# SAGE: Bridging Semantic and Actionable Parts for Generalizable Manipulation of Articulated Objects

Haoran Geng*[1,2], Songlin Wei*[2,3], Congyue Deng[1], Bokui Shen[1], He Wang†[2,3], Leonidas Guibas†[1]
Department of Computer Science, Stanford University[1]
CFCS, School of Computer Science, Peking University[2]
Beijing Academy of Artificial Intelligence[3]
* Equal contribution. † Equal advising.
Corresponding author: hewang@pku.edu.cn

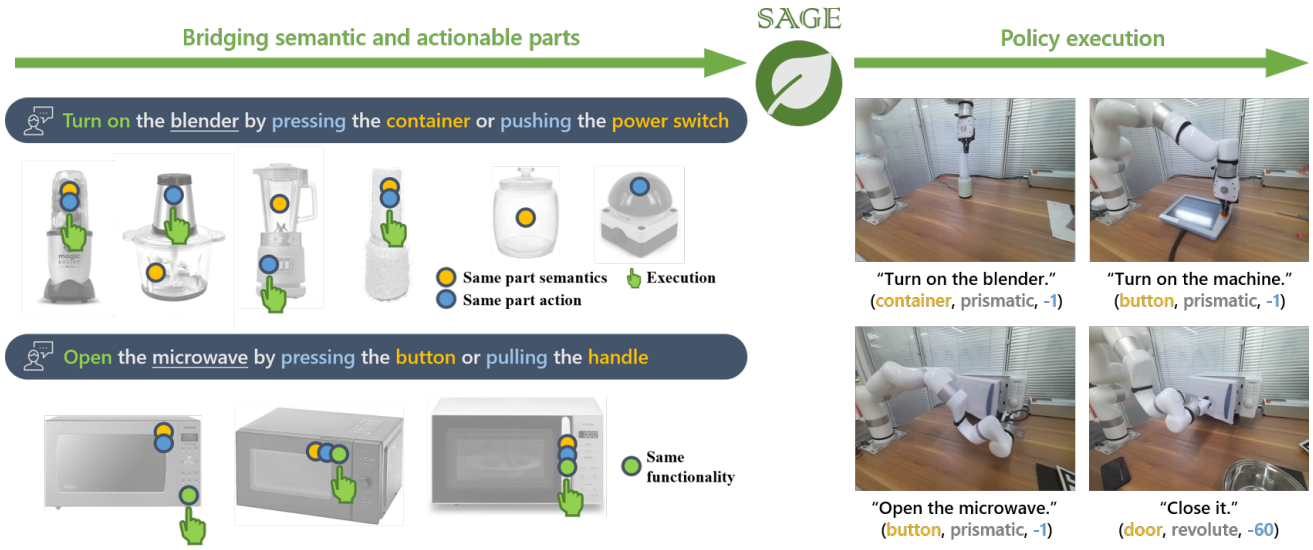



Fig. 1: **Overview.** We present SAGE, a framework bridging the understanding of semantic and actionable parts for generalizable manipulation of articulated objects. **Left:** We give two examples of human instructions illustrating the concept of part semantics, part functionalities, and corresponding actions. **Right:** examples of real-world tasks and our results. Our framework can tackle diverse manipulation tasks on various articulated objects.

***Abstract*—To interact with daily-life articulated objects of diverse structures and functionalities, understanding the object parts plays a central role in both user instruction comprehension and task execution. However, the possible discordance between the semantic meaning and physics functionalities of the parts poses a challenge for designing a general system. To address this problem, we propose SAGE, a novel framework that bridges semantic and actionable parts of articulated objects to achieve generalizable manipulation under natural language instructions. More concretely, given an articulated object, we first observe all the semantic parts on it, conditioned on which an instruction interpreter proposes possible action programs that concretize the natural language instruction. Then, a part-grounding module maps the semantic parts into so-called Generalizable Actionable Parts (GAParts), which inherently carry information about part motion. End-effector trajectories are predicted on the GAParts, which, together with the action program, form an executable policy. Additionally, an interactive feedback module is incorporated to respond to failures, which closes the loop and increases the robustness of the overall framework. Key to the success of our framework is the joint proposal and knowledge fusion between a large vision-language model (VLM) and a small domain-specific model for both context comprehension and part perception, with the former providing general intuitions and the latter serving as expert facts. Both simulation and real-robot experiments show our effectiveness in handling a large variety of articulated objects with diverse language-instructed goals.**

## I. Introduction

From furniture to home appliances, articulated object prevails in our daily lives. While being long studied in the robotics literature, their diverse object structures, functionalities, and manipulation goals in real-world scenarios remain challenging for most robot systems. The design of a general system that accommodates such object diversities falls into the problem of generalizable object manipulation, which involves the transfer of manipulation skills across object shapes [36, 20], object categories[13, 14, 56, 51], or even tasks [60]. Specifically, a line of works has focused on learning generalizable manipulation skills of articulated objects [36, 20, 13, 14, 17, 15]. However, even these works are still limited to a few common object categories, leaving aside the problem of how to model the resemblance between seemingly irrelevant articulated objects, from a simple cabinet to a multifunctional blender with intricate mechanical structures and electricity-powered functionalities.

In this work, we approach this problem by separately modeling the cross-category part commonality in their semantics and actions. Taking the blender in Fig. 1 top left as an example, to mince the food ingredients in it, one needs to

press its container from the top to turn it on. The part you press on is semantically a "container", but considering your interaction with it, its action resembles a "button" which can be pressed and trigger other mechanical motions (such as the blade rotating). To exert the functionality of this object, both part semantics and actions should be well understood. This also fits into the two different types of affordances in Gibson's theory of distributed cognition [61]: physical affordance and cognitive affordance, which have different emphases in human cognition of interactable objects. Physical affordance focuses on physical structures such as the joint conditions and part motions of articulated objects; cognitive affordances, on the other hand, are provided by cultural conventions and involve a high-level semantic understanding of the object.

Following this inspiration, we introduce SAGE, a framework for generalizable manipulation of articulated objects under natural language instructions by bridging semantic and actionable parts (Fig. 1 right). Our key insight is that large Visual-Language Models (VLMs) possess general knowledge of part semantics, while small domain-specific models present higher accuracy in predicting part actions, which can serve as "expert facts". Different from prior works that separately assign VLMs and small models to different sub-tasks[24], we fuse their predictions in both context comprehension and part perception, which achieves a good balance of generality and exactness.

Concretely, given a manipulation goal specified by natural language and an RGBD image as visual observation, an instruction interpreter first translates the language instruction into programmatic action representation. These action programs are composed of so-called action units, which are represented by 3-tuples of object semantic parts, joint types, and state changes. We then convert the action programs defined on the object semantic parts into executable policies through a part grounding module that maps the semantic parts into so-called Generalizable Actionable Parts (GAParts), which is a cross-category definition of parts according to their actionabilities. From the detected GAParts, we can generate physically plausible part actions indicated by the pre-obtained action programs. Finally, we also introduce an interactive feedback module that actively responds to failed action steps and adjusts the overall policy accordingly, which enables the framework to act robustly under environmental ambiguities or failure.

We demonstrate the effectiveness of our method both in simulation environments and on real robots. Specifically, we also assess the crucial components of language-guided articulated-object manipulation: scene descriptions, part perceptions, and manipulation policies. Quantitatively, our method surpasses all baseline approaches. This success is attributed to our blend of a comprehensive general-purpose model and a precise domain specialist, endowing our method with both stellar performance and robust generalization capabilities. Qualitatively, we further illustrate the proficiency of our method through its performance on several demanding tasks.

To summarize, our key contributions are:

- We bridge the notion of semantic and actionable parts to model the commonality across articulated objects of highly diverse structures and functionalities.
- We build a robot system for generalizable manipulation of articulated objects under language instructions.
- We design a knowledge fusion mechanism between VLMs and small domain-specific models to incorporate expert facts into general perception and comprehension.
- We demonstrate our strong generalizability on a variety of different objects under diverse language instructions in both simulation environments and on real robots.

## II. Related Work

**Articulated-object manipulation**. Articulated object manipulation presents significant challenges due to diverse object geometries and physical characteristics. Although benchmarks by [49, 36] have been introduced, their scope remains limited. While methods like [39, 25, 4] have delved into motion planning, visual affordance learning has also gained attention [35, 54, 52, 62, 15]. Other works [7, 58] have proposed special representations for manipulation, though they are tailored mainly for suction grippers. On the other hand, [13, 14] introduce generalizable parts as a novel representation but are limited to single-part interaction and can not handle natural instructions.

**Generalizable object manipulation**. Achieving generalization in robot applications is essential yet challenging. Various works [8, 47, 18, 12, 57] employ supervised learning with motion planning to tackle generalizable tasks such as grasping. Nevertheless, these techniques often necessitate task-specific architectures and may falter in intricate manipulations. Though reinforcement learning holds promise for complex challenges [40, 2], the lack of generalizability remains unresolved [30, 16]. The ManiSkill benchmark [36] pioneers category-level object manipulation in simulations, while [42] utilizes imitation learning with RL-trained demonstrations. However, in both cases, the generalization ability is still unclear. [13, 14] first tackles generalizable manipulation in a cross-category manner, but is still limited to predefined part classes. [56, 51] have universal generalization ability but only for grasping tasks. In summary, Generalizable object manipulation is still extremely challenging.

**Robot control with large language models**. Incorporating pre-trained language models into robot systems has been gaining tremendous attention recently. A notable segment of these works concentrates on the grounding of language to manipulation skills, essential for executing long-horizon instructions [26, 46, 34, 5, 41, 1, 17, 19, 43, 21, 59]. A hierarchical approach, seen in another subset of studies, hinges on a two-fold process: establishing a skill set and subsequently strategizing over it using large language models (LLMs) [3, 23, 45, 22, 32, 28]. However, this method is limited by the huge data collection effort, is still constrained to the predefined skill sets, and lacks visual understanding. In contrast, LLMs are employed to spawn code for low-level skills through APIs, though predicting these skills' continuous parameters remains challenging [31]. Some employ a way-point-based method

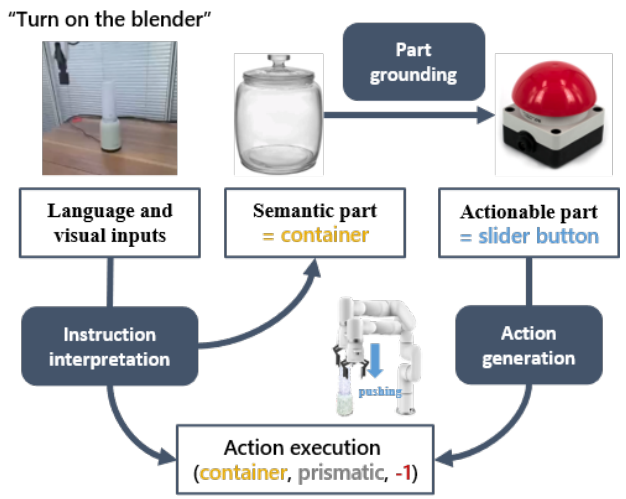



Fig. 2: **Framework overview.** Given a natural language input conditioned on visual observation, we first obtain the semantic parts and interpret the language instructions on them. Then we link the semantic parts to actionable parts through a part grounding module, on which executable actions can be predicted.

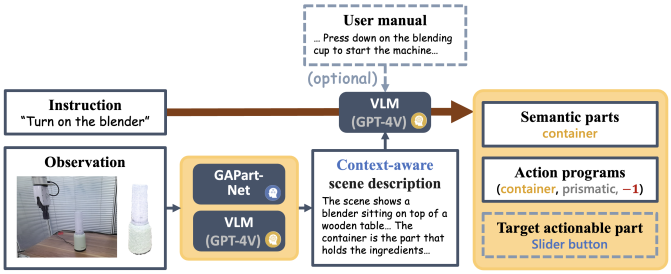



Fig. 3: **Context-aware instruction interpretation.** With instruction and observation (RGBD image) as input, the interpreter first generates a scene description with VLM and GAPartNet[13]. Then LLM(GPT-4) takes both instruction and scene description as input and generates semantic parts and action programs. Optionally, we can input a specific user manual, and LLM will generate a target actionable part; see Sec.IV-C for more details about the actionable part.

for efficiency, sidestepping low-level robot actions in favor of trajectory hand poses [44, 17]. But they also require lots of data and have no detailed planning strategy. VoxPoser [24] performs motion planning based on affordance and obstacle maps generated by LLM, but still lacks visual understanding. For articulated objects with diverse geometries, intricate structures, and physical constraints, language-guided manipulation remains a challenging and underexplored problem.

## III. Problem Formulation

### A. Semantic and Actionable Parts

As shown in Fig 1, we characterize a *semantic part* based on its human-defined identity or its functional role in articulated objects. During the interaction, for a *actionable part*, we adopt the concept of "Generalizable Actionable Part (GAPart)" as formulated by GAPartNet [13]. GAPart categorizes parts across different objects based on their actionability, where parts within the same GAPart category exhibit similar geometry and functionality. For instance, a button designed for pressing and a knob meant for rotation would not fall under the same GAPart category, even if they appear similar. Our approach emphasizes the importance of understanding parts at *both semantic and action* levels for generalized manipulation of articulated objects.

### B. Problem Definition and Framework Overview

**Problem definition.** Our task focuses on open-vocabulary manipulation of articulated objects. The input consists of a human language instruction and a single-view RGB-D image. A fixed base robot arm is required to follow human instructions and manipulate the articulated object within the scene to achieve the goal specified in the instructions.

**Framework overview (Fig. 2).** We propose SAGE, a comprehensive framework designed for open-vocabulary articulated object manipulation in the wild. By combining the strength of the general-purpose Visual Language Model (VLM) and a domain-specific 3D part perception and manipulation model, SAGE is uniquely capable of understanding object parts on both semantic and action levels, facilitating more effective and adaptable manipulation in real-world scenarios.

In the SAGE system, we employ the classic *perception-decision-execution-feedback* loop to generalizable articulate object manipulation: More specifically, (1) we first use GPT-4V to process the input RGB image to obtain a scene description that contains task-related information, such as *semantic parts*, part interaction possibilities, and object states. We further detect more *actionable parts* with GAPartNet and fuse both outputs to obtain the final part-aware scene description. (2) Then, we again prompt GPT-4V to comprehend the scene description together with the human instruction to act like a global planner. It outputs action programs step by step and makes decisions based on execution feedback. (3) The action programs are executed with a motion planner which controls the robot arm to follow a predefined trajectory grounded on the actionable part. (4) Finally, we interactively detect the part and object state changes and feedback on the updates to the global planner.

## IV. Method

### A. Part-aware Scene Perception

In daily-life scenarios, human language instructions could be diverse and even ambiguous. To facilitate large language models to truly understand human intention and plan accordingly, it is beneficial to add comprehensive scene descriptions to the prompt text inspired by the success of Chain-of-thought [53] prompting.

**Scene Description**. Chain-of-thought[53] is proven useful for prompting large models. Here, we also notice that, compared to outputting an action program (See Sec. IV-B) directly using a Visual-Language Model, it is better to first generate a scene description $\tilde{D}_{scene}$ containing task-related information (such as parts, object states, interaction possibilities). However, we observe that VLM provides rich context but lacks accuracy in part-related perception as illustrated in Fig. 4. This example

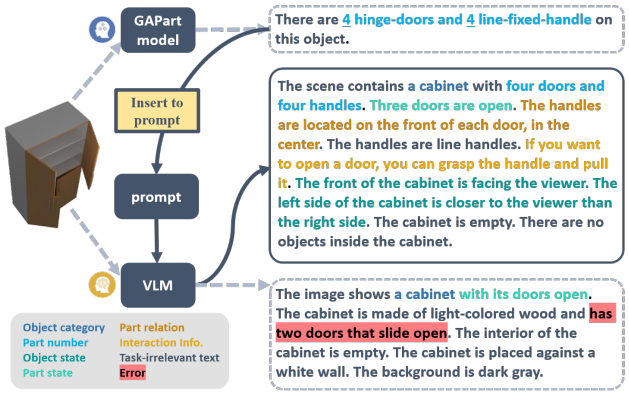



Fig. 4: **Scene description.** To better help with action generation, the scene description should contain object information, part information, and some interaction-related information. We use expert GAPart model[10] to generate some expert descriptions as part of the VLM prompt and then generate the scene descriptions, which works well and absorbs the advantages of both models.

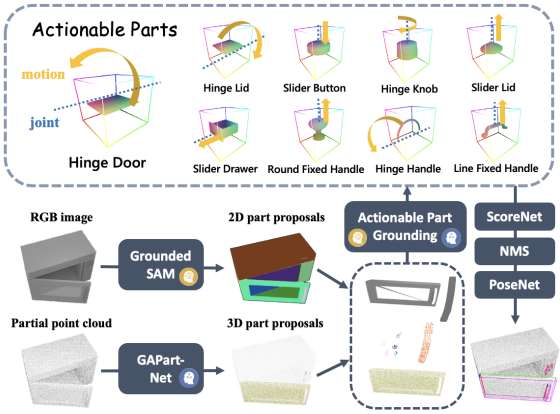



Fig. 5: **Part grounding.** With the observation as input, our method ensembles 2D proposals from GroundedSAM and 3D proposals from GAPartNet. then the proposals are fed to the ground as an actionable part. Leveraging ScoreNet, NMS, and PoseNet, we then present the perception results. Notice that: (1) For the part perception evaluation benchmark, we directly employ SAM[29]. However, within our manipulation pipeline, we utilize GroundedSAM, which also considers the semantic part as input. (2) As highlighted in Fig. 3, if the LLM produces a target actionable part, the grounding process is bypassed.

illustrates that employing the Visual-Language Model directly can result in the omission of crucial information related to parts and tasks, or may even introduce factual inaccuracies.

**Actionable Part Detection**. We deploy the 3D GAPart Model [13] to enhance part perception. This model interprets the partial RGBD point cloud $X$ to yield its set of 3D part proposals, which are masked point clouds denoted as $\mathcal{M}^{3D} = \{l_j, M_j^{3D}\}_{j=1}^{n}$, where $l$ is the actionable part label, $M$ is the masked partial point cloud, and $j$ indexes the detected actionable parts from 1 to $n$. We reframe the actionable parts with a template sentence as shown in Fig. 4, eg., *There are 4 hinge-door and 4 line-fixed-handle on the object*. The sentences that contain actionable parts are then appended to obtain the part-enhanced scene description $D_{scene}$. We empirically found this significantly boosts the performance of perception. More details are given in the experiments.

### B. Instruction Interpretation and Global Planner

Given the part-aware scene description $D_{scene}$, we here again employ GPT-4V [37] as the instruction interpreter to translate natural-language instruction $\tilde{I}$ into executable actions that will be sent to the downstream execution module. We devise a representation named *action unit* to encapsulate the output actions.

**Action Units**. We define the most basic manipulation on one single part of articulated objects as an "action unit", which can be represented by a 3-tuple $\boldsymbol{a} = (\tilde{p}, j, \Delta s)$ with part name $\tilde{p}$, joint type $j$, and the change of part state $\Delta s$. The part name $\tilde{p}$ is a noun in natural language. $j$ refers to the joint directly linked to part $\tilde{p}$ and indicates whether it is a revolute or a prismatic joint; $\Delta s$ is the state change of joint $j$ under the action, which is either an angle $\pm\theta$ if $j$ is revolute or a relative translation $\pm t$ *w.r.t.* the part bounding box if $j$ is prismatic. The final output of the interpreter should be in one of the following formats:

- a single action unit,
- an unordered union of multiple action units,
- an ordered list of multiple action units,

or be a finite combination of the three formats. Following the convention of programming languages (PL), this can be viewed as a typing system with expressions:

$$\boldsymbol{a} \mid \texttt{Union}\{\boldsymbol{a}, \boldsymbol{a}\} \mid \texttt{List}[\boldsymbol{a}, \boldsymbol{a}]$$

Here the `Union` expression is for non-deterministic policy generation when multiple action units can reach the same goal, e.g., both pulling the door and pressing a button result in the microwave door being opened. `List` is for sequential action generation when no single-step solution exists, e.g., to open a door, a knob must be first rotated to unlock the door.

**Global Planner**. Throughout the interaction, tracking of both the target gripper and part states is maintained by a global planner, where details are given in Algorithm 1. If significant discrepancies arise, the planner has the discretion to select from one of four states: "continue", "transition to the next step", "halt and replan", or "success". For instance, if the gripper is set to rotate 60 degrees along a joint, yet the door only opens to 15 degrees, the LLM planner would opt to "halt and re-plan". This interaction tracking model ensures that the LLM remains cognizant of developments during interactions, permitting strategy adjustments and, where necessary, recovery from unforeseen setbacks.

### C. Part Grounding and Execution

An execution module is implemented to act accordingly to the action unit outputted by the global planner. Like any language-guided manipulation task, we first need to ground the action onto the corresponding physical actionable part.

**Actionable part grounding**. To this end, We first collect a dataset of actionable part features and part labels $\{F_{\text{part}}^{\text{dino}}, l_{\text{part}}\}$. This dataset contains the pre-defined interaction policies of each typical GAPart. The policies are a series of pre-defined end-effector trajectories to complete a certain motion for a specific part. It serves as our domain-specific expert which is the key to the success of our generalizable system. In

particular, we extract a good image feature map $F^{\text{dino}}$ from DINOv2 [38]. Then, for the target actionable part denoted as $p_j$, we use its mask to get the part feature $F_j^{\text{dino}}$. We first do max-pooling to obtain the part feature and run KNN algorithm to find the grounded actionable part label denoted as $l_j$

$$F_j^{\text{dino}} = \text{MaxPooling}(F^{\text{dino}}[M_j])$$
$$l_j = \text{KNN}(\{F_{\text{part}}^{\text{dino}}, l_{\text{part}}\}, F_j^{\text{dino}}).$$

**Trajectory Generation**. Once we have grounded the semantic part to the actionable GAPart, we can generate executable manipulations on this part. We first estimate the part pose $P_j$ as defined in GAPartNet [13]. We also compute the joint state (part axis and position) and plausible motion direction based on the joint type (prismatic or revolute). Then we generate the part actions according to these estimations. We first predict an initial gripper pose as the primary action. It is followed by a motion induced from a pre-determined strategy defined in GAPartNet [13] according to the part poses and joint states. For example, to open a door with a revolute joint, the starting position can be on the door's rim or handle, with the trajectory being a circular arc oriented along the door's hinge.

### D. Interactive Feedback

**Algorithm 1** Global Planner

**Input** Gripper target and current state $\Delta g_t$, $\delta g_t$ and part target and estimated movement $\Delta s_t$, $s_t$.

$prompt$ = template$(\Delta g_t, \delta g_t, \Delta s_t, s_t)$
$result$ = call VLM with $prompt$
**switch** $result$ **do**
    **case** `"continue"`
        Continue the current strategy
    **case** `"transition to next step"`
        Finish the current action tuple and transition to the next action.
    **case** (`"halt and replan"`)
        Stop the current execution and replan ▷ Some errors happened.
    **case** (`"success"`)
        Success and terminate.

**Algorithm 2** Interactive perception

**Input** initial and current part point cloud $X_{\text{part}}^0, X_{\text{part}}^t$
**Output** joint axis $j_t$ and movement $s_t$

$t_t, R_t = \text{RANSAC}(\text{Umeyama}(X_{\text{part}}^0, X_{\text{part}}^t))$
$j_t, s_t = \text{GeometryInference}(R_t, t_t)$

Up to now, we have only utilized a single initial observation $I_{rgbd}^0$ for generating open-loop interactions. We now introduce a mechanism to further leverage the observations acquired during the interaction process, which can update the perception results and adjust the manipulations accordingly. Toward this goal, we introduce a two-section feedback mechanism during interaction.

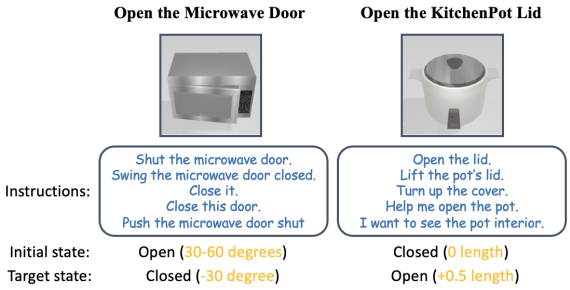



Fig. 6: **Task examples for experiments in simulation.**

**Interactive perception.** It should be noted that occlusion and estimation errors may arise during the perception of the first observation. To address these challenges, we propose a model that leverages interactive observations to enhance operation (see Algo. 2). Consider two distinct observations, $I_{rgbd}^0$ and $I_{rgbd}^t$, captured during an interaction. A part's motion can be detected between these two observations. We subsequently employ the mask of the moved part to deduce the actual joint state during the interaction. Applying RANSAC [9] for outlier removal and Umeyama algorithm [48], we estimate the rotation and translation between the two frames of the part's point cloud $X_{\text{part}}^0, X_{\text{part}}^t$. This computation subsequently allows us to ascertain the joint state $j_t$ and the current part movement states $s_t$. Finally, we send the joint state and part movement state back to the global planner (see Sec. IV-B, thereby completing the main loop.

## V. Experiments

### A. Simulation Experiments

**Setup**. We conducted our simulations using the SAPIEN environment [55] and designed 12 language-guided articulated object manipulation tasks. An example task is given in Fig. 6. A comprehensive breakdown of these tasks, along with detailed statistics, is presented in Table I II. For each category of *Microwave*, *StorageFurniture* and *Cabinate*, we devised 3 tasks including opening from the initial open and close state and closing from the initial open state. The remaining tasks for *KitchenPot*, *Remote* and *Blender* are *Open the lid*, *Press the button* and *Turn on the blender* respectively. For each task, we carefully curated more than 5 distinct objects sourced from the GAPartNet dataset [13] that were well-suited for the respective actions. For example, we selected 5 different models of *Microwaves* from GAPartNet, chosen specifically for tasks like *pulling the door open* and *pushing the door closed*. Likewise, we chose 20 different *StorageFurniture* objects and 10 *Blenders*, each tailored to their corresponding tasks. To assess the robustness of our part interaction module, we conducted over 20 trials for each task. During each trial, we introduced randomization in both camera position and initial joint states, ensuring the variability of scenarios. Specifically, for the task *Close the door*, the initial positions of the articulated object doors were randomized within the range of $(30, 60)$ degrees. To facilitate fair comparisons, we leveraged

| Category | Microwave | | | StorageFurniture | | | Cabinet | | | KitchenPot | Remote | Blender |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Task ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| VoxPoser*[24] | - | - | 13.0 | - | - | 15.0 | - | - | 14.2 | - | - | - |
| GAPartNet*[13] | 87.5 | 75.5 | 58.6 | 53.3 | 76.9 | 81.3 | 50.0 | 66.3 | 73.8 | - | - | - |
| Ours | **98.0** | **96.0** | **94.1** | **83.3** | **80.0** | **95.0** | **79.3** | **89.6** | **83.1** | **85.7** | **60.7** | **42.9** |

TABLE I: **Success rates (%) under language instructions.** Comparison between our method with VoxPoser [24] and GAPartNet[13]. We evaluate 12 tasks with 6 different articulated objects. "-" means the task is not implemented by the baseline. GAPartNet* shares the same execution policy as our method, while VoxPoser* is adapted from the unofficial version of the authors.

| Category | #Task | #Init. state | #Tgt. state | #Instruct. |
|---|---|---|---|---|
| Microwave | 100 | 3 | 2 | 5 |
| StorageFurniture | 40 | 3 | 2 | 5 |
| Cabinet | 40 | 3 | 2 | 5 |
| KitchenPot | 20 | 1 | 1 | 5 |
| Remote | 18 | 1 | 1 | 5 |
| Blender | 20 | 1 | 1 | 5 |

TABLE II: **Benchmark statistics.** For each object category, we created tasks that were randomly initialized.

| | | |
|---|---|---|
| Perception | Scene Description | <1s |
| | Part Detection | 13s |
| Decision | Instruction Interpretation | <10s |
| | Global Planninng | <1s |
| Execution | Part Grounding | <5s |
| | Motion Planning | 45s |
| Feedback | Interactive Perception | <1s |

TABLE III: **Average runtime of each module.**

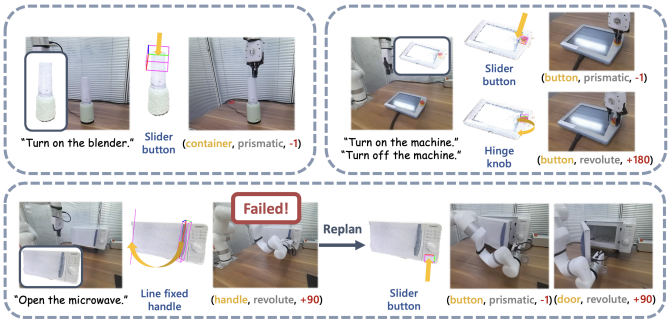



Fig. 7: **Explanations of real-world results. Top:** "turn on the blender" and "turn on/off the machine". Our method accurately understands the part semantics and actions that are not aligned. **Bottom:** "Open the microwave". The mechanical structures of the microwave prevent the robot from directly pulling open the door but instead require the button to be pressed, resulting in a failure. However, our interactive feedback model can detect the failure and recognize that it should try pressing the button instead and subsequently complete the task.

| MOS ↑ | GAPartNet | VLM (Bard) | **Ours** (Bard) | VLM (GPT4V) | **Ours** (GPT4V) |
|---|---|---|---|---|---|
| Part description | 3.8 | 4.1 | 6.3 | 7.2 | **9.6** |
| Part accuracy | 8.7 | 4.0 | 6.5 | 6.9 | **9.5** |
| Part state precision | 0.0 | 3.7 | 5.6 | 7.7 | **7.8** |
| Object & scene descr. | 0.0 | 2.7 | 6.2 | 6.4 | **8.0** |
| Interaction info. | 0.3 | 7.3 | 7.6 | 7.6 | **8.7** |
| Overall performance | 2.6 | 4.4 | 6.4 | 7.2 | **8.7** |

TABLE IV: **User study results for scene description.** We solicited volunteers to assess the quality of scene descriptions. Participants were tasked with evaluating the following aspects: (1) Performance in the overall part description. (2) Accuracy in the enumeration and naming of parts. (3) Precision in part-state depiction. (4) Depiction of objects and the overall scene. (5) Description of interaction-related information. (6) Overall performance.

pre-trained weights from GAPartNet to identify actionable parts within the scene. Subsequently, we applied the same motion generation policy as our pipeline.

**Results**. The results of our experiments are summarized in Table I, showcasing the superior performance of our methods across nearly all tasks. The key drivers of our success can be attributed to our enhanced perception capabilities, which benefit from the fusion of the specialized model from GAPartNet and the generalist model of LLM. The average runtime breakdown of each module in the SAGE system is shown in Table. III.

### *B. Real-Robot Experiments*

**Setup**. In our real-world experiments, we establish an experimental setup with the UFACTORY xArm 6 and several different articulated objects for manipulation.

**Results**. Fig. 7 shows our results on real-robot execution, and more results can be found in the supplementary materials. Three challenging cases are highlighted in Fig. 7 with detailed explanations and intermediate outputs. The top left is a blender whose top part is perceived as a container for containing juices but functions as a button to be pressed down. Our framework effectively links its semantic and action understandings and successfully executes the task. The top right shows an emergency stop button for robots, which requires a press (down) to halt an operation and a rotation (up) to restart it. With the auxiliary input of the user manual, our method completes both of these two tasks. The bottom shows a rather challenging case where the microwave door cannot be directly pulled open. Instead, it requires first pressing the button to initiate a slight door opening. In this case, our method first tries the most straightforward solution of pulling the door and fails. The interactive feedback module then detects this failure and informs the global planner to replan a second strategy that adapts to the current environment, finally completing the task.

### *C. Generalizable Visual Perception*

As described in Sec. IV, we design knowledge fusion mechanisms to join the force VLMs and small domain-specific models both in our context comprehension and part perception. In this section, we evaluate our intermediate results on scene



(a) Nudge the microwave's door open



(b) Open the door, please



(c) Lift the pod's lid



(d) Open the pod



(e) Turn on the blender



(f) Pull the top drawer



(g) Open the microwave door

Fig. 8: **Real-robot results.** We show the keyframes of various tasks in our experiments. More results can be found in the supplementary materials.

| Part Perception Accuracy | In-distribution | | Unseen states | | Unseen objects | | Unseen categories | |
|---|---|---|---|---|---|---|---|---|
| | AP@50 | mAP | AP@50 | mAP | AP@50 | mAP | AP@50 | mAP |
| PointGroup[27] | 69.70 | 60.58 | 69.54 | 60.48 | 58.26 | 46.29 | 24.57 | 19.40 |
| SoftGroup[50] | 69.59 | 60.54 | 70.02 | 59.59 | 59.20 | 47.10 | 28.18 | 22.50 |
| AutoGPart[33] | 66.81 | 57.63 | 67.60 | 56.69 | 55.30 | 43.50 | 26.24 | 20.38 |
| GAPartNet[13] | 81.42 | **72.55** | 80.80 | 71.73 | 63.18 | 53.94 | 36.39 | 27.40 |
| PartGroundedSAM [29, 38] | 73.73 | 61.75 | 72.36 | 62.02 | 66.25 | 38.64 | 41.59 | 28.45 |
| **Ours** | **83.04** | 72.23 | **82.39** | **71.91** | **72.17** | **58.04** | **47.69** | **34.57** |

TABLE V: **Part perception results.** We have curated a novel evaluation dataset for part perception, enriched with more comprehensive data. Our method is benchmarked against 3D point cloud-based techniques, including PointGroup[27], SoftGroup[50], AutoGPart[33], and GAPartNet[13]. In alignment with the 2D branch of our approach, we also employ SAM[29] and DINOv2[38] to establish a 2D-centric baseline. For evaluation, we adopt AP@50 and mAP as our primary metrics.

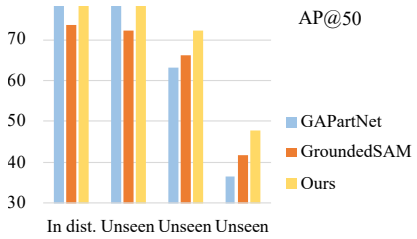



Fig. 9: **Part perception results.** Our method consistently achieved the highest AP@50 score for each level of generalization.

description (Sec. IV-A) and part perception (Sec. IV-C) and show that our generalizable perception modules achieve a better balance between generalization and exactness compared to existing methods or its ablated versions.

**Scene description (Sec. IV-A)**. In our instruction interpreter, a scene description is generated from the visual observations to inform the other modules of the scene context. To evaluate the quality of scene descriptions generated by our method which joins the forces of generalist and specialist models, we curate an image dataset specifically for evaluating scene descriptions for manipulation purposes. The dataset includes 145 images from 15 object categories with varying part states. We also designed six interaction-oriented metrics for assessing the qualities of the descriptions: part description, part accuracy, part state precision, object&scene description, interaction-related information, and overall performance. With these metrics, we can collect the Mean Opinion Score (MOS) through carefully designed user studies, as in [11].

We evaluate the scene descriptions generated by five methods: (1) GAPart Model: detection results are translated into natural language descriptions through a human-crafted format; (2)(3) VLM (Bard, GPT-4V), craft descriptions from the input image using Bard and GPT-4V; (4)(5) Our methods (based on Bard and GPT-4V). In the user study, participants are provided an image and the descriptions generated by the three methods at each time, and they are asked to rate the descriptions using a scale of 0-10 (with 10 being the best) according to the 6 metrics. 17 participants have been invited, and each person is asked to rate about 40 scenes on average.

Table IV shows the results of this user study. We found that domain-specific models can hardly provide useful object and scene-level descriptions due to their lack of general knowledge. On the other hand, VLM provides rich context but lacks accuracy in domain-related information. Our method benefits from both of them and achieves much better performances by joining their strengths.

**Part perception (Sec. IV-C)**. In our part perception task, we utilize a single RGBD image as input to predict part-related information, including part semantic segmentation, pose and state estimation. To evaluate the methods, we introduce a new benchmark for part perception tasks. In comparison with GAPartNet, our benchmark is more comprehensive and suitable for manipulation tasks. For instance, we incorporate a greater number of objects with closed parts, accounting for 12.5%, which is not included in GAPartNet[13]. To enable the evaluation of generalizations at different levels, the test data are divided into 4 subsets: in-distribution, unseen (articulation) states, unseen objects, and unseen object categories. We use the average precision AP@50 and mAP as evaluation metrics for part segmentation, which are widely adopted in prior 3D semantic/instance segmentation benchmarks such as ScanNet[6] and GAPartNet[13],

Table V presents the results of our method in comparison to various baselines for part perception. We consider GAPartNet[13], a modified version of PointGroup[27], SoftGroup[50], and AutoGPart [33] as our primary baselines. Additionally, we adapted GroundedSAM, rebranding it as PartGroundedSAM for this context. The performance metrics in Table V indicate that our approach surpasses other baselines. Notably, we observed that methods based on 3D tend to underperform on out-of-domain data, whereas 2D-centric methods yield subpar results for parts. Our methodology derives advantages from both the 2D and 3D realms, resulting in optimal performance and superior generalization capabilities.

## VI. Conclusions

In this paper, we introduce a novel framework for language-guided manipulation of articulated objects. Bridging the understanding of object semantics and actionability at the part level, we can ground language-implied actions to executable manipulations. Throughout our framework, we also study the combination of general-purpose large vision/language models and domain-specialist models for enhancing the richness and correctness of network predictions. better handle these tasks and achieve state-of-the-art performance. We demonstrate our strong generalizability across diverse object categories and tasks. We also provide a new benchmark for language-instructed articulated-object manipulations.

**Limitations and future work**. Since the SAGE system is built upon existing large vision-language models, it suffers from the drawbacks of demanding massive data and massive computing power. Although in our case we use APIs provided by GPT-4V, the long inference time and high cost could be a concern. On the other hand, we rely on the expert model GAPartNet to detect parts and its prior knowledge to manipulate the parts. The policy is not always optimal by blindly following the

pre-defined end-effector trajectories. Such a motion-planning-based execution policy is not as responsive as a reinforcement learning or imitation learning agent. One future exploration direction could be fine-tuning existing large models to directly output the desired low-level action of the end-effector to increase responsiveness.

## REFERENCES


[1] Ahmed Akakzia, Cédric Colas, Pierre-Yves Oudeyer, Mohamed Chetouani, and Olivier Sigaud. Grounding language to autonomously-acquired skills via goal generation. In *ICLR 2021-Ninth International Conference on Learning Representation*, 2021.

[2] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[3] Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning*, pages 287–318. PMLR, 2023.

[4] Felix Burget, Armin Hornung, and Maren Bennewitz. Whole-body motion planning for manipulation of articulated objects. In *2013 IEEE International Conference on Robotics and Automation*, pages 1656–1662. IEEE, 2013.

[5] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4693–4700, 2018. doi: 10.1109/ICRA.2018.8460487.

[6] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[7] Ben Eisner, Harry Zhang, and David Held. Flowbot3d: Learning 3d articulation flow to manipulate articulated objects. *arXiv preprint arXiv:2205.04382*, 2022.

[8] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11444–11453, 2020.

[9] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[10] Samir Yitzhak Gadre, Kiana Ehsani, and Shuran Song. Act the part: Learning interaction strategies for articulated object part discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15752–15761, 2021.

[11] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022.

[12] Wei Gao and Russ Tedrake. kpam-sc: Generalizable manipulation planning using keypoint affordance and shape completion. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6527–6533. IEEE, 2021.

[13] Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. *arXiv preprint arXiv:2211.05272*, 2022.

[14] Haoran Geng, Ziming Li, Yiran Geng, Jiayi Chen, Hao Dong, and He Wang. Partmanip: Learning cross-category generalizable part manipulation policy from point cloud observations, 2023.

[15] Yiran Geng, Boshi An, Haoran Geng, Yuanpei Chen, Yaodong Yang, and Hao Dong. End-to-end affordance learning for robotic manipulation. *arXiv preprint arXiv:2209.12941*, 2022.

[16] Dibya Ghosh, Jad Rahme, Aviral Kumar, Amy Zhang, Ryan P Adams, and Sergey Levine. Why generalization in rl is difficult: Epistemic pomdps and implicit partial observability. *Advances in Neural Information Processing Systems*, 34:25502–25515, 2021.

[17] Ran Gong, Jiangyong Huang, Yizhou Zhao, Haoran Geng, Xiaofeng Gao, Qingyang Wu, Wensi Ai, Ziheng Zhou, Demetri Terzopoulos, Song-Chun Zhu, et al. Arnold: A benchmark for language-grounded task learning with continuous states in realistic 3d scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[18] Minghao Gou, Hao-Shu Fang, Zhanda Zhu, Sheng Xu, Chenxi Wang, and Cewu Lu. Rgb matters: Learning 7-dof grasp poses on monocular rgbd images. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13459–13466. IEEE, 2021.

[19] Prasoon Goyal, Raymond J. Mooney, and Scott Niekum. Zero-shot task adaptation using natural language. *CoRR*, abs/2106.02972, 2021. URL https://arxiv.org/abs/2106.02972.

[20] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. *arXiv preprint arXiv:2302.04659*, 2023.

[21] Hengyuan Hu, Denis Yarats, Qucheng Gong, Yuandong Tian, and Mike Lewis. Hierarchical decision making by generating and following natural language instructions. *Advances in neural information processing systems*, 32, 2019.

[22] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and

Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR, 2022.

[23] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Tomas Jackson, Noah Brown, Linda Luu, Sergey Levine, Karol Hausman, and brian ichter. Inner monologue: Embodied reasoning through planning with language models. In *6th Annual Conference on Robot Learning*, 2022. URL https://openreview.net/forum?id=3R3Pz5i0tye.

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

[25] Advait Jain and Charles C Kemp. Pulling open novel doors and drawers with equilibrium point control. In *2009 9th ieee-ras international conference on humanoid robots*, pages 498–505. IEEE, 2009.

[26] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[27] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4867–4876, 2020.

[28] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts. *arXiv preprint arXiv:2210.03094*, 2022.

[29] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[30] Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of generalisation in deep reinforcement learning. *arXiv preprint arXiv:2111.09794*, 2021.

[31] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *arXiv preprint arXiv:2209.07753*, 2022.

[32] Kevin Lin, Christopher Agia, Toki Migimatsu, Marco Pavone, and Jeannette Bohg. Text2motion: From natural language instructions to feasible plans. *arXiv preprint arXiv:2303.12153*, 2023.

[33] Xueyi Liu, Xiaomeng Xu, Anyi Rao, Chuang Gan, and Li Yi. Autogpart: Intermediate supervision search for generalizable 3d part segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11624–11634, 2022.

[34] Oier Mees, Lukas Hermann, and Wolfram Burgard. What matters in language conditioned robotic imitation learning. *arXiv preprint arXiv:2204.06252*, 2022.

[35] Kaichun Mo, Leonidas J Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6813–6823, 2021.

[36] Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. *arXiv preprint arXiv:2107.14483*, 2021.

[37] OpenAI. Gpt-4 technical report, 2023.

[38] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[39] L Peterson, David Austin, and Danica Kragic. High-level control of a mobile manipulator for door opening. In *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113)*, volume 3, pages 2333–2338. IEEE, 2000.

[40] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

[41] Lin Shao, Toki Migimatsu, Qiang Zhang, Karen Yang, and Jeannette Bohg. Concept2robot: Learning manipulation concepts from instructions and human demonstrations. *The International Journal of Robotics Research*, 40(12-14):1419–1434, 2021.

[42] Hao Shen, Weikang Wan, and He Wang. Learning category-level generalizable object manipulation policy via generative adversarial self-imitation learning from demonstrations. *arXiv preprint arXiv:2203.02107*, 2022.

[43] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[44] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[45] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. *arXiv preprint arXiv:2209.11302*, 2022.

[46] Simon Stepputtis, Joseph Campbell, Mariano Phielipp, Stefan Lee, Chitta Baral, and Heni Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. *Advances in Neural Information Processing Systems*, 33:13139–13150, 2020.

[47] Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13438–13444. IEEE, 2021.

[48] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380, 1991.

[49] Yusuke Urakami, Alec Hodgkinson, Casey Carlin, Randall Leu, Luca Rigazio, and Pieter Abbeel. Doorgym: A scalable door opening environment and baseline agent. *arXiv preprint arXiv:1908.01887*, 2019.

[50] Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, and Chang D Yoo. Softgroup for 3d instance segmentation on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2708–2717, 2022.

[51] Weikang Wan, Haoran Geng, Yun Liu, Zikang Shan, Yaodong Yang, Li Yi, and He Wang. Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning. *arXiv preprint arXiv:2304.00464*, 2023.

[52] Yian Wang, Ruihai Wu, Kaichun Mo, Jiaqi Ke, Qingnan Fan, Leonidas J Guibas, and Hao Dong. Adaafford: Learning to adapt manipulation affordance for 3d articulated objects via few-shot interactions. In *European Conference on Computer Vision*, pages 90–107. Springer, 2022.

[53] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[54] Ruihai Wu, Yan Zhao, Kaichun Mo, Zizheng Guo, Yian Wang, Tianhao Wu, Qingnan Fan, Xuelin Chen, Leonidas Guibas, and Hao Dong. VAT-mart: Learning visual action trajectory proposals for manipulating 3d ARTiculated objects. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=iEx3PiooLy.

[55] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *CVPR*, 2020.

[56] Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, et al. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. *arXiv preprint arXiv:2303.00938*, 2023.

[57] Zhenjia Xu, Beichun Qi, Shubham Agrawal, and Shuran Song. Adagrasp: Learning an adaptive gripper-aware grasping policy. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4620–4626. IEEE, 2021.

[58] Zhenjia Xu, Zhanpeng He, and Shuran Song. Universal manipulation policy network for articulated objects. *IEEE Robotics and Automation Letters*, 7(2):2447–2454, 2022.

[59] Yang You, Bokui Shen, Congyue Deng, Haoran Geng, He Wang, and Leonidas Guibas. Make a donut: Language-guided hierarchical emd-space planning for zero-shot deformable object manipulation, 2023.

[60] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Avnish Narayan, Hayden Shively, Adithya Bellathur, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning, 2021.

[61] Jiajie Zhang and Vimla L Patel. Distributed cognition, representation, and affordance. *Pragmatics & Cognition*, 14(2):333–341, 2006.

[62] Yan Zhao, Ruihai Wu, Zhehuan Chen, Yourong Zhang, Qingnan Fan, Kaichun Mo, and Hao Dong. Dualafford: Learning collaborative visual affordance for dual-gripper object manipulation. *arXiv preprint arXiv:2207.01971*, 2022.