International Journal of Control, Automation, and Systems 22(8) (2024) 2341-2384
http://dx.doi.org/10.1007/s12555-024-0438-7
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Unlocking Robotic Autonomy: A Survey on the Applications of Foundation Models

Dae-Sung Jang , Doo-Hyun Cho , Woo-Cheol Lee , Seung-Keol Ryu , Byeongmin Jeong , Minji Hong , Minjo Jung , Minchae Kim , Minjoon Lee , SeungJae Lee , and Han-Lim Choi*

Abstract: The advancement of foundation models, such as large language models (LLMs), vision-language models (VLMs), diffusion models, and robotics foundation models (RFMs), has become a new paradigm in robotics by offering innovative approaches to the long-standing challenge of building robot autonomy. These models enable the development of robotic agents that can independently understand and reason about semantic contexts, plan actions, physically interact with surroundings, and adapt to new environments and untrained tasks. This paper presents a comprehensive and systematic survey of recent advancements in applying foundation models to robot perception, planning, and control. It introduces the key concepts and terminology associated with foundation models, providing a clear understanding for researchers in robotics and control engineering. The relevant studies are categorized based on how foundation models are utilized in various elements of robotic autonomy, focusing on 1) perception and situational awareness: object detection and classification, semantic understanding, mapping, and navigation; 2) decision making and task planning: mission understanding, task decomposition and coordination, planning with symbolic and learning-based approaches, plan validation and correction, and LLM-robot interaction; 3) motion planning and control: motion planning, control command and reward generation, and trajectory generation and optimization with diffusion models. Furthermore, the survey covers essential environmental setups, including real-world and simulation datasets and platforms used in training and validating these models. It concludes with a discussion on current challenges such as robustness, explainability, data scarcity, and real-time performance, and highlights promising future directions, including retrieval augmented generation, on-device foundation models, and explainability. This survey aims to systematically summarize the latest research trends in applying foundation models to robotics, bridging the gap between the state-of-the-art in artificial intelligence and robotics. By sharing knowledge and resources, this survey is expected to foster the introduction of a new research paradigm for building generalized and autonomous robots.

Keywords: Decision making, foundation models, large language models (LLMs), motion planning, perception, robotic autonomy, task planning, vision-language models (VLMs).

1. INTRODUCTION

1.1. A paradigm shift in robotic autonomy

The pursuit of increased robot autonomy and minimized human intervention through intelligent agents has been a long-standing aspiration in the field of robotics. While automated robots offer advantages such as high productivity and reduced costs, their limitations in handling complex situations and uncertainties, along with the need for expert modeling and interpretation, have restricted their scalable and versatile integration. Traditionally, achieving safe and sustained robot operation has heavily relied on human intervention and decision-making. The ability of robots to autonomously perceive and understand complex contexts, similar to human cognition, and dynamically plan and execute tasks under di-

Manuscript received May 28, 2024; accepted June 7, 2024. Recommended by Editor-in-Chief Hyo-Sung Ahn. This research was partly supported by Unmanned Vehicles Core Technology Research and Development Program through the National Research Foundation of Korea (NRF) (2020M3C1C1A0108237512), and by Gyonggi-do Regional Research Centre (GRRC) funded by Gyonggi Province (GRRC Aerospace 2023-B01).

Dae-Sung Jang is with the Department of Aeronautical and Astronautical Engineering, Korea Aerospace University, 76 Hanggongdaehak-ro, Deogyang-gu, Goyang-si, Gyeonggi-do 10540, Korea (e-mail: dsjang@kau.ac.kr). Doo-Hyun Cho and Seungjae Lee are with the AI Team, D.Notitia Inc, 1 Gangnam-daero 51-gil, Seocho-gu, Seoul 06628, Korea (e-mails: {dhcho, seungjae.lee}@dnotitia.com). Woo-Cheol Lee is with the Extreme Robotics Team, Korea Atomic Energy Research Institute, 111 Daedeok-daero 989 beon-gil, Yuseong-gu, Daejeon 34057, Korea (e-mail: wclee@kaeri.re.kr). Seung-Keol Ryu, Byeongmin Jeong, Minji Hong, Minjo Jung, Minchae Kim, and Han-Lim Choi are with the Department of Aerospace Engineering, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Korea (e-mails: {skryu, bmjeong, mjhong, mjjung, mckim, hanlimc}@lics.kaist.ac.kr). Minjoon Lee is with the Space Systems Team, Defense Agency for Technology and Quality, 70 Saneopdanji-ro, Daedeok-gu, Daejeon, Korea (e-mail: mjlee@dtaq.re.kr).
* Corresponding author.

verse conditions has remained a distant goal.

Traditional approaches in robot planning and control, such as PID control, model predictive control [1,2], optimal control techniques [3,4], planning domain definition language (PDDL) [5], and rapidly-exploring random tree (RRT) [6,7], have long relied on proper modeling of the robot systems and environment, adjusted for the specific tasks at hand. These methods often require significant expertise in robot kinematics, dynamics, and control theory, along with a deep understanding of the specific robot platform and its environment. Consequently, these approaches can be time-consuming and expensive to implement, and their reliance on accurate models limits their ability to generalize to new tasks or environments.

To address these limitations and cope with uncertainties inherent in real-world scenarios, learning-based methods have emerged as promising alternatives. Research in this area had focused on supervised learning and reinforcement learning techniques for robot planning and control [8-12]. However, these methods often require large amounts of task-specific training data, limiting their ability to generalize to new tasks or robots and necessitating the continuous collection and labeling of new datasets for each new scenario. Therefore, while offering improvements in adaptability compared to classical methods, the challenge of limited transferability remained largely unresolved.

The emergence of foundation models, such as large language models (LLMs) [13-16], has led to a rapid transformation in the robotics landscape, challenging conventional knowledge and paradigms in robotics and control engineering. Foundation models, trained on massive datasets, enable the generalization of knowledge, thereby overcoming the limitations of traditional supervised learning methods. The comprehensive and sophisticated embedded representations learned from the extensive datasets empower these models to understand and reason about new situations, thereby facilitating zero-shot generalization to novel tasks without the need for further task-specific training.

LLMs exhibit remarkable capabilities in natural language understanding, reasoning, generation, and context comprehension, with their development advancing at an accelerated pace. This has unlocked new avenues for seamless communication between humans and machines through natural language, enabling the utilization of LLMs as intelligent agents capable of analyzing situations, planning tasks, and executing actions based on human instructions. Recent research [17-21] exploring the integration of LLMs into robot task planning demonstrate their potential for enabling agents to independently understand natural language input, set goals, generate and decompose tasks, and determine appropriate action sequences. Furthermore, LLMs exhibit remarkable potential for self-correction [22-29], incorporating user feedback or refining their own outputs to enhance performance.

Additionally, vision-language models (VLMs) [30-32] that associate visual data with natural language texts have enabled robots to achieve higher-level situational awareness through semantic understanding, reasoning, mapping, and navigation, using visual information from sensors. Multimodal foundation models trained on diverse data, including image-text or image-text-voice with action [33,34], image-text-audio for 3D mapping [35,36], and robot-specific data [17,37,39-41], allows for the integration of perception, reasoning, planning, and actuation, leading to comprehensive robot autonomy. The advancements in foundation models and the realization of intelligent autonomous agents hold significant promise for fulfilling the long-held ambition of achieving highly autonomous robots within the robotics field. This prospect has fueled an explosion of research in recent years, paving the way for a transformative shift in robotic capabilities and applications.

1.2. Review of existing surveys

The utilization of foundation models to construct intelligent agents and their applications in robotics has spurred extensive research from various perspectives. Among the foundation models, LLMs play a pivotal role in implementing reasoning functions, leading to various surveys on LLM-based intelligent autonomous agents. Research on LLM-based agents has primarily focused on operations within web-based or simulated environments, but extensive explorations into the use of LLMs for reasoning and task planning have been conducted. These explorations facilitate the transfer of such techniques to robotic decision-making and task planning.

Cheng *et al*. [42] have analyzed the definitions and components of LLM-based agents in single and multi-agent systems, categorizing existing studies based on functionalities such as planning, memory, rethinking, and interaction, as well as their operational environments. These agents have also been applied in fields such as mathematics, natural sciences, social sciences, and in practical tasks including administrative assistance and policy-making. Huang *et al*. [43] classified research on LLM-based agent planning into five distinct categories: task decomposition, multi-plan selection, memory-augmented planning, reflection, and external planner-aided planning. Li *et al*. [44] analyzed goal-oriented prompt design in LLM prompt engineering, which directs the model to think and respond in a human-like manner. These techniques are further divided into task decomposition, action selection/implementation, evaluation of sub-goal results, and the selection of valuable sub-goals. Moreover, Yang *et al*. [45] focused on enhancing the capabilities of intelligent agents by utilizing LLM's code generation and analysis features, presenting methods to leverage the consistency, executability, and modularity of codes

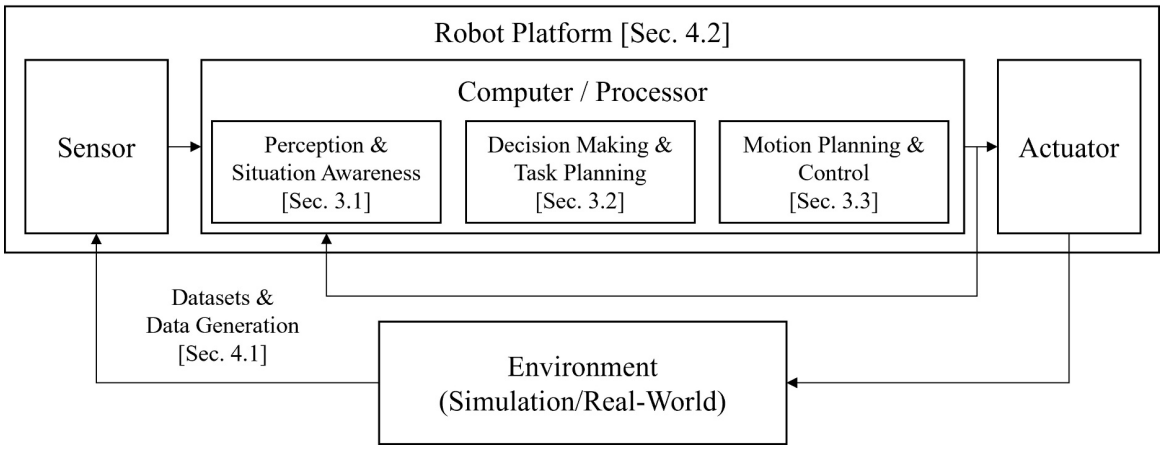


Fig. 1. A generic architecture of robotic system operations and autonomy.

to enhance LLM's reasoning, connectivity with external tools, automation, and self-improvement.

Other surveys of robotics applications [46-49] have also highlighted the various foundation models necessary for interacting with environment. Kawaharazuka *et al*. [46] categorized and analyzed foundation models used in robotics from modality and input-output perspectives. They classified the application cases in robot perception and planning into low-level and high-level layers and also summarized studies utilizing foundation models for data augmentation. Firoozi *et al*. [47] reviewed the use of foundation models in robot perception, decision-making, planning, and control, while Wang *et al*. [48] categorized related research into mission planning, manipulation, and reasoning. Hu *et al*. [49] conducted the most systematic analysis of foundation models used in robotics. They examined functionalities required for robot autonomy such as perception, decision-making, planning, and action generation and also addressed the associated challenges in robotics. They introduced how various modality foundation models are utilized in these functions, and discussed robotic foundation models (RFM) trained on robot data and prompting techniques.

1.3. Contribution of this work

This paper presents a comprehensive and systematic analysis of the existing literature on applying foundation models in robotics. This survey aims to bridge the knowledge gap and facilitate the adoption of this new paradigm by researchers in robotics and control engineering. While several existing surveys share a similar goal, they often cater to experts in artificial intelligence and machine learning and lack specific and in-depth guidance for direct application in robotics. This paper includes a preliminary section that elucidates the fundamental concepts, definitions, characteristics, and basic usage techniques of foundation models.

It then categorizes relevant research based on how foundation models address autonomy issues related to each component of a robotic system. Broadly classifying robotic components into sensors, onboard computers, and actuators, foundation models can revolutionize existing functionalities and enable unprecedented autonomous capabilities in onboard component interactions (Fig. 1). Therefore, this paper classifies research on foundation models in robotics into three widely recognized and familiar elements of robotic autonomy: 1) perception and situational awareness, 2) decision making and task planning, and 3) motion planning and control.

The sub-classification within each element is designed to cover aspects previously overlooked or insufficiently addressed in existing surveys. This approach aims to provide valuable research indicators for researchers tackling specific issues in robotics. In the domains of perception and situational awareness, this paper analyzes research cases focusing on object detection, classification, semantic understanding, and notably on mapping and navigation—key areas of interest for robotics researchers. Relevant techniques in decision making and task planning are further divided into four key categories: mission understanding, planning, validation/correction, and LLM-robot interaction. Mission understanding entails comprehending missions, setting goals, and identifying feasible skills, while planning is centered on structuring task sequences. This paper also investigates research on how LLMs self-evaluate and adjust their generated plans, as well as techniques for LLM interaction with robots. Studies related to motion planning encompass not only the utilization of LLMs and VLMs but also the examples of diffusion model applications, a topic seldom explored in prior surveys.

Furthermore, this paper compiles elements of the research environment relevant to foundation models applied

in robotics to provide practical and useful information. This analysis offers the most detailed and extensive survey compared to currently available studies. Recognizing the need for extensive data in training and performance validation, the paper offers a focused and comprehensive survey on datasets and data generation research employed in robotics. It also provides an informative overview of various simulation environments and real robot systems referred in the literature.

Finally, this paper addresses the challenges and future prospects in this field, offering perspectives not previously presented in existing studies. This includes addressing issues like robustness, safety, reliability, data scarcity, and real-time performance. Moreover, the paper outlines directions on topics of significant interest to robotics researchers, such as on-device foundation models necessary for implementing mobile robot systems and retrieval augmented generation for enhancing system robustness and consistency.

The organization of this paper is as follows: Section 2 provides a preliminary, including the fundamental concepts, characteristics, and utilization techniques of foundation models. Section 3 classifies and analyzes how various foundation models are applied to robotic autonomy. The research covered is broadly divided into three categories: perception and situational awareness, decision-making and task planning, and motion planning and control. Section 4 presents a survey of the datasets necessary for training and validating foundation models, the methods for augmenting such datasets, and the simulation environments and robot platforms used in related research. Section 5 discusses the various challenges associated with implementing foundation models in robotics and explores potential future research directions. The paper concludes with Section 6.

2. PRELIMINARIES

This section introduces the essential background knowledge for understanding the integration of foundation models into robotics applications. We will first explore key concepts related to foundation models and give an overview of prominent foundation models relevant to robotics.

2.1. Key concepts and terminology

To facilitate comprehension for researchers in robotics and control engineering, we provide clear explanations of key terms and concepts related to foundation models and their utilization.

1) **Embedding:** Embedding refers to the process and outcome of representing data, such as words or images, as real-valued vectors in a high-dimensional space. This representation enables the computation of semantic relationships and similarities between different data within the vector space, allowing models to learn complex patterns and relationships, which is essential for tasks like language understanding, image recognition, and robotic control. In LLMs, text is typically embedded at the word level, often using techniques like word2vec [50] or GloVe [51] to transform words into vectors. VLMs, on the other hand, utilize methods like contrastive learning to embed both images and text into a joint multimodal embedding space.
2) **Tokenization:** Tokenization is the process of breaking down raw data, such as text or images, into smaller units called tokens. The unit of a token can vary depending on the type of data and the tokenization method employed in the model. For text, semantic units like words or subwords are often chosen. Both LLMs and VLMs tokenize raw data into a sequence of tokens before the embedding process.
3) **Grounding:** Grounding refers to the process of associating a model's data to concrete representations in the real world. In the context of robotics, grounding links the inputs and outputs of LLMs and VLMs, i.e., natural language text and image, to the robot's physical actions and environmental situation. This connection is crucial for a robot to perceive objects in its environment, understand their states and relationships, and generate affordable actions.
4) **Zero-shot and few-shot learning:** Zero-shot learning describes a model's ability to perform tasks involving unseen data, not included in its training data, by leveraging its learned generalization capabilities. This is achieved through the model's ability to generalize knowledge and rules acquired during pre-training on vast and diverse datasets. Few-shot learning, on the other hand, involves presenting a model with a small number of relevant examples to facilitate its performance on a new task. Given that robots may encounter situations and tasks outside the scope of their pre-trained data, the ability for zero-shot and few-shot learning is highly valuable. In LLMs, few-shot learning can be implemented by providing examples within the input text, a process known as prompting.
5) **In-context learning:** In-context learning is a prompting technique that provides demonstration within the input prompt to guide LLMs in understanding the desired approach or rules for performing a new task. This technique allows the model to achieve generalization for the new task by incorporating a small amount of data within the prompt, without modifying the model itself, which would require further training.
6) **Pre-training:** Pre-training refers to the process of training a foundational model on a large dataset.

Models pre-trained in this manner can be applied to a variety of tasks to solve, known as downstream tasks. For example, downstream tasks applicable to a pre-trained LLM that has been trained on internet-collected text may include understanding the context of a document, generating text, and translating. Due to the need to train large foundational models that exhibit generalization capabilities, pre-training requires substantial amounts of data, thus incurring significant time and cost expenditures.

7) **Fine-tuning:** Fine-tuning denotes the process of further training a pre-trained model on a relatively small dataset tailored to specific downstream tasks. Throughout the fine-tuning, adjustments are made to the model's parameters to optimize performance on specific downstream tasks. While re-training a foundational model from scratch can be costly, fine-tuning offers a cost-effective way to adapt the model for desired tasks and is widely employed. However, fine-tuning may lead to overfitting on specific tasks' datasets and can result in a degradation of performance in other tasks, a phenomenon known as catastrophic forgetting.

8) **Prompting:** Prompting involves designing text input prompts to enhance the quality of outputs generated by LLMs or VLMs. Various techniques have been proposed to enable language models to better understand tasks and produce effective results. This includes specifying detailed contexts, examples, and reasoning methods within the prompt. A notable technique is the chain-of-thought [52], where the model is guided through a step-by-step reasoning process that leads to the final conclusion. This enables the model to tackle complex tasks by breaking them down into smaller, manageable sub-goals. Other prompting techniques include few-shot and tree-of-thought prompting [53].

2.2. Foundation models for robotics

The advent of foundation models, particularly LLMs and VLMs, has presented a groundbreaking solution for achieving general-purpose robots and enhanced autonomy. These foundation models learn from massive amounts of data, thereby constructing vast knowledge and generalizations through the embedded relationships between the data. In this process, techniques such as self-supervised learning are commonly applied, as the training of foundation models often involves much more unlabeled or poorly labeled data. The capability for generalization of knowledge enables the effective application of foundation models to a variety of untrained tasks with minimal or no need for additional task-specific training.

2.2.1 Large language models (LLMs)

Recent major LLMs such as GPT-3 [13], Llama2 [14], LaMDA [15], and PaLM2 [16] have been developed based on the generative pre-trained transformer (GPT) architecture [54]. Whereas encoder-only models like BERT [55] are designed to understand the contextual relationships between sentences, GPT is a decoder-only model trained to predict the most suitable subsequent word in text sequences, thereby rendering it highly effective for generating new text. As the various issues with decoder-only models have been overcome, GPT-based LLMs have become the current state-of-the-art.

GPT models are trained through a self-supervised learning method that involves predicting the next word using text datasets. They utilize the self-attention mechanism [56] to learn the relationships among words within the input sequence. GPT is pre-trained on a web-scale corpus, and recent LLMs have billions of parameters. The pre-trained model is then fine-tuned for specific natural language processing tasks such as question answering or text generation. This process involves additional training on task-specific datasets to optimize performance on the tasks. GPT-3.5 [57] has been further trained through reinforcement learning by human users to prevent inappropriate or unethical responses and produce more human-like outputs.

These trained LLMs can understand natural language context, generate text, and even mimic logical and creative intellectual human activities that are based on language, such as inference, analysis, prediction, and planning. Consequently, they are employed in various applications, including question answering, translation, document generation, task assistance, intelligent agents, code generation, and problem-solving. Their application extends across diverse fields such as healthcare, law, finance, and education [42], demonstrating their vast potential. In robotics, these models are utilized for a variety of tasks including commonsense reasoning [58-63], selecting and assessing affordable actions [17,18,64,65], decomposing and planning tasks [20,21,66,67], generating robotic control commands [17,68] and reward functions [69-71], and validating and correcting generated plans [24-29].

2.2.2 Vision-language models (VLMs)

VLMs are foundational models that learn to associate text with images. Notable examples include contrastive learning-based models such as CLIP [30], ALIGN [31], and BLIP [32]. The most widely used model, CLIP, has been developed with a training dataset of 0.4 billion image-text pairs sourced from the internet. CLIP utilizes encoders to extract features from both image and text inputs. For image features, it employs a transformer encoder structure similar to ViT [72], segmenting the image into fixed-size patches that are then linearized into vectors.

The features extracted by the encoders are projected into an $n \times n$ joint latent space where they share the same dimensions. In the joint latent space, the model is trained to minimize a contrastive loss function designed to increase the similarity of corresponding image-text pairs, simultaneously lowering the similarity of incorrect pairs.

By learning images and texts together, CLIP can infer semantic information about untrained images, thereby excelling in tasks such as zero-shot or open-vocabulary image classification, image caption generation, and image-based question answering. Open vocabulary refers to the ability to recognize and categorize objects beyond the fixed set of categories specifically trained on, allowing for more flexibility and adaptability in dynamic environments. In robotics, VLMs like CLIP are used for situational awareness of robots, primarily for object classification, semantic understanding, scene representation, and mapping.

The open-vocabulary capability of VLMs is also beneficial in object detection. Object detection involves identifying the locations and types of objects in an image, yet the number of datasets for detection is significantly smaller compared to classification training data. Since VLMs can perform open-vocabulary classification on images, they enable the detection and classification of new objects not present in the detection training data. VLMs used for this purpose include GroundingDINO [73], vision transformer for open-world localization (OWL-ViT) [74], vision and language knowledge distillation (ViLD) [75], detector with image classes (Detic) [76], and grounded language-image pre-training (GLIP) [77]. GroundingDINO and OWL-ViT modify the contrastive learning architecture similar to CLIP to address object detection, while Detic utilizes the embeddings from CLIP with object class labels to train models on general images without specific bounding boxes. ViLD employs a knowledge distillation approach where a student model learns to align region embeddings of detected bounding boxes with the text-image embeddings of a teacher model for open-vocabulary image classification, like CLIP. GLIP utilizes the additional semantic information in text about images to combine phrase grounding with object detection, self-training grounding boxes for detected objects.

2.2.3 Diffusion models

Diffusion models [78-82] are among the generative models that produce high-quality samples. These models generate samples by reversing a process analogous to actual physical diffusion. The process consists of two opposing phases: the forward diffusion (noise addition) and the reverse diffusion (noise removal). During the forward diffusion process, predefined noise (typically Gaussian noise) is progressively added to data samples such as images, transforming them into random noise. The reverse diffusion process, where actual learning occurs, is more complex. In this phase, the model is trained to iteratively denoise the data, restoring it back to its original sample. Unlike the forward diffusion process, this reverse process is stochastic, involving random variations as the model's parameters are learned to restore the noisy data. The training of the model involves predicting the amount of noise added at each stage of the forward process and reversing this process. The goal is to ensure that the model's output closely matches the original noise-free data.

Diffusion models have proven notably successful in tasks such as realistic image generation and damaged image restoration [82]. These models can handle complex data patterns without estimating detailed probability distributions. The ability of diffusion models to systematically transform random noise into structured and valuable data through a methodical learning process distinguishes them from other generative models. They use a controlled reverse process to transform a high-entropy initial state into an organized, desired state. In robotics, diffusion models are primarily utilized for motion planning. Leveraging their capability to generate sample data, they are used to sample trajectories in reinforcement learning for trajectory optimization [83], and to sample from the posterior trajectory distribution of manipulators, which is conditioned on specific task goals [84].

2.2.4 Robotic foundation models (RFMs)

Unlike LLMs or VLMs, which are primarily trained on vast amounts of data collected from the internet, RFMs [17,37,39-41] are foundational models that learn from data gathered within robot operating environments. Leveraging the generalization capabilities of LLMs or VLMs often presents challenges in grounding when applied to actual robots. Consequently, research has been conducted to construct models better suited to real robotic missions by training on raw data acquired from robot sensors and control commands applicable to robot actuators.

PaLM-E [37] has been trained using a pre-trained LLM, called PaLM [85]. It integrates user's text commands and images from a manipulator robot's camera sensor, into PaLM's language embedding space and outputs text for task and motion planning. RoboVQA [39] has been trained to generate text outputs for situational awareness, mission planning, and future predictions, using video-text training data collected from both robots and humans.

Known as the robotics transformer, RT-1 [40] has been trained on an extensive dataset of long-term collected robotic data using a transformer-based policy model. This model receives images and text commands as inputs and outputs discrete robot actions. In RT-2 [17], a pre-trained VLM has been fine-tuned with robot data and original training datasets, capable of outputting specific action commands such as the position and angles of the end-effector, enabling low-level closed-loop control. RoboCat [41], a model based on Gato [86], is trained on a

Table 1. Summary table of foundation models for robotics.

Model type	Key characteristics	Robotics applications
LLMs	Trained on massive text datasets, excels in natural language understanding, generation, and reasoning.	Mission understanding, task decomposition, plan generation, action selection, reward function design, plan validation, and human-robot interaction.
VLMs	Trained on image-text pairs, capable of associating visual data with natural language.	Object detection and classification, semantic understanding, scene representation, mapping, and navigation.
RFMs	Trained on robot-specific data, such as sensor readings and control commands.	Grounding LLM/VLM outputs to robot actions, enabling more robust and adaptable robot behavior.
Diffusion Models	Generative models that learn to denoise data by progressively adding and removing noise.	Sampling trajectories for motion planning and control, enabling robots to adapt to new tasks and environments.
LMMs	Combine multiple modalities (e.g., text, images, audio, robot actions).	Comprehensive robot autonomy, encompassing perception, planning, and action execution.

large dataset composed of images and action sequences of robotic arms. Gato is a multimodal model that processes vision-language-action in both simulated and physical environments, enabling RoboCat to perform new tasks in various manipulators with relatively few demonstrations.

2.2.5 Large multimodal models (LMMs)

Foundation models that integrate multiple modalities, such as text, images, videos, audio, robotic actions, thermal imagery, and other sensory data, are referred to as large multimodal models (LMMs), such as GPT-4 [87], GPT-4o [88], Gemini [89]. VLM, which combines images and natural language, and RFM, which is trained using various robotic data, are prime examples of LMMs. Additionally, COMPASS [90] has proposed a contrastive learning architecture for various spatiotemporal multimodal signals. For robotic systems, several models have been designed to receive multimodal inputs and produce robotic actions: VIMA [33] takes images and text as inputs and MUTEX [34] processes images, videos, text, and voice. Furthermore, LMMs have been developed for 3D mapping and navigation. AVLMaps [35] associates text, depth, pose, image, and audio for 3D mapping. ConceptFusion [36] integrates image, audio, and text features into a 3D map. ULIP-2 [38] performs classification and captioning, similar to VLM, but on 3D input data.

3. APPLICATIONS IN ROBOTICS

This section examines the diverse applications of foundation models and diffusion models in robotics, focusing on three main areas: perception and situational awareness, decision-making and task planning, and motion planning and control. Leveraging these advanced techniques, robots can better understand their environment, make smarter decisions, and navigate complex situations more autonomously and efficiently. We provide specific examples in each area, demonstrating how these approaches can significantly enhance robotic systems.

3.1. Perception and situational awareness

This part explores the advancements in robotics that enhance a robot's capability to perceive, understand, and navigate complex environments. It focuses on three key areas: object detection and classification, where robots use open-vocabulary technologies to identify a diverse array of objects; recognition, mapping, and navigation, which involve the integration of semantic information to create detailed scene representations and enhance navigation capabilities; and perception-focused embodied models in robotics, emphasizing how foundation models enhance sensory processing for decision-making in dynamic interactions with the environment. They are tightly linked together, generating valuable information for decision-making and planning (Fig. 2). Each section collectively illustrates how cutting-edge technological integrations are pushing the boundaries of robotic capabilities, enabling them to perform with greater adaptability and accuracy.

3.1.1 Object detection and classification

Recently, significant strides in object detection and classification have been made through the integration of vision and language foundation models, enhancing robots' abilities to interact with their surroundings. This development allows robots to recognize and categorize objects in diverse environments, beyond traditional fixed categories. We explore the application of these advanced models in robotic perception, examining their impact from multiple perspectives.

3.1.1.1 Open-vocabulary object detection

Object detection is a vital perceptual capability for robots, allowing them to locate and recognize objects in their surroundings. This skill is essential for many robotic tasks, including grasping, manipulation, navigation, and interaction with humans. Traditional object detectors, which rely on predefined categories and labeled datasets, often fail to adapt to the varied and dynamic real world. Recent advancements have introduced open-vocabulary object detection, using LLMs and VLMs to

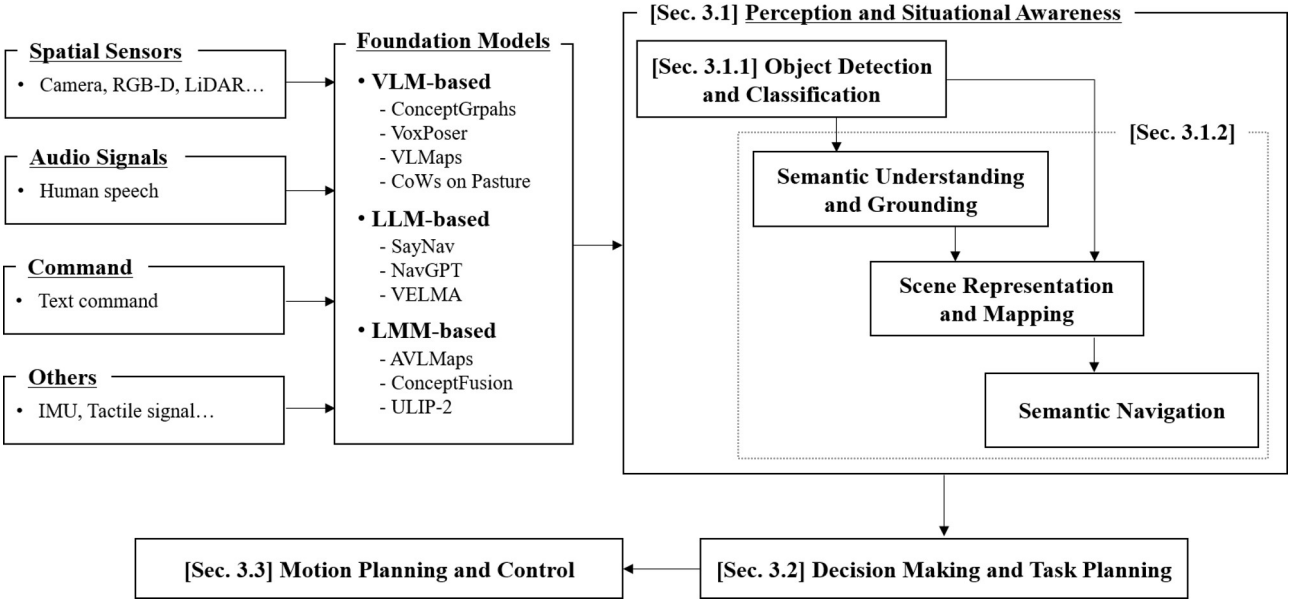


Fig. 2. This flowchart illustrates the perception process of a robot. It begins with various sensory inputs, such as spatial data, audio, and commands, which are processed using foundation models like VLM, LLM, and LMM. This information is then used to build an understanding of the environment. This process starts with object detection and classification, followed by semantic understanding and grounding. This then leads to scene representation and mapping, culminating in semantic navigation, ultimately contributing to the robot's perception and situational awareness. Finally, the robot uses this gathered knowledge for decision making, planning, and control.

significantly broaden the range of objects that robots can identify.

These recent works highlight the growing trend of using VLMs for more nuanced and sophisticated robotic manipulation tasks. While ConceptGraphs [91] uses an LLM for scene graph generation and leverages GroundingDINO [73] for object detection in its CG-D variant, VLFM [92] employs GroundingDINO for object detection to aid in zero-shot object navigation. Similarly, Yang *et al*. [93] utilize GroundingDINO and SAM [94] for object detection and segmentation to create language-reasoning masks for robot manipulation. MOKA [95] takes this a step further, employing GroundedSAM, [96] which combines GroundingDINO and SAM, for open-vocabulary object segmentation and keypoint proposal for point-based motion generation.

Several of the papers leverage OWL-ViT [74] VLM for open-vocabulary object detection as a key component in their robotic systems. OK-Robot [97] and CoWs on Pasture [98] both employ OWL-ViT for detecting objects in novel environments. OK-Robot incorporates OWL-ViT detection output into a voxel map representation for navigation and manipulation, while CoWs on Pasture uses OWL-ViT to identify the target object during exploration, guiding the robot towards it. Similarly, MOO [99] and VoxPoser [100] utilize OWL-ViT to extract spatial information from language instructions. MOO uses OWL-ViT to ground object descriptions in images, providing object locations as input to a learned manipulation policy, while VoxPoser leverages OWL-ViT to identify the objects mentioned in instructions, using this information to generate 3D value maps for planning. Grounded decoding [101] takes a slightly different approach, utilizing OWL-ViT's object detection capabilities as a grounding function to score the likelihood of language model-generated tokens for planning actions in a real-world kitchen setting. Notably, PHYSOBJECTS [102] does not directly utilize OWL-ViT for physical reasoning. Instead, it focuses on fine-tuning a VLM using a novel dataset of physical object annotations. OWL-ViT is used solely for initial object detection in their planning evaluation, but its role is limited to scene understanding rather than directly contributing to physical reasoning.

CLIP-Fields [103] leverages Detic [76] as an open-label object detector to provide semantic labels for training their 3D scene representation, enabling CLIP-Fields to learn semantic representations without explicit human annotations. Similarly, HomeRobot [104] utilizes Detic to provide object segmentation masks for both heuristic and reinforcement learning-based agents, enabling the robot to identify and locate objects within the scene, and highlighting that Detic's performance significantly impacts overall task success, thereby emphasizing the importance of robust object detection in real-world robotic applications.

ProgPrompt [20] employs ViLD [75] on a real robot setup to identify and segment objects in the scene. These detected objects are then incorporated into the ProgPrompt framework, allowing the LLM to generate plans grounded in the actual objects present in the robot's environment. Similarly, NLMap [105] leverages ViLD alongside CLIP to extract features for each proposed region of interest (ROI) in the scene during its object query phase. The paper highlights that ViLD is particularly effective at detecting common objects, while CLIP excels at recognizing less common or out-of-distribution objects. By combining these models, NLMap creates a more comprehensive and robust scene representation for its robot planning system.

In ESC [62], GLIP [77] is employed to detect common objects and rooms within the environment in an open-world setting. It takes an image and a text prompt (e.g., "chair, table, picture, cabinet") as input and outputs detected objects, their bounding boxes, and confidence scores. Lastly, Statler [106] utilizes MDETR [107] in its real-world robot experiments for open-vocabulary object segmentation, providing segmentation masks for objects that are used with depth information to obtain object point clouds and estimate their positions for manipulation tasks.

Object detection provides the initial step in understanding the scene, but segmentation takes it a step further, allowing robots to delineate object boundaries and extract more detailed information about their shapes and properties. Powerful segmentation models like SAM [94] and LSeg [108] allow robots to identify and understand objects and scenes with greater precision and flexibility. VoxPoser [100] and OK-Robot [97] employ SAM to refine initial object detections, creating accurate object masks to enhance their 3D understanding of the environment. OK-Robot incorporates these masks into a voxel-based map representation for navigation and manipulation, while VoxPoser utilizes them to generate point clouds for constructing 3D value maps specifically for manipulation tasks. ConceptGraphs [91] also leverages SAM to create class-agnostic instance masks from RGB images, serving as the foundation for identifying and reconstructing 3D objects which are then represented as nodes in a scene graph. Along with SAM, VLMaps [109] employs LSeg to generate semantically rich pixel-level embeddings from RGB images, which are then fused with 3D reconstruction data to create a spatial map enabling natural language queries for robot navigation.

3.1.1.2 FM-based object classification

Classification is essential for object detection, helping robots categorize and understand objects. TidyBot [110] uses CLIP [30] to recognize objects from close-up images taken by its egocentric camera, with an LLM providing target object categories based on user preferences, allowing generalization to novel objects and personalized tidying. Similarly, Khandelwal *et al*. [111] leverage CLIP embeddings for navigation in embodied AI, using CLIP's object classification to guide agents towards target objects and understand spatial relationships for reachability and free space estimation. CDUL [112] utilizes CLIP to generate pseudo-labels for unannotated images, training a classification network and achieving results comparable to weakly supervised methods without manual annotations. Kanazawa *et al*. [113] also use CLIP to recognize various food states (e.g., "boiling water," "melted butter") through natural language prompts, creating a flexible classification system for cooking robots.

Beyond 2D image classification, CLIP has been extended to handle 3D point cloud data, enhancing robotic perception and manipulation in 3D environments. PointCLIP [114] introduces zero-shot 3D point cloud classification by projecting point clouds into multi-view depth maps and using CLIP's visual and textual encoders to align these with textual category prompts, achieving promising results and improving fully-supervised 3D models. ULIP [115] and ULIP-2 [38] advance this by learning a unified representation of images, text, and 3D point clouds through pre-training on object triplets. ULIP aligns the 3D feature space with the image-text space using synthesized triplets, enhancing performance on both zero-shot and standard 3D classification tasks. ULIP-2 scales up this approach, using LMMs to generate holistic language descriptions for 3D shapes, which enables training on larger datasets and achieves state-of-the-art results in 3D classification and promising performance in 3D captioning.

3.1.2 Recognition, mapping, and navigation

The foundation models enable robots to achieve unprecedented levels of semantic understanding, situational awareness, and adaptability by interpreting and connecting human language instructions to the visual scene, building rich scene representations, and navigating based on high-level semantic cues. This part explores cutting-edge research and techniques in semantic understanding and grounding, scene representation and mapping, and semantic navigation, showcasing how foundation models are transforming robot perception and paving the way for more intelligent and versatile robotic systems.

3.1.2.1 Semantic understanding and grounding

Semantic understanding and grounding enable robots to connect human language with the visual world, allowing them to perceive objects and scenes and interpret their meaning and context. This includes recognizing objects, places, and their relationships, as well as understanding the affordances and constraints for task execution in complex environments. Advances in LLMs and VLMs have significantly improved semantic understanding and grounding, enhancing the robustness and efficiency of robotic perception systems.

Open-vocabulary understanding, the ability of robots to reason about novel objects and concepts beyond their training data, is a critical aspect of robust robotic perception. This capability is crucial for real-world deployment, where robots may encounter unfamiliar objects or situations. By moving beyond a closed set of predefined concepts, robots can adapt to new situations and build more versatile perception systems. Research in this area has explored the use of foundation models to create open-set multimodal 3D maps and open-vocabulary 3D scene graphs for perception and planning. Some methods [36,91] show that robots can expand their understanding of the world by integrating information from multiple modalities and representing scenes in a structured, flexible manner, allowing them to adapt to new situations.

Object grounding, the ability to link natural language descriptions to specific objects in a scene, plays a crucial role in enabling robots to understand the properties and functionalities of objects within their environment. This involves not only recognizing objects but also comprehending their affordances and constraints based on natural language instructions. For instance, methods in [92,100] showcase the ability to extract affordances and constraints from natural language instructions and represent them spatially, allowing robots to reason about how to interact with objects and perform manipulation tasks effectively. Additionally, VLMs have been utilized to ground natural language descriptions of unique objects, as seen in LGX [60], enabling robots to recognize and locate objects based on detailed descriptions rather than relying solely on predefined categories. This capability opens up possibilities for more complex and nuanced robot behaviors grounded in a deeper understanding of object properties and functionalities.

3.1.2.2 Scene representation and mapping

Scene representation and mapping play a pivotal role in enabling robots to navigate and interact with their surroundings effectively. Recent research has shifted from traditional geometric maps towards incorporating semantic information, creating richer and more informative representations that facilitate a deeper understanding of the environment. Similar to Subsection 3.1.2.1, this progress is largely driven by the advancements in foundation models which offer the ability to reason about the visual world in ways that were previously impossible.

Semantic mapping has progressed from the pioneering work called SemanticFusion [116], which combined CNN-based segmentation with 3D mapping but had limitations due to fixed vocabularies and large training data requirements. VLMs like CLIP enabled open-vocabulary approaches, leading to methods like VLMaps [109] that integrate pre-trained visual-language features into 3D maps for natural language-based queries and navigation, allowing interaction with unseen objects or locations. FM-Fusion [117] fuses object detections from vision-language foundation models into instance-aware semantic maps, using probabilistic label fusion and instance refinement to address open-set labels and inconsistent segmentation. Multi-scale CLIP features are embedded into 3D maps [118] for real-time object exploration.

Building queryable scene representations enhances the utility of semantic maps by enabling robots to retrieve information about their environment and the objects within it using natural language queries. This makes robot systems more flexible and adaptable, allowing them to effectively interact with their surroundings based on natural language instructions. Two methods that exemplify this approach are NLMap [105] and OpenFusion [119]. NLMap uses VLMs like CLIP and ViLD [75] to create an open-vocabulary scene representation that can be queried using natural language. This allows robots to retrieve information about object locations and availability, aiding in planning and decision-making. Similarly, OpenFusion employs a pre-trained vision-language fusion model (VLFM) and truncated signed distance function (TSDF) for 3D scene reconstruction. This method generates a semantically aware 3D map that can also be queried with an open vocabulary, enhancing the robot's ability to understand and interact with its environment. CLIP-Fields [103] takes a different approach by learning a mapping from spatial locations to semantic embedding vectors, trained with weak supervision from web-image and web-text trained models. This method enables open-vocabulary segmentation, instance identification, semantic search, and view localization.

Several works explore the concept of multimodal mapping, integrating information from different sensory modalities to provide a more comprehensive understanding of the environment. AVLMaps [35] introduces a unified 3D spatial map representation that incorporates audio, visual, and language cues. By fusing features from pre-trained multimodal foundation models into a 3D voxel grid, AVLMaps enables robots to index goals based on multimodal queries, including textual descriptions, images, or audio snippets of landmarks. This approach improves goal disambiguation in complex scenarios, allowing robots to leverage complementary information from different senses. ConceptFusion [36] extends the concept of queryable 3D maps by supporting interactions through text, image, audio, and clicks, enabling robots to perform zero-shot spatial reasoning and manage long-tailed concepts. This approach creates open-set multimodal 3D maps, enabling robots to perform zero-shot spatial reasoning and showcasing the potential of multimodal mapping for real-world robotics tasks.

Object-centric 3D representations offer another compelling approach for scene representation. ConceptGraphs [91] proposes an open-vocabulary graph-structured representation where objects are represented as nodes with

both geometric and semantic features, and relationships between them are encoded as edges in the graph. This structure enables efficient scene representation and facilitates downstream tasks like object search and manipulation. OK-Robot [97] showcases the integration of various Open Knowledge models for robotics tasks, emphasizing the importance of carefully combining these systems with robotic modules to achieve high performance in complex real-world scenarios. AnyLoc [120] tackles the challenge of building robust and universal visual place recognition systems, using foundation models to extract per-pixel features and then aggregating them with VLAD or GeM pooling to achieve state-of-the-art performance across diverse environments without retraining or fine-tuning.

3.1.2.3 Semantic navigation

Semantic navigation enables robots to understand and interact with their environment in a way that mirrors human-like perception and reasoning. Instead of depending on detailed waypoints or low-level navigation commands, semantic navigation allows robots to interpret high-level instructions related to specific objects, rooms, or tasks. This shift supports more intuitive human-robot communication and helps robots form a deeper understanding of their surroundings, thereby improving their situational awareness. Recent progress in foundation models has accelerated this development, allowing robots to process and reason about their environment through complex language cues.

One of the key applications of foundation models in semantic navigation is language-guided exploration. Methods like CoWs on Pasture [98], LGX [60], ESC [62], VLFM [92], and L3MVN [121] demonstrate the use of LLMs and VLMs to reason about potential target object locations based on commonsense knowledge and language priors. These methods utilize visual scene descriptions and object labels to formulate LLM prompts, allowing the models to infer promising areas for exploration based on their understanding of object relationships and spatial reasoning. This capability significantly enhances a robot's perception of its environment by providing semantic context and guiding exploration toward areas relevant to the task at hand. Further enhancing situational awareness, dynamic planning with LLMs allows robots to continuously generate and refine navigation plans based on real-time observations and instructions. SayNav [122], and March in Chat [123] employ LLMs as high-level planners, generating step-by-step instructions and incorporating feedback from the environment to adapt to changing situations and complete complex tasks. The capability of dynamic planning allows robots to continuously update their understanding of the environment and adjust their plans accordingly, contributing to a more robust and flexible form of navigation that is crucial for maintaining situational awareness in dynamic real-world scenarios. NavGPT [124] further explores the reasoning capabilities of GPT models for high-level navigation planning, showcasing their ability to decompose instructions, integrate commonsense knowledge, and handle exceptions, demonstrating the potential of LLMs for complex navigation tasks.

Several works address the challenge of navigating to unseen objects and locations with minimal or no training. ViNT [125] presented a promising approach in visual navigation, trained on a diverse set of real-world robot navigation data to achieve strong zero-shot generalization and adaptable to various downstream tasks and goal modalities through fine-tuning or prompt-tuning. Zero-shot navigation methods like ZSON [126] and OVRL-V2 [127] demonstrate the capability of navigating to unseen objects and locations without explicit task-specific training, enabling robots to quickly adapt to new environments and tasks, further enhancing their perception and situational awareness. Open-set navigation methods like OpenFMNav [128] expand the scope of semantic navigation by allowing robots to handle free-form natural language instructions with open-set objects, enabling them to perceive and understand complex instructions and navigate to objects not included in the pre-defined vocabulary. Structured scene representations, such as those employed by StructNav [129], VoroNav [130], and ConceptGraphs [91], also play a vital role in enhancing perception and situational awareness by providing efficient and semantically rich scene understanding.

Enhancing navigation robustness and efficiency is crucial for real-world deployment and contributes directly to improved perception and situational awareness. TriHelper [131] addresses challenges like collision avoidance, exploration efficiency, and target misidentification, leading to improved object goal navigation (ObjectNav) benchmark performance. Reasoning about the Unseen [132] focuses on efficient outdoor object navigation, utilizing LLMs and topological graphs to plan routes in complex outdoor settings. OK-Robot [97] demonstrates the importance of integrating Open Knowledge models with robotic modules for practical applications. Expanding semantic navigation to diverse environments and modalities, VELMA [133] focuses on urban vision and language navigation in Street View, showcasing the potential for robot perception and situational awareness. By leveraging LLMs and visual data through CLIP, this method achieves strong performance in urban visual language navigation tasks, using verbalized trajectory inputs and visual landmark identification to guide its navigation. This demonstrates the effectiveness of language-driven interfaces for embodied AI in complex real-world scenarios.

3.1.3 Enhanced perception in embodied models

The integration of foundation models in robotic systems has led to a new generation of robots with advanced

perception and interpretation capabilities. These embodied models use the semantic understanding of foundation models to process visual information, allowing robots to recognize objects, understand their relationships, and respond appropriately.

PaLM-E [37] utilizes the PaLM [85] model with 562B parameters and a 22B VIT to enhance robots' perception and situational awareness through advanced multimodal data processing. By encoding visual and state information interleaved with natural language descriptions, PaLM-E enables robots to perform tasks with a deep understanding of both visual elements and textual instructions. This integration allows for more grounded and contextually relevant responses, critical for tasks requiring a detailed understanding of the physical world. Similarly, Steve-Eye [134] integrates an LMM to process visual and textual inputs, enhancing the perceptual and situational awareness capabilities of robotic agents. Steve-Eye utilizes a visual encoder to transform visual inputs into a format compatible with the LLM, enabling a holistic understanding of the environment. Its two-stage training strategy aligns multimodal features and uses instruction tuning based on an open-world dataset, allowing for effective navigation and operation in complex and dynamically changing environments.

SayCan [135] integrates an LLM with robotic systems to enhance perceptual capabilities by translating natural language instructions into actionable tasks. Using the PaLM model with 540B parameters, it grounds instructions through learned affordance functions that evaluate task feasibility based on the robot's state and capabilities. This ensures effective perception and interpretation of the environment, facilitating appropriate actions. Similarly, RT-1 [40] employs a specialized Transformer model that processes multimodal data, including visual inputs from cameras and textual instructions. Using EfficientNet-B3 for image processing, RT-1 integrates visual and textual data for a comprehensive understanding of tasks and environments, enhancing real-time decision-making. Extending this approach, RT-2 [17] integrates VLMs into robotic control systems. RT-2 processes visual inputs through a pretrained VLM, combining images and textual descriptions to determine robotic actions, enabling accurate perception and contextually appropriate actions.

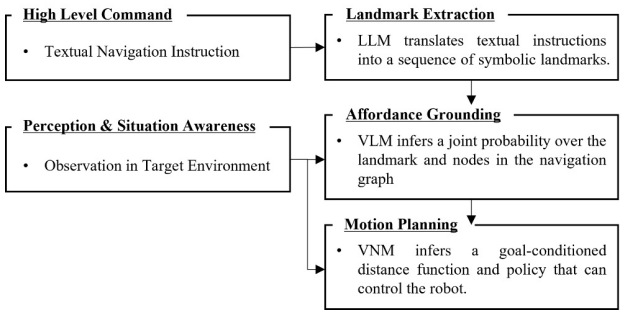
The Open X-Embodiment project [136] discusses RT-X models designed to handle high-dimensional, heterogeneous data from diverse robotic platforms to enhance perception and situational awareness. Pre-trained on large-scale datasets from various robots, RT-X models learn generalist robot policies. Integrating visual sensors and text descriptions, these models accurately perceive and interpret contextual information, enabling effective task execution across different robots and environments. AutoRT [137] significantly enhances robotic perception and situational awareness by integrating large-scale data processing capabilities with sensory inputs. This system autonomously generates and collects robot learning data, which accelerates AI robot development. By combining the RT-1 and RT-2 models, AutoRT allows robots to accurately perceive their environment and understand complex tasks based on visual and textual information. This improved integration of sensory data enables better situational awareness and precise responses to environmental cues, facilitating advanced data mining and analysis for more effective decision-making in dynamic environments. PaLI-X [138] introduces an enhanced VLM with large-scale, multilingual capabilities for complex tasks. Using a ViT-22B model tuned for OCR capabilities, PaLI-X processes images and multilingual textual data, allowing sophisticated perception and understanding of visual content. Multimodal training on diverse datasets enables PaLI-X to excel in tasks requiring detailed perception of visual and textual inputs, significantly improving situational awareness.

The ALOHA [139] system presents a low-cost platform for bimanual robotic teleoperation, focusing on enabling effective perception through advanced teleoperation interfaces and visual sensors. Building on this, ALOHA-2 [140] introduces significant enhancements, including the use of Intel RealSense D405 cameras for high-resolution RGB and depth data capture. The system integrates user inputs through teleoperation interfaces that mimic robotic arm movements, enhancing perceptual capabilities and responsiveness. ALOHA-2's improvements in ergonomic design and sensory input integration enable more effective perception and situational awareness in teleoperation tasks.

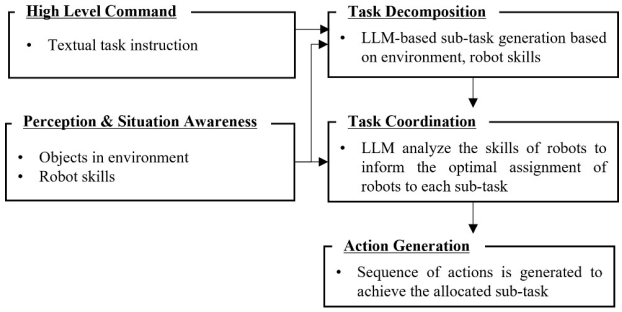
3.2. Decision making and task planning

This section focuses on the critical role of LLMs in facilitating robotic decision-making and task planning. Effective robot operation in complex environments necessitates the ability to understand missions, decompose them into sub-goals, plan actions, and adapt based on feedback or human interaction. LLMs, with their ability to process and generate human-like text, can be incorporated into the decision making and task planning frameworks in diverse ways (Fig. 3). We will explore this topic through four key components of decision and task planning (Fig. 4).

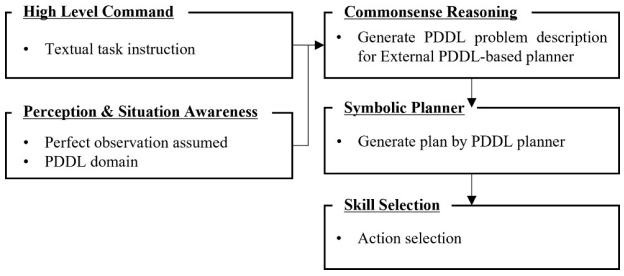
1) **Mission understanding:** This subsection focuses on how LLMs can be used to extract key information from natural language instructions, reason about the environment and objects within it, and select appropriate skills to achieve the desired goals. This includes techniques such as landmark extraction, commonsense reasoning, skill selection and affordance grounding, and search space reduction.
2) **Planning:** Here, we investigate how LLMs can be directly involved in generating action sequences or



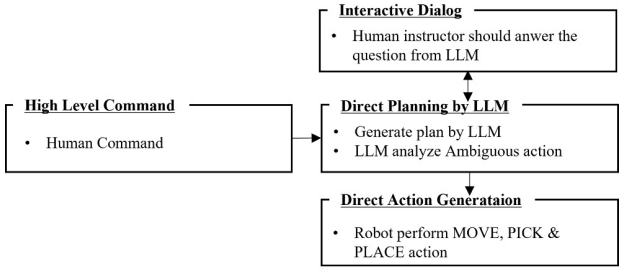
(a) LM-Nav [141].



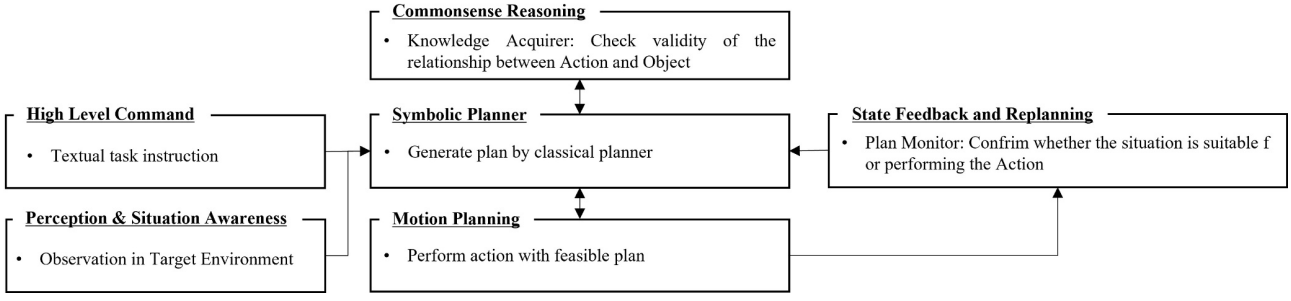
(b) SMART-LLM [21].



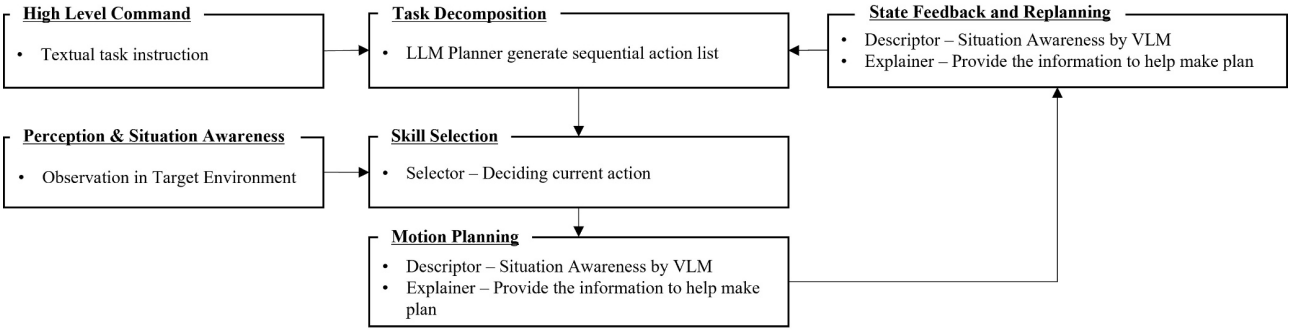
(c) LLM-DP [25].



(d) Hori *et al*. [142].



(e) COWP [27].



(f) DEPS [29].

Fig. 3. This figure showcases six exemplary diagrams illustrating the diverse ways LLMs are incorporated into robot task planning frameworks. Each diagram represents a different research paper, highlighting the specific mechanism of the proposed method. The diagrams depict various components of the task planning process, including mission understanding, planning, validation and correction, and robot-LLM interaction. These examples demonstrate the versatility of LLMs in enhancing task planning, showcasing their potential to empower robots with greater autonomy, flexibility, and adaptability in handling complex tasks (It's important to note that the inclusion of these specific papers does not imply superiority of these methods over others).

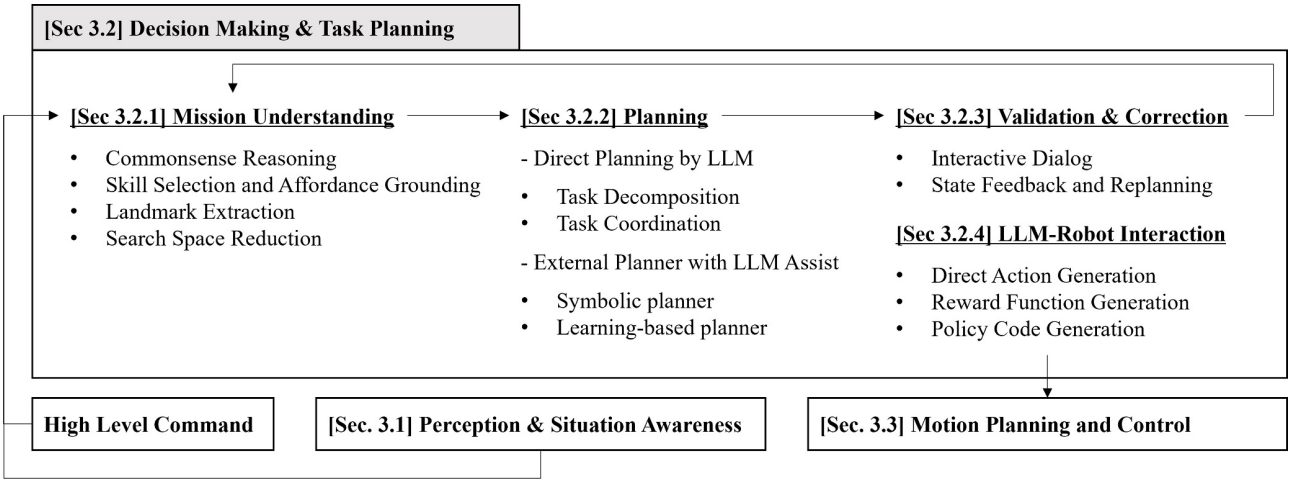


Fig. 4. Decision making and task planning framework that leverages LLMs at various stages. LLMs contribute to mission understanding by reasoning with commonsense knowledge, extracting landmarks, selecting skills, and reducing the search space for plans. They further support planning through direct plan generation or by guiding traditional symbolic and learning-based planners. Validation and correction are facilitated through interactive dialogue and replanning based on robot feedback. Finally, LLMs enable seamless LLM-robot interaction by generating direct action commands, reward functions, and policy code.

assist external planners in creating more robust and adaptable plans. This covers direct planning by LLMs and approaches where LLMs augment the capabilities of traditional planning algorithms.

3) **Validation and correction:** This subsection explores the use of LLMs in ensuring the validity and effectiveness of generated plans. We examine techniques such as interactive dialogue between the robot and user, state feedback mechanisms for plan adaptation, and other miscellaneous approaches for plan correction and refinement.

4) **LLM-robot interaction:** Finally, we describe the various ways LLMs can interact with robots beyond planning. This includes the direct generation of robot actions, generating reward functions for planners such as reinforcement learning, and even producing code for robot policies.

3.2.1 Mission understanding

Before a robot can embark on a task, it needs to first understand what is required. This involves processing natural language instructions, identifying relevant objects and landmarks in the environment, and reasoning about the actions needed to achieve the desired goals. LLMs play a crucial role in this mission understanding in various phases (Table 2).

3.2.1.1 Commonsense reasoning

Understanding the implicit relationships and properties of objects, as well as the general principles of how the world works, is crucial for effective task execution. LLMs, trained on vast amounts of text data, can provide commonsense reasoning capabilities to robots, allowing them to make inferences and predictions beyond explicit instructions.

One group of papers focuses on using LLMs for spatial reasoning and world modeling. Prompter [58] uses LLMs to predict object locations based on spatial relationships with landmarks, guiding robot search. RT-2 [17] demonstrates commonsense reasoning by understanding object relationships (e.g., "put strawberry in the correct bowl"), recognizing precarious situations (e.g., "pick up the bag about to fall off"), and selecting appropriate objects for improvised tools (e.g., "I need to hammer a nail, what object might be useful?"). Similarly, LLM-GROP [59] and LGX [60] leverage LLMs to reason about object relationships and guide object search based on common sense knowledge. Some works highlight the strength of LLMs in creating and maintaining explicit world models [145,146]. LLM-MCTS [148] utilizes LLMs to generate a commonsense world model that informs Monte Carlo tree search planning. LanguageMPC [144] leverages LLM to interpret high-dimensional information in autonomous driving scenarios and helps make common sense judgments based on this.

Another set of papers focuses on applying LLMs to enhance planning and execution with common sense reasoning. Xie *et al.* [147] use LLMs to fill in missing details in human instructions based on common sense. Jarvis-1 [61] leverages LLMs for self-checking and self-explanation during planning, ensuring feasibility and adaptability. ESC [62] employs LLMs to reason about spatial relation-

Table 2. LLM research on mission understanding: A categorical overview.

	Commonsense reasoning	Landmark extraction	Skill selection affordance grounding	Search space reduction
ESC [62], BeyondText [63], COWP [27], LGX [60], SafetyChip [143], LanguageMPC [144], Guan *et al*. [145], LLM+P [146], LLM-GROP [59], Prompter [58], Wei *et al*. [52], Xie *et al*. [147]	✓			
RT-2 [17], Jarvis-1 [61]	✓		✓	
LLM-MCTS [148]	✓			✓
VELMA [133], LM-Nav [141]		✓	✓	
SayCan [18], Lynch *et al*. [23], LLM-Planner [24], AutoGPT+P [64], VOYAGER [149], LLMs4Plan [65], STEVE-1 [150]			✓	
Valmeekam *et al*. [151], RAP [152], Silver *et al*. [153]				✓

ships between objects and rooms, aiding in task execution.

Finally, some papers emphasize the role of LLMs in ensuring safety and robustness. BeyondText [63] utilizes vocal cues to interpret the uncertainty in human instructions, adding a layer of common sense interpretation. The "safety chip" concept [143] uses LLMs as external modules to enforce safety constraints [52]. Chain-of-thought prompting, as seen in COWP [27], enhances LLM reasoning by integrating common sense knowledge for dynamic action augmentation, checking preconditions, and predicting action effects.

3.2.1.2 Skill selection and affordance grounding

LLMs can assist in selecting relevant skills from a predefined library based on the given instruction. This can be further enhanced by incorporating value functions that capture the affordances of each skill, ensuring that the selected skills are feasible in the current environment.

A common theme is combining LLMs with affordance-based methods for robust skill selection. SayCan [18] exemplifies this by using LLMs to score skill relevance to instructions while affordance functions evaluate skill feasibility. RT-2 [17] leverages its training data to select appropriate skills (picking, placing, etc.) and grounds those skills in relation to perceived objects and their affordances. AutoGPT+P [64] similarly leverages LLMs to derive Object Affordance Models, enabling adaptive planning and object substitution based on environmental constraints. LLM-Planner [24], similar to SayCan, utilizes LLMs for high-level planning and uses k-nearest neighbors to estimate future actions based on previous successful ones.

Several papers focus on grounding LLM-based decisions through visual affordances and landmark identification. LLM-GROP [154] utilizes LLMs to extract common sense knowledge about object placement and spatial relationships, enabling human-like rearrangement decisions. LM-Nav [141] combines CLIP and ViNG [155] to ground landmarks in visual observations, facilitating navigation skill selection based on environmental affordances. VELMA [133] uses CLIP to determine landmark visibility in panoramic images, informing LLM-based action selection based on visual affordances.

Other papers emphasize the integration of LLMs with skill libraries and low-level execution mechanisms. LLMs4Plan [65] uses LLMs to enhance graph planning algorithms by selecting promising actions from a vast pool of candidates. Jarvis-1 [61] grounds high-level plans into executable actions within a Minecraft environment, utilizing controllers like STEVE-1 [150] for skill execution and affordance grounding. VOYAGER [149] similarly relies on a skill library for retrieving situation-relevant skills, effectively grounding actions to the agent's capabilities and the environment. Lynch *et al*. [23] emphasizes learning and utilizing a large repertoire of natural-language-conditioned skills, grounding language commands to robot actions and assessing their feasibility.

3.2.1.3 Landmark extraction

Identifying and labeling key objects or locations within the environment mentioned in the instructions is essential for grounding language to the physical world. LLMs can be used to extract such landmarks, enabling the robot to build a spatial representation of its surroundings. This is demonstrated in LM-Nav [141]. VELMA [133] utilizes GPT-3 to extract landmarks from the natural language instructions, similar to LM-Nav. This helps in identifying and tracking relevant visual cues for navigation.

3.2.1.4 Search space reduction

This section focuses on methods that leverage LLMs to significantly reduce the search space explored by downstream planners. These methods aim to guide the planning process, preventing unnecessary exploration and enabling more efficient decision-making, particularly in scenarios with limited information.

One approach involves using LLM-generated plans as heuristic guidance for inherently reliable planners, as

Table 3. LLM research on planning: A categorical overview.

	Direct planning by LLM		External planner with LLM assist	
	Task decomposition	Task coordination	Symbolic planner	Learning based planner
ProgPrompt [20], LLM-State [156], Parsel [28], LgTS [70], 3P-LLM [157], Chen *et al*. [158], TPTU-v2 [159], Silver *et al*. [160], MLDT [161], ADaPT [19], Huang *et al*. [162], TaPA [163], Zhen *et al*. [164], DEPS [29]	✓			
SMART-LLM [21], Octopus [66], AgentLite [22], Text2Motion [165]	✓	✓		
RT-1 [40], VELMA [133], LanguageMPC [144], AgentCoord [166], Roco [67], Zhou *et al*. [167]		✓		
Guan *et al*. [145], LLM+P [146], Sayplan [26], LLM-DP [25], DELTA [168], LLM+ASP [169]			✓	
Chalvatzaki *et al*. [170], LLM-Personalize [171], SwiftSage framework [172]				✓

demonstrated by Valmeekam *et al*. [151]. They further refine the LLM prompts based on feedback from external validators to enhance plan effectiveness. Similarly, Silver *et al*. [153] utilize LLM outputs as initial plans for search-based planners, thereby minimizing the number of nodes explored. LLM-MCTS [148], employs LLMs to create a policy model that functions as a heuristic to guide the search algorithm. This approach effectively reduces the search space and enhances the overall search process of traditional planners. RAP [152] presents a novel approach by reconstructing LLMs into a world model and a reasoning agent. It utilizes an MCTS-based planning algorithm to strategically explore the inference space, with the LLM incrementally building an inference tree. This strategy balances exploration and exploitation to pinpoint inference paths yielding high rewards.

3.2.2 Planning

Once the robot has a clear understanding of the mission, the next step involves formulating a sequence of actions to achieve the desired goal. This planning stage can be approached in two primary ways with LLMs: direct planning by LLM or using external planner with LLM assist. The reviewed papers utilizing LLMs in various task planning phases are shown in Table 3.

3.2.2.1 Direct planning by LLM

LLMs can be used to directly generate action plans based on their understanding of the task, environment, and robot capabilities. By leveraging their knowledge and reasoning abilities, LLMs can produce a series of steps that the robot can follow to accomplish its objectives. This approach can be particularly useful for simple tasks or situations where a high-level plan is sufficient. This section focuses on methods that leverage LLMs directly for robot task planning, exploring two key sub-areas: task decomposition and task coordination.

1) Task decomposition: LLMs can break down high-level instructions into smaller, manageable sub-tasks by identifying relevant objects and actions within the instructions. This information helps generate a sequence of sub-tasks to achieve the overall goal, often combined with skill selection and affordance grounding. Additionally, LLMs can reason about the temporal order and logical dependencies between sub-tasks, ensuring their correct and efficient execution.

Several papers demonstrate the use of LLMs to generate sub-tasks from high-level instructions. SMART-LLM [21] utilizes environmental information, robot skills, and previous task examples to guide LLM-based sub-task generation. Similarly, Huang *et al*. [162] show that LLMs like GPT-3 can decompose tasks even without explicit training on step-by-step examples. ProgPrompt [20] incorporates natural language comments within code structures to represent sub-tasks, while Zhen *et al*. [164] use structured expert knowledge for hierarchical task decomposition. TaPA [163] relies on multi-modal data and object detection to break down instructions into executable actions. TPTU-v2 [159] employs fine-tuned LLMs to generate subtasks and select relevant APIs for complex systems. LLM-State [156] utilizes LLM as a policy to decompose natural language commands into sub-tasks that fit object states. Octopus [66] and Parsel [28] both generates code for sub-tasks to achieve a larger goal. MLDT [161] highlight the role of LLMs in hierarchical planning and code generation for sub-tasks. The paper addresses long-horizon tasks by decomposing them into goal-level, task-level, and action-level components. AgentLite [22] uses a LLM-based manager agent to decompose tasks and assign sub-tasks to individual agents. DEPS [29] performs task decomposition while considering environmental context and previous plan failures. Chen *et al*.

[158] utilize LLMs for step-by-step decision-making in multi-agent path planning, focusing on immediate actions rather than full paths.

Some papers explore adaptive task decomposition and skill sequencing using LLMs. ADaPT [19] uses an LLM planner to decompose tasks into simpler sub-tasks based on LLM capabilities. LgTS [70] employs LLMs to generate potential sub-goal sequences, constructing a directed acyclic graph for a reinforcement learning agent. Text2Motion [165] focuses on decomposing instructions into executable robot skills, while 3P-LLM [157] breaks down navigation instructions into actionable steps. Silver *et al*. [160] apply a three-stage decomposition approach to enable generalized planning with LLMs in the PDDL domain.

2) **Task coordination:** After decomposing a task into sub-tasks, LLMs can coordinate their execution among single or multiple robots. This involves assessing each robot's skills and capabilities to assign sub-tasks efficiently, leveraging the strengths of each robot. Additionally, LLMs can form coalitions or teams of robots when a single robot lacks the necessary skills for a sub-task, enabling efficient utilization of the multi-robot system and completion of complex tasks.

   Chen *et al*. [67] investigate four methods for task coordination: decentralized, centralized, and two hybrid approaches. Decentralized methods empower each robot with its own LLM, allowing them to communicate and decide on actions independently. In contrast, centralized approaches use a single LLM to plan and assign actions for all robots. The hybrid approaches combine elements of both, aiming to improve planning efficiency and task success rates.

   RT-1 [40] acts as the central brain, coordinating various tasks based on visual inputs and language instructions. It maps these inputs to appropriate actions, effectively coordinating the robot's movements and manipulations. VELMA [133] similarly operates as the agent's brain, receiving verbalized observations and past trajectory information to predict the next action for efficient navigation.

   LLMs can assess the skills of individual robots and coordinate their actions to complete tasks efficiently. SMART-LLM [21] focuses on leveraging LLMs to analyze the skills of individual robots. This analysis informs the optimal assignment of robots or robot teams to each sub-task, taking into account factors such as parallel execution possibilities, skill sets, and environmental or robot constraints. The LLM then generates executable code dictating the actions and sequence for each robot, ensuring efficient and coordinated task completion. Similarly, Octopus [66] utilizes an LLM-based manager agent to coordinate actions within a hierarchical multi-agent system, while AgentLite [22] facilitates communication and collaboration among agents, promoting efficient completion of the overall task.

   LLMs can facilitate negotiation and coordinate actions among robots, ensuring tasks are completed according to requirements. Chen *et al*. [67] leverages dialogue between LLMs to negotiate and coordinate actions, taking into account individual robot capabilities and overall task requirements.

   LLMs can leverage formal task representations and coordinate multi-agent systems for various applications. Zhou *et al*. [167] utilize the PDDL format for task and action generation. This approach involves a translator to convert natural language to PDDL files, a planner to generate plans, and a validator for self-refinement. Beyond robotic manipulation, LLMs like LanguageMPC [144] coordinate individual vehicle actions for smooth traffic flow in multi-vehicle scenarios, while Text2Motion [165] coordinates the execution of robot skills, ensuring feasibility and resolving geometric dependencies. AgentCoord [166] analyzes agent capabilities to allocate the optimal agent for each task, planning collaborative processes for task completion.

3.2.2.2 External planner with LLM assist

For more complex tasks or dynamic environments, LLMs can be used to assist external planning algorithms. The LLM can provide valuable information such as landmark understanding, object affordances, and commonsense knowledge to enhance the planner's capabilities. This collaboration can lead to more robust, adaptable, and efficient plans that can better handle unforeseen situations and changes in the environment.

1) **Symbolic planner:** Symbolic planners, relying on formal models like PDDL, have long been central to automated planning. Recently, LLMs have been integrated with symbolic planners to enhance their capabilities. This summary categorizes recent research based on the specific role LLMs play in the planning process.

   One approach leverages LLMs for translating between natural language and PDDL. LLM+P [146] uses LLMs to translate a given problem into a PDDL representation, which is then solved by a classical planner. The resulting PDDL plan is subsequently translated back into natural language. Similarly, LLM-DP [25] employs LLMs to formulate PDDL problems based on domain, goal, and environment descriptions. The classical planner then generates actions based on this PDDL formulation.

   Another approach utilizes LLMs to enhance plan generation. Sayplan [26] employs a two-phase approach: semantic search and iterative replanning. In

the semantic search phase, LLMs search for relevant items within a 3D scene graph using memory. The iterative replanning phase involves LLMs generating high-level plans, while classical planners generate executable plans based on state feedback. DELTA [168] leverages LLMs to construct the PDDL domain and problem. Subsequently, the LLM performs sub-goal decomposition, and the classical planner generates plans for each sub-goal.

Finally, LLMs can also be used for improving the accuracy and expressiveness of PDDL models. Guan *et al*. [145] employ LLMs to construct and correct errors within initial PDDL models. LLM+ASP [169] goes beyond PDDL, using LLMs for semantic mapping to represent stories in logic and then utilizes answer set programming (ASP) to find solutions.

2) **Learning-based planner:** Learning-based planners, unlike their symbolic planner, leverage the power of data-driven learning to generate plans. This summary examines recent research utilizing LLMs for learning-based planning, focusing on different architectural choices and training paradigms.

One approach focuses on fine-tuning LLMs for detailed plan generation. Chalvatza *et al*. [170] fine-tuned a GPT-2 model on the ALFRED dataset to generate detailed plans from simple natural language commands. This demonstrates the potential of directly leveraging pre-trained LLMs for end-to-end planning.

Another approach involves combining LLMs with specialized modules for hierarchical planning. LLM-Personalize [171] consists of three components: a Context Generator, an LLM-Planner, and a Controller. The Context Generator processes sensory information, the LLM-Planner generates high-level actions, and the Controller executes low-level actions. Notably, the LLM-Planner is personalized through imitation learning and user preferences. Similarly, the SwiftSage framework [172] employs two modules: Swift for rapid, specialized thinking and Sage for deeper, reflective thinking. Swift utilizes a compact language model, while Sage guides larger LLMs like GPT-4 in subgoal planning. This highlights the potential of combining LLMs with specialized modules for efficient and nuanced planning.

Both approaches, i.e., 'direct' and 'assist', leverage the strengths of LLMs in different ways to improve the robot's planning capabilities. The choice between direct LLM planning and LLM-assisted external planning depends on the complexity of the task, the capabilities of the robot, and the desired level of autonomy and adaptability.

3.2.3 Validation and correction

Even with careful planning, unforeseen circumstances or incomplete information can lead to deviations from the intended course of action. To ensure task success, it is crucial to have mechanisms for validating and correcting plans in real-time. As shown in Table 4, LLMs can play a significant role in this process through: 1) interactive dialogue and 2) state feedback and replanning.

3.2.3.1 Interactive dialogue

Enabling robots to engage in natural language dialogue with users allows for clarification of instructions, feedback on plan progress, and adjustments based on user input. LLMs can facilitate this interaction by processing user queries, understanding the intent, and generating appropriate responses that guide the robot's actions.

Several LLMs demonstrate this capability. For instance, ChatGPT [174] can engage in dialogue with users, allowing for feedback and correction of the robot's behavior. Wake *et al*. [173] utilize ChatGPT to generate executable robot actions through multi-step instructions and feedback, with the ability to adjust these based on natural language feedback about the provided plan and changing environments. Similarly, AgentLite [22] enables interactive dialogue between agents and users, allowing for clarification and refinement of instructions during task execution.

Addressing ambiguity and missing information in user instructions is another critical aspect of interactive dialogue. Hori *et al*. [142] highlight the challenges of ambiguity and lack of information in natural language instructions provided to LLMs. In their approach, the LLM analyzes instructions, identifies missing information, and poses questions to humans to gather the necessary details, enabling the execution of precise robot action plans. Meanwhile, Lynch *et al*. [23] propose an interactive language framework for real-time interaction, where users can provide feedback and correct the robot's behavior during task execution. This approach enables dynamic replanning and adjustment of actions based on user input.

3.2.3.2 State feedback and replanning

Robots can utilize sensors and mechanisms to monitor their progress and detect deviations from the plan. LLMs can analyze this state feedback in conjunction with the original plan to identify discrepancies and trigger replanning processes. This allows the robot to adapt to changing circumstances and maintain its course towards achieving the desired goals. By incorporating feedback from the environment and the robot's sensors, LLMs can dynamically update their plans or policies to adapt to changing situations and recover from failures. This can be achieved by monitoring the state of the environment and the robot.

Several papers present methods for incorporating state feedback into the planning process. LLM-Planner [24]

Table 4. LLM research on validation and correction and LLM-robot interaction: a categorical overview.

	Validation & correction		LLM-robot interaction		
	Interactive dialogue	State feedback and replanning	Direct action gen.	Reward func. gen.	Policy code gen.
AgentLite [22]	✓	✓	✓		
Lynch *et al.* [23], Hori *et al.* [142]	✓		✓		
Wake *et al.* [173], ChatGPT [174]	✓				
ProgPrompt [20]		✓	✓		✓
Sayplan [26], VELMA [133], RT-1 [40], ADaPT [19], COWP [27], LGX [60], SafetyChip [143], LM-Nav [141], LLM-Planner [24], Raman *et al.* [175]		✓	✓		
LgTS [70], BeyondText [63]		✓		✓	
Octopus [66], LLM-State [156], Parsel [28]		✓			✓
LLM-DP [25], Zhou *et al.* [167], DEPS [29], Jarvis-1 [61], AutoGPT+P [64], VOYAGER [149], LLMs4Plan [65], Inner Monologue system [176]		✓			
Chalvatzaki *et al.* [170], Text2Motion [165], LanguageMPC [144], Huang *et al.* [162], TaPA [163], Zhen *et al.* [164], RT-2 [17], SayCan [18]			✓		
Yu *et al.* [69]				✓	
ESC [62], CaP [177]					✓

uses a grounded replanning algorithm triggered by execution difficulties, prompting the LLM to generate a revised plan based on updated environmental information. Similarly, the Inner Monologue system [176] integrates environmental feedback into the LLM's prompts to allow for plan correction. LLM-DP [25] dynamically adjusts plans in response to real-world environmental changes. SayPlan [26] employs an iterative replanning mechanism, using feedback from a scene graph simulator to identify and rectify errors in planned actions. Raman *et al.* [175] propose using "precondition error" messages from the robot as feedback to recalibrate the plan. AutoGPT+P [64] leverages LLM to represent actionability-based scenes, and helps complete tasks by proposing alternatives or setting partial goals so that tasks can be executed even when objects are not in the environment.

Other papers focus on how LLMs can learn and adapt through feedback loops. COWP [27] uses feedback to improve planning abilities over time by querying the LLM for new knowledge when encountering unfamiliar situations. Octopus [66] assesses the success of actions through environmental feedback, triggering replanning when necessary. Parsel [28] demonstrates self-correction through state feedback and replanning in both HumanEval and robotic planning tasks. AgentLite [22] utilizes a memory module to store action-observation histories for strategy adaptation. In LLMs4Plan [65], infeasible action is recorded in constraint and LLM avoid the constraint to make feasible action.

Several papers explore the role of feedback in error handling and generating better plans. DEPS [29] uses an explainer to provide feedback and generate improved plans, while LGX [60] uses updated state information to trigger replanning when exploration fails. LLM-based systems can incorporate assertions and recovery actions within generated code for structured error handling [20].

Other papers focus on real-time control using continuous feedback. RT-1 [40] and LM-Nav [141] both operate in closed-loop, adjusting actions and replanning trajectories based on continuous visual feedback. VELMA [133] utilizes verbalized observations from panorama images and street intersections for route replanning. ISR-LLM [167] proposes self-validation through an LLM or a customized validator to ensure feasible actions.

Some demonstrates how feedback drives replanning in hierarchical control architectures. JARVIS-1 [61] uses feedback to re-plan and generate improved action sequences. ADaPT [19] relies on a self-generated success heuristic to determine the need for replanning. LgTS [70] employs a Teacher-Student framework where feedback leads to replanning by selecting promising sub-tasks. BeyondText [63] and SafetyChip [143] both utilize feedback in the form of error messages to trigger replanning when safety constraints are violated.

The final group of papers propose specific mechanisms for efficient replanning based on feedback. An iterative prompting mechanism incorporates feedback for code refinement and plan improvement [149], while a closed-loop execution with geometric feasibility planning enables replanning after each step [156]. LLM-State [156] utilizes

dynamically updated state representations automatically constructed by LLMs for object state tracking and inference, contributing to efficient replanning by leveraging information about object attributes and changes.

3.2.4 LLM-robot interaction

The interaction between LLMs and robots for decision-making and task planning can be implemented through several methods, each with its own advantages and limitations. With the categorical overview in Table 4, this subsection explores several ways LLMs can be integrated into robotic systems: 1) direct action generation, 2) reward function generation, and 3) policy code generation.

3.2.4.1 Direct action generation

In this approach, the LLM directly predicts the robot's next action based on the current state and past actions. This can be implemented by prompting the LLM to output either natural language instructions or executable code.

Several papers utilize LLMs to generate high-level action plans in natural language, which are then translated into robot-executable instructions. Huang *et al*. [162], VELMA [133], SayCan [18], Zhen *et al*. [164], LLM-GROP [154], Hori *et al*. [142], and AgentLite [22] fall under this category. These systems allow for flexible and intuitive interaction with the robot using natural language commands.

Other approaches, such as RT-1 [40], RT-2 [17], and LM-Nav [141], directly generate discrete action tokens or code that correspond to specific movements and manipulations of the robot. This approach allows for precise and efficient control of the robot's actions.

LLM-Planner [24], TaPA [163], and LGX [60] utilize LLMs to guide the selection of actions based on the robot's capabilities and the current situation. This allows for more informed decision-making and efficient task completion. ProgPrompt [20], ADaPT [19], LanguageMPC [144], and the safety chip agent [143] demonstrate the use of LLMs for reactive planning and safety. These systems can respond to dynamic environments and ensure safe robot operation. Some systems [23,170], like those presented in Raman *et al*. [175], and Text2Motion [165], utilize natural language processing techniques to directly generate action plans. These systems offer a more intuitive and human-like approach to robot control.

Finally, systems like SayPlan [26] and COWP [27] use symbolic planning techniques to generate direct plans composed of pre-defined actions. This approach leverages the power of symbolic reasoning for efficient plan generation.

3.2.4.2 Reward function generation

LLMs can be used to generate or refine reward functions based on natural language descriptions of desired outcomes or behaviors. This can simplify the design of systems, which need to be fine-tuned, and make them more adaptable to different tasks and environments. The LLM effectively defines the objective for the robot, and a separate optimization algorithm finds a policy that maximizes the expected reward.

Yu *et al*. [69] propose the method that offers flexibility and allows for complex behaviors, but it can be challenging to design appropriate reward functions and ensure that the learned policy aligns with the intended behavior. The LgTS [70] utilizes the LLM to generate a graphical representation of sub-goals, which implicitly defines the reward structure for the reinforcement learning agent. This representation guides the agent towards the goal state, as it receives rewards based on its progress through the graph. In BeyondText [63], the system influences the reward by adjusting the weight matrix of the MPC based on the LLM's interpretation of the human's vocal cues. This indirectly guides the robot's actions towards safer and more reliable navigation.

3.2.4.3 Policy code generation

Leveraging their knowledge of programming languages and control structures, LLMs can generate codes representing robot policies. This allows for expressive and complex policies, including those with feedback loops and reactive behavior.

Examples of works using this method include CaP [177], LLM-State [156], and ProgPrompt [20]. However, this approach requires careful prompt engineering to ensure the correctness and safety of the generated code. Octopus [66] focuses on generating executable code that serves as the policy for the robot's actions within the environment. Parsel [28] utilizes LLMs to generate code implementations for each function within the Parsel program, ultimately creating executable code that acts as the robot's policy. In ESC [62], The LLM's output is incorporated into a frontier-based exploration method to guide the robot's navigation, effectively generating policy code.

3.3. Motion planning and control

LLMs and VLMs, along with generative models like diffusion models, excel at interpreting and generating complex data, which enhances robot performance in dynamic and unpredictable environments.

This section explores how foundation models and diffusion models are integrated into robotic systems to enable sophisticated motion planning and low-level control. Foundation models such as LLMs and VLMs adeptly process vast amounts of language and visual data, strengthening decision-making and facilitating the execution of complex tasks. Diffusion models optimize and sample trajectories for motion planning of robots. In each iteration, the quality of trajectories from diffusion models get better. We divide the topic into two main subsections: Foundation

Models (Subsection 3.3.1) and Diffusion Models (Subsection 3.3.2).

3.3.1 Foundation models for motion planning and control

In this subsection, we review the application of LLMs and VLMs for motion planning and issuing control commands. LLMs and VLMs are foundational models that exhibit exceptional capabilities in understanding and generating complex information. These abilities can enhance the performance of robots in unpredictable and complicated environments. This section introduces research that utilizes these foundational models to enable robots to execute advanced motion planning and achieve low-level control.

3.3.1.1 Foundation models for motion planning

LLMs and VLMs not only excel in natural language processing but also play a critical role in error recovery processes through their ability to handle complex data and pattern recognition. Due to these advantages, both models are effectively utilized in task and motion planning (TAMP). In the field of robotics, while motion planning is often studied independently, a more integrated approach combining motion planning and task planning is necessary for complex and advanced tasks. Hence, this subsection delineates two main approaches. The first approach involves analyzing cases where LLMs and VLMs are applied independently to motion planning. The second approach explores how these models are integrated into TAMP while it particularly focuses on aspects of motion planning.

1) **Motion planning:** Conventional motion planning algorithms primarily focus on path planning and obstacle avoidance, but they are often designed to specific tasks or environments, making them difficult to apply to new situations. Additionally, they require interacting with users using a limited set of commands, which restricts the user experience. To overcome these limitations, recent research has proposed approaches using foundation models. Jiao *et al*. [178] used a LLM to generate waypoints for choreographing a swarm of drones. To ensure safety, they use Swarm-GPT with a safety filter to determine the timing and positioning of each drone according to beats extracted from music data and conducted path analysis for the drones. RoCo [179] exploits LLMs to create collision-free paths for multi-robot collaboration. The LLM facilitates dialogue among robots to plan individual subtasks and generates 3D waypoint paths for their execution. These paths are then processed using RRT to determine the final motion sequences in the robots' joint spaces, enabling effective task execution by the robots.

   Research has been conducted on combining LLMs and VLMs to formulate the motion paths of robots. VoxPoser [100] utilizes these two foundation models to form 3D value maps for motion planning. Initially, the LLM analyzes natural language instructions to recognize the required actions' affordances and constraints. Then, using the VLM, it obtains the location and geometric information of objects, which are used to assign spatial values to the value map. The motion planner then uses this value map to plan the robot's motion. LATTE [180] also applies two foundational models, BERT [55] and CLIP [30], to plan robotic actions. By analyzing natural language commands through these models, a semantic cost map is formed. This map is then processed through the transformer's encoder and decoder to generate specific motion instructions. This complex input allows for effective adjustment of the robot's motion path.

2) **Task and motion planning (TAMP):** Traditional TAMP methods are designed using modules specialized for specific environments or fields, which presents a significant limitation in their adaptability to changes in the environment. To overcome this limitation, new research has been conducted utilizing the feedback processing, reasoning, and generalization capabilities of LLMs.

   LLM3 [181] employs LLMs to generate symbolic action sequences that abstractly describe the steps a robot must perform, and based on these, it derives specific action parameters necessary for real-world environments. Additionally, it analyzes feedback from potential failures during execution and continuously adjusts the action sequences and parameters. This process ensures that TAMP can be effectively applied across various environments. Text2Motion [165] utilizes the reasoning capabilities of LLMs to plan skill sequences for executing given commands. These sequences are then transformed into low-level motion plans through a motion planning process, enabling the robot to perform the tasks in real-world scenarios.

3.3.1.2 Foundation models for control

LLMs and VLMs excel as high-level planners in robotic tasks, demonstrating strong capabilities in text generation, problem-solving, visual recognition, and object-element interaction. However, these models have limitations in directly controlling intricate manipulations or precise robotic movements, tasks that require handling complex sub-operations. To address these issues, research into designing intermediate interfaces for indirect control between Foundation Models and robotic commands is actively underway. These studies categorize the use of Foundation Models in robot control into two main approaches: Direct control and Indirect control. This section introduces

research that utilizes these approaches, further dividing Indirect Control into categories such as reward generation for control optimization and action translation.

1) **Direct control:** In this approach, foundation models generate low-level control commands to enable the system to achieve a given task. Wang *et al*. [68] proposed a study where they used LLMs to generate low-level control commands necessary for a robot to perform dynamic tasks. In this study, they design text prompts to teach the robot walking motions, and enabled the LLM to determine target joint positions based on observations of the physical environment and provided information, which could then be directly executed through a PD controller.

   RT-2 [17] exploits VLM to implement direct control of robots. In this model, The robot's actions are output as numbers or minimal token strings, represented through text instructions describing the tasks to be performed and image data. The output includes the 6 degrees of freedom (6 DOF) for position and rotation, the gripper extension level, and special discrete commands to terminate the episode, making it immediately applicable for actual robot control.

2) **Reward generation for control optimization:** Reinforcement learning is used to maximize robot performance in dynamic environments. Traditionally, designing reward functions for each task manually required significant time and cost. To address this, methods have been developed using LLMs to automatically design reward functions [182,183]. This approach simplifies system design based on natural language descriptions and enables easier adaptation to various environments. However, these methods mostly utilize basic movements to perform robot actions, making them unsuitable for tasks requiring fine manipulation.

   To handle this issue, Yu *et al*. [69] developed a method that connects the LLM and the robot through reward functions, enabling the LLM to infer and control robot actions. The LLM interprets the user's commands, translating them into reward parameters, and maps these to a motion controller using MPC. This allows for optimized actions at each control point. EUREKA [71] develop a reward design algorithm using a code-generating LLM for low-level manipulation tasks. By utilizing LLMs to analyze task descriptions, they designed a reward function that integrates with the robot's sensor data. This function enables real-time adjustment of the robot's behavior, guiding it to take optimal actions. Song *et al*. [184] proposed a method for designing reward functions for continuous control tasks in robotics. A LLM designs the initial reward function, and as the robot performs tasks based on this reward function, it collects data. This data is then used to continuously refine the reward function. Through this process, the robot can perform tasks with greater precision, and the reward function can be continuously improved.

3) **Action translation:** This part introduces research that utilizes LLMs to transform natural language commands into executable robotic actions, and further, to control robots based on these translated actions. SayTap [185] generates a foot contact pattern with an LLM and controller to manage the movement and positioning of a quadruped robot. Through LLM, it analyzes natural language instructions, then uses a pattern generator to define how and when the robot's feet should touch the ground in a binary format of 0s and 1s to control the robot's actions. The mobility controller must maintain the given speed while adjusting to ensure that the foot contact pattern aligns with the required pattern. InCoRo system [186] utilizes an LLM controller that integrates a traditional control feedback loop with LLMs, enabling the robot to operate in dynamic environments. The pre-processor breaks down the natural language commands into atomic actions, after which the LLM within the LLM controller transforms these into low-level commands that the robot can understand. These commands are then used to control the robot's sensors and actuators.

   Cao *et al*. [187] proposed a robot control method using task frame formalism (TFF) and LLM. This method uses LLM to process natural language commands and extract necessary information. Then, it employs TFF to generate task frames for planning movements and utilizes inverse kinematics to convert the movement path into control commands for the robot. CaP [177] employs LLM to ensure system flexibility without relying on predefined policies. It generates language model programs (LMPs) by converting given commands into python code via LLM. These LMPs determine actions based on information from the robot's sensors and parameterize control primitive APIs, enabling low-level control commands.

3.3.2 Diffusion models in motion planning

Diffusion models view motion planning as a trajectory optimization problem, where a trajectory is a sequence of states and actions. This perspective allows diffusion models to be adapted to the robotics domain.

3.3.2.1 Leveraging diffusion models for effective action generation

Motion planning and action generation in robotics are critical for enabling robots to execute precise physical maneuvers [83,84,188-194]. Traditionally, these tasks have been addressed using methods that explicitly model the

tasks and their environments. However, these approaches often fail in complex and unpredictable real-world environments.

Recent advancements in machine learning, particularly diffusion models, offer solutions to these challenges. Diffusion models operate by progressively refining disorganized inputs into structured outputs, but so far, both inputs and outputs are images and videos. This process can be adapted for robotics, where the goal is to evolve a sequence of robotic actions from an initial state (usually purely random) to a desired end state. Similar to image generative models, sampled trajectories are followed by the denoising process, which improves with each iteration.

The strength of diffusion models is that they use a controlled reverse process to transform a high-entropy initial state into an organized, desired state.

3.3.2.2 Trajectory optimization and generation with diffusion models

Janner *et al*. [83] introduce a novel approach that integrates planning and modeling in reinforcement learning by utilizing a diffusion probabilistic model for trajectory optimization. By employing a diffusion model of trajectories, the planning process becomes closely aligned with sampling, enhancing flexibility and adaptability during decision-making. The proposed Diffuser model demonstrates the ability to handle sparse rewards and adapt to new reward structures without retraining. They compose trajectories from in-distribution subsequences. The perturbed distribution used in planning allows for the optimization of functions such as rewards or costs. It enables the reuse of a single diffusion model for multiple tasks within the same environment. Overall, the paper presents a learning-based approach to planning inspired by past work in trajectory optimization, which offers a new direction for diffusion-based planning procedures in reinforcement learning.

Similar to Janner *et al*., many papers [84,189,190,194] have used diffusion models as trajectory generative models. For example, Cavalho *et al*. [84] proposed motion planning diffusion (MPD) models, which introduce the concept of learning diffusion models as priors for motion planning. The models can generate samples from the posterior trajectory distribution which is based on task objectives. The objectives are based on utilizing the denoising mechanism of diffusion models. The method has great performance in encoding high-dimensional trajectory distributions of robot motions. They demonstrate superior performance compared to several baselines in 7 DOF robot arm manipulator environments (Franka Emika). MPD's ability to generalize to environments with previously unseen obstacles highlights its robustness and potential for real-world applications

The recent papers on diffusion models improve the strong point of diffusion models (flexibility) and alleviate the weakness (computation time) of diffusion models [195,196]. Saha *et al*. [195] proposed the Ensemble-of-costs-guided diffusion for motion planning (EDMP) approach, which combines diffusion models with gradient-based cost functions to enhance motion planning. The biggest contributions are leveraging diffusion models to learn a prior, which are kinematically feasible trajectories, and at the same time, incorporating scene-specific guidance through gradient-based cost functions. The novel cost-guided approach in EDMP offers significant benefits by enabling the generation of diverse multimodal trajectories based on scene-specific constraints, such as collision costs. This approach enhances flexibility and efficiency in motion planning by considering multiple cost functions simultaneously. Overall, the cost-guided approach is more versatile and capable of handling complex real-world scenarios with improved trajectory generation capabilities.

Zhou *et al*. [196] focus on replanning with diffusion models, addressing when plans should be regenerated when the environment changes. They suggest a method for deciding when to replan using plan likelihood and replanning trajectories that maintain the original goal. This method evaluates the feasibility of a plan based on the likelihood function learned by the diffusion model, with high probabilities indicating feasible trajectories and low probabilities indicating the need for replanning. This approach significantly improves the performance of diffusion planners, resulting in a 38% improvement in Maze2D tasks

3.3.2.3 Integrating LLMs and diffusion models

Also, there are a few attempts to incorporate LLMs and diffusion models [197]. Liu *et al*. introduce StructDiffusion, a model that synthesizes physically valid structures from unseen objects guided by language instructions. In each iteration, the model analyzes 3D object embeddings and task specifications described in language to predict the target poses of the objects.

Specifically, StructDiffusion utilizes this combination to generate diverse high-level motion goals for language-guided rearrangement tasks. It carefully predicts possible goals that comply with physical constraints, such as detouring collisions between objects. By leveraging object-centric representations, the method navigates both the global constraints of the scene and the local interactions between objects. The model's capacity to generate a variety of structures stems from its ability to sample from Gaussian noises at different scales. This feature is important for resolving ambiguities in language instructions and ensuring that the rearrangements are physically feasible.

4. ENVIRONMENTAL SETUPS IN ROBOTICS

This section provides a fundamental understanding of environmental setups for foundation models in robotics

research. Subsection 4.1 is about the datasets to train robots and data generation. The datasets in this field are of two types: real world (Subsection 4.1.1.1) and simulation (Subsection 4.1.1.2). Subsection 4.2 is about real-world robotic platforms and simulation environments commonly used in robotics research using foundation models.

4.1. Datasets and data generation

This subsection examines the essential datasets for training foundational models for robots. The introduction encompasses datasets collected using actual robots and those generated through simulations, along with their characteristics. Additionally, various studies are explored, demonstrating the utilization of foundational models to expand datasets or generate new data. Through this exploration, readers can attain an understanding of the data environment crucial for training robot foundation models and find assistance in selecting suitable datasets.

4.1.1 Robotics datasets

The lack of datasets for training foundation models for robotics is apparent. While fields like computer vision and natural language processing have access to large amounts of text and image data on an internet scale for training large models, robotics datasets are notably scarce. Additionally, relying solely on text and image datasets presents limitations in training foundation models for robotics. This is because essential physics-related knowledge such as friction, pressure, and weight is difficult to learn from these datasets alone. Therefore, for robots to generalize across more diverse environments and tasks, larger and more diverse multimodal robotics datasets are needed and should be constructed. Typically, to build such datasets, real robots are operated to collect data or simulations are used for data collection.

4.1.1.1 Real world datasets

1) **RoboNet** [198]**:** RoboNet is a massive dataset consisting of 15 million video frames collected from 7 different vision-based robotic manipulation platforms. The dataset includes 162,000 trajectories and takes into account various environmental variables during data collection, such as viewpoint, object, table, and lighting conditions, in addition to robot platform differences.
2) **Bridge V1** [199], **Bridge V2** [200]**:** Bridge V1 is a dataset of 7,200 demonstrations collected using the WidowX 250 robot, designed to facilitate research on the generalization of manipulation tasks in various kitchen environments. It comprises data from 10 distinct environments and features 71 different tasks. Bridge V2 significantly expands upon V1 by providing 60,096 trajectories collected across 24 environments. Additionally, Bridge V2 incorporates data collected not only through human teleoperation but also through a randomized pick-and-place policy. Importantly, all data in V2 is annotated with natural language labels, enabling open-vocabulary task specification through either goal images or natural language instructions.
3) **Language-table** [23]**:** Language-table is a dataset for 2D manipulation tasks, such as moving, using the xArm6 robot. It comprises around 600,000 trajectories and 200 natural language instructions, enabling robots to understand and execute natural language instructions.
4) **RH20T** [201]**:** RH20T is a multimodal robotic dataset that includes not only visual information but also tactile, audio, and proprioception data. It features a variety of robot configurations using 4 different robot arms, 4 different grippers, and 3 types of force-torque sensors, resulting in 7 distinct robot setups. The dataset contains approximately 110 K contact-rich robot manipulation sequences, along with 110 K corresponding human demonstration videos, amounting to over 50 million images.
5) **RT-1** [40]**:** In RT-1, 130 K episodes were collected over 17 months using 13 Google mobile manipulation robots, encompassing over 700 diverse tasks. Each episode is annotated with natural language instructions describing the task performed by the robot.
6) **Open X-embodiment (OXE) datasets** [136]**:** OXE dataset is a large-scale dataset containing 22 different robot platforms and 527 skills. It provides 1,000,000 episodes, including approximately 160,266 tasks, spanning a wide range of robot platforms, from single-arm robots to bimanual robots and quadruped robots.
7) **General navigation model (GNM)** [202]**:** GNM aims to facilitate the learning of a general navigation model that can be applied to new robots by leveraging navigation data from various robots. It comprises 70 hours of navigation data collected across 6 different robots in diverse environments.
8) **Ego4D** [203]**:** Ego4D consists of 3,670 hours of daily-life activity video captured by 931 unique camera wearers across 74 worldwide locations and 9 countries, spanning hundreds of scenarios (household, outdoor, workplace, leisure, etc.). Ego4D is an action-free dataset that doesn't include actual robot actions, which makes it easily applicable to different types of robot platforms, but limits the model's ability to directly learn low-level actions.

4.1.1.2 Simulation datasets

1) **Blocksworld** [204]**:** Blocksworld is a classic problem in the field of AI planning where the task involves moving blocks on a table to match the structure of a target block. Despite its simple and intuitive nature,

Blocksworld has the characteristic of being able to generate problems of various difficulty levels. Consequently, it is used in various simulation environments as a benchmark not only for researching different planning algorithms but also for evaluating the planning and manipulation abilities of manipulation robots.

2) **Action learning from realistic environments and directives (ALFRED)** [205]**:** ALFRED is a benchmark dataset for mapping natural language instructions and egocentric vision into sequences of actions for household tasks. ALFRED covers a total of 120 indoor environments and 84 objects across 7 tasks. Using the AI2THOR simulator, 8,055 expert demonstrations containing a total of 25,743 natural language directives are collected. Each directive includes both high-level goals and low-level natural language instructions. The dataset is partitioned into training, validation, and test folds, with the test fold primarily used as a benchmark to evaluate agent generalization capabilities.
3) **Robot learning benchmark (RLBench)** [206]**:** RLBench is a benchmark and learning environment for robot learning. It includes data for 100 unique manipulation tasks, ranging from simple tasks to multi-stage tasks, using CoppeliaSim/V-REP. This allows for comprehensive evaluation of vision-based manipulation robot learning algorithm. RLBench offers infinite demonstrations at waypoints for each task using motion planners, expanding the potential for demonstration-based learning. Additionally, it provides tools for users to easily create and verify new tasks and demonstrations, making it a benchmark widely used across various research endeavors.
4) **Ravens** [207]**:** Ravens is a simulation-based robot learning environment founded on PyBullet, designed to evaluate Transporter Networks. It offers a benchmark consisting of 10 discrete-time tabletop manipulation tasks for the Universal Robot UR5e equipped with a suction gripper. The benchmark tasks include block insertion, towers of Hanoi, manipulating rope, and more, providing a diverse range of difficulty levels for evaluating robot learning algorithms comprehensively. Additionally, users can define new tasks to integrate into Ravens, allowing for the evaluation of not only Transporter Networks but also the performance of various other algorithms.
5) **Composing actions from language and vision (CALVIN)** [208]**:** CALVIN is an open-source simulated benchmark designed to learn long-horizon daily manipulation tasks using human language instructions and onboard sensor information. Based on the PyBullet [209], it encompasses approximately 24 hours of demonstration data for 34 tasks across four indoor environments, along with about 20,000 language instructions. Additionally, it provides evaluation metrics for multi-task language control (MTLC) measuring the performance of agents across the 34 manipulation tasks and long-horizon MTLC (LH-MTLC) evaluating how well they execute 1000 instruction chains composed of five consecutive tasks. Hence, CALVIN enables the evaluation of agents in zero-shot scenarios with novel language instructions and environments.
6) **VIMA-Bench** [33]**:** VIMA-Bench is a benchmark developed for training and evaluating robot agents using multimodal prompts, based on Ravens [207] simulator. It provides over 650,000 successful robot trajectory data through scripted oracles, and consists of 17 meta-tasks that can be instantiated into approximately 1,000 individual tasks. Additionally, some objects are separated for texture generalization evaluation, and four of the 17 meta-tasks are reserved for zero-shot generalization evaluation. VIMA-Bench employs a four-stage evaluation protocol of varying difficulty levels to systematically measure the agents' generalization capabilities.
7) **BEHAVIOR-1K** [210]**:** BEHAVIOR-1K is a human-centered, embodied AI benchmark featuring 1,000 everyday activities and realistic simulation. It includes 1,000 everyday activities instantiated in 50 fully interactive scenes with over 9,000 objects, selected through surveys to reflect human-preferred robot daily activities, making it a more human-centric benchmark. Moreover, it utilizes the OmniGibson simulator based on NVidia Omniverse and PhysX 5, allowing for more realistic physical phenomena and providing high-quality rendering.

4.1.2 Data feneration

Existing robot data is primarily collected through human teleoperation or engineered data collection schemes, which often require significant time and resources. For instance, in the case of RT-1 [40], obtaining 130 K episodes required 13 robots and 17 months. Consequently, there is a burgeoning research effort to expand or generate robot datasets beyond real robot data collection. Particularly, there is a considerable focus on leveraging the data generation capabilities of large-scale pretrained models such as text-to-image generation models and LLMs to increase the diversity of training data, enabling robots to learn more robust and generalizable skills.

The related studies are classified into 5 categories according to the method of augmenting datasets. The classification criteria and related papers for each category are as follows:

4.1.2.1 Visual augmentation

Visual augmentation focuses on modifying the visual attributes of images, such as color, texture, and lighting, to increase data variability.

1) **CACTI** [211]**:** CACTI is a framework that focuses on collecting expert demonstration data and expanding it through visual augmentation, enabling robots to learn robustly across various tasks and scene variations. It utilizes an in-painting approach, inserting various plausible objects into pre-defined masked regions using Stable Diffusion [212]. This helps robots become more robust to variations in visual input but does not address semantic changes in the environment.
2) **GenAug** [213]**:** GenAug modifies object textures, colors and background with pre-defined masked regions and it uses depth-guided image model for photorealism.

4.1.2.2 Semantic augmentation

Semantic augmentation focuses on modifying the semantic content of images, introducing new objects, backgrounds, and distractor configurations to enhance the robot's ability to adapt to new tasks and environments.

1) **GenAug** [213]**:** Besides visual augmentation, GenAug also performs semantic augmentation by generating images with in-category and cross-category object substitutions and new visual distractors through a text-to-image generative model. This helps robots learn to interact with various objects they haven't seen before.
2) **ROSIE** [214]**:** ROSIE utilizes OWL-ViT [74], an open vocabulary segmentation model, to automatically generate masks for regions of interest by detecting objects, backgrounds, or distractors mentioned in the task description. Then, it leverages LLMs like GPT-3 to propose text prompts for image augmentation. Finally, ROSIE employs Imagen Editor [215], a text-to-image diffusion model, to generate new objects, distractors, or background changes within the masked regions based on the provided text prompts.
3) **SuSIE** [216]**:** Using a pre-trained image-editing diffusion model, SuSIE generates future sub-goals' images based on language commands, effectively creating new tasks and scenarios for the robot to learn. This enables zero-shot generalization for tasks that are not explicitly present in the training data.

4.1.2.3 Language-based augmentation

Language-based augmentation is applied to increase the linguistic diversity of data using a text-to-image generative model or LLM and to learn for robots to understand and follow various language instructions. In general, there are methods of creating new instruction in an existing dataset using VLM or LLM, or decomposing complex tasks into sub-tasks using LLM and creating language labels for each sub-task.

1) **DIAL** [217]**:** A fine-tuned VLM learns the relationship between visual scenes and language instructions using a small dataset of robot data and corresponding language instructions. Instruction augmentation is performed by utilizing the fine-tuned VLM to generate new instructions for a larger dataset of unlabeled robot demonstrations. Through this augmented dataset, the language-conditioned robot policy becomes more robust to diverse language instructions.
2) **Scaling up and distilling down** [218]**:** Scaling up and distilling down is a single-system framework that combines the advantages of LLM, random sampling planner, and policy learning. It utilizes LLM for high-level task planning and generates various robot trajectories using a sampling-based planner to execute them. Then, it labels the success/failure of each trajectory using LLM and learns visuo-linguo-motor policies based on the collected data. Scaling up and distilling down provides richer language context, enhancing the robot's understanding of the task structure.

4.1.2.4 Simulation-based augmentation

Simulation-based augmentation focuses on generating simulation environments and tasks for data augmentation, rather than directly augmenting data itself. In other words, it utilizes LLMs to automatically create tasks, scenes, and training supervisors within the simulation environment. Then, the data collected from these generated simulations is used for robot learning. This approach enables the collection of a vast amount of data with minimal cost and effort compared to real-world data collection.

1) **GenSim** [219]**:** GenSim leverages the grounding and code generation capabilities of LLMs to automatically create new simulation tasks, thereby improving the policy's generalization ability at the task level rather than the scene level. GenSim consists of three main components: a task creator that proposes a tasks for solving a given task or expands existing tasks to generate new tasks and demonstrations, a task library for storing and retrieving generated tasks, and an LLM-supervised multitask policy that enhances task-level generalization by utilizing the generated task data.
2) **RoboGen** [220]**:** RoboGen is a framework that utilizes LLMs and generative models to automatically generate training data for robot learning, including various tasks and simulation environments. This enables efficient expansion of skill learning across different robot platforms. Using an LLM like GPT-4, it

suggests tasks based on the type of robot and given object information and generates simulation scenes suitable for the proposed tasks through a generative model. Additionally, it decomposes the suggested tasks into smaller subtasks using the LLM and selects appropriate learning algorithms for each subtask. Finally, robots learn these skills in a simulation environment.

4.1.2.5 No augmentation

Some papers don't utilize any augmentation techniques for robot data generation. Instead, there exists research that creates robot data itself, rather than augmenting existing data.

1) **RT-Trajectory** [221]**:** To address the challenge of generalization at the task level in existing language-conditioned and goal-conditioned policies, RT-Trajectory specifies tasks through 2D trajectory sketches and trains trajectory-conditioned policies. These trajectory sketches can be provided through human-drawn sketches, human demonstration videos, code generation via LLM prompts, and image generation models. Policies learned from trajectory data enable robots to generalize to new tasks and behaviors not present in the data.

Data generation techniques significantly impact the performance and generalization capabilities of robot learning frameworks. Carefully considering the strengths and weaknesses of each technique is crucial for selecting the appropriate approach for specific tasks and environments.

4.2. Platforms

The papers on robotics using foundation models can be grouped into two sets based on which platforms are used: real robots and simulations (Table 5).

4.2.1 Real robots

Most real robots used in robotics are manipulators. They usually have 7 DOFs, and some are fixed, and others are mounted on movable parts to move freely. Mobile robots are classified into two types: wheeled and quadruped. Lastly, a few robots are semi-humanoids that can make high-level decisions.

4.2.1.1 UR series [240]

UR series are developed by Universal Robots, and they are a standard type of manipulator. There are many products (UR3e, UR5e, UR10e, UR16e, ..., etc.) that fit users' needs, but usually, UR5e is commonly used. The UR series is also known for its ease of programming and versatility. These robots integrate smoothly with the robot operating system (ROS), enhancing their utility in research and industrial applications. They are characterized by a user-friendly interface, facilitating quick changes in deployment and lowering barriers to automation. The collaborative capabilities of the UR series allow them for a wide range of applications, such as working side-by-side with human operators and assembling tasks on a small scale.

4.2.1.2 Sawyer

Sawyer, developed by Rethink Robotics, is an industrial robotic arm engineered for tasks that demand precision, such as machine tending and circuit board testing. Sawyer has 7 DOFs, embedded sensors, and the vision system which enable adaptive behavior and precise task execution in dynamic environments. Sawyer has a payload capacity of 4 kilograms and includes an intuitive user interface.

4.2.1.3 LBR iiwa

The KUKA LBR iiwa, part of the KUKA robots lineup, stands out due to its distinctive features oriented towards sensitive and precise automation tasks. The acronym "iiwa" represents "intelligent industrial work assistant," which emphasizes its design for complex industrial applications. The LBR iiwa is available in different payload capacities, including versions that handle up to 7 and 14 kilograms. Notably, the LBR iiwa's capabilities are enhanced by its lightweight construction and 7 DOFs, allowing for greater flexibility and range of movement in various industrial environments.

4.2.1.4 Franka Emika [241]

Franka Emika, notably the Panda robot, is developed by Wego Robotics. The Panda robot is designed to be user-friendly and highly adaptable. Panda is characterized by its 7-axis arm, which is lightweight and capable of handling a payload of up to 3 kilograms. Its integrated torque sensors in each joint enable it to perform tasks that require fine motor skills and a gentle touch. The design of Panda focuses on safe human-robot interaction, which is essential for environments where close cooperation between humans and machines is necessary.

4.2.1.5 Everyday Robots [242]

Everyday Robots project aims to develop robotic systems capable of executing practical tasks within unstructured human environments. These robots combine advanced manipulative abilities with machine learning and sensory technologies to interact with dynamic everyday settings. The robot used in SayCan [135] and RT-1 [40] is a mobile manipulator with 7 DOFs, a two-fingered gripper, and a movable base. There is no clearly defined name for that in the mentioned papers.

4.2.1.6 WidowX 250s

WidowX 250s, made by Trossen Robotics, is a robotic manipulator designed primarily for research purposes. It features 6 DOFs, providing a good balance of flexibility

Table 5. List of robotic platforms in studies on foundation models.

Platform's name [Robot manufacturer]	Real robot type / simulation	Literature
UR series [Universal Robots]	Manipulator	[27], [36], [177], [219], [222], [223], [224], [225]
Sawyer [Rethink Robotics]	Manipulator	[198], [226]
LBR iiwa [KUKA Roboticss]	Manipulator	[198]
Franka Emika [Wego Robotics]	Manipulator	[20], [26], [35], [84], [165], [198], [226], [227], [228], [229], [60]
Everyday Robots [Google]	Manipulator	[40], [105], [135]
WidowX 250s [Trossen robotics]	Manipulator	[199], [200]
HSR [Toyota]	Manipulator	[230]
xARM-7 [uFactory]	Manipulator	[219], [231]
TurtleBot series [TurtleBot]	Wheeled	[60]
Jackal UGV [Clearpath robotics]	Wheeled	[91], [141], [202]
Spot [Boston Dynamics]	Quadruped	[92], [143]
A1 [Unitree]	Quadruped	[185]
NiCOL	Semi-humanoid	[232], [233]
V-REP (CoppeliaSim)	Simulation	[231], [232], [233], [234]
PyBullet	Simulation	[67], [177], [181], [197], [227]
Mujoco	Simulation	[69], [179]
NVIDIA Isaac Sim	Simulation	[167]
Virtual Home	Simulation	[20], [143], [145], [156], [161], [162], [173], [175], [235], [236]
ALFWorld	Simulation	[19], [24], [25], [145] [29], [237], [238]
RoboTHOR and AI2-THOR	Simulation	[21], [36], [60], [62], [104], [122], [163], [170], [224]
Habitat	Simulation	[35], [62], [103], [104], [121], [123], [126], [127], [128], [129], [130], [131]
Minecraft	Simulation	[29], [61], [134]
Overcooked Environment	Simulation	[239]

and functionality for various automated tasks. The manipulator can handle a payload of up to 250 grams for lightweight operations. Its construction allows for a wide range of motion, facilitating complex maneuvers in constrained spaces. The WidowX 250s is widely used in universities for studies in robotics and automation.

4.2.1.7 Human support robot (HSR) [243]

HSR by Toyota Robotics is designed to assist individuals with mobility challenges, especially in domestic environments. It features a telescopic arm and a mobile base, enabling it to perform various tasks such as retrieving objects, opening doors, and manipulating small items. Its programmable characteristics allow for customization to specific user needs or research applications, making it a versatile tool for developing assistive technologies.

4.2.1.8 xARM-7 [244]

xARM-7, developed by UFactory, is another robotic arm notable for its utility in industrial and research settings. It has 7 DOFs, enhancing its dexterity and ability to perform more intricate tasks. With a payload capacity of up to 5 kilograms, the xARM-7 can be employed in simple pick-and-place operations and more sophisticated assembly tasks. The user-friendly interface promotes ease of integration and operation. Its characteristics make the xARM-7 suitable not only for industrial environments but also for research institutions exploring robotic automation.

4.2.1.9 TurtleBot [245]

TurtleBot series developed as an open-source platform, serves as a foundational tool for robotics education and research. These robots are known for their modular design. The popular model, such as the TurtleBot 3, features a lightweight and compact design, weighing approximately 2 kilograms and measuring about 138 mm in diameter and 199 mm in height. TurtleBot 3 is equipped with a 360-degree LiDAR sensor, essential for navigation and obstacle detection. TurtleBot 3 can extend to mount additional sensors or devices with a payload capacity of up to 250 grams. Its capacity is well-suited for mapping and interactive tasks.

4.2.1.10 Jackal UGV

Jackal unmanned ground vehicle (UGV), produced by Clearpath Robotics, is engineered for use as a mobile platform in both outdoor and indoor research settings. With dimensions of 508 mm by 430 mm by 250 mm and weighing around 17 kilograms, it can carry additional payloads

of up to 20 kilograms. The robot has four wheels that operate independently, allowing a maximum speed of 2 m/s. Equipped with integrated sensors and computational capabilities, the Jackal UGV is particularly useful for projects requiring autonomous navigation and mapping.

4.2.1.11 Spot

Spot, developed by Boston Dynamics, is an advanced mobile robot designed for a variety of industrial and commercial applications. This quadruped robot is characterized by its ability to navigate complex terrains and environments. Spot features an advanced array of sensors that facilitate autonomous navigation and effective obstacle detection. It is engineered to handle payloads up to 14 kilograms and can maneuver through challenging landscapes, including stairs. When fully charged, Spot operates for about 90 minutes.

4.2.1.12 A1

A1, developed by UniTree, is a robotic platform designed for agility and autonomous navigation across various environments. A1 has a payload capacity of up to 5 kilograms and features high-speed capabilities, reaching speeds of up to 3.3 meters per second. The A1's design incorporates a lightweight structure, weighing approximately 12 kilograms, which contributes to its agility and performance. Its battery provides a runtime of roughly 2.5 hours, which is enough for outdoor experiments.

4.2.1.13 NiCOL robots [246]

NiCOL robots, by researchers in Japan, represent an advancement in the field of semi-humanoid robotics. NiCOL focuses primarily on non-verbal communication. It has 25 DOFs, distributed across its head, arms, legs, and torso, enabling sophisticated movement patterns. Furthermore, NiCOL employs advanced deep-learning algorithms to interpret human gestures and expressions. This integration of robotic mobility and cognitive computing creates a more intuitive communication interface.

4.2.2 Simulation environments

Simulation environments offer significant advantages for robotics research by providing a controlled setting where numerous experiments can be conducted with ease and at a lower cost. Simulation environments allow for rapid, numerous iterations and testing of robotic algorithms without the physical wear and risk associated with real-world testing.

4.2.2.1 V-REP (CoppeliaSim) [247]

V-REP is utilized in both academia and industry to simulate robots. This software employs a modular approach that enables users to create, combine, and modify components to achieve realistic simulations. The capabilities of V-REP include a comprehensive suite of features that support the development and testing of algorithmic and robotic solutions. It can also analyze robotic behaviors within controlled environments.

4.2.2.2 PyBullet [209]

PyBullet is an open-source robotics simulator. It features advanced physics simulation capabilities suitable for projects involving complex dynamics and real-time simulations. PyBullet offers an intuitive Python API that eases the process of robot control and learning algorithm development. It has many robotic platforms and is frequently utilized in research focused on reinforcement learning and humanoid robotics.

4.2.2.3 Multi-joint dynamics with contact (Mujuco) [248]

Mujoco is a high-performance physics engine for simulating complex robotic systems that involve multiple contacts. Its attributes of speed and accuracy in dynamic simulations make it a popular tool for the development and testing of advanced control algorithms. Mujoco is widely used in academic research, particularly for applications in robotic autonomous systems that require precise interactions with environments.

4.2.2.4 NVIDIA Isaac Sim [249]

NVIDIA Isaac Sim, constructed on the Omniverse platform, is crafted for simulating and testing AI-driven robotics. It utilizes universal scene description (USD) to offer exceptionally realistic simulation settings. The platform is versatile because it can be applied to diverse robotics applications, including detailed physics simulation, photorealistic rendering, and AI-based perception. Isaac Sim is seamlessly integrated with NVIDIA's extensive suite of AI and robotics development tools, including deep learning and computer vision libraries.

4.2.2.5 Virtual Home [250,251]

Virtual Home is a simulation platform that replicates everyday household environments to examine interactions between robots and complex indoor spaces. It enables researchers to simulate and assess scenarios where robots perform tasks such as cooking, cleaning, and organizing, thus supporting the advancement of domestic robots.

4.2.2.6 ALFWorld [252]

ALFWorld combines the text-based game engine TextWorld [253] with the visual-based robot simulator ALFRED (Subsection 4.1.1.2). By providing both text-based abstract representations and visual-based concrete representations of the same environment, agents can experience and learn from both modalities simultaneously. In TextWorld, agents interact with environment information represented in text and engage with high-level text commands. However, in ALFRED, they interact with environments rendered as high-quality images and perform low-level robot actions. This dual-environment structure

is highly effective for learning both language-based reasoning abilities and visual-based action capabilities simultaneously.

4.2.2.7 RoboTHOR [254] and AI2-THOR [255]

RoboTHOR and AI2-THOR are both simulation environments developed to enhance research in visual AI and embodied agents. One commonality between them is their focus on providing detailed, interactive indoor environments that facilitate the experimentation of AI models. However, RoboTHOR is distinct in its approach of bridging the gap between simulation environments and real-world applications, which enables consistent experiments across both simulated and physical robots within similar setups. In contrast, AI2-THOR offers a more extensive manipulation of environmental variables and objects, which focuses on dynamic interaction within virtual spaces to improve autonomous navigation and visual perception capabilities. This distinction highlights RoboTHOR's emphasis on practical deployment readiness, while AI2-THOR concentrates on broadening the scope of interactive simulations for AI research.

4.2.2.8 Habitat [256-258]

Habitat is a simulation platform intended to experiment AI agents within realistic 3D environments. It focuses on high-performance simulations that emphasize visual navigation and manipulation tasks in daily life scenarios. Habitat facilitates the training of AI models through the use of photo-realistic environments that can be procedurally generated. This platform is vital for advancing the capabilities of virtual testing environments for autonomous agents.

4.2.2.9 Minecraft [61] and Overcooked environment [259]

Minecraft environment and the Overcooked environment are both used in robotics and AI research to simulate complex tasks requiring strategic planning and teamwork. Overcooked focuses on the coordination and cooperation between agents to efficiently manage a kitchen and prepare meals under time constraints. In contrast, the Minecraft environment presents a sandbox-style world where agents engage in tasks such as building, crafting, and surviving which needs creative problem-solving and resource management. Both environments are valuable for studying multi-agent systems and the interaction between AI agents and dynamic, changing environments. Despite their different thematic focuses, they share the core feature of providing a rich, interactive platform for developing and testing advanced AI strategies.

5. FUTURE WORKS

While foundation models have demonstrated significant potential in advancing robot autonomy, several critical challenges must be addressed for their widespread adoption in real-world applications.

Current robotic systems struggle to handle dynamic and unpredictable environments, lacking the robust perception and adaptation capabilities necessary for real-world deployment. The computational demands of foundation models pose significant challenges for real-time operation on resource-constrained robot platforms. Furthermore, ensuring the safety and reliability of robots driven by complex foundation models requires enhanced explainability, interpretability, and robust validation mechanisms. The development of truly generalizable robot skills is also hindered by the scarcity of diverse and comprehensive robotics datasets. Future research efforts must focus on overcoming these limitations to fully realize the potential of foundation models in robotics. This section outlines promising research directions in three key areas.

1) **Improving robot capabilities:** This involves developing methods for handling dynamic environments, integrating multi-modal sensing, improving motion planning and control, and enhancing zero-shot and few-shot task generalization.
2) **Enhancing model efficiency and robustness:** This encompasses integrating retrieval augmented generation, developing on-device foundation models, optimizing task planning architectures, and addressing computational limitations for real-time control.
3) **Advancing dataset development and simulation:** This includes expanding and diversifying robotics datasets, improving simulation realism, and overcoming challenges associated with data generation techniques.

5.1. Improving robot capabilities

1) **Dynamic environments and object permanence:** Existing approaches predominantly focus on static scenes, limiting their applicability to real-world scenarios. Future research should focus on integrating dynamic object detection and tracking, enabling robots to understand and interact with dynamic surroundings. Reasoning about object permanence, i.e., understanding that objects exist even when out of sight, will further enhance situational awareness and planning capabilities in complex environments.
2) **Multimodal fusion and interaction:** Expanding multimodal mapping beyond vision and language by incorporating audio, haptic feedback, and proprioceptive sensing will offer a more comprehensive understanding of the environment. This requires developing new methods for fusing data from diverse modalities and reasoning about the world using richer sensory inputs. Research on human perception and multi-sensory integration can provide valuable insights for this direction.

3) **Explainability and interpretability:** Building trust and ensuring safe and reliable robot behavior necessitates understanding the decision-making process of foundation models. Future work should focus on enhancing the explainability and interpretability of these models, especially in robotic tasks. This can be achieved through techniques from explainable AI (XAI), visualizing internal representations, and developing methods for tracing decisions back to their origins within the model.
4) **Human-robot collaboration and intention understanding:** For seamless human-robot collaboration, robots need to understand human intentions and adapt their behavior accordingly. Future robotic systems should focus on developing methods for shared autonomy, natural language interaction, and reasoning about human behavior and goals. This can be facilitated through research in human-robot interaction, collaborative robotics, and cognitive science.
5) **Motion planning and control enhancements:** Bridging the gap between high-level language understanding and low-level control requires further advancements in motion planning and control. This includes developing methods for accurate mapping of natural language instructions to robot actions, generalizing motion planning across diverse environments, improving real-time response capabilities, and enhancing precise action execution. Incorporating real-time feedback mechanisms, collecting diverse interaction data, and developing algorithms that effectively handle complex tasks and generalize across various environments is crucial for widespread adoption.
6) **Zero-shot and few-shot task generalization:** While foundation models have shown promising generalization capabilities, their performance in zero-shot and few-shot scenarios, where they encounter novel tasks or environments not seen during training, remains a significant challenge for practical robotic deployment [17,40]. This is because real-world applications demand robots to adapt to situations that go beyond the specific examples or instructions encountered during training. For instance, models struggle with truly novel compositions of skills, exhibit sensitivity to visual and semantic shifts in objects and instructions, and are limited by the scarcity and bias inherent in real-world robot data [40,137]. Overcoming these limitations requires developing techniques that improve the models' ability to reason about unseen scenarios, adapt to visual and semantic variations, and leverage external knowledge sources to complement limited training data. This might involve exploring novel training paradigms that encourage compositional representations of skills, incorporating data augmentation or regularization methods for robustness to distribution shifts, and integrating external knowledge bases for more informed decision-making.

5.2. Enhancing model efficiency and robustness

1) **Integrating retrieval augmented generation (RAG):** To address LLM hallucination and ensure consistent, grounded decisions, future research can explore integrating RAG systems. By combining LLM generative capabilities with information retrieval from external knowledge bases, robots can access factual information and domain-specific knowledge during planning. This will enable more reliable and robust decision-making in complex and dynamic environments.
2) **On-device foundation models:** Enabling real-time robot operation and autonomy in network-constrained environments necessitates deploying foundation models directly on robot hardware. Future work should focus on developing lightweight and efficient foundation models through techniques like model compression, quantization, knowledge distillation, and specialized hardware architectures.
3) **Asynchronous LLM integration:** To address the computational burden of synchronous LLM usage, future robotic systems should incorporate asynchronous LLM integration. This involves pre-training LLMs on specific task domains, developing tiered decision-making systems that utilize computationally cheaper algorithms for routine tasks, and introducing asynchronous planning strategies where LLMs work in the background while the robot executes previously planned actions.
4) **Optimizing task planning architectures:** Exploring optimal task planning architectures is crucial for efficient and adaptable robot behavior. This includes investigating flexible hierarchical structures where LLMs can dynamically adjust task hierarchies based on environmental feedback and robot state, exploring hybrid approaches combining traditional planning methods with LLM capabilities, and integrating human-in-the-loop feedback for continuous improvement of LLM decision-making.
5) **Addressing limitations in foundation model-based control:** Overcoming the limitations of direct and indirect control with LLMs and VLMs requires enhancing robot cognitive abilities and processing speeds. Current direct control methods, where LLMs or VLMs directly generate low-level commands, struggle with tasks that demand fine motor skills or dexterous manipulation, primarily due to the limited compositionality and semantic grounding of individual joint angles or actuator commands [17]. Indirect control approaches, which often involve using LLMs to

design reward functions for RL or generate high-level plans, face challenges in ensuring the alignment of LLM outputs with desired robot behavior and achieving real-time performance due to the computational demands of LLMs [47,49]. To address these limitations, future research should focus on developing more complex natural language processing algorithms for accurate and swift execution of human intentions, conducting extensive testing across diverse robot types and environments for greater versatility, integrating real-time feedback mechanisms, collecting and incorporating diverse interaction data, and developing algorithms capable of handling complex tasks and generalizing across diverse environments.

5.3. Advancing dataset development and simulation

1) **Expanding and diversifying datasets:** Building larger and more diverse robotics datasets is crucial for improving model generalization. This includes addressing the challenges of real-world data collection, such as time and resource constraints, bias towards specific tasks or environments, privacy concerns, and ethical considerations. Advancements in large-scale data processing and storage technologies are essential for managing growing datasets, and standardization of data formats, labeling criteria, and evaluation metrics is critical for effective dataset construction and sharing.
2) **Improving simulation realism:** Bridging the gap between simulation and real-world scenarios is vital for effective robot training. Future work should focus on developing more realistic physics engines and simulators that accurately model real-world physics and environmental interactions. This will ensure that skills learned in simulation translate effectively to real-world applications.
3) **Data generation challenges:** While data generation techniques increase data diversity, they can also compromise data integrity and introduce biases. Future research should explore methods for generating data without losing crucial information and avoiding bias, ensuring the generated data is representative of real-world scenarios and does not exacerbate overfitting.
4) **Overcoming diffusion model limitations:** Addressing the computational cost of diffusion models in training and deployment is crucial for their wider adoption in robotics. Future research should investigate efficient training and inference methods, exploring model compression, quantization, and specialized hardware architectures. Additionally, developing techniques for controlling the output of diffusion models, ensuring consistency with task requirements and desired specifications, will be essential for reliable and precise robot behavior.

6. CONCLUSION

This survey provides a comprehensive and insightful exploration of the rapidly advancing foundation models and the applications in robotics, focusing on their innovative capabilities to achieve robot autonomy. It explores how various applications of foundation models in relevant studies, including LLMs, VLMs, RFMs, diffusion models, and LMMs, contribute to essential elements of robot autonomy, such as perception, planning, and control.

By integrating natural language reasoning and context understanding with multi-modal processing capabilities, foundation models enable robots to perceive and comprehend complex environments similarly to humans, facilitating sophisticated decision-making and task execution. Furthermore, these models with grounded knowledge effectively assist robots in interacting with the world. In this paper, these advancements are systematically categorized to provide researchers with a clear understanding of how foundation models are utilized to address the challenges of achieving robotic autonomy, including object detection, semantic mapping, navigation, task decomposition and coordination, motion planning, and action generation. Additionally, this paper summarizes the real-world and simulation environments, datasets, data generation techniques, and major robot platforms used in the literature to train the models and evaluate their applications. This summary aims to share necessary information and resources with researchers, aiding in various studies for securing more robust and generalizable robotic autonomy.

The remarkable capabilities of foundation models offer new opportunities for paradigm shifts in both the field of artificial intelligence and robotics. However, building independent, flexible, and general robotic autonomy through foundation models, and applying them to actual robots, still presents many challenges. Issues such as robustness, reliability, data scarcity, real-time performance limitations, and the assurance of explainability and safety remain critical areas for future research. Nevertheless, promising approaches such as retrieval-augmented generation, on-device foundation models, and asynchronous LLM integration offer noteworthy directions to address these challenges and further extend the boundaries of robotic autonomy.

This survey bridges the gap between advances in artificial intelligence and robotic autonomy, guiding researchers to expand their understanding of this rapidly evolving technology and apply it to their respective fields. We anticipate that ongoing research and application of foundation models in robotics will ultimately provide practical and effective solutions that enable robots to perform complex tasks in dynamic and unpredictable environments and to collaborate safely and reliably with humans.

CONFLICTS OF INTEREST

The authors declare that there is no competing financial interest or personal relationship that could have appeared to influence the work reported in this paper. The corresponding author, Han-Lim Choi, is a senior editor of this journal.

REFERENCES

[1] J. H. Lee, "Model predictive control: Review of the three decades of development," *International Journal of Control, Automation, and Systems*, vol. 9, pp. 415-424, 2011.

[2] C. Jing, H. Shu, and Y. Song, "Model predictive control for integrated lateral stability and rollover prevention based on a multi-actuator control system," *International Journal of Control, Automation, and Systems*, vol. 21, no. 5, pp. 1518-1537, 2023.

[3] Y. Zhang, S. Li, and L. Liao, "Near-optimal control of nonlinear dynamical systems: A brief survey," *Annual Reviews in Control*, vol. 47, pp. 71-80, 2019.

[4] K. Prag, M. Woolway, and T. Celik, "Toward data-driven optimal control: A systematic review of the landscape," *IEEE Access*, vol. 10, pp. 32190-32212, 2022.

[5] Y.-Q. Jiang, S.-Q. Zhang, P. Khandelwal, and P. Stone, "Task planning in robotics: An empirical comparison of PDDL-and ASP-based systems," *Frontiers of Information Technology & Electronic Engineering*, vol. 20, pp. 363-373, 2019.

[6] L. G. D. Véras, F. L. Medeiros, and L. N. Guimaráes, "Systematic literature review of sampling process in rapidly-exploring random trees," *IEEE Access*, vol. 7, pp. 50933-50953, 2019.

[7] S. Lim and S. Jin, "Safe trajectory path planning algorithm based on RRT* while maintaining moderate margin from obstacles," *International Journal of Control, Automation, and Systems*, vol. 21, no. 11, pp. 3540-3550, 2023.

[8] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 740-759, 2020.

[9] B. Singh, R. Kumar, and V. P. Singh, "Reinforcement learning in robotic applications: A comprehensive survey," *Artificial Intelligence Review*, vol. 55, no. 2, pp. 945-990, 2022.

[10] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Motion planning and control for mobile robot navigation using machine learning: A survey," *Autonomous Robots*, vol. 46, no. 5, pp. 569-597, 2022.

[11] L. Le Mero, D. Yi, M. Dianati, and A. Mouzakitis, "A survey on imitation learning techniques for end-to-end autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14128-14147, 2022.

[12] S. Choi, S. Kim, and H. Jin Kim, "Inverse reinforcement learning control for trajectory tracking of a multirotor UAV," *International Journal of Control, Automation, and Systems*, vol. 15, pp. 1826-1834, 2017.

[13] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877-1901, 2020.

[14] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[15] R. Thoppilan, D. De Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du *et al.*, "LaMDA: Language models for dialog applications," *arXiv preprint arXiv:2201.08239*, 2022.

[16] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen *et al.*, "PaLM 2 technical report," *arXiv preprint arXiv:2305.10403*, 2023.

[17] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, "RT-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[18] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng, "Do as I can, not as I say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[19] A. Prasad, A. Koller, M. Hartmann, P. Clark, A. Sabharwal, M. Bansal, and T. Khot, "ADaPT: As-needed decomposition and planning with language models," *arXiv preprint arXiv:2311.05772*, 2023.

[20] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "ProgPrompt: Generating situated robot task plans using large language models," *Proc. of 2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 11523-11530, 2023.

[21] S. S. Kannan, V. L. N. Venkatesh, and B.-C. Min, "SMART-LLM: Smart multi-agent robot task planning using large language models," *arXiv preprint arXiv:2309.10062*, 2023.

[22] Z. Liu, W. Yao, J. Zhang, L. Yang, Z. Liu, J. Tan, P. K. Choubey, T. Lan, J. Wu, H. Wang *et al.*, "AgentLite: A lightweight library for building and advancing task-oriented LLM agent system," *arXiv preprint arXiv:2402.15538*, 2024.

[23] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence, "Interactive language: Talking to robots in real time," *IEEE Robotics and Automation Letters*, 2023.

[24] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, "LLM-Planner: Few-shot grounded planning for embodied agents with large language models," *Proc. of the IEEE/CVF International Conference on Computer Vision*, pp. 2998-3009, 2023.

[25] G. Dagan, F. Keller, and A. Lascarides, "Dynamic planning with a LLM," *arXiv preprint arXiv:2308.06391*, 2023.

[26] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf, "SayPlan: Grounding large language models using 3d scene graphs for scalable robot task planning," *Proc. of Conference on Robot Learning (CoRL)*, 2023.

[27] Y. Ding, X. Zhang, S. Amiri, N. Cao, H. Yang, A. Kaminski, C. Esselink, and S. Zhang, "Integrating action knowledge and LLMs for task planning and situation handling in open worlds," *Autonomous Robots*, 2023.

[28] E. Zelikman, Q. Huang, G. Poesia, N. Goodman, and N. Haber, "Parsel: Algorithmic reasoning with language models by composing decompositions," *Advances in Neural Information Processing Systems*, vol. 36, pp. 31466-31523, 2023.

[29] Z. Wang, S. Cai, A. Liu, X. Ma, and Y. Liang, "Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents," *arXiv preprint arXiv:2302.01560*, 2023.

[30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," *Proc. of International Conference on Machine Learning*, PMLR, pp. 8748-8763, 2021.

[31] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," *Proc. of International Conference on Machine Learning*, PMLR, pp. 4904-4916, 2021.

[32] J. Li, D. Li, C. Xiong, and S. Hoi, "BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation," *Proc. of International Conference on Machine Learning*, PMLR, pp. 12888-12900, 2022.

[33] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan, "VIMA: General robot manipulation with multimodal prompts," *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.

[34] R. Shah, R. Martín-Martín, and Y. Zhu, "MUTEX: Learning unified policies from multimodal task specifications," *Proc. of 7th Annual Conference on Robot Learning*, 2023.

[35] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Audio visual language maps for robot navigation," *arXiv preprint arXiv:2303.07522*, 2023.

[36] K. M. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, A. Maalouf, S. Li, G. Iyer, S. Saryazdi, N. Keetha *et al.*, "ConceptFusion: Open-set multimodal 3d mapping," *arXiv preprint arXiv:2302.07241*, 2023.

[37] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu *et al.*, "PaLM-E: An embodied multimodal language model," *arXiv preprint arXiv:2303.03378*, 2023.

[38] L. Xue, N. Yu, S. Zhang, A. Panagopoulou, J. Li, R. Martin-Martin, J/ Wu, C. Xiong, R. Xu, J. C. Niebles, and S. Savarese, "Ulip-2: Towards scalable multimodal pre-training for 3d understanding", *arXiv preprint arXiv:2305.08275*, 2023.

[39] P. Sermanet, T. Ding, J. Zhao, F. Xia, D. Dwibedi, K. Gopalakrishnan, C. Chan, G. Dulac-Arnold, S. Maddineni, N. J. Joshi *et al.*, "RoboVQA: Multimodal long-horizon reasoning for robotics," *arXiv preprint arXiv:2311.00899*, 2023.

[40] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "RT-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[41] K. Bousmalis, G. Vezzani, D. Rao, C. Devin, A. X. Lee, M. Bauza, T. Davchev, Y. Zhou, A. Gupta, A. Raju *et al.*, "RoboCat: A self-improving foundation agent for robotic manipulation," *arXiv preprint arXiv:2306.11706*, 2023.

[42] Y. Cheng, C. Zhang, Z. Zhang, X. Meng, S. Hong, W. Li, Z. Wang, Z. Wang, F. Yin, J. Zhao *et al.*, "Exploring large language model based intelligent agents: Definitions, methods, and prospects," *arXiv preprint arXiv:2401.03428*, 2024.

[43] X. Huang, W. Liu, X. Chen, X. Wang, H. Wang, D. Lian, Y. Wang, R. Tang, and E. Chen, "Understanding the planning of LLM agents: A survey," *arXiv preprint arXiv:2402.02716*, 2024.

[44] H. Li, J. Leung, and Z. Shen, "Towards goal-oriented large language model prompting: A survey," *arXiv preprint arXiv:2401.14043*, 2024.

[45] K. Yang, J. Liu, J. Wu, C. Yang, Y. R. Fung, S. Li, Z. Huang, X. Cao, X. Wang, Y. Wang *et al.*, "If LLM is the wizard, then code is the wand: A survey on how code empowers large language models to serve as intelligent agents," *arXiv preprint arXiv:2401.00812*, 2024.

[46] K. Kawaharazuka, T. Matsushima, A. Gambardella, J. Guo, C. Paxton, and A. Zeng, "Real-world robot applications of foundation models: A review," *arXiv preprint arXiv:2402.05741*, 2024.

[47] R. Firoozi, J. Tucker, S. Tian, A. Majumdar, J. Sun, W. Liu, Y. Zhu, S. Song, A. Kapoor, K. Hausman *et al.*, "Foundation models in robotics: Applications, challenges, and the future," *arXiv preprint arXiv:2312.07843*, 2023.

[48] J. Wang, Z. Wu, Y. Li, H. Jiang, P. Shu, E. Shi, H. Hu, C. Ma, Y. Liu, X. Wang *et al.*, "Large language models for robotics: Opportunities, challenges, and perspectives," *arXiv preprint arXiv:2401.04334*, 2024.

[49] Y. Hu, Q. Xie, V. Jain, J. Francis, J. Patrikar, N. Keetha, S. Kim, Y. Xie, T. Zhang, Z. Zhao *et al.*, "Toward general-purpose robots via foundation models: A survey and meta-analysis," *arXiv preprint arXiv:2312.08782*, 2023.

[50] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[51] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532-1543, 2014.

[52] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24824-24837, 2022.

[53] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan, "Tree of Thoughts: Deliberate problem solving with large language models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[54] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[55] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[57] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27730-27744, 2022.

[58] Y. Inoue and H. Ohashi, "Prompter: Utilizing large language model prompting for a data efficient embodied instruction following," *arXiv preprint arXiv:2211.03267*, 2022.

[59] Y. Ding, X. Zhang, C. Paxton, and S. Zhang, "Task and motion planning with large language models for object rearrangement," *Proc. of 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 2086-2092, 2023.

[60] V. S. Dorbala, J. F. Mullen Jr, and D. Manocha, "Can an embodied agent find your "cat-shaped mug"? LLM-based zero-shot object navigation," *IEEE Robotics and Automation Letters*, 2023.

[61] Z. Wang, S. Cai, A. Liu, Y. Jin, J. Hou, B. Zhang, H. Lin, Z. He, Z. Zheng, Y. Yang *et al.*, "JARVIS-1: Open-world multi-task agents with memory-augmented multimodal language models," *arXiv preprint arXiv:2311.05997*, 2023.

[62] K. Zhou, K. Zheng, C. Pryor, Y. Shen, H. Jin, L. Getoor, and X. E. Wang, "ESC: Exploration with soft commonsense constraints for zero-shot object navigation," *Proc. of International Conference on Machine Learning*, PMLR, 2023, pp. 42829-42842.

[63] X. Sun, H. Meng, S. Chakraborty, A. S. Bedi, and A. Bera, "Beyond Text: Utilizing vocal cues to improve decision making in llms for robot navigation tasks," *arXiv preprint arXiv:2402.03494*, 2024.

[64] T. Birr, C. Pohl, A. Younes, and T. Asfour, "AutoGPT+P: Affordance-based task planning with large language models," *arXiv preprint arXiv:2402.10778*, 2024.

[65] H. H. Zhuo, X. Chen, and R. Pan, "On the roles of llms in planning: Embedding llms into planning graphs," *arXiv preprint arXiv:2403.00783*, 2024.

[66] J. Yang, Y. Dong, S. Liu, B. Li, Z. Wang, C. Jiang, H. Tan, J. Kang, Y. Zhang, K. Zhou *et al.*, "Octopus: Embodied vision-language programmer from environmental feedback," *arXiv preprint arXiv:2310.08588*, 2023.

[67] Y. Chen, J. Arkin, Y. Zhang, N. Roy, and C. Fan, "Scalable multi-robot collaboration with large language models: Centralized or decentralized systems?" *arXiv preprint arXiv:2309.15943*, 2023.

[68] Y.-J. Wang, B. Zhang, J. Chen, and K. Sreenath, "Prompt a robot to walk with large language models," *arXiv preprint arXiv:2309.09969*, 2023.

[69] W. Yu, N. Gileadi, C. Fu, S. Kirmani, K.-H. Lee, M. G. Arenas, H.-T. L. Chiang, T. Erez, L. Hasenclever, J. Humplik *et al.*, "Language to rewards for robotic skill synthesis," *arXiv preprint arXiv:2306.08647*, 2023.

[70] Y. Shukla, W. Gao, V. Sarathy, A. Velasquez, R. Wright, and J. Sinapov, "LgTS: Dynamic task sampling using LLM-generated sub-goals for reinforcement learning agents," *arXiv preprint arXiv:2310.09454*, 2023.

[71] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," *arXiv preprint arXiv:2310.12931*, 2023.

[72] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[73] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu *et al.*, "Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[74] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, X. Wang, X. Zhai, T. Kipf, and N. Houlsby, "Simple open-vocabulary object detection," *Proc. of European Conference on Computer Vision*, pp. 726-755, 2022.

[75] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui, "Open-vocabulary object detection via vision and language knowledge distillation," *arXiv preprint arXiv:2104.13921*, 2021.

[76] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting twenty-thousand classes using image-level supervision," *Proc. of European Conference on Computer Vision*, Springer, pp. 350-368, 2022.

[77] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang *et al*., "Grounded language-image pre-training," *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10965-10975, 2022.

[78] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," *Proc. of International Conference on Machine Learning*, PMLR, pp. 2256-2265, 2015.

[79] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[80] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840-6851, 2020.

[81] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *arXiv preprint arXiv:2011.13456*, 2020.

[82] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780-8794, 2021.

[83] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," *arXiv preprint arXiv:2205.09991*, 2022.

[84] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters, "Motion planning diffusion: Learning and planning of robot motions with diffusion models," *Proc. of 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 1916-1923, 2023.

[85] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al*., "PaLM: Scaling language modeling with pathways," *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1-113, 2023.

[86] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg *et al*., "A generalist agent," *arXiv preprint arXiv:2205.06175*, 2022.

[87] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al*., "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[88] (2023) GPT-4 model documentation. [Online]. Available: https://platform.openai.com/docs/models/gpt-4

[89] M. Reid, N. Savinov, D. Teplyashin, D. Lepikhin, T. Lillicrap, J.-b. Alayrac, R. Soricut, A. Lazaridou, O. Firat, J. Schrittwieser *et al*., "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *arXiv preprint arXiv:2403.05530*, 2024.

[90] S. Ma, S. Vemprala, W. Wang, J. K. Gupta, Y. Song, D. McDufft, and A. Kapoor, "Compass: Contrastive multimodal pretraining for autonomous systems," *Proc. of 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 1000-1007, 2022.

[91] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa *et al*., "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," *arXiv preprint arXiv:2309.16650*, 2023.

[92] N. H. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, "Vlfm: Vision-language frontier maps for zero-shot semantic navigation," *Proc. of 2nd Workshop on Language and Robot Learning: Language as Grounding*, 2023.

[93] J. Yang, W. Tan, C. Jin, B. Liu, J. Fu, R. Song, and L. Wang, "Pave the way to grasp anything: Transferring foundation models for universal pick-place robots," *arXiv preprint arXiv:2306.05716*, 2023.

[94] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al*., "Segment anything," *Proc. of the IEEE/CVF International Conference on Computer Vision*, pp. 4015-4026, 2023.

[95] F. Liu, K. Fang, P. Abbeel, and S. Levine, "MOKA: Open-vocabulary robotic manipulation through mark-based visual prompting," *arXiv preprint arXiv:2403.03174*, 2024.

[96] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan *et al*., "Grounded sam: Assembling open-world models for diverse visual tasks," *arXiv preprint arXiv:2401.14159*, 2024.

[97] P. Liu, Y. Orru, C. Paxton, N. M. M. Shafiullah, and L. Pinto, "OK-robot: What really matters in integrating open-knowledge models for robotics," *arXiv preprint arXiv:2401.12202*, 2024.

[98] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation," *Proc. of Proc of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23171-23181, 2023.

[99] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, S. Kirmani, B. Zitkovich, F. Xia *et al*., "Open-world object manipulation using pre-trained vision-language models," *arXiv preprint arXiv:2303.00905*, 2023.

[100] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.

[101] W. Huang, F. Xia, D. Shah, D. Driess, A. Zeng, Y. Lu, P. Florence, I. Mordatch, S. Levine, K. Hausman *et al*., "Grounded decoding: Guiding text generation with grounded models for robot control," *arXiv preprint arXiv:2303.00855*, 2023.

[102] J. Gao, B. Sarkar, F. Xia, T. Xiao, J. Wu, B. Ichter, A. Majumdar, and D. Sadigh, "Physically grounded vision-language models for robotic manipulation," *arXiv preprint arXiv:2309.02561*, 2023.

[103] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam, "Clip-fields: Weakly supervised semantic fields for robotic memory," *arXiv preprint arXiv:2210.05663*, 2022.

[104] S. Yenamandra, A. Ramachandran, K. Yadav, A. Wang, M. Khanna, T. Gervet, T.-Y. Yang, V. Jain, A. W. Clegg, J. Turner *et al.*, "Homerobot: Open-vocabulary mobile manipulation," *arXiv preprint arXiv:2306.11565*, 2023.

[105] B. Chen, F. Xia, B. Ichter, K. Rao, K. Gopalakrishnan, M. S. Ryoo, A. Stone, and D. Kappler, "Open-vocabulary queryable scene representations for real world planning," *Proc. of 2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 11509-11522, 2023.

[106] T. Yoneda, J. Fang, P. Li, H. Zhang, T. Jiang, S. Lin, B. Picker, D. Yunis, H. Mei, and M. R. Walter, "Statler: State-maintaining language models for embodied reasoning," *arXiv preprint arXiv:2306.17840*, 2023.

[107] A. Kamath, M. Singh, Y. LeCun, G. Synnaeve, I. Misra, and N. Carion, "Mdetr-modulated detection for end-to-end multi-modal understanding," *Proc. of the IEEE/CVF International Conference on Computer Vision*, pp. 1780-1790, 2021.

[108] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven semantic segmentation," *arXiv preprint arXiv:2201.03546*, 2022.

[109] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," *Proc. of 2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 10608-10615, 2023.

[110] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "Tidybot: Personalized robot assistance with large language models," *Autonomous Robots*, vol. 47, no. 8, pp. 1087-1102, 2023.

[111] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, "Simple but effective: Clip embeddings for embodied AI," *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14829-14838, 2022.

[112] R. Abdelfattah, Q. Guo, X. Li, X. Wang, and S. Wang, "Cdul: Clip-driven unsupervised learning for multi-label image classification," *Proc. of the IEEE/CVF International Conference on Computer Vision*, pp. 1348-1357, 2023.

[113] N. Kanazawa, K. Kawaharazuka, Y. Obinata, K. Okada, and M. Inaba, "Recognition of heat-induced food state changes by time-series use of vision-language model for cooking robot," *arXiv preprint arXiv:2309.01528*, 2023.

[114] R. Zhang, Z. Guo, W. Zhang, K. Li, X. Miao, B. Cui, Y. Qiao, P. Gao, and H. Li, "Pointclip: Point cloud understanding by clip," *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8552-8562, 2022.

[115] L. Xue, M. Gao, C. Xing, R. Martín-Martín, J. Wu, C. Xiong, R. Xu, J. C. Niebles, and S. Savarese, "Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding," *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1179-1189, 2023.

[116] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," *Proc. of 2017 IEEE International Conference on Robotics and automation (ICRA)*, IEEE, pp. 4628-4635, 2017.

[117] C. Liu, K. Wang, J. Shi, Z. Qiao, and S. Shen, "FM-Fusion: Instance-aware semantic mapping boosted by vision-language foundation models," *IEEE Robotics and Automation Letters*, 2024.

[118] S. Taguchi and H. Deguchi, "Online embedding multi-scale CLIP features into 3D maps," *arXiv preprint arXiv:2403.18178*, 2024.

[119] K. Yamazaki, T. Hanyu, K. Vo, T. Pham, M. Tran, G. Doretto, A. Nguyen, and N. Le, "Open-Fusion: Real-time open-vocabulary 3D mapping and queryable scene representation," *arXiv preprint arXiv:2310.03923*, 2023.

[120] N. Keetha, A. Mishra, J. Karhade, K. M. Jatavallabhula, S. Scherer, M. Krishna, and S. Garg, "AnyLoc: Towards universal visual place recognition," *IEEE Robotics and Automation Letters*, 2023.

[121] B. Yu, H. Kasaei, and M. Cao, "L3MVN: Leveraging large language models for visual target navigation," *Proc. of 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 3554-3560, 2023.

[122] A. Rajvanshi, K. Sikka, X. Lin, B. Lee, H.-P. Chiu, and A. Velasquez, "SayNav: Grounding large language models for dynamic planning to navigation in new environments," *arXiv preprint arXiv:2309.04077*, 2023.

[123] Y. Qiao, Y. Qi, Z. Yu, J. Liu, and Q. Wu, "March in Chat: Interactive prompting for remote embodied referring expression," *Proc. of the IEEE/CVF International Conference on Computer Vision*, pp. 15758-15767, 2023.

[124] G. Zhou, Y. Hong, and Q. Wu, "NavGPT: Explicit reasoning in vision-and-language navigation with large language models," *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 7, pp. 7641-7649, 2024.

[125] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "ViNT: A foundation model for visual navigation," *arXiv preprint arXiv:2306.14846*, 2023.

[126] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra, "ZSON: Zero-shot object-goal navigation using multimodal goal embeddings," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32340-32352, 2022.

[127] K. Yadav, A. Majumdar, R. Ramrakhya, N. Yokoyama, A. Baevski, Z. Kira, O. Maksymets, and D. Batra, "OVRL-V2: A simple state-of-art baseline for ImageNav and ObjectNav," *arXiv preprint arXiv:2303.07798*, 2023.

[128] Y. Kuang, H. Lin, and M. Jiang, "OpenFMNav: Towards open-set zero-shot object navigation via vision-language foundation models," *arXiv preprint arXiv:2402.10670*, 2024.

[129] J. Chen, G. Li, S. Kumar, B. Ghanem, and F. Yu, "How to not train your dragon: Training-free embodied object goal navigation with semantic frontiers," *arXiv preprint arXiv:2305.16925*, 2023.

[130] P. Wu, Y. Mu, B. Wu, Y. Hou, J. Ma, S. Zhang, and C. Liu, "VoroNav: Voronoi-based zero-shot object navigation with large language model," *arXiv preprint arXiv:2401.02695*, 2024.

[131] L. Zhang, Q. Zhang, H. Wang, E. Xiao, Z. Jiang, H. Chen, and R. Xu, "TriHelper: Zero-shot object navigation with dynamic assistance," *arXiv preprint arXiv:2403.15223*, 2024.

[132] Q. Xie, T. Zhang, K. Xu, M. Johnson-Roberson, and Y. Bisk, "Reasoning about the unseen for efficient outdoor object navigation," *arXiv preprint arXiv:2309.10103*, 2023.

[133] R. Schumann, W. Zhu, W. Feng, T.-J. Fu, S. Riezler, and W. Y. Wang, "VELMA: Verbalization embodiment of LLM agents for vision and language navigation in street view," *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, pp. 18924-18933, 2024.

[134] S. Zheng, Y. Feng, Z. Lu *et al*., "Steve-Eye: Equipping LLM-based embodied agents with visual perception in open worlds," *Proc. of The Twelfth International Conference on Learning Representations*, 2023.

[135] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al*., "Do as I can, not as I say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[136] A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan *et al*., "Open X-Embodiment: Robotic learning datasets and RT-X models," *arXiv preprint arXiv:2310.08864*, 2023.

[137] M. Ahn, D. Dwibedi, C. Finn, M. G. Arenas, K. Gopalakrishnan, K. Hausman, B. Ichter, A. Irpan, N. Joshi, R. Julian *et al*., "AutoRT: Embodied foundation models for large scale orchestration of robotic agents," *arXiv preprint arXiv:2401.12963*, 2024.

[138] X. Chen, J. Djolonga, P. Padlewski, B. Mustafa, S. Changpinyo, J. Wu, C. R. Ruiz, S. Goodman, X. Wang, Y. Tay *et al*., "PaLI-X: On scaling up a multilingual vision and language model," *arXiv preprint arXiv:2305.18565*, 2023.

[139] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile ALOHA: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," *arXiv preprint arXiv:2401.02117*, 2024.

[140] J. Aldaco, T. Armstrong, R. Baruch, J. Bingham, S. Chan, K. Draper, D. Dwibedi, C. Finn, P. Florence, S. Goodrich *et al*., "ALOHA 2: An enhanced low-cost hardware for bimanual teleoperation," *arXiv preprint arXiv:2405.02292*, 2024.

[141] D. Shah, B. Osiński, S. Levine *et al*., "LM-Nav: Robotic navigation with large pre-trained models of language, vision, and action," *Proc. of Conference on Robot Learning*, PMLR, pp. 492-504, 2023.

[142] K. Hori, K. Suzuki, and T. Ogata, "Interactively robot action planning with uncertainty analysis and active questioning by large language model," *Proc. of 2024 IEEE/SICE International Symposium on System Integration (SII)*, IEEE, pp. 85-91, 2024.

[143] Z. Yang, S. S. Raman, A. Shah, and S. Tellex, "Plug in the safety chip: Enforcing constraints for LLM-driven robot agents," *arXiv preprint arXiv:2309.09919*, 2023.

[144] H. Sha, Y. Mu, Y. Jiang, L. Chen, C. Xu, P. Luo, S. E. Li, M. Tomizuka, W. Zhan, and M. Ding, "LanguageMPC: Large language models as decision makers for autonomous driving," *arXiv preprint arXiv:2310.03026*, 2023.

[145] L. Guan, K. Valmeekam, S. Sreedharan, and S. Kambhampati, "Leveraging pre-trained large language models to construct and utilize world models for model-based task planning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 79081-79094, 2023.

[146] B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, and P. Stone, "LLM+P: Empowering large language models with optimal planning proficiency," *arXiv preprint arXiv:2304.11477*, 2023.

[147] Y. Xie, C. Yu, T. Zhu, J. Bai, Z. Gong, and H. Soh, "Translating natural language to planning goals with large-language models," *arXiv preprint arXiv:2302.05128*, 2023.

[148] Z. Zhao, W. S. Lee, and D. Hsu, "Large language models as commonsense knowledge for large-scale task planning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[149] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, "VOYAGER: An open-ended embodied agent with large language models," *arXiv preprint arXiv:2305.16291*, 2023.

[150] S. Lifshitz, K. Paster, H. Chan, J. Ba, and S. McIlraith, "Steve-1: A generative model for text-to-behavior in minecraft," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[151] K. Valmeekam, M. Marquez, S. Sreedharan, and S. Kambhampati, "On the planning abilities of large language models-a critical investigation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[152] S. Hao, Y. Gu, H. Ma, J. J. Hong, Z. Wang, D. Z. Wang, and Z. Hu, "Reasoning with language model is planning with world model," *arXiv preprint arXiv:2305.14992*, 2023.

[153] T. Silver, V. Hariprasad, R. S. Shuttleworth, N. Kumar, T. Lozano-Pérez, and L. P. Kaelbling, "PDDL planning with pretrained large language models," *Proc. of NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.

[154] Y. Ding, X. Zhang, C. Paxton, and S. Zhang, "Leveraging commonsense knowledge from large language models for task and motion planning," *Proc. of RSS 2023 Workshop on Learning for Task and Motion Planning*, 2023.

[155] D. Shah, B. Eysenbach, G. Kahn, N. Rhinehart, and S. Levine, "ViNG: Learning open-world navigation with visual goals," *Proc. of 2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 13215-13222, 2021.

[156] S. Chen, A. Xiao, and D. Hsu, "LLM-State: Expandable state representation for long-horizon task planning in the open world," *arXiv preprint arXiv:2311.17406*, 2023.

[157] E. Latif, "3P-LLM: Probabilistic path planning using large language model for autonomous robot navigation," *arXiv preprint arXiv:2403.18778*, 2024.

[158] W. Chen, S. Koenig, and B. Dilkina, "Why solving multi-agent path finding with large language model has not succeeded yet," *arXiv preprint arXiv:2401.03630*, 2024.

[159] Y. Kong, J. Ruan, Y. Chen, B. Zhang, T. Bao, S. Shi, G. Du, X. Hu, H. Mao, Z. Li *et al.*, "TPTU-v2: Boosting task planning and tool usage of large language model-based agents in real-world systems," *arXiv preprint arXiv:2311.11315*, 2023.

[160] T. Silver, S. Dan, K. Srinivas, J. B. Tenenbaum, L. Kaelbling, and M. Katz, "Generalized planning in PDDL domains with pretrained large language models," *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 18, pp. 20256-20264, 2024.

[161] Y. Wu, J. Zhang, N. Hu, L. Tang, G. Qi, J. Shao, J. Ren, and W. Song, "MLDT: Multi-level decomposition for complex long-horizon robotic task planning with open-source large language model," *arXiv preprint arXiv:2403.18760*, 2024.

[162] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," *Proc. of International Conference on Machine Learning*, PMLR, pp. 9118-9147, 2022.

[163] Z. Wu, Z. Wang, X. Xu, J. Lu, and H. Yan, "Embodied task planning with large language models," *arXiv preprint arXiv:2307.01848*, 2023.

[164] Y. Zhen, S. Bi, L. Xing-tong, P. Wei-qin, S. Haipeng, C. Zi-rui, and F. Yi-shu, "Robot task planning based on large language model representing knowledge with directed graph structures," *arXiv preprint arXiv:2306.05171*, 2023.

[165] K. Lin, C. Agia, T. Migimatsu, M. Pavone, and J. Bohg, "Text2Motion: From natural language instructions to feasible plans," *Autonomous Robots*, vol. 47, no. 8, pp. 1345-1365, 2023.

[166] B. Pan, J. Lu, K. Wang, L. Zheng, Z. Wen, Y. Feng, M. Zhu, and W. Chen, "AgentCoord: Visually exploring coordination strategy for llm-based multi-agent collaboration," *arXiv preprint arXiv:2404.11943*, 2024.

[167] Z. Zhou, J. Song, K. Yao, Z. Shu, and L. Ma, "ISR-LLM: Iterative self-refined large language model for long-horizon sequential task planning," *arXiv preprint arXiv:2308.13724*, 2023.

[168] Y. Liu, L. Palmieri, S. Koch, I. Georgievski, and M. Aiello, "DELTA: Decomposed efficient long-term robot task planning using large language models," *arXiv preprint arXiv:2404.03275*, 2024.

[169] Z. Yang, A. Ishay, and J. Lee, "Coupling large language models with logic programming for robust and general reasoning from text," *arXiv preprint arXiv:2307.07696*, 2023.

[170] G. Chalvatzaki, A. Younes, D. Nandha, A. T. Le, L. F. R. Ribeiro, and I. Gurevych, "Learning to reason over scene graphs: A case study of finetuning GPT-2 into a robot language model for grounded task planning," *Frontiers in Robotics and AI*, 2023.

[171] D. Han, T. McInroe, A. Jelley, S. V. Albrecht, P. Bell, and A. Storkey, "LLM-Personalize: Aligning LLM planners with human preferences via reinforced self-training for housekeeping robots," *arXiv preprint arXiv:2404.14285*, 2024.

[172] B. Y. Lin, Y. Fu, K. Yang, F. Brahman, S. Huang, C. Bhagavatula, P. Ammanabrolu, Y. Choi, and X. Ren, "SwiftSage: A generative agent with fast and slow thinking for complex interactive tasks," *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023.

[173] N. Wake, A. Kanehira, K. Sasabuchi, J. Takamatsu, and K. Ikeuchi, "ChatGPT empowered long-step robot control in various environments: A case application," *IEEE Access*, 2023.

[174] S. H. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor, "ChatGPT for robotics: Design principles and model abilities," *IEEE Access*, 2024.

[175] S. S. Raman, V. Cohen, E. Rosen, I. Idrees, D. Paulius, and S. Tellex, "Planning with large language models via corrective re-prompting," *Proc. of NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.

[176] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar *et al.*, "Inner Monologue: Embodied reasoning through planning with language models," *arXiv preprint arXiv:2207.05608*, 2022.

[177] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as Policies: Language model programs for embodied control," *Proc. of 2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 9493-9500, 2023.

[178] A. Jiao, T. P. Patel, S. Khurana, A.-M. Korol, L. Brunke, V. K. Adajania, U. Culha, S. Zhou, and A. P. Schoellig, "Swarm-GPT: Combining large language models with safe motion planning for robot choreography design," *arXiv preprint arXiv:2312.01059*, 2023.

[179] Z. Mandi, S. Jain, and S. Song, "RoCo: Dialectic multi-robot collaboration with large language models," *arXiv preprint arXiv:2307.04738*, 2023.

[180] A. Bucker, L. Figueredo, S. Haddadin, A. Kapoor, S. Ma, S. Vemprala, and R. Bonatti, "LaTTe: Language trajectory transformer," *Proc. of 2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 7287-7294, 2023.

[181] S. Wang, M. Han, Z. Jiao, Z. Zhang, Y. N. Wu, S.-C. Zhu, and H. Liu, "LLM3: Large language model-based task and motion planning with motion failure reasoning," *arXiv preprint arXiv:2403.11552*, 2024.

[182] T. Xie, S. Zhao, C. H. Wu, Y. Liu, Q. Luo, V. Zhong, Y. Yang, and T. Yu, "TEXT2REWARD: Reward shaping with language models for reinforcement learning," *arXiv preprint arXiv:2309.11489*, 2023.

[183] D. M. Proux, C. Roux, M. Niemaz *et al.*, "LARG2, language-based automatic reward and goal generation," 2023.

[184] J. Song, Z. Zhou, J. Liu, C. Fang, Z. Shu, and L. Ma, "Self-refined large language model as automated reward function designer for deep reinforcement learning in robotics," *arXiv preprint arXiv:2309.06687*, 2023.

[185] Y. Tang, W. Yu, J. Tan, H. Zen, A. Faust, and T. Harada, "SayTap: Language to quadrupedal locomotion," *arXiv preprint arXiv:2306.07580*, 2023.

[186] J. Y. Zhu, C. G. Cano, D. V. Bermudez, and M. Drozdzal, "InCoRo: In-context learning for robotics control with feedback loops," 2024.

[187] Y. Cao and C. G. Lee, "Ground manipulator primitive tasks to executable actions using large language models," *Proc. of the AAAI Symposium Series*, vol. 2, no. 1, pp. 502-507, 2023.

[188] H. He, C. Bai, K. Xu, Z. Yang, W. Zhang, D. Wang, B. Zhao, and X. Li, "Diffusion model is an effective planner and data synthesizer for multi-task reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[189] J. Chang, H. Ryu, J. Kim, S. Yoo, J. Seo, N. Prakash, J. Choi, and R. Horowitz, "Denoising heat-inspired diffusion with insulators for collision free motion planning," *arXiv preprint arXiv:2310.12609*, 2023.

[190] H. Ryu, J. Kim, J. Chang, H. S. Ahn, J. Seo, T. Kim, J. Choi, and R. Horowitz, "Diffusion-EDFs: Bi-equivariant denoising generative modeling on se (3) for visual robotic manipulation," *arXiv preprint arXiv:2309.02685*, 2023.

[191] J. Urain, N. Funk, J. Peters, and G. Chalvatzaki, "SE(3)-DiffusionFields: Learning smooth cost functions for joint grasp and motion optimization through diffusion," *Proc. of 2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 5923-5930, 2023.

[192] J. Carvalho, M. Baierl, J. Urain, and J. Peters, "Conditioned score-based models for learning collision-free trajectory generation," *Proc. of NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

[193] Z. Wu, S. Ye, M. Natarajan, and M. C. Gombolay, "Diffusion-reinforcement learning hierarchical motion planning in adversarial multi-agent games," *arXiv preprint arXiv:2403.10794*, 2024.

[194] C. Jiang, A. Cornman, C. Park, B. Sapp, Y. Zhou, D. Anguelov *et al.*, "MotionDiffuser: Controllable multi-agent motion prediction using diffusion," *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9644-9653, 2023.

[195] K. Saha, V. Mandadi, J. Reddy, A. Srikanth, A. Agarwal, B. Sen, A. Singh, and M. Krishna, "EDMP: Ensemble-of-costs-guided diffusion for motion planning," 2023.

[196] S. Zhou, Y. Du, S. Zhang, M. Xu, Y. Shen, W. Xiao, D.-Y. Yeung, and C. Gan, "Adaptive online replanning with diffusion models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[197] W. Liu, Y. Du, T. Hermans, S. Chernova, and C. Paxton, "StructDiffusion: Language-guided creation of physically-valid structures using unseen objects," *arXiv preprint arXiv:2211.04604*, 2022.

[198] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn, "RoboNet: Large-scale multi-robot learning," *arXiv preprint arXiv:1910.11215*, 2019.

[199] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine, "Bridge Data: Boosting generalization of robotic skills with cross-domain datasets," *arXiv preprint arXiv:2109.13396*, 2021.

[200] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du *et al.*, "BridgeData V2: A dataset for robot learning at scale," *Proc. of Conference on Robot Learning*, PMLR, pp. 1723-1736, 2023.

[201] H.-S. Fang, H. Fang, Z. Tang, J. Liu, J. Wang, H. Zhu, and C. Lu, "RH20T: A comprehensive robotic dataset for learning diverse skills in one-shot," *Proc. of RSS 2023 Workshop on Learning for Task and Motion Planning*, 2023.

[202] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine, "GNM: A general navigation model to drive any robot," *Proc. of 2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 7226-7233.

[203] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu *et al.*, "Ego4D: Around the world in 3,000 hours of egocentric video," *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995-19012, 2022.

[204] J. Slaney and S. Thiébaux, "Blocks world revisited," *Artificial Intelligence*, vol. 125, no. 1-2, pp. 119-153, 2001.

[205] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "ALFRED: A benchmark for interpreting grounded instructions for everyday tasks," *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10740-10749, 2020.

[206] S. James, Z. Ma, D. Rovick Arrojo, and A. J. Davison, "RLBench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, 2020.

[207] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani *et al.*, "Transporter Networks: Rearranging the visual world for robotic manipulation," *Proc. of Conference on Robot Learning*, PMLR, pp. 726-747, 2021.

[208] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7327-7334, 2022.

[209] J. Panerati, H. Zheng, S. Zhou, J. Xu, A. Prorok, and A. P. Schoellig, "Learning to fly—a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control," *Proc. of 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 7512-7519, 2021.

[210] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun *et al.*, "BEHAVIOR-1K: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation," *Proc. of Conference on Robot Learning*, PMLR, pp. 80-93, 2023.

[211] Z. Mandi, H. Bharadhwaj, V. Moens, S. Song, A. Rajeswaran, and V. Kumar, "CACTI: A framework for scalable multi-task multi-scene visual imitation learning," *arXiv preprint arXiv:2212.05711*, 2022.

[212] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684-10695, 2022.

[213] Z. Chen, S. Kiami, A. Gupta, and V. Kumar, "GenAug: Retargeting behaviors to unseen situations via generative augmentation," *arXiv preprint arXiv:2302.06671*, 2023.

[214] T. Yu, T. Xiao, A. Stone, J. Tompson, A. Brohan, S. Wang, J. Singh, C. Tan, J. Peralta, B. Ichter *et al.*, "Scaling robot learning with semantically imagined experience," *arXiv preprint arXiv:2302.11550*, 2023.

[215] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36479-36494, 2022.

[216] K. Black, M. Nakamoto, P. Atreya, H. Walke, C. Finn, A. Kumar, and S. Levine, "Zero-shot robotic manipulation with pretrained image-editing diffusion models," *arXiv preprint arXiv:2310.10639*, 2023.

[217] T. Xiao, H. Chan, P. Sermanet, A. Wahid, A. Brohan, K. Hausman, S. Levine, and J. Tompson, "Robotic skill acquisition via instruction augmentation with vision-language models," *arXiv preprint arXiv:2211.11736*, 2022.

[218] H. Ha, P. Florence, and S. Song, "Scaling Up and Distilling Down: Language-guided robot skill acquisition," *Proc. of Conference on Robot Learning*, PMLR, pp. 3766-3777, 2023.

[219] L. Wang, Y. Ling, Z. Yuan, M. Shridhar, C. Bao, Y. Qin, B. Wang, H. Xu, and X. Wang, "GenSim: Generating robotic simulation tasks via large language models," *arXiv preprint arXiv:2310.01361*, 2023.

[220] Y. Wang, Z. Xian, F. Chen, T.-H. Wang, Y. Wang, K. Fragkiadaki, Z. Erickson, D. Held, and C. Gan, "RoboGen: Towards unleashing infinite data for automated robot learning via generative simulation," *arXiv preprint arXiv:2311.01455*, 2023.

[221] J. Gu, S. Kirmani, P. Wohlhart, Y. Lu, M. G. Arenas, K. Rao, W. Yu, C. Fu, K. Gopalakrishnan, Z. Xu *et al.*, "RT-Trajectory: Robotic task generalization via hindsight trajectory sketches," *arXiv preprint arXiv:2311.01977*, 2023.

[222] A. Z. Ren, A. Dixit, A. Bodrova, S. Singh, S. Tu, N. Brown, P. Xu, L. Takayama, F. Xia, J. Varley *et al.*, "Robots that ask for help: Uncertainty alignment for large language model planners," *arXiv preprint arXiv:2307.01928*, 2023.

[223] C. Kassab, M. Mattamala, L. Zhang, and M. Fallon, "Language-EXtended Indoor SLAM (LEXIS): A versatile system for real-time visual scene understanding," *arXiv preprint arXiv:2309.15065*, 2023.

[224] Z. Liu, A. Bahety, and S. Song, "REFLECT: Summarizing robot experiences for failure explanation and correction," *arXiv preprint arXiv:2306.15724*, 2023.

[225] G. Tatiya, J. Francis, and J. Sinapov, "Cross-tool and cross-behavior perceptual knowledge transfer for grounded object recognition," *arXiv preprint arXiv:2303.04023*, 2023.

[226] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3M: A universal visual representation for robot manipulation," *arXiv preprint arXiv:2203.12601*, 2022.

[227] A. Z. Ren, B. Govil, T.-Y. Yang, K. R. Narasimhan, and A. Majumdar, "Leveraging language for accelerated learning of tool manipulation," *Proc. of Conference on Robot Learning*, PMLR, pp. 1531-1541, 2023.

[228] M. Shridhar, L. Manuelli, and D. Fox, "CLIPort: What and where pathways for robotic manipulation," *Proc. of Conference on robot learning*, PMLR, pp. 894-906, 2022.

[229] L.-H. Lin, Y. Cui, Y. Hao, F. Xia, and D. Sadigh, "Gesture-informed robot assistance via foundation models," *Proc. of 7th Annual Conference on Robot Learning*, 2023.

[230] R. Mirjalili, M. Krawez, and W. Burgard, "FM-Loc: Using foundation models for improved vision-based localization," *Proc. of 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 1381-1387, 2023.

[231] Y. Ze, G. Yan, Y.-H. Wu, A. Macaluso, Y. Ge, J. Ye, N. Hansen, L. E. Li, and X. Wang, "GNFactor: Multi-task real robot learning with generalizable neural feature fields," *Proc. of Conference on Robot Learning*, PMLR, pp. 284-301, 2023.

[232] K. Chu, X. Zhao, C. Weber, M. Li, W. Lu, and S. Wermter, "Large language models for orchestrating bimanual robots," *arXiv preprint arXiv:2404.02018*, 2024.

[233] X. Zhao, M. Li, C. Weber, M. B. Hafez, and S. Wermter, "Chat with the environment: Interactive multimodal perception using large language models," *Proc. of 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 3590-3596, 2023.

[234] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki, "Act3D: 3d feature field transformers for multi-task robotic manipulation," *Proc. of 7th Annual Conference on Robot Learning*, 2023.

[235] M. Gramopadhye and D. Szafir, "Generating executable action plans with environmentally-aware language models," *Proc. of 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 3568-3575, 2023.

[236] M. Hu, Y. Mu, X. Yu, M. Ding, S. Wu, W. Shao, Q. Chen, B. Wang, Y. Qiao, and P. Luo, "Tree-Planner: Efficient close-loop task planning with large language models," *arXiv preprint arXiv:2310.08582*, 2023.

[237] Z. Liu, H. Hu, S. Zhang, H. Guo, S. Ke, B. Liu, and Z. Wang, "Reason for future, act for now: A principled framework for autonomous LLM agents with provable sample efficiency," *arXiv preprint arXiv:2309.17382*, 2023.

[238] J. Yu, R. He, and R. Ying, "Thought Propagation: An analogical approach to complex reasoning with large language models," *arXiv preprint arXiv:2310.03965*, 2023.

[239] J. Brawer, K. Bishop, B. Hayes, and A. Roncone, "Towards a natural language interface for flexible multi-agent task assignment," *Proc. of the AAAI Symposium Series*, vol. 2, no. 1, pp. 167-171, 2023.

[240] T. T. Andersen, "Optimizing the universal robots ros driver." 2015.

[241] S. Haddadin, S. Parusel, L. Johannsmeier, S. Golz, S. Gabl, F. Walch, M. Sabaghian, C. Jähne, L. Hausperger, and S. Haddadin, "The Franka Emika robot: A reference platform for robotics research and education," *IEEE Robotics & Automation Magazine*, vol. 29, no. 2, pp. 46-64, 2022.

[242] F. Kaplan, "Everyday robotics: Robots as everyday objects," *Proc. of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-aware Services: Usages and Technologies*, pp. 59-64, 2005.

[243] U. Yamaguchi, F. Saito, K. Ikeda, and T. Yamamoto, "HSR, human support robot as research and development platform," *Proc. of The Abstracts of the international conference on advanced mechatronics: Toward evolutionary fusion of IT and mechatronics: ICAM 2015.6*, The Japan Society of Mechanical Engineers, pp. 39-40, 2015.

[244] G. Elias, M. Schuenck, Y. Negócio, J. Dias Jr, and S. M. Filho, "X-ARM: An asset representation model for component repository systems," *Proc. of the 2006 ACM symposium on Applied computing*, pp. 1690-1694, 2006.

[245] R. Amsters and P. Slaets, "Turtlebot 3 as a robotics education platform," *Proc. of Robotics in Education: Current Research and Innovations 10*, Springer, pp. 170-181, 2020.

[246] M. Kerzel, P. Allgeuer, E. Strahl, N. Frick, J.-G. Habekost, M. Eppe, and S. Wermter, "NICOL: A neuro-inspired collaborative semi-humanoid robot that bridges social interaction and reliable manipulation," *IEEE Access*, vol. 11, pp. 123531-123542, 2023.

[247] E. Rohmer, S. P. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," *Proc. of 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, pp. 1321-1326, 2013.

[248] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," *Proc. of 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, pp. 5026-5033, 2012.

[249] J. Liang, V. Makoviychuk, A. Handa, N. Chentanez, M. Macklin, and D. Fox, "Gpu-accelerated robotic simulation for distributed reinforcement learning," *Proc. of Conference on Robot Learning*, PMLR, pp. 270-282, 2018.

[250] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba, "VirtualHome: Simulating household activities via programs," *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8494-8502, 2018.

[251] X. Puig, T. Shu, S. Li, Z. Wang, J. B. Tenenbaum, S. Fidler, and A. Torralba, "Watch-And-Help: A challenge for social perception and human-AI collaboration," 2020.

[252] M. Shridhar, X. Yuan, M.-A. Côté, Y. Bisk, A. Trischler, and M. Hausknecht, "ALFWorld: Aligning text and embodied environments for interactive learning," *Proc. of the International Conference on Learning Representations (ICLR)*, 2021.

[253] M.-A. Côté, A. Kádár, X. Yuan, B. Kybartas, T. Barnes, E. Fine, J. Moore, R. Y. Tao, M. Hausknecht, L. E. Asri, M. Adada, W. Tay, and A. Trischler, "TextWorld: A learning environment for text-based games," *CoRR*, vol. abs/1806.11532, 2018.

[254] M. Deitke, W. Han, A. Herrasti, A. Kembhavi, E. Kolve, R. Mottaghi, J. Salvador, D. Schwenk, E. VanderBilt, M. Wallingford, L. Weihs, M. Yatskar, and A. Farhadi, "RoboTHOR: An open simulation-to-real embodied ai platform," *Proc. of CVPR*, 2020.

[255] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "AI2-THOR: An interactive 3d environment for visual AI," *arXiv preprint arXiv:1712.05474*, 2017.

[256] X. Puig, E. Undersander, A. Szot, M. D. Cote, R. Partsey, J. Yang, R. Desai, A. W. Clegg, M. Hlavac, T. Min, T. Gervet, V. Vondruš, V.-P. Berges, J. Turner, O. Maksymets, Z. Kira, M. Kalakrishnan, J. Malik, D. S. Chaplot, U. Jain, D. Batra, A. Rai, and R. Mottaghi, "Habitat 3.0: A co-habitat for humans, avatars and robots," 2023.

[257] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. Chaplot, O. Maksymets, A. Gokaslan, V. Vondrus, S. Dharur, F. Meier, W. Galuba, A. Chang, Z. Kira, V. Koltun, J. Malik, M. Savva, and D. Batra, "Habitat 2.0: Training home assistants to rearrange their habitat," *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[258] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A platform for embodied ai research," *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[259] M. Carroll, R. Shah, M. K. Ho, T. Griffiths, S. Seshia, P. Abbeel, and A. Dragan, "On the utility of learning about humans for human-ai coordination," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

Dae-Sung Jang received his B.S. and Ph.D. degrees in aerospace engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2008 and 2015, respectively. He is an Associate Professor with the Department of Aeronautical and Astronautical Engineering, Korea Aerospace University (KAU), Goyang, Korea. Before joining KAU in 2018, he worked at KAIST and then NASA Ames Research Center as a Postdoctoral Researcher. His research interests include multi-agent system decision making and task assignment/scheduling, combinatorial optimization and approximation algorithms, sensor system resource management, navigation and planning of autonomous robots, and cooperative estimation/control.

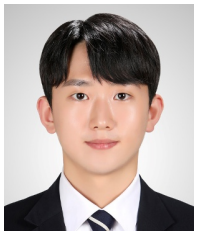
Doo-Hyun Cho received his B.S., M.S., and Ph.D. degrees in aerospace engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2013, 2015, and 2019, respectively. He is currently with D.Notitia, having joined the organization in 2023. Prior to this, he worked at Samsung Electronics DS and later at Stradvision as a research engineer. His research interests include foundation model-based multi-agent system decision-making, task assignment/scheduling, system resource optimization, and vision-related AI model optimization.

Woo-Cheol Lee received his B.S. degree in aerospace engineering from Korea Aerospace University (KAU), Goyang, Korea, in 2015. He received his M.S. and Ph.D. degrees in aerospace engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2017 and 2022, respectively. He is a Senior Researcher with the Extreme Robotics Team, Korea Atomic Energy Research Institute (KAERI), Daejeon, Korea. Before joining KAERI in 2023, he worked at Samsung Electronics and then Hyundai Motor Company as a Senior Robotics Researcher. His research interests include navigation, perception, and task management.

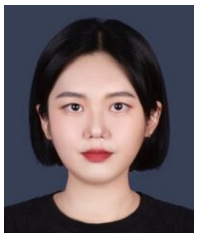
Seung-Keol Ryu is a post-master researcher at Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. He received his B.S. degree (double majors) in aerospace engineering and physics from KAIST in 2022. He received an M.S. degree in aerospace engineering from KAIST in 2024. He will begin his Ph.D. studies in aerospace engineering at the University of Colorado Boulder (CU Boulder) in fall 2024. His research interests include motion planning, aerial robotics, and diffusion models.

Byeongmin Jeong is currently a Ph.D. stundent of aerospace engineering at Korea Advanced Institute of Science and Technology (KAIST). He received his B.S. degree in mechanical engineering from Korea Aerospace University (KAU) in 2012. He received an M.S. degree in aerospace engineering from KAIST in 2014. His research interests inlcude multi-agent systems and path planning.

Minji Hong received her B.S. degree in aerospace engineering from Korea Aerospace University (KAU), Goyang, Korea, in 2022. She is a Ph.D. student of aerospace engineering at Korea Advanced Institute of Science and Technology (KAIST). Her research interests include multi agent path planning and task assignment.



Minjo Jung received his B.S. degree in mechanical engineering from Korea Aerospace University (KAU), Goyang, Korea, in 2023. He is an M.S. student of aerospace engineering at Korea Advanced Institute of Science and Technology (KAIST). His research interests inlcude task assignment, multi agent path planning and robot motion planning.



Minchae Kim received her B.S. degree in aerospace engineering from Inha University, Incheon, Korea, in 2023. She is an M.S. student of aerospace engineering at Korea Advanced Institute of Science and Technology (KAIST). Her research interests include decision-making under uncertainty, astrodynamics, and spacecraft autonomous control.

Minjoon Lee received his B.S. and M.S. degrees in aerospace engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2020 and 2022, respectively. He is a Researcher with Space Systems Team, Defense Agency for Technology and Quality. His research interests include task assignment/scheduling and spacecraft autonomous control.

SeungJae Lee received his B.S. and M.S. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2004 and 2006, respectively. He obtained his Ph.D. in electrical and computer engineering from the University of California Irvine, CA, USA, in 2016. He has held various engineering roles, including serving as a principal engineer at Samsung Electronics. He later worked as a computer vision/perception engineer at autonomous driving companies 42dot and Stradvision. Currently, he is with D.notitia, leading the AI team and focusing on research in physics-informed neural networks, large language models (LLM/VLM), and their applications to solving diverse real-world problems.

Han-Lim Choi is a Professor of aerospace engineering at Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. He received his B.S. and M.S. degrees in aerospace engineering from KAIST, in 2000 and 2002, respectively, and his Ph.D. degree in aeronautics and astronautics from Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2009. He then studied at MIT as a postdoctoral associate until he joined KAIST in 2010. His current research interests include decision-making for multi-agent systems, decision under uncertainty and learning, and intelligent aerospace systems.