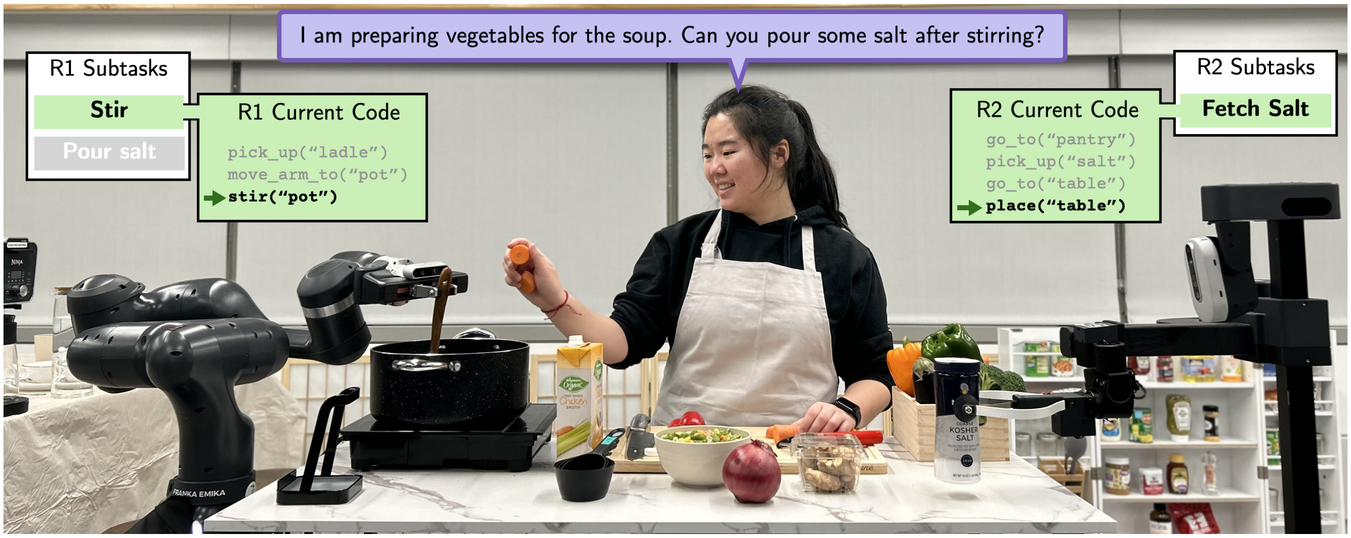
MOSAIC: A Modular System for Assistive and Interactive Cooking

Huaxiaoyue Wang*, Kushal Kedia*, Juntao Ren*, Rahma Abdullah, Atiksh Bhardwaj, Angela Chao, Kelly Y Chen, Nathaniel Chin, Prithwish Dan, Xinyi Fan, Gonzalo Gonzalez-Pumariega, Aditya Kompella, Maximus Adrian Pace, Yash Sharma, Xiangwan Sun, Neha Sunkara, Sanjiban Choudhury

Cornell University

https://portal-cornell.github.io/MOSAIC/


I am preparing vegetables for the soup. Can you pour some salt after stirring?
R1 Subtasks
Stir
Pour salt
R1 Current Code
R2 Current Code
R2 Subtasks
Fetch Salt

```
pick_up("ladle")
move_arm_to("pot")
stir("pot")
```

```
go_to("pantry")
pick_up("salt")
go_to("table")
place("table")
```

Fig 1. **MOSAIC cooking in the kitchen.** (top) MOSAIC interacts with a user via natural language and controls a tabletop manipulator (R1) and a mobile manipulator (R2) to prepare vegetable soup with the user. (bottom) We evaluate MOSAIC on multiple recipes, involving a range of robot skills that interact with the human user and everyday objects.

***Abstract*— We present MOSAIC, a modular architecture for home robots to perform complex collaborative tasks, such as cooking with everyday users. MOSAIC tightly collaborates with humans, interacts with users using natural language, coordinates multiple robots, and manages an open vocabulary of everyday objects. At its core, MOSAIC employs modularity: it leverages multiple large-scale pre-trained models for general tasks like language and image recognition, while using streamlined modules designed for task-specific control. We extensively evaluate MOSAIC on 60 end-to-end trials where two robots collaborate with a human user to cook a combination of recipes. We also extensively test individual modules with 180 episodes of visuomotor picking, 60 episodes of human motion forecasting, and 46 online user evaluations of the task planner. We show that MOSAIC is able to efficiently collaborate with humans, interpret and execute complex tasks, and adapt to new tasks with minimal reconfiguration. Finally, we discuss limitations of the current system and exciting open challenges in this domain. The project's website is at https://portal-cornell.github.io/MOSAIC/**

I. Introduction

Collaborative tasks in home environments requiring a coordinated medley of skills pose significant challenges for robots. These tasks require robots to have natural interactions with human users, possess the ability to learn a diverse set of skills, and perform them in a collaborative manner. Prior systems in this domain [1]–[4] have demonstrated impressive capabilities. However, they typically have one of two limitations: either they operate in isolation and lack meaningful collaboration with humans, or they interact with humans in a highly scripted manner, and are therefore only capable of completing a narrow set of predefined tasks. In this paper, we aim to overcome both of these limitations by designing a system that fluidly collaborates with humans and performs a wide range of tasks.

We identify three key desiderata for the system: *(1)* interact with users via natural language, *(2)* perform a range of skills that require manipulating everyday objects, and *(3)* collaborate seamlessly with humans. Consider the scenario in Figure 1, where a human user collaborates with two robots to prepare a meal. The user should be able to effortlessly interact with the system via natural language to decide on a recipe. The robots in turn should perform the necessary skills to make the recipe, such as fetching a range of ingredients and cooking with them. Finally, the robots must fluidly collaborate with humans, such as handing over items.

One of the key challenges in building a collaborative agent that functions seamlessly in the wild is ensuring that it is able to act safely across an expansive set of possible inputs. While a single end-to-end model works well for tasks like language

[*] Denotes equal contribution.

understanding where large amounts of data are available, such an approach is difficult for robot controls, where less data is available and extreme precision is important. Our key insight is that ***by modularizing our architecture, we can segment out parts of the framework that require broad generalization, such as language and image recognition, from the portions that require task-specific control.*** This division of work means that strong overall performance can be achieved through specialization: we can use large *pre-trained models* to extract useful information from large and unstructured input spaces and *task-specific models* to make safe and precise decisions.

We apply this modular approach in building MOSAIC (**Mo**dular **S**ystem for **A**ssistive and **I**nteractive **C**ooking): a modular architecture for home robots that integrates multiple large-scale pre-trained models. In particular, we use large language models (LLMs) for interactive task planning, vision language models (VLMs) for visuomotor skills, and motion forecasting models for predicting human intents for collaboration. To the best of our knowledge, this is the first system to integrate multiple large-scale models in such a way that enables multiple home robots to collaborate with a human user to tackle complex, long-horizon tasks such as cooking.

While the principle of modularity has been central to developing robust real-world robotic systems (e.g. in autonomous driving), such systems often rely on meticulously engineered components. We introduce several key innovations to create an adaptive, scalable system that collaborates fluidly with humans. Our contributions can be organized as:

1) **Interactive Task Planner.** We propose an architecture that embeds Large Language Models (LLMs) within a *behavior tree*. Prior work [1], [5]–[8] attempts to directly use LLMs for task planning. However, LLMs often make mistakes and are difficult to control. To reduce errors, we partition the action space and reasoning process as nodes in the tree.
2) **Visuomotor Skills.** We propose a lightweight architecture that uses a pre-trained VLM for object identification and an RL policy learnt in simulation. In contrast to prior work [9]–[12], our method does not require *any* online demonstrations nor training large networks.
3) **Human Motion Forecasting.** We develop a method for forecasting human motion that allows robots to seamlessly collaborate with humans in manipulation tasks. Unlike prior works [13], [14] that model humans as static entities, we utilize large-scale human motion data [15] to train a forecasting model.
4) **Comprehensive Evaluation.** We conduct 60 end-to-end trials where two robots collaborate with a human user to cook complex, long-horizon recipes. We also extensively test individual modules with 180 episodes of visuomotor picking, 60 episodes of human motion forecasting, and 46 online user evaluations of the task planner. We run our system end-to-end with a real human user, completing 68.3% (41/60) collaborative cooking trials of 6 different recipes with an average subtask completion rate of 91.6%.

II. Approach

We present MOSAIC, **Mo**dular **S**ystem for **A**ssistive and **I**nteractive **C**ooking, a modular architecture that combines multiple large-scale pre-trained models to solve collaborative cooking tasks. Fig. 2 shows an overview of MOSAIC. It consists of three main components: *1) Interactive Task Planner (II-A):* a module that interacts with real users via natural language to plan a diverse set of tasks and coordinate subtasks during the cooking process. *2) Visuomotor Skill (II-B):* a module that generalizes robot skills to a diverse set of kitchen objects and environments. *3) Human Motion Forecasting (II-C):* a module that leverages motion forecasting models to predict human motion, ensuring that robots can collaborate safely and fluidly with humans.

A. Interactive Task Planner

The goal of the task planner is to continuously interact with a human user using natural language, delegate subtasks to different robots or the user, and monitor progress. Concretely, the task planner interacts with the user to determine a task(e.g. “Prepare vegetable soup”). It represents the task $\mathcal{T}$ as a directed acyclic graph (DAG), which models temporal dependencies between different subtasks and determines available subtasks that can be assigned. The task planner also assigns and maintains a queue of subtasks for each robot. To execute a subtask (e.g. “fetch salt”), the task planner generates a code snippet that issues a series of API calls such as `go_to("pantry")`, `pick("pepper")`, etc.

While many recent approaches [1], [5]–[8] directly use LLMs for task planning, we observe two main challenges. First, even with chain-of-thought prompting [16], since the action space is large and the reasoning process is complex, the LLMs make mistakes such as misinterpreting the observation or choosing incorrect actions. More importantly, the LLMs tend to violate safety constraints that the developer specifies, such as assigning subtasks without confirming with the user. Second, the developer has little control over the LLMs’ behavior other than specifying the rules and constraints in one monolithic prompt, which is challenging to debug and scale. To overcome both challenges, we propose an architecture that embeds LLMs within a behavior tree (BT) [17]. Each behavior partitions the action space and reasoning process, thereby reducing the complexity and potential error rate of the LLMs. Moreover, the modular nature of BT makes it easy to scale to multiple behaviors.

B. Visuomotor Skills

The goal of the visuomotor skills module is to execute subtasks assigned by the task planner. A common approach to train visuomotor skills is to imitate human demonstrations on a suite of tasks via end-to-end training [9]–[12], [18]–[20]. However, state-of-the-art methods using this approach generally require (1) good coverage of states and (2) expert action labels from those states. This includes data that shows the robot how to recover after making errors. Taken together, this leads to algorithms that require up to hundreds of hours of expert demonstrations, which is infeasible to collect.

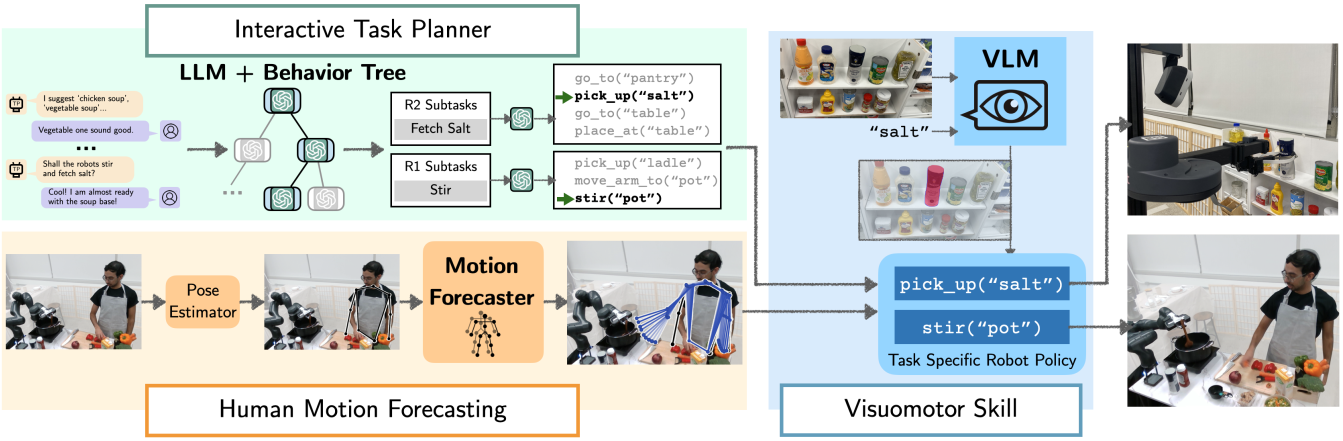


Fig. 2: **MOSAIC System Overview.** The *Interactive Task Planner* module communicates with the user via natural language to decide on a recipe. It assigns subtasks to each robot accordingly. The *Human Motion Forecasting* extracts and converts the human's 2D post to 3D coordinates, which it uses to predict future human motion. Simultaneously, a VLM takes image and language as input and produces a 3D grasp pose around the object of interest. Combined, all three are taken by the execution policy of the *Visuomotor Skill* module to produce a final robot action.

Instead, we partition the end-to-end architecture into object-identification and action-execution modules. We offload object identification to pre-trained VLMs that can generalize to many objects, and we solve action execution by searching for a policy purely in simulation using reinforcement learning. In doing so, we have addressed both challenges without needing to collect any additional data.

Object detection via pre-trained models. Given our input image and language condition, we pass both through a pre-trained OwlViT [21] model, giving us a set of bounding boxes. To handle robot-specific viewpoints (that may be less common in the training data of these large VLMs), we and take the bounding-box coordinate with the highest CLIP similarity score [22].

Grasp-pose generation via point-cloud segmentation. In the next phase of our pipeline, we use FastSAM [23] to obtain a more accurate segmentation of the object within the bounding box and back-project the segmented pixels through the depth camera's point cloud. We take our grasp-pose to be the center-of-mass of this projection.

Action prediction via model-based reinforcement learning. To predict the final actions, we design a simulator and reward function to train any general RL agent that takes as input some privileged information of the world, in this case the 3D grasp-pose, and outputs actions to reach that position without violating some set constraints.

C. Human Motion Forecasting

Safe and effective coordination with humans requires forecasting human motion and adapting robot plans accordingly. Accurate forecasts are critical for collaborative cooking, where robots work in close proximity to humans. For instance, observe the robot stirring a pot alongside a human partner in Figure 2. When the human moves in to put vegetables in the pot, the robot should anticipate that movement and make way for the human by retracting its arm back. Our goal is to use forecasts of human motion to guide the robot's decision making.

Pre-training on Large-scale Data. We first pre-train our model on large-scale human activity data to generate smooth predictions of human motion given a history of joint positions as input. We use AMASS [15], a large dataset of human activity, encompassing over 300 subjects and 40 hours of motion capture data.

Fine-tune on Interaction Data. To ensure the forecaster's motion predictions are helpful for the robot to plan its actions around humans in the kitchen, we utilize the Collaborative Manipulation Dataset (CoMaD) [24], a dataset of human-human interactions in a kitchen setting.

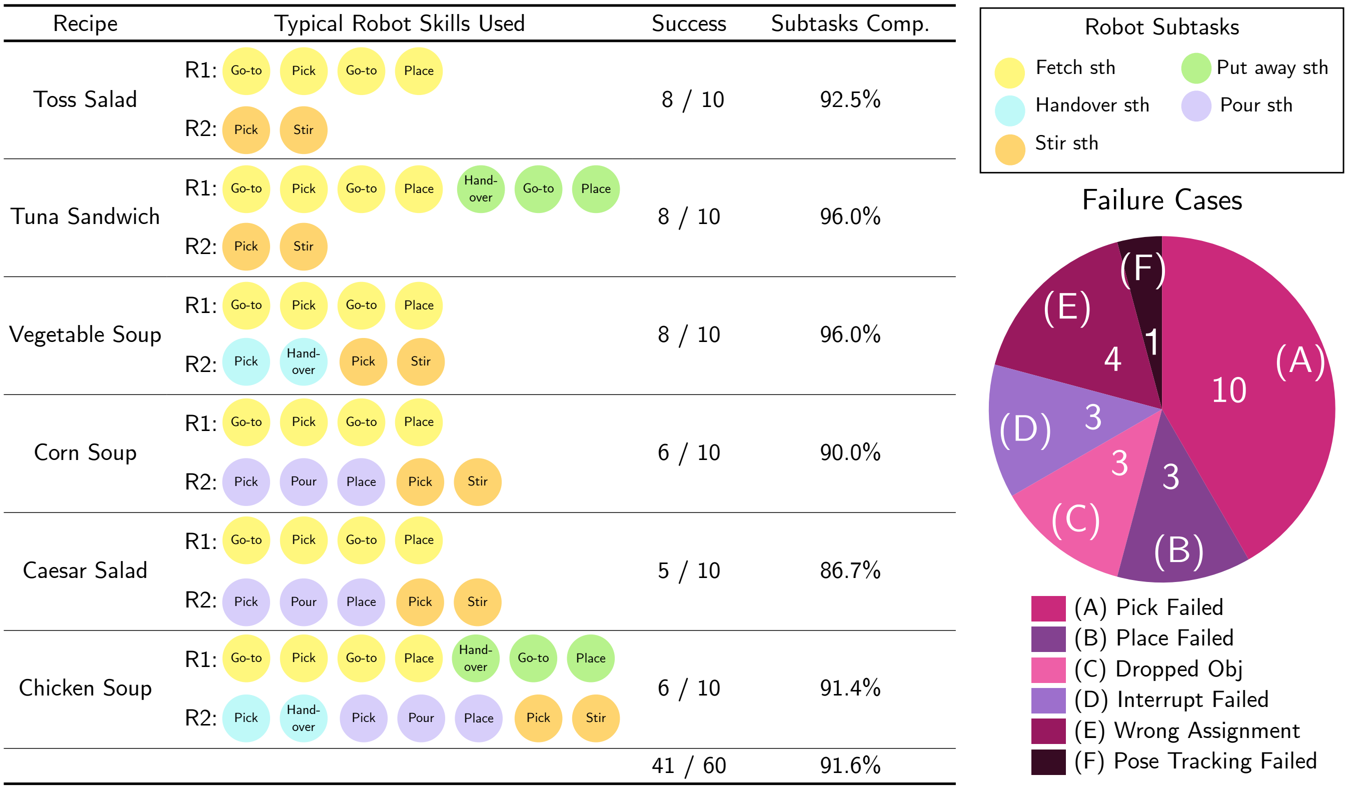
Inference Time: Real-time, Vision-based Forecasting and Planning. A single RGB-D camera aimed at the human's torso is used to detect their upper-body pose. The human joint locations are then identified on the RGB image using MediaPipe [25], a 2D pose detector. These locations are then back-projected to 3D world coordinates using the image depth map. Finally, the human poses are used to generate real-time motion forecasts used by the robot.

III. Experiments

We conduct a total of 60 end-to-end trials with two robots and a user collaboratively making 6 recipes. In all experiments, the tabletop manipulator (R1) is a 7-DoF Franka Emika Research 3 [26] and the mobile manipulator (R2) is a 6-DoF Stretch Robot RE1 [27]. The kitchen also has two overhead RGB-D cameras that can perceive the workspace and capture a human's motion. To allow users to interact with the task planner, we use Google's speech-to-text APIs [28] to transcribe user's verbal instructions and its text-to-speech APIs to vocalize the task planner's responses.

A. End-to-end Trials

Figure 3 shows a table with the different recipes (tasks), the different subtasks, and the robot skills involved. Each recipe involves a different combination of robot skills and different types of interaction with the user. For example, users provide vague instructions, interrupt a robot's subtask,



Recipe		Typical Robot Skills Used	Success	Subtasks Comp.
Toss Salad	R1:	Go-to, Pick, Go-to, Place	8 / 10	92.5%
	R2:	Pick, Stir		
Tuna Sandwich	R1:	Go-to, Pick, Go-to, Place, Hand-over, Go-to, Place	8 / 10	96.0%
	R2:	Pick, Stir		
Vegetable Soup	R1:	Go-to, Pick, Go-to, Place	8 / 10	96.0%
	R2:	Pick, Hand-over, Pick, Stir		
Corn Soup	R1:	Go-to, Pick, Go-to, Place	6 / 10	90.0%
	R2:	Pick, Pour, Place, Pick, Stir		
Caesar Salad	R1:	Go-to, Pick, Go-to, Place	5 / 10	86.7%
	R2:	Pick, Pour, Place, Pick, Stir		
Chicken Soup	R1:	Go-to, Pick, Go-to, Place, Hand-over, Go-to, Place	6 / 10	91.4%
	R2:	Pick, Hand-over, Pick, Pour, Place, Pick, Stir		
			41 / 60	91.6%

Fig. 3: **End-to-end results.** On-policy results for 6 recipes, where each recipe is tested through 10 trials. Each recipe contains various subtasks involving different robot skills. We report the number of trials that are completed without any errors and the individual subtask completion rate. We also categorize the failure cases. MOSAIC is able to complete 41/60 tasks with an average subtask completion rate of 91.6%.

and add new subtasks that are not in the recipe. For each trial, we compute two metrics: was the trial successful, and the subtask completion rate. Overall, MOSAIC completes 41/60 (68.3%) collaborative cooking trials of 6 different recipes with an average subtask completion rate of 91.6%. We analyze two specific questions:

How does MOSAIC scale with longer horizon tasks? We test a range of recipes, from "Toss Salad", which involves 6 skills, to "Chicken Soup", which involves 14 skills. While MOSAIC's success rate drops with the increasing horizon as one would expect, it does not fall off exponentially and stays above 50%. A key reason is that each module in MOSAIC is trained to be robust to errors in incoming input (e.g. the task planner handles delays made by a robot, the visuomotor skills `pick()` handles errors from `go_to()`, the forecasting handles errors from pose estimation, and so on).

Does modularity help localize failures to specific modules? As each module has sub-modules, each with a clear input/output contract, localizing an error is easily automated. We use this to cluster failures into the following 5 categories, shown also in Figure 3:

(A) *[Visuomotor Skill] Failed to pick up the object:* Sometimes, the VLM selects an incorrect object for picking.

(B) *[Visuomotor Skill] Failed to successfully place the object:* The robot releases an object from an incorrect height, causing it to topple.

(C) *[Visuomotor Skill] Dropped the object during a skill:* The `stir()` and `pour()` skill may drop an object due to an insufficiently stable grip.

(D) *[Interactive Task Planner] Failed to interrupt a subtask:* The speech-to-text module sometimes fails to correctly transcribe user's short command.

(E) *[Interactive Task Planner] Assigned an incorrect subtask:* The task planner misunderstands the user's command and re-assigns a completed subtask to the robot.

(F) *[Human Motion Forecasting] Pose Tracking Failed:* The human's pose moved outside the camera's view, causing a tracking error while forecasting motion.

IV. Discussion

In this paper, we present a modular system capable of controlling two robots to interactively cook a variety of recipes with a human user. Leveraging an ensemble of large-scale, pre-trained models, our system communicates with the user, forecasts their intents, and completes a series of visuomotor skills. To validate our design decisions, we conduct extensive experiments in the real world with multiple human users. We architect a set of modular frameworks that utilizes large-scale, pre-trained models to quickly equip multi-agent systems with generalizable skills. These characteristics make MOSAIC a desirable foundation for collaborative human-robot systems in complex home environments and for future work that further refine and expand this system's capability. Furthermore, this process of modular evaluation has been instrumental in uncovering potential failure modes.

References

[1] 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.

[2] M. Bajracharya, J. Borders, R. Cheng, D. M. Helmick, L. Kaul, D. Kruse, J. Leichty, J. Ma, C. Matl, F. Michel, C. Papazov, J. Petersen, K. Shankar, and M. Tjersland, "Demonstrating mobile manipulation in the wild: A metrics-driven approach," in *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, K. E. Bekris, K. Hauser, S. L. Herbert, and J. Yu, Eds., 2023. [Online]. Available: https://doi.org/10.15607/RSS.2023.XIX.055

[3] N. M. M. Shafiullah, A. Rai, H. Etukuru, Y. Liu, I. Misra, S. Chintala, and L. Pinto, "On bringing robots home," *arXiv preprint arXiv:2311.16098*, 2023.

[4] 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.

[5] 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," 2023.

[6] 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," 2023.

[7] B. Li, P. Wu, P. Abbeel, and J. Malik, "Interactive task planning with language models," 2023.

[8] Z. Mandi, S. Jain, and S. Song, "Roco: Dialectic multi-robot collaboration with large language models," 2023.

[9] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "Bc-z: Zero-shot task generalization with robotic imitation learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 991–1002.

[10] 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.

[11] 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.

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

[13] W. Yang, B. Sundaralingam, C. Paxton, I. Akinola, Y.-W. Chao, M. Cakmak, and D. Fox, "Model predictive control for fluid human-to-robot handovers," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6956–6962.

[14] E. A. Sisbot and R. Alami, "A human-aware manipulation planner," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1045–1057, 2012.

[15] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "AMASS: Archive of motion capture as surface shapes," in *International Conference on Computer Vision*, Oct. 2019, pp. 5442–5451.

[16] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," 2023.

[17] M. Colledanchise and P. Ögren, "Behavior trees in robotics and AI: an introduction," *CoRR*, vol. abs/1709.00084, 2017. [Online]. Available: http://arxiv.org/abs/1709.00084

[18] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *arXiv preprint arXiv:2303.04137*, 2023.

[19] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 894–906.

[20] 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.

[21] G. Heigold, M. Minderer, A. Gritsenko, A. Bewley, D. Keysers, M. Lučić, F. Yu, and T. Kipf, "Video owl-vit: Temporally-consistent open-world localization in video," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 13 802–13 811.

[22] 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," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[23] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang, "Fast segment anything," *arXiv preprint arXiv:2306.12156*, 2023.

[24] K. Kedia, A. Bhardwaj, P. Dan, and S. Choudhury, "Interact: Transformer models for human intent prediction conditioned on robot actions," *ArXiv*, vol. abs/2311.12943, 2023.

[25] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. L. Zhu, F. Zhang, and M. Grundmann, "Blazepose: On-device real-time body pose tracking," *ArXiv*, vol. abs/2006.10204, 2020.

[26] "Franka research 3," Franka Robotics, 2022. [Online]. Available: https://franka.de/documents

[27] C. C. Kemp, A. Edsinger, H. M. Clever, and B. Matulevich, "The design of stretch: A compact, lightweight mobile manipulator for indoor human environments," 2022.

[28] [Online]. Available: https://cloud.google.com/speech-to-text/