts.

In this pursuit, integrating generative artificial intelligence can provide a promising direction. Generative models can be used to create diverse and complex training scenarios within simulations, better preparing LLM agents for the range of situations they might encounter in reality. Additionally, combining general AI principles with generative AI can enhance the adaptability and robustness of these agents, allowing them to generalize learned behaviors and knowledge more effectively. Research into developing seamless Sim2Real transfer methods and incorporating generative models into training regimes is crucial for advancing the practical applicability of LLM-empowered embodied intelligence.

8.2 Inspirations from Neuroscience, Behavioral Science, and Social Sciences

Neuroscience offers valuable insights into how biological systems achieve intelligence, which can inform the design of LLM-empowered agents. Understanding the neural mechanisms underlying human cognition, perception, and action can inspire algorithms that mimic these processes, enhancing the agents' ability to learn and adapt. For instance, concepts such as neuroplasticity and the hierarchical organization of the brain can guide the development of adaptive learning frameworks and multi-layered neural networks in AI systems. By aligning AI models more closely with biological principles, we can create more efficient and robust embodiments of intelligence.

Behavioral science and social sciences also provide critical perspectives on how agents should interact with their environment and other agents. Insights into human behavior, decision-making, and social interactions can inform the development of LLM agents that exhibit more natural and effective social behaviors. This includes understanding the role of emotions, motivations, and social dynamics in human interactions, which can be translated into algorithms for more sophisticated and context-aware AI agents. By integrating these interdisciplinary insights, researchers can create LLM agents that are not only technically proficient but also exhibit more human-like social and adaptive behaviors.

8.3 From physical embodiment to social embodiment

While physical embodiment focuses on enabling AI agents to interact with the physical world through sensors and actuators, social embodiment extends this capability to social interactions and relationships. Physical embodiment allows LLM agents to perceive and act within their environment, providing a foundation for tasks such as navigation, manipulation, and sensory integration. However, for truly human-like intelligence, it is equally important for these agents to understand and participate in social contexts, interpreting social cues, and responding appropriately to human emotions and behaviors.

Social embodiment involves equipping LLM agents with the ability to engage in meaningful social interactions, understand social norms, and build relationships. This requires advances in natural language understanding, sentiment analysis, and social reasoning. By developing agents that can navigate both physical and social environments, we can create more versatile and effective AI systems capable of operating in a wide range of real-world scenarios. Future research should focus on integrating physical and social embodiment, enabling LLM agents to transition seamlessly between interacting with objects and people, thus achieving a more comprehensive form of embodied intelligence.

8.4 Embodied intelligence in open city environment

Open city environments present a unique and complex challenge for embodied intelligence due to their dynamic, unpredictable, and multifaceted nature. LLM-empowered agents operating in such environments must be capable of real-time perception, decision-making, and interaction, handling tasks ranging from navigation and transportation to social interaction and service provision. These agents need to integrate multiple sources of information, adapt to changing conditions, and interact seamlessly with humans and other agents in the city.

Achieving embodied intelligence in open city environments requires advancements in several areas, including robust sensory integration, real-time data processing, and adaptive learning algorithms. Additionally, these agents must adhere to ethical and legal standards, ensuring safe and beneficial interactions within the urban context. Future research should focus on developing scalable and flexible AI frameworks that can handle the complexity of city environments, leveraging advancements in LLMs, robotics, and urban informatics. By addressing these challenges, we can create AI systems that enhance the quality of life in urban settings, providing intelligent services and improving the overall efficiency and sustainability of city operations.

9 Conclusion

In this paper, we take the pioneering step to systematically review the recent advances of large language model agents in the research area of embodied intelligence. For the embodied intelligence for which the tasks are various, we first present the basic taxonomy. We then elaborate on those recent advances with large language model agents, following the above taxonomy. We further discuss the remaining open problems and the promising research. We believe the survey paper can help the readers quickly grasp the recent advances and inspire the following research.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[2] Michael Ahn, Debidatta Dwibedi, Chelsea Finn, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Karol Hausman, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, et al. Autort: Embodied foundation models for large scale orchestration of robotic agents. *arXiv preprint arXiv:2401.12963*, 2024.

[3] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19129–19139, 2022.

[4] Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[6] Shaofei Cai, Bowei Zhang, Zihao Wang, Xiaojian Ma, Anji Liu, and Yitao Liang. Groot: Learning to follow instructions by watching gameplay videos. *arXiv preprint arXiv:2310.08235*, 2023.

[7] Nicolas Harvey Chapman, Feras Dayoub, Will Browne, and Chris Lehnert. Enhancing embodied object detection through language-image pre-training and implicit object memory. *arXiv preprint arXiv:2402.03721*, 2024.

[8] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[9] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10850–10869, 2023.

[10] Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, Yang Zhou, Kaizhao Liang, Jintai Chen, Juanwu Lu, Zichong Yang, Kuei-Da Liao, et al. A survey on multimodal large language models for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 958–979, 2024.

[11] Vishnu Sashank Dorbala, Prasoon Goyal, Robinson Piramuthu, Michael Johnston, Dinesh Manocha, and Reza Ghanadhan. S-eqa: Tackling situational queries in embodied question answering. *arXiv preprint arXiv:2405.04732*, 2024.

[12] Vishnu Sashank Dorbala, James F Mullen Jr, and Dinesh Manocha. Can an embodied agent find your "cat-shaped mug"? llm-based zero-shot object navigation. *IEEE Robotics and Automation Letters*, 2023.

[13] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

[14] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35:18343–18362, 2022.

[15] Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, et al. Threedworld: A platform for interactive multi-modal physical simulation. *arXiv preprint arXiv:2007.04954*, 2020.

[16] Chuang Gan, Siyuan Zhou, Jeremy Schwartz, Seth Alter, Abhishek Bhandwaldar, Dan Gutfreund, Daniel LK Yamins, James J DiCarlo, Josh McDermott, Antonio Torralba, et al. The threedworld transport challenge: A visually guided task-and-motion planning benchmark towards physically realistic embodied ai. In *2022 International conference on robotics and automation (ICRA)*, pages 8847–8854. IEEE, 2022.

[17] Chen Gao, Xiaochong Lan, Zhihong Lu, Jinzhu Mao, Jinghua Piao, Huandong Wang, Depeng Jin, and Yong Li. S ˜3: Social-network simulation system with large language model-empowered agents. *arXiv preprint arXiv:2307.14984*, 2023.

[18] Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. *arXiv preprint arXiv:2309.16650*, 2023.

[19] Dongge Han, Trevor McInroe, Adam Jelley, Stefano V Albrecht, Peter Bell, and Amos Storkey. Llm-personalize: Aligning llm planners with human preferences via reinforced self-training for housekeeping robots. *arXiv preprint arXiv:2404.14285*, 2024.

[20] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023.

[21] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862, 2023.

[22] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023.

[23] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pages 9118–9147. PMLR, 2022.

[24] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

[25] Syed Mahmudul Huq, Rytis Maskeliūnas, and Robertas Damaševičius. Dialogue agents for artificial intelligence-based conversational systems for cognitively disabled: A systematic review. *Disability and Rehabilitation: Assistive Technology*, 19(3):1059–1078, 2024.

[26] Md Mofijul Islam, Alexi Gladstone, Riashat Islam, and Tariq Iqbal. Eqa-mx: Embodied question answering using multimodal expression. In *The Twelfth International Conference on Learning Representations*, 2023.

[27] Shi Jinxin, Zhao Jiabao, Wang Yilei, Wu Xingjiao, Li Jiawen, and He Liang. Cgmi: Configurable general multi-agent interaction framework. *arXiv preprint arXiv:2308.12503*, 2023.

[28] Shyam Sundar Kannan, Vishnunandan LN Venkatesh, and Byung-Cheol Min. Smart-llm: Smart multi-agent robot task planning using large language models. *arXiv preprint arXiv:2309.10062*, 2023.

[29] Siddharth Karamcheti, Suraj Nair, Annie S Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. *arXiv preprint arXiv:2302.12766*, 2023.

[30] Blanka Klimova, Marcel Pikhart, Alice Delorme Benites, Caroline Lehr, and Christina Sanchez-Stockhammer. Neural machine translation in foreign language teaching and learning: a systematic review. *Education and Information Technologies*, 28(1):663–682, 2023.

[31] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

[32] Klemen Kotar and Roozbeh Mottaghi. Interactron: Embodied adaptive object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14860–14869, 2022.

[33] Klemen Kotar, Aaron Walsman, and Roozbeh Mottaghi. Entl: Embodied navigation trajectory learner. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10863–10872, 2023.

[34] Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, et al. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. *arXiv preprint arXiv:2108.03272*, 2021.

[35] Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large scale language model society. *arXiv preprint arXiv:2303.17760*, 2023.

[36] Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18061–18070, 2024.

[37] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023.

[38] Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*, 2023.

[39] Lizi Liao, Grace Hui Yang, and Chirag Shah. Proactive conversational agents in the post-chatgpt world. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3452–3455, 2023.

[40] Haonan Luo, Guosheng Lin, Fumin Shen, Xingguo Huang, Yazhou Yao, and Hengtao Shen. Robust-eqa: robust learning for embodied question answering with noisy labels. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[41] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. *arXiv preprint arXiv:2210.07474*, 2022.

[42] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

[43] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16488–16498, 2024.

[44] So Yeon Min, Devendra Singh Chaplot, Pradeep Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov. Film: Following instructions in language with modular methods. *arXiv preprint arXiv:2110.07342*, 2021.

[45] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[46] Adrián Núñez-Marcos, Olatz Perez-de Viñaspre, and Gorka Labaka. A survey on sign language machine translation. *Expert Systems with Applications*, 213:118993, 2023.

[47] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22, 2023.

[48] Xavi Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Ruslan Partsey, Jimmy Yang, Ruta Desai, Alexander William Clegg, Michal Hlavac, Tiffany Min, Theo Gervet, Vladimir Vondrus, Vincent-Pierre Berges, John Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars and robots, 2023.

[49] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8494–8502, 2018.

[50] Xavier Puig, Tianmin Shu, Shuang Li, Zilin Wang, Yuan-Hong Liao, Joshua B Tenenbaum, Sanja Fidler, and Antonio Torralba. Watch-and-help: A challenge for social perception and human-ai collaboration. *arXiv preprint arXiv:2010.09890*, 2020.

[51] Zekun Qi, Runpei Dong, Shaochen Zhang, Haoran Geng, Chunrui Han, Zheng Ge, Li Yi, and Kaisheng Ma. Shapellm: Universal 3d object understanding for embodied interaction. *arXiv preprint arXiv:2402.17766*, 2024.

[52] Zekun Qi, Runpei Dong, Shaochen Zhang, Haoran Geng, Chunrui Han, Zheng Ge, Li Yi, and Kaisheng Ma. Shapellm: Universal 3d object understanding for embodied interaction. *arXiv preprint arXiv:2402.17766*, 2024.

[53] Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 2023.

[54] Maria Abi Raad, Arun Ahuja, Catarina Barros, Frederic Besse, Andrew Bolt, Adrian Bolton, Bethanie Brownfield, Gavin Buttimore, Max Cant, Sarah Chakera, et al. Scaling instructable agents across many simulated worlds. *arXiv preprint arXiv:2404.10179*, 2024.

[55] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18890–18900, 2022.

[56] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[57] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[58] Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D'Arpino, Shyamal Buch, Sanjana Srivastava, Lyne Tchapmi, et al. igibson 1.0: A simulation environment for interactive tasks in large realistic scenes. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7520–7527. IEEE, 2021.

[59] Lingdong Shen, Chunlei Huo, Nuo Xu, Chaowei Han, and Zichen Wang. Learn how to see: Collaborative embodied learning for object detection and camera adjusting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4793–4801, 2024.

[60] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[61] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

[62] Kunal Pratap Singh, Jordi Salvador, Luca Weihs, and Aniruddha Kembhavi. Scene graph contrastive learning for embodied navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10884–10894, 2023.

[63] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3009, 2023.

[64] Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Elliott Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, Karen Liu, et al. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In *Conference on robot learning*, pages 477–490. PMLR, 2022.

[65] Andrew Szot, Max Schwarzer, Harsh Agrawal, Bogdan Mazoure, Rin Metcalf, Walter Talbott, Natalie Mackraz, R Devon Hjelm, and Alexander T Toshev. Large language models as generalizable policies for embodied tasks. In *The Twelfth International Conference on Learning Representations*, 2023.

[66] Sinan Tan, Mengmeng Ge, Di Guo, Huaping Liu, and Fuchun Sun. Knowledge-based embodied question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):11948–11960, 2023.

[67] Weihao Tan, Ziluo Ding, Wentao Zhang, Boyu Li, Bohan Zhou, Junpeng Yue, Haochong Xia, Jiechuan Jiang, Longtao Zheng, Xinrun Xu, et al. Towards general computer control: A multimodal agent for red dead redemption ii as a case study. *arXiv preprint arXiv:2403.03186*, 2024.

[68] Ruoyu Wang, Zhipeng Yang, Zinan Zhao, Xinyan Tong, Zhi Hong, and Kun Qian. Llm-based robot task planning with exceptional handling for general purpose service robots. *arXiv preprint arXiv:2405.15646*, 2024.

[69] Justin Wasserman, Karmesh Yadav, Girish Chowdhary, Abhinav Gupta, and Unnat Jain. Last-mile embodied visual navigation. In *Conference on Robot Learning*, pages 666–678. PMLR, 2023.

[70] Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas Funkhouser. Tidybot: Personalized robot assistance with large language models. *Autonomous Robots*, 47(8):1087–1102, 2023.

[71] Zhenyu Wu, Ziwei Wang, Xiuwei Xu, Jiwen Lu, and Haibin Yan. Embodied task planning with large language models. *arXiv preprint arXiv:2307.01848*, 2023.

[72] Zhen Xing, Qijun Feng, Haoran Chen, Qi Dai, Han Hu, Hang Xu, Zuxuan Wu, and Yu-Gang Jiang. A survey on video diffusion models. *arXiv preprint arXiv:2310.10647*, 2023.

[73] Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. Exploring large language models for communication games: An empirical study on werewolf. *arXiv preprint arXiv:2309.04658*, 2023.

[74] Zhixuan Xu, Chongkai Gao, Zixuan Liu, Gang Yang, Chenrui Tie, Haozhuo Zheng, Haoyu Zhou, Weikun Peng, Debang Wang, Tianyi Chen, et al. Manifoundation model for general-purpose robotic manipulation of contact synthesis with arbitrary objects and robots. *arXiv preprint arXiv:2405.06964*, 2024.

[75] Zhiyuan Xu, Kun Wu, Junjie Wen, Jinming Li, Ning Liu, Zhengping Che, and Jian Tang. A survey on robotics with foundation models: toward embodied ai. *arXiv preprint arXiv:2402.02385*, 2024.

[76] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

[77] Sherry Yang, Jacob Walker, Jack Parker-Holder, Yilun Du, Jake Bruce, Andre Barreto, Pieter Abbeel, and Dale Schuurmans. Video as the new language for real-world decision making. *arXiv preprint arXiv:2402.17139*, 2024.

[78] Zeyuan Yang, Jiageng Liu, Peihao Chen, Anoop Cherian, Tim K Marks, Jonathan Le Roux, and Chuang Gan. Rila: Reflective and imaginative language agent for zero-shot semantic audio-visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16251–16261, 2024.

[79] Zhao Yang, Jiaxuan Liu, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. Appagent: Multimodal agents as smartphone users. *arXiv preprint arXiv:2312.13771*, 2023.

[80] Bangguo Yu, Hamidreza Kasaei, and Ming Cao. Co-navgpt: Multi-robot cooperative visual semantic navigation using large language models. *arXiv preprint arXiv:2310.07937*, 2023.

[81] Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3mvn: Leveraging large language models for visual target navigation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3554–3560. IEEE, 2023.

[82] Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. *arXiv preprint arXiv:2307.02485*, 2023.

[83] Yue Zhang, Ziqiao Ma, Jialu Li, Yanyuan Qiao, Zun Wang, Joyce Chai, Qi Wu, Mohit Bansal, and Parisa Kordjamshidi. Vision-and-language navigation today and tomorrow: A survey in the era of foundation models. *arXiv preprint arXiv:2407.07035*, 2024.

[84] Zheyuan Zhang, Daniel Zhang-Li, Jifan Yu, Linlu Gong, Jinchang Zhou, Zhiyuan Liu, Lei Hou, and Juanzi Li. Simulating classroom education with llm-empowered agents. *arXiv preprint arXiv:2406.19226*, 2024.

[85] Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 7641–7649, 2024.

[86] Kaiwen Zhou, Kaizhi Zheng, Connor Pryor, Yilin Shen, Hongxia Jin, Lise Getoor, and Xin Eric Wang. Esc: Exploration with soft commonsense constraints for zero-shot object navigation. In *International Conference on Machine Learning*, pages 42829–42842. PMLR, 2023.

[87] Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, et al. Ghost in the minecraft: Generally capable agents for open-world enviroments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144*, 2023.

[88] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2911–2921, 2023.