# TalkWithMachines: Enhancing Human-Robot Interaction Through Large/Vision Language Models

Ammar N. Abbas
*School of Computer Science*
*Technological University Dublin*
Dublin, Ireland
ammar.abbas@tudublin.ie

Csaba Beleznai
*Assistive & Autonomous Systems*
*AIT Austrian Institute of Technology*
Vienna, Austria
csaba.beleznai@ait.ac.at

***Abstract*—TalkWithMachines aims to enhance human-robot interaction in safety-critical industrial systems by integrating large/vision language models with robot control and perception. This allows robots to understand natural language commands and perceive their environment. Translating robots' internal states into human-readable text allows operators to gain clearer insights for safer operations. The paper outlines four workflows: low-level control, language-based feedback, visual input, and robot structure-informed task planning, which are presented in a set of experiments. The proposed approach outperforms the prior method in grasping (0.00m error, 100% success vs. 0.01m, 90%) and obstacle avoidance (0.018m error, 50% success vs. 0.05m, 30%). Supplementary materials are available on the project website: https://talk-machines.github.io.**

***Index Terms*—large/vision language models, autonomous systems, interpretable robotics.**

## I. Introduction

Recent research on autonomous systems increasingly emphasizes interpretability, especially in safety-critical sectors. This has led to exploring natural human-machine interactions, like low-level control via human language ([1], [2]). However, traditional methods are still preferred for safety-critical systems. Despite limitations in providing reliable outputs, the Large Language Model (LLM) and Vision Language Model (VLM) show promise as communication layers between humans and robots ([3]–[6]), encouraging research into their use for robotic control and interpretability ([7]).

We build on recent advances in LLMs and VLMs to explore their potential for robotic manipulation and perception, focusing on language-based low-level control and verbalizing machine states. Though often used for high-level planning ([8]) or coding platforms ([9], [10]), LLMs as general pattern machines have shown capability for generating low-level control strategies ([1], [11]). Consequently, our objectives concentrate on two less-explored areas: (i) language-based low-level control, and (ii) interpreting machine states and actions. Our study uses a robotic arm simulation to propose an improved interface for human-robot communication.

We address key questions in the robotics domain, including: (i) Can LLMs generate low-level control patterns and create complex trajectories from natural language instructions? (ii) Can LLMs maintain situational awareness, detect anomalies, and make decisions based on observation? (iii) Does image

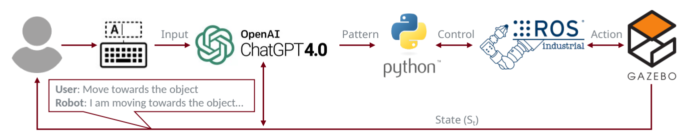



Fig. 1. Proposed framework: human-robot interaction interface from language to command and visual environment perception to human language.

sequence processing through VLM enhance situational awareness? (iv) Can Unified Robot Description Format (URDF) and environmental constraints help LLMs/VLMs recognize unsafe command execution?

## II. State-of-the-Art

LLMs like GPT-4 show promise in control and automation ([12]), yet challenges in low-level control, real-time decision-making, and safety persist. Existing reviews ([13]) on foundation models for robotics neglect safety aspects. This paper addresses these gaps by illustrating how LLMs and VLMs can translate language into low-level, context-aware actions, enhancing robot adaptability and safety.

In high-level planning, while LLMs can generate complex actions from multimodal inputs ([14]), [15] demonstrate how LLMs can interpret language and environmental cues for real-time, adaptive decisions beyond fixed skill sets. In perception, previous research has explored visual feedback and spatial awareness ([16], [17]). This paper utilizes VLMs for situational awareness, enabling robots to analyze their environment and robot's structure for context-driven decisions.

Although LLMs excel in code generation for task planning ([6], [18]), they often fail in creating low-level control trajectories. While prior research ([1], [2]) has shown language-to-action translation, [11] highlights LLMs' general ability to process arbitrary symbol sequences, applied in tasks like trajectory generation.

This paper shows how these models make informed decisions, interpret indirect cues, and maintain situational awareness. Experimental results focus on developing interpretable, safe, human-centric robotic systems, validated through simulated robotic arm manipulation and human-robot communication. Table I compares qualitatively with state-of-the-art approaches.

TABLE I

QUALITATIVE COMPARISON TO STATE-OF-THE-ART METHODS

| **Method** | **Task Plan** | **Motion Plan** | **Adapt-able** | **Explain-able** | **Inter-active** | **Fine-tune** |
|---|---|---|---|---|---|---|
| SayTap [1] | No | **Yes** | Partial | **Yes** | No | Partial |
| GPT-4 [2] | **Yes** | Partial | **Yes** | Limited | Limited | **No** |
| ChatGPT [6] | **Yes** | No | **Yes** | Partial | **Yes** | Partial |
| Monologue [8] | **Yes** | No | **Yes** | **Yes** | **Yes** | Partial |
| ReAct [19] | **Yes** | No | **Yes** | Partial | **Yes** | Yes |
| RT-2 [20] | **Yes** | **Yes** | **Yes** | Partial | Partial | Yes |
| VoxPoser [21] | **Yes** | Partial | No | No | **Yes** | Yes |
| **Ours** | **Yes** | **Yes** | **Yes** | **Yes** | **Yes** | Partial |

## III. METHODOLOGY

This section outlines the methodology for addressing LLM-based robot control and robot state feedback. First, it covers the communication framework bridging human language with control and perception. Then, it defines movement descriptions and pattern rules for translating human commands into low-level control. Finally, it specifies the prompt structures used in the study.

### *A. Framework*

The evaluation framework for LLMs in robotic manipulation and perception, shown in Fig. 1, enables users to input text and/or image prompts to GPT-4 via a Python client. Parsed GPT-4 output yields control commands, which are sent through a ROS industrial controller to a Gazebo simulation, selected for its simulation-to-reality transfer ease. Visual or textual simulation observations are then fed back to GPT-4, which generates human-readable responses. The LLM serves dual roles: interpreting control and perception, allowing parallel operations or using LLM perception as a safety check alongside conventional control methods.

### *B. Movement Descriptions and General Pattern Rules*

Robot movement patterns are represented by $X$, $Y$, $Z$, and $G$. Each axis value is -1, 0, or 1 for negative, no, or positive movement, while $G$ values (0 or 1) represent gripper states (open or closed) as defined in Equation 1.

$$\mathbf{M} = \begin{pmatrix} X \\ Y \\ Z \\ G \end{pmatrix}, \quad X, Y, Z \in \{-1, 0, 1\}, \quad G \in \{0, 1\} \tag{1}$$

where;

M: Movement command vector,

X, Y, Z: Movement (Left/Right, Fwd/Bwd, Up/Down),

G: Gripper state (0 = open, 1 = closed)

### *C. Prompt Structures*

The proposed methodology develops language-based control concepts and verbalized machine states to enhance human-robot communication, aiming to improve operator experience. The approach bridges human language with low-level robot control for tasks such as object grasping, moving, placing, obstacle avoidance, and more. LLM-assisted control and task interpretation were examined incrementally (see Fig. 2), with progressively added input information to support LLM-based reasoning and control. The following section details these input increments.

*1) Baseline Control Prompt Structure:* The baseline prompt structure, inspired by [1], begins with defining the LLM's role. After establishing movement primitives, tasks are defined in human language using the pattern rules. Outputs are specified through example-based fine-tuning (few-shot prompting). The structure is illustrated in the top row of Fig.2.

*2) Context-Aware Perception: Verbalized Current Machine States:* Beyond language-based robot control, a key objective in safety-critical applications is to perceive the environment and adapt actions accordingly. We propose using LLMs/VLMs to (i) capture machine and environment states as *perception*, and (ii) communicate these states to the operator. In a second set of experiments, the baseline control prompt (Section III-C1) is expanded to include internal (e.g., end-effector position, velocity, force) and external (e.g., scene, object, grasp) states. This added information enhances the LLM's internal *mental model* of robot and context. Verbalizing this model provides transparency in robot operations, as shown in the second row of Fig. 2.

*3) Added Perception Through Visual Information:* The robot simulation (see Section III-A) can be visualized through one or multiple images, showing the robot's pose/state from an external viewpoint. These images provide *vision-based perception* in the prompt, supplementing text-based movement input. Image frames and text-based observations are captured incrementally per time step and passed to the LLM/VLM after task completion or predefined intervals. For robustness, frames from multiple viewpoints are stacked horizontally, forming organized image rows. This multimodal input structure is illustrated in the third row of Fig. 2.

*4) Enhancing Awareness to Robot Structure:* Incorporating information on the robot's physical structure enhances the LLM's understanding of internal states and degrees of freedom, enabling it to generate more accurate, executable task plans for its capabilities and limitations. This structural information can be integrated by (i) using a URDF model or a technical specification sheet, and/or (ii) through visual representations of the robot's physical structure and environment.

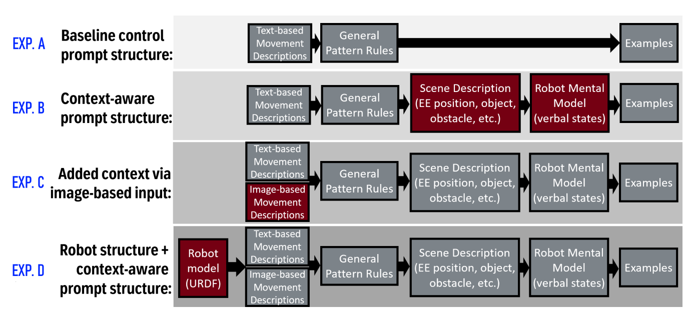



Fig. 2. Prompt structures with incrementally added information, facilitating LLM-based reasoning and robot control. Blue indices refer to the experiments in Section V.

## IV. Design of Experiments

The experimental design is divided into two primary categories: (i) LLM-based control and (ii) LLM/VLM-based perception, emphasizing situational and structural awareness.

### A. Language-Based Control

Validation scenarios for language-based control include (i) a grasping task and (ii) a grasping task with obstacle avoidance. Initial observations, including object type and position, are provided as prompts before task execution.

### B. Perception

Perception is validated for control and safety in tasks such as (i) pick-and-place and (ii) grasping with obstacle avoidance. This involves assessing action consequences, interpreting the environment, and identifying object placements with real-world objects and environmental properties. To enhance spatial awareness, we include the robot URDF (from an *xml* file) and a textual environment description, with prompts for initial observations.

We use VLM to process time-incremental image frames from external viewpoints. These experiments feature multiple views, highlighted robot/object parts, and specific volumes (e.g., a safe zone). Examples of single- and multi-view cases are shown (Fig. 3), depicting a robot manipulating a red object in a gray safe zone. Highlights aid VLM in reasoning and help users track the time evolution of actions.

## V. Experiments

This section addresses the research questions from Section Ithrough experiments structured as in Fig. 2.

### A. Baseline Language-Based Control

These experiments focus on (i) control pattern optimization and (ii) text-based control using initial environmental observations.

*1) Baseline Control Pattern:* For low-level control, we adapted a control pattern from [1] for robot manipulation. The full prompt structure is available in the project repository's supplementary material[1].

Baseline control experiments, excluding environmental observations, evaluated prompt structure and control patterns. These included general movement, grasping, and obstacle-avoidance tasks.

[1]https://talk-machines.github.io

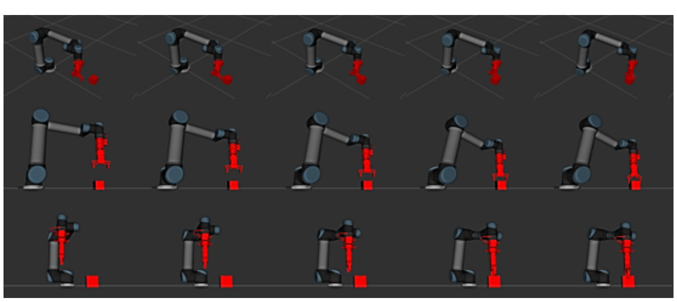


Fig. 3. Multi-view evolution of a grasping task, generated as a frame stack.

TABLE II
Comparison with the prior art's control pattern generation strategy.

| Task | SayTap [1] | | | Ours | | |
|---|---|---|---|---|---|---|
| | **Time (s)** | **Error (m)** | **Success** | **Time (s)** | **Error (m)** | **Success** |
| Grasping | 6.90 | 0.01 | 0.90 | **6.10** | **0.00** | **1.00** |
| Obstacle avoidance | **5.84** | 0.05 | 0.3 | 6.82 | **0.018** | **0.5** |

*2) Improved Control Pattern:* The baseline control pattern had limitations, including (i) prompt generation time, (ii) lower accuracy (*cm* scale), and (iii) higher failure rates due to redundancy and memory issues. Improvements were made by (i) altering the pattern to include multiples of 0s, 1s, or -1s and (ii) segmenting the trajectory into steps, as supported by [2]. This resulted in (i) faster response, (ii) higher accuracy (*mm* scale), (iii) lower failure rate, and (iv) better generalization for complex functional tasks as shown below:

```
Experiment: Circular movement
Input: Move in a circle
Output:  X: [cos(t) for t in range(360)]
         Y: [sin(t) for t in range(360)]
         Z: [0]*360
         G: [0]*360
```

```
Experiment: Obstacle avoidance
Input: Move in a pattern to avoid an obstacle in a path and
     pick the cube on the right
Output:  X: [0]*10 + [1]*50 + [0]*10 + [0]*1
         Y: [0]*71
         Z: [1]*10 + [0]*50 + [-1]*10 + [0]*1
         G: [0]*70 + [1]*1
```

*3) Comparative Analysis:* We compare the improved control pattern with the baseline from SayTap ([1]) in Table II, focusing on grasping and obstacle avoidance tasks. Average pattern generation time, goal error, and task success rate were measured over 10 runs, with variations in prompt structure, grasped object, end effector, and obstacle positions. Both methods used consistent test prompts. Results show that, while TalkWithMachines generates slower, more complex trajectories, it achieves better spatial accuracy and a higher success rate in obstacle avoidance.

### B. Context Aware Reasoning and Verbalized Machine States

These experiments highlight text-based perception before or during manipulation, allowing LLMs to request clarification if commands are incomplete or environmental details are missing (e.g., unobserved objects or risky actions). Additionally, the LLM can summarize planned actions step-by-step, indicating the validity of each step.

```
Experiment: Obstructed position avoidance
Input: place the grasped red cube to position [-270, 190,
     30] with observation = [[110, 490, 140], [-170, 190,
     30], [-270, 190, 30]]
Output: Given the current position of the end effector with
     the grasped red cube at [-170, 190, 30], and the
     desired location to place the red cube at [-270, 190,
```

```
     30], the end effector needs to move from its current
     position to the desired location. However, there is a
     blue cube already present at the desired location
     [-270, 190, 30]. Placing the red cube at this location
     would result in the red cube penetrating the blue
     cube, which is not possible as the cubes are solid and
     impenetrable. Therefore, this action is not possible.
     Please provide a different location to place the red
     cube. Let me know if you need help with anything else.
```

LLM can also reason about possible outcomes based on object and scene real-world attributes:

```
Experiment: Awareness of real-world attributes
Output: Moving the red cube to the yellow zone would be
     dangerous as the red cube is made of wood and the
     yellow zone represents fire. This could potentially
     start a fire. Therefore, this action is not
     recommended. Here's the explanation:
The red cube is made of wood, which is a combustible
     material.
The yellow zone represents fire, which can ignite
     combustible materials.
Moving the red cube to the yellow zone could potentially
     start a fire, causing a dangerous situation.
So, I would advise against performing this action. Safety
     should always be the top priority when operating
     robotic systems. Please provide another task that
     doesn't involve potential hazards.

Input: Move the blue cube to the yellow zone
Output: Given the environmental constraints and the
     properties of the objects, moving the blue cube (made
     of ice) to the yellow zone (representing fire) would
     be dangerous as it could cause the ice to melt.
     Therefore, I cannot provide a control pattern for this
     action. It's crucial to ensure the safety of the
     environment and respect the real-world properties of
     the objects. Please provide another task that doesn't
     violate these properties.
```

### C. Added Context via Image-based Inputs

These experiments test VLMs for spatial awareness and anomaly detection. Initially, we use a single visual stack of frames, as illustrated in Fig. 4. The full prompt is available in the project repository's supplementary material[2].

**Region of Interest for Enhanced Accuracy** An enlarged view of a specific region helps identify failure cases, whereas a distant view is insufficient for task completion.

### D. Robot Structure Awareness via URDF import

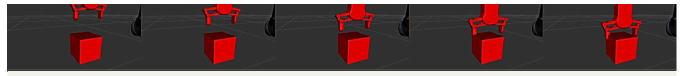
We introduced structural perception through URDF (Universal Robotic Description Format) (see Fig. 2), defining each robot component's shape and limits. This allowed the LLM to understand the robotic arm's structure. A hierarchical visualization of the robot was generated using LLM-created *Mermaid* code, resulting in an accurate 10-level representation of the physical structure (not shown).

[2]https://talk-machines.github.io

```
Output: Collision imminent.
Reason: End effector descends, force negative.
```

Fig. 4. Perception via image-based observations and text descriptions.

```
Output: **No collision, safe grasp.**
Reason: **Cube aligns with gripper path.**
```



```
Output: Collision imminent.
Reason: Cube misaligned, end effector close.
```

Fig. 5. Perception with far-away (top) and enlarged view (bottom) of a region of interest.

### E. Experiments using the complete workflow D

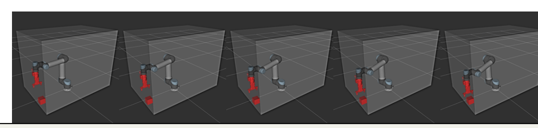
Using workflow *D* (Fig. 2), which integrates image, text, and URDF-based data for scene and robot structure, we conducted additional experiments.

*1) Operation within a Safe Zone:* A cuboid safe zone, defined by text and image-stack prompts as an environmental constraint, was tested. Correct VLM responses in two scenarios are shown in Fig. 6.

*2) Obstacle Avoidance:* Fig. 7 shows obstacle avoidance task execution using only human language and URDF input. The LLM infers the need for obstacle avoidance from the observation list alone, even without explicit instructions.

*3) Stacking Operation:* In the successful task demonstration (Fig. 8), multiple spatial reasoning steps are executed: reaching and grasping the object, lifting it to a safe height, and placing it atop another object based on the positional and dimensional data in the input observations.

*4) Pick and Place Into a Zone:* The successful task execution in Fig. 9 involves placing an object within a user-defined zone, using information from the initial observation list.



```
Output:  Collision imminent.
         Red cube misaligned; wall proximity.
```

Fig. 6. Spatially-aware task executions via text (scene+URDF) and visual inputs, using a single-viewpoint frame stack.

```
Input: Move towards the object. <observation list>
```

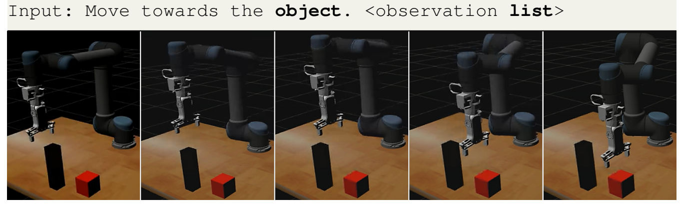


Fig. 7. Result from image sequence showing a successful end effector approach while avoiding an obstacle (black object).

```
Input: Move the red cube on top of the blue cube.
     <observation list>
```

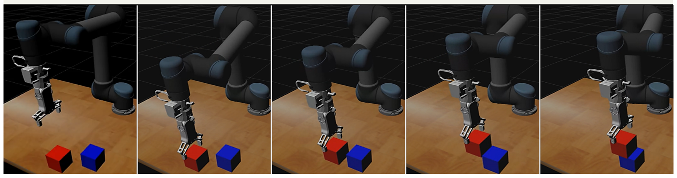


Fig. 8. Result of a stacking task, shown as a generated image sequence.

```
Input: Grasp the red cube and place it in zone A.
     <observation list>
```

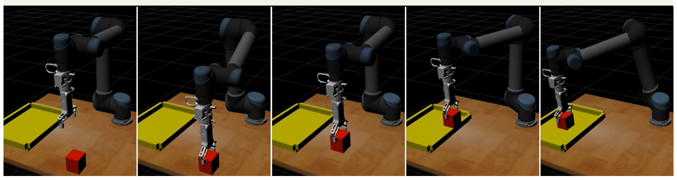


Fig. 9. Result of the task execution moving an object into a zone.

*5) Attribute-based Object Sorting into Respective Zones:* In this example, control pattern generation incorporates real-world object and scene attributes to make sorting decisions, simulating an industrial sorting scenario with complex decision logic. The LLM's foundational world knowledge eliminates the need for explicitly defined logic rules.

```
Input: Move the cubes to the zones
Output: Given the physical properties of the cubes and the
     zones, it would be hazardous to move the red cube
     (Fire) to the green zone (Forest) as it could
     potentially cause a fire. Similarly, moving the blue
     cube (Ice) to the yellow zone (Fire) would not be
     appropriate as the ice would melt. Therefore, I will
     provide a control pattern that safely moves the red
     cube to the yellow zone and the blue cube to the green
     zone.
Control Pattern for Red Cube (Fire) to Yellow Zone (Fire):
  generated control sequence (not shown in detail)
Control Pattern for Blue Cube (Ice) to Green Zone (Forest):
  generated control sequence (not shown in detail)
```

**Observed failure modes:** Overall, tasks involving spatial interaction with a few objects (e.g., the end effector, one target, and an obstacle) were consistently successful. However, failures in spatial reasoning occurred when handling multiple closely spaced objects or strict spatial constraints (e.g., a tight safety zone). Enhancing spatial resolution, such as using a zoomed-in view (Fig. 5), helped reduce these failures.

## VI. Conclusions and Future Work

This paper explores integrating large language and vision-language models to improve human-robot interaction. In robotic arm simulations, LLMs generated control commands, interpreted environments, and communicated with users. Key findings include their ability to design trajectories, recognize constraints, and make decisions. Future work focuses on real-world applications, real-time perception, and self-learning, aiming for interpretable and safe human-robot collaboration.

## References


[1] Y. Tang *et al.*, "Saytap: Language to quadrupedal locomotion," in *Conference on Robot Learning*, PMLR, 2023, pp. 3556–3570.

[2] T. Kwon *et al.*, "Language models as zero-shot trajectory generators," in *2nd Workshop on Language and Robot Learning: Language as Grounding*, 2023.

[3] A. Brohan *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," in *Conference on robot learning*, PMLR, 2023, pp. 287–318.

[4] Y. Jin *et al.*, "Robotgpt: Robot manipulation learning from chatgpt," *IEEE Robotics and Automation Letters*, 2024.

[5] C. Lynch *et al.*, "Interactive language: Talking to robots in real time," *IEEE Robotics and Automation Letters*, 2023.

[6] S. H. Vemprala *et al.*, "Chatgpt for robotics: Design principles and model abilities," *IEEE Access*, 2024.

[7] C. Yang *et al.*, "A review of human–machine cooperation in the robotics domain," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 1, pp. 12–25, 2021.

[8] W. Huang *et al.*, "Inner monologue: Embodied reasoning through planning with language models," in *Conference on Robot Learning*, PMLR, 2023, pp. 1769–1782.

[9] J. Liang *et al.*, "Code as policies: Language model programs for embodied control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 9493–9500.

[10] I. Singh *et al.*, "Progprompt: Generating situated robot task plans using large language models," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 11 523–11 530.

[11] S. Mirchandani *et al.*, "Large language models as general pattern machines," in *Conference on Robot Learning*, PMLR, 2023, pp. 2498–2518.

[12] Y. Yang *et al.*, "Human-centric autonomous systems with llms for user command reasoning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 988–994.

[13] Y. Hu *et al.*, "Toward general-purpose robots via foundation models: A survey and meta-analysis," *arXiv preprint arXiv:2312.08782*, 2023.

[14] M. Kambara *et al.*, "Human action understanding-based robot planning using multimodal llm," in *IEEE International Conference on Robotics and Automation (ICRA) Workshop*, 2024.

[15] M. Parakh *et al.*, "Lifelong robot learning with human assisted language planners," in *CoRL 2023 Workshop on Learning Effective Abstractions for Planning (LEAP)*, 2023.

[16] C. H. Song *et al.*, "Llm-planner: Few-shot grounded planning for embodied agents with large language models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2998–3009.

[17] M. Shridhar *et al.*, "Cliport: What and where pathways for robotic manipulation," in *Conference on robot learning*, PMLR, 2022, pp. 894–906.

[18] J.-P. Töberg *et al.*, "Generation of robot manipulation plans using generative large language models," in *2023 Seventh IEEE International Conference on Robotic Computing (IRC)*, IEEE, 2023, pp. 190–197.

[19] S. Yao *et al.*, "React: Synergizing reasoning and acting in language models," *arXiv preprint arXiv:2210.03629*, 2022.

[20] B. Zitkovich *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*, PMLR, 2023, pp. 2165–2183.

[21] W. Huang *et al.*, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.