Inner Monologue: Embodied Reasoning through Planning with Language Models

Wenlong Huang[†]**, Fei Xia**[†]**, Ted Xiao**[†]**, Harris Chan, Jacky Liang, Pete Florence,**
Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet,
Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, Brian Ichter
Robotics at Google, † equal contribution and alphabetically listed
Project website: https://innermonologue.github.io

Abstract: Recent works have shown how the reasoning capabilities of Large Language Models (LLMs) can be applied to domains beyond natural language processing, such as planning and interaction for robots. These embodied problems require an agent to understand many semantic aspects of the world: the repertoire of skills available, how these skills influence the world, and how changes to the world map back to the language. LLMs planning in embodied environments need to consider not just what skills to do, but also how and when to do them - answers that change over time in response to the agent's own choices. In this work, we investigate to what extent LLMs used in such embodied contexts can reason over sources of feedback provided through natural language, without any additional training. We propose that by leveraging environment feedback, LLMs are able to form an *inner monologue* that allows them to more richly process and plan in robotic control scenarios. We investigate a variety of sources of feedback, such as success detection, scene description, and human interaction. We find that closed-loop language feedback significantly improves high-level instruction completion on three domains, including simulated and real table top rearrangement tasks and long-horizon mobile manipulation tasks in a kitchen environment in the real world.

1 Introduction

Intelligent and flexible embodied interaction requires robots to be able to deploy large repertoires of basic behaviors in appropriate ways, sequence these behaviors as needed for long horizon tasks, and also recognize when to switch to a different approach if a particular behavior or plan is unsuccessful. High-level planning, perceptual feedback, and low-level control are just a few of the sub-tasks that would need to be seamlessly combined together to perform the sort of reasoning required for an embodied agent, such as a robot, to intelligently act in the world. While conventionally these challenges have been approached from the perspective of planning (e.g., TAMP [1]) or hierarchical learning (e.g., HRL [2]), effective high-level reasoning about complex tasks also requires semantic knowledge and understanding of the world.

One of the remarkable observations in recent machine learning research is that large language models (LLMs) can not only generate fluent textual descriptions, but also appear to have rich internalized knowledge about the world [3, 4, 5, 6, 7]. When appropriately conditioned (e.g., prompted), they can even carry out some degree of deduction and respond to questions that appear to require reasoning and inference [8, 9, 10, 11, 12, 13]. This raises an intriguing possibility: beyond their ability to interpret natural language instructions, can language models further serve as reasoning models that combine multiple sources of feedback and become interactive problem solvers for embodied tasks, such as robotic manipulation?

Prior studies show that language helps humans internalize our knowledge and perform complex relational reasoning through *thinking in language* [14, 15, 16, 17, 18]. Imagine the "inner monologue" that happens when a person tries to solve some task: "I have to unlock the door; let me try to pick up the key and put it in the lock... no, wait, it doesn't fit, I'll try another one... that one worked, now I can turn the key." The thought process in this case involves choices about the best immediate action to solve the high-level task ("pick up the key"), observations about the outcomes of attempted actions ("it doesn't fit"), and corrective actions that are taken in response to these observations ("I'll try another one"). Inspired by the human thought process, we propose that such an inner monologue is a natural framework for incorporating feedback for LLMs.

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

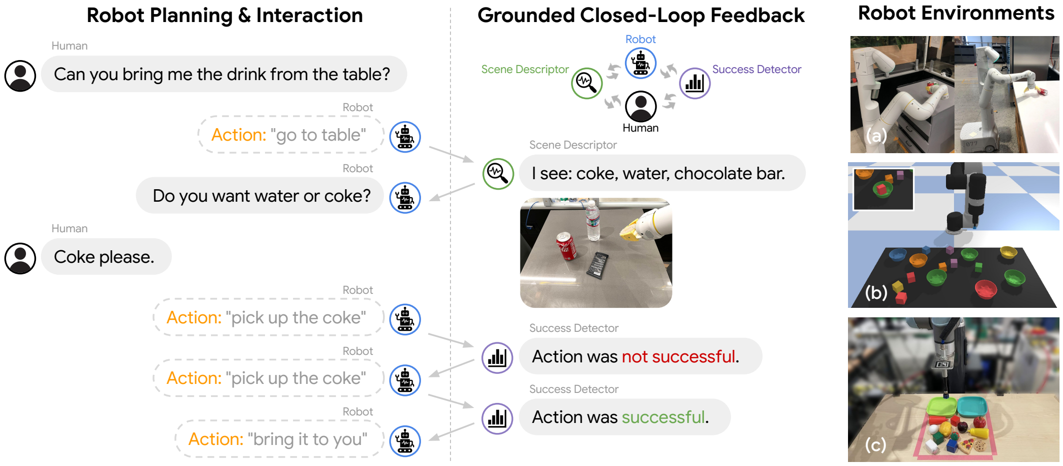

Robot Planning & Interaction
Human
Can you bring me the drink from the table?
Robot
Action: "go to table"
Robot
Do you want water or coke?
Human
Coke please.
Robot
Action: "pick up the coke"
Robot
Action: "pick up the coke"
Robot
Action: "bring it to you"
Grounded Closed-Loop Feedback
Robot
Scene Descriptor
Success Detector
Human
Scene Descriptor
I see: coke, water, chocolate bar.
Success Detector
Action was not successful.
Success Detector
Action was successful.
Robot Environments
(a)
(b)
(c)

Figure 1: Inner Monologue enables grounded closed-loop feedback for robot planning with large language models by leveraging a collection of perception models (e.g., scene descriptors and success detectors) in tandem with pretrained language-conditioned robot skills. Experiments show our system can reason and replan to accomplish complex long-horizon tasks for (a) mobile manipulation and (b,c) tabletop manipulation in both simulated and real settings.

Our work studies these questions by combining LLMs with various sources of textual feedback, only utilizing few-shot prompting without any additional training. We observe that similarly to recent work [19], natural language provides a universal and interpretable interface for such grounding of model communication and allows them to incorporate their conclusions in an overarching inner monologue driven by a language model. While prior work has investigated using language models as planners [20, 21] or incorporating multimodal-informed perception through language [19], to the best of our knowledge no work has studied the critical link of not only planning with language, but also informing *embodied feedback with language*, which we investigate in this work.

Specifically, we study methods and sources of feedback for closing the agent-environment loop via an inner monologue and their impact on downstream execution success and new capabilities arising from such interaction. In particular, we combine multiple perception models that perform various tasks such as language-conditioned semantic classification or language-based scene description, together with feedback provided by a human user that the robot is cooperating with. To execute the commands given by a user, the actions are chosen from a set of pre-trained robotic manipulation skills together with their textual descriptions that can be invoked by a language model. Our proposed system Inner Monologue chains together these various components (perception models, robotic skills, and human feedback) in a shared language prompt, enabling it to successfully perform user instructions.

Finally, we show that Inner Monologue, without requiring additional training beyond a frozen language model and pre-trained robotic skills, can accomplish complex, long-horizon, and unseen tasks in simulation as well as on two real-world robotic platforms. Notably, we show that it can efficiently retry under observed stochastic failure, replan under systematic infeasibility, or request human feedback for ambiguous queries, resulting in significantly improved performance in dynamical environments. As a demonstration of the versatility of LLMs and grounded closed-loop feedback, we additionally show several surprising capabilities emerging from the inner monologue formulation, including continued adaptation to new instructions, self-proposed goals, interactive scene understanding, multilingual interactions, and more.

2 Related Work

Task and Motion Planning. Task and motion planning [22, 23] requires simultaneously solving a high-level, discrete task planning problem [24, 25, 26], and a low-level, continuous motion planning problem [27]. Traditionally, this problem has been solved through optimization [28, 29] or symbolic reasoning [24, 26], but more recently machine learning has been applied to aspects of the problem via learned representations, learned task-primitives, and more [30, 31, 32, 33, 34, 35, 36, 37, 38]. Some works utilize language for planning and grounding [39, 40, 41, 42, 43, 44]. Others have approached the problem through hierarchical learning [45, 46, 34, 47, 48, 49, 50]. In this work, we leverage pre-trained LLMs and their semantic knowledge, along with trained low-level skills, to find feasible plans.

Task Planning with Language Models. Various prior works have explored using language as a space for planning [51, 52, 53, 20, 54, 21]. Some methods use prompt structure, self-talk, or discussion between

Success Detection in context

Robot Action: Pick up coke
Success: True
Robot Action: Bring it to user

Passive Scene Description in context:

Robot Action: Go to table
Scene: lime soda, coke, energy bar
Robot Action: pick up energy bar

Active Scene Description in context:

Robot Action: Go to drawers
Robot Ask: Is the drawer open?
Human: The drawer is closed.
Robot Action: Open the drawer

Figure 2: Various types of textual feedback. Success Detection gives task-specific task completion information, Passive Scene Description gives structured semantic scene information at every planning step, and Active Scene Description gives unstructured semantic information only when queried by the LLM planner.

experts to reason about plans or semantic concepts [55, 19, 10, 11]. Similar to ours are recent task planning approaches that leverage pre-trained autoregressive LLMs to decompose abstract, high-level instructions into a sequence of low-level steps executable by an agent [20, 21] in a zero-shot manner. Specifically, Huang et al. [20] prompt GPT-3 [9] and Codex [56] to generate action plans for embodied agents, where each action step is semantically translated to an admissible action with a Sentence-RoBERTa model [57, 58]. SayCan [21] instead grounds the actions by multiplying each candidate action's probability under FLAN [59] with the action's value function, which serves as a proxy for affordance [34]. However, both approaches effectively produce the plan while assuming that each proposed step is executed successfully by the agent. As a result, these approaches may not be robust in handling intermediate failures in dynamic environments or with poor lower level policies. We explore in Inner Monologue ways to incorporate grounded feedback from the environment into the LLM as we produce each step in the plan.

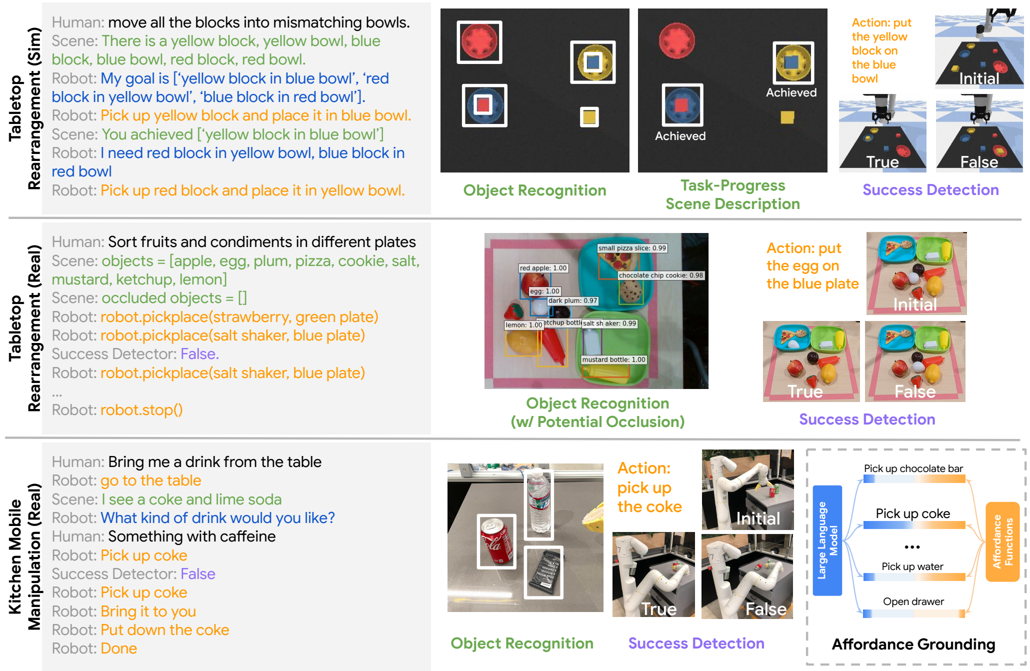
Fusing Vision, Language, and Control in Robotics. Various works have investigated strategies for the challenging problem of fusing vision, language, and control [60, 61, 62, 63, 64, 65, 66]. Some works have been trained directly for language-based interaction in robotic tasks [67, 68, 69, 70]. Recent large visual-language models (e.g., CLIP [71]) have been trained on joint image(s) and corresponding text captions via variants of a masked language modeling objective [72, 73, 74, 75], a contrastive loss [76, 77, 71] or other supervised objectives[78, 79]. CLIP has been employed in several robotics and embodied settings in zero-shot manner [80], or combined with Transporter networks [81] as in CLIPort [82]. Finally, Socratic Models [19] proposes the combination of different foundation models (e.g., GPT-3 [9], ViLD [83]) and language-conditioned policies, using language as the common interface. While Socratic Models has been demonstrated on a tabletop object manipulation task, Inner Monologue examines additional challenges for robots operating in dynamic environments, which require closed-loop feedback to the planner.

3 Leveraging Embodied Language Feedback with Inner Monologue

We consider the setting where an embodied robotic agent attempts to perform a high-level natural language instruction i. This robotic agent is only capable of executing short-horizon skills from a library of previously trained policies $\pi_k \in \Pi$ with short language descriptions ℓ_k, which may be trained with reinforcement learning or behavioral cloning. The "planner," which is a pretrained LLM [20, 21], attempts to find a sequence of skills to accomplish the instruction. To observe the environment, the planner has access to textual feedback o from the environment that can be appended to the instruction or requested by the planner. Our work studies to what extent the LLM planner is able to reason over and utilize such feedback to "close the loop" with the environment and improve planning.

3.1 Inner Monologue

We formulate an "inner monologue" by continually injecting information from the various sources of feedback into the LLM planning language prompts as the robot interacts with the environment. While LLMs have demonstrated exceptional planning capabilities for embodied control tasks [20], prior works have found it crucial to ground LLM predictions with external components such as affordance functions [21] in order to produce useful plans that are executable by robots. However, LLMs used in this context have thus far remained one-directional – providing a list of skills, without making corrections or leveraging opportunities to replan accordingly. In contrast, Inner Monologue studies settings where grounded environment feedback



Human: move all the blocks into mismatching bowls.
Scene: There is a yellow block, yellow bowl, blue block, blue bowl, red block, red bowl.
Robot: My goal is ['yellow block in blue bowl', 'red block in yellow bowl', 'blue block in red bowl'].
Robot: Pick up yellow block and place it in blue bowl.
Scene: You achieved ['yellow block in blue bowl']
Robot: I need red block in yellow bowl, blue block in red bowl
Robot: Pick up red block and place it in yellow bowl.

Human: Sort fruits and condiments in different plates
Scene: objects = [apple, egg, plum, pizza, cookie, salt, mustard, ketchup, lemon]
Scene: occluded objects = []
Robot: robot.pickplace(strawberry, green plate)
Robot: robot.pickplace(salt shaker, blue plate)
Success Detector: False.
Robot: robot.pickplace(salt shaker, blue plate)
...
Robot: robot.stop()

Human: Bring me a drink from the table
Robot: go to the table
Scene: I see a coke and lime soda
Robot: What kind of drink would you like?
Human: Something with caffeine
Robot: Pick up coke
Success Detector: False
Robot: Pick up coke
Robot: Bring it to you
Robot: Put down the coke
Robot: Done

Figure 3: Different instantiations of Inner Monologue in three distinct domains – simulated tabletop rearrangement (*top*), real-world tabletop rearrangement (*middle*), and real-world kitchen mobile manipulation (*bottom*). Each domain uses different prompts and different feedback models. Sharing across the domains is the same Inner Monologue formulation that uses a pre-trained langauge model to take in a **human instruction** and decompose it into a sequence of **actionable steps** by the agent, while accounting for injected embodied feedback from different models, such as **object recognizers** and **success detectors**. In real-world kitchen mobile manipulation domain (*bottom*), we additionally ground the actions using pre-trained affordance functions built in [21], which do not communicate back to the language model.

is provided directly to the LLM in a closed-loop fashion. This promotes improved LLM reasoning in complex long-horizon settings, even before any external affordance-based grounding methods are applied.

Our analysis assumes textual feedback is provided to the planner, but does not assume a single specific method of fusing LLM planning with low-level robotic control or a specific method of extracting environment feedback into language. Rather than focusing on a particular algorithmic implementation, our aim is to provide a case study on the value of incorporating different types of feedback into closed-loop LLM-based planning. Thus, Inner Monologue in Sec 4 utilizes language feedback within separate systems that incorporate different LLMs, different methods of fusing planning with control, different environments and tasks, and different methods of acquiring control policies. We note that in our specific implementations of Inner Monologue, we use pre-trained LLMs for planning that are not finetuned, but rather evaluated solely with few-shot prompting; the full prompts can be found in the Appendix.

3.2 Sources of Feedback

In theory any type of environment feedback can inform the LLM planner, as long as it can be expressed through language. We focus on the specific forms of feedback shown in Fig 2: (1) task-specific feedback, such as success detection, and (2) scene-specific feedback (either "passive" or "active"), which describes the scene. Specific instantiations and implementation details of each type of feedback can be found in Sec 4.1, Sec 4.2, and Sec 4.3 respectively for each domain.

Success Detection. The *Success* feedback gives binary "yes" or "no" response in language form, specifying whether the low-level skill π_k has succeeded. Engineered success detectors can operate on ground-truth state in simulation, while learned success detectors can be trained on real examples of successes and failures in the real world [84, 85, 86, 87, 88].

Passive Scene Description. We refer broadly to any sources of scene feedback that are consistently and automatically injected into the LLM prompt as Passive Scene Description, which also typically follow some structure. One common type of such feedback is object recognition [89, 90, 91, 92] that returns a list of present objects, to which we refer as *Object* feedback. We also demonstrate the use of a task-progress scene description in the simulated tabletop rearrangement environment, to which we refer as *Scene* feedback.

Active Scene Description. As the proactive counterpart, Active Scene Description encompasses sources of feedback that are provided directly in response to active queries by the LLM planner, which are answered either by a person, or by another pretrained model, such as a Visual Question Answering (VQA) model [93, 94, 95, 96]. Unlike the passive counterpart which are strictly structured and narrow in their scope, this feedback allows the planner to actively gather information relevant to the scene, the task, or even preferences of the user. The combined output we send to the LLM planner includes both the LLM-generated question along with the response. As we aim to investigate *whether* and *how* a LLM planner can incorporate such feedback and wish to study both structured VQA-style human feedback as well as unstructured human preferences feedback, we only consider human-provided response in this work, which we refer to as *Human* feedback.

4 Experimental Results

In order to study how different sources of environment feedback can support a rich inner monologue that enables complex robotic control, we study different Inner Monologue implementations in three environments, each with different LLM and different sources of feedback from the environment: 1) simulated tabletop manipulation (Sec 4.1), 2) real-world tabletop manipulation (Sec 4.2), and 3) real-world mobile manipulation in an office kitchen (Sec 4.3). For more details about the experiment setup and results, please refer to the Appendix.

4.1 Simulated Tabletop Rearrangement

We experiment with Ravens-based [81] environment, where a robotic arm with a gripper is tasked with rearranging blocks and bowls in some desired configuration, specified by natural language. We evaluate each method on four seen tasks and four unseen tasks, where seen tasks may be used for training (in the case of supervised baseline) or used as few-shot prompting.

This instantiation of Inner Monologue uses (i) InstructGPT [9, 97] for planning [20, 21], (ii) scripted modules to provide language feedback in the form of object recognition (*Object*), success detection (*Success*), and task-progress scene description (*Scene*), and (iii) a pre-trained language-conditioned pick-and-place primitive (similar to CLIPort [82] and Transporter Nets [81]). *Object* feedback informs the list of present objects and *Success* feedback informs the success/failure of the most recent action. However, consider the task of stacking multiple blocks, because the unfinished tower of blocks may be knocked over by the robot, it is also critical to reason about overall task progress. Therefore, task-progress scene description (*Scene*) describes the semantic sub-goals inferred by the LLM towards completing the high-level instruction that are achieved by the agent so far.

We additionally compare to a multi-task CLIPort directly trained on long-horizon task instructions. Because CLIPort is a single-step policy and does not terminate spontaneously during policy rollout, we report CLIPort evaluations with oracle termination (i.e., repeat until oracle indicates task completion) and fixed-step termination (i.e., repeat for 15 steps). To simulate real-world disturbances and evaluate the system's robustness to disturbances, we add Gaussian noise to multiple levels of the system at test time: $\mathcal{N}(0,3)$ for pixel observation, $\mathcal{N}(0,2.5)$ for policy primitive (i.e., pick-place pixel heatmaps), $\mathcal{N}(0,0.02m)$ for place locations.

				+LLM	+Inner Monologue	
	Tasks	**CLIPort**	+oracle	*Object*	*Object + Success*	*Object + Scene*
Seen Tasks	"Pick and place"	24.0%	74.0%	80.0%	90.0%	**94.0%**
	"Stack all the blocks"	2.0%	32.0%	4.0%	10.0%	**26.0%**
	"Put all the blocks on the [x] corner/side"	2.0%	32.0%	**30.0%**	28.0%	**30.0%**
	"Put all the blocks in the [x] bowl"	32.0%	94.0%	52.0%	46.0%	**56.0%**
Unseen Tasks	"Put all the blocks in different corners"	0.0%	0.0%	20.0%	20.0%	**26.0%**
	"Put the blocks in their matching bowls"	0.0%	0.0%	56.0%	70.0%	**82.0%**
	"Put the blocks on mismatched bowls"	0.0%	0.0%	62.0%	76.0%	**86.0%**
	"Stack all the blocks on the [x] corner/side"	0.0%	0.0%	0.0%	4.0%	**6.0%**

Table 1: Success rates averaged across 50 episodes in simulated pick-and-place. CLIPort + oracle indicates that CLIPort was provided a "termination" oracle. LLM-informed feedback effectively enable retrying/replanning in the presence of test-time disturbances, while enjoying the generalization benefits of LLMs to unseen tasks.

Analysis. As shown in Table 1, Inner Monologue effectively enables retrying and replanning in the face of test-time disturbances, where *Object* + *Scene* performs the best because of its ability to keep track of sub-goal conditions. Furthermore, this performance directly translates to unseen tasks by leveraging rich semantic knowledge of LLM. Finally, we observe that non-hierarchical and solitary systems such as CLIPort (i) struggle at generalizing to unseen long-horizon tasks under test-time disturbances, and (ii) on training tasks, an oracle is also often required to indicate task completion for good performance.

4.2 Real-World Tabletop Rearrangement

We evaluate Inner Monologue on a real-world robot platform designed to resemble the simulation experiments. This instantiation uses (i) InstructGPT [9, 97] for planning, (ii) MDETR [98] for open-vocab object recognition (*Object*) (iii) heuristics on the object bounding box predictions from MDETR for Success Detection (*Success*), and (iv) a suction-based pick-and-place motion primitive that uses an LLM to parse target objects from a language command (e.g., given by the planner).

We investigate two tasks: (i) a 3-block stacking task where 2 blocks are already pre-stacked, and (ii) a long-horizon sorting task to place food in one plate and condiments in another (where categorizing food versus condiments is autonomously done by the LLM planner). In additional to additional challenges of real-world perception and clutter, we artificially inject Gaussian noise into the policy actions (i.e., add standard deviation σ=4mm clipped at 2σ) to stress test recovery from failures via replanning with grounded closed-loop feedback. Results are presented in Table 2.

	LLM	**+Inner Monologue**		
Task Family	*Object*	*Object*	*Success*	*Object* + *Success*
Finish 3-block stacking	20%	40%	40%	**100%**
Sort fruits from bottles	20%	50%	40%	**80%**
Total	20%	45%	40%	**90%**

Table 2: Success rates averaged across 10 runs in real-world pick-and-place. We observe significant improvement in Inner Monologue with *Object* and *Success* feedback, with the two feedback being complementary to each other.

Analysis. We compare to variants with only *Object* or *Success* feedback, as well as an open-loop variant ("LLM Object") that only runs object recognition once at the beginning of the task (similar to the system demonstrated in [19]). The partial 3-block stacking task highlights an immediate failure mode of the open-loop baseline, where the initial scene description struggles to capture a complete representation of the scene (due to clutter and occlusion) to provide as input to the multi-step planner. As a result, the system only executes one pick-and-place action – and cannot recover from mistakes. To address these shortcomings, Inner Monologue (*Object* + *Success*) leverages closed-loop scene description and success detection after each step, which allows it to successfully replan and recover from policy mistakes.

4.3 Real-World Mobile Manipulator in a Kitchen Setting

We implement Inner Monologue in a robotic system using the kitchen environment and task definitions described in SayCan [21]. The Everyday Robots robot, a mobile manipulator with RGB observations, is placed in an office kitchen to interact with common objects using concurrent [99] continuous closed-loop control.

The baseline, SayCan [21], is a method that plans and acts in diverse real world scenarios by combining an LLM with value functions of control policies. While SayCan creates plans that are grounded by the affordances of value functions, the LLM predictions in isolation are never given any closed-loop feedback.

We use an instantiation of Inner Monologue that uses (i) PALM [8] for planning, (ii) value functions from pre-trained control policies for affordance grounding [21], (iii) a learned visual classification model for *Success* feedback, (iv) human-provided *Object* feedback, and (v) pre-trained control policies for relevant skills in the scene. We also perform a case study where we allow the agent to ask questions and source *Human* feedback directly; results are shown in Fig 5a and the Appendix.

We evaluate on 120 runs over three task families: (1) four manipulation tasks, (2) two dexterous manipulation tasks utilizing drawers, and (3) two long-horizon combined manipulation and navigation tasks. We consider both cases with and without manually-added adversarial disturbances during control policy executions that cause skill policy rollouts to fail. While these failures occur naturally even without perturbances, the adversarial disturbances creates a consistent comparison between methods that requires retrying or replanning to accomplish the original instruction.

		+Inner Monologue	
Task Family	**SayCan**	*Success*	*Object + Success*
No Disturbances			
Manipulation	50.0%	62.5%	**75.0%**
Mobile Manipulation	50.0%	50.0%	**75.0%**
Drawers	83.3%	83.3%	**100.0%**
With Disturbances			
Manipulation	12.5%	25.0%	**33.3%**
Mobile Manipulation	0.0%	25.0%	**75.0%**
Drawers	0.0%	**44.4%**	**44.4%**
Total	30.8%	48.7%	**60.4%**

Table 3: Averaged success rate across 120 evaluations on several task families in our real-world mobile manipulation environment. We consider a standard setting and adversarial setting with external human disturbances. In all cases, LLM-informed embodied feedback is shown to be effective in improving robustness of the system, especially when low-level policies are prone to failures.

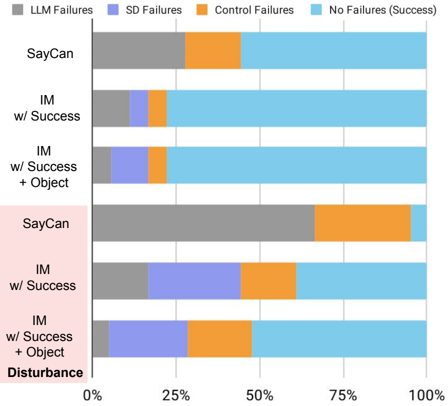


Figure 4: Failure causes on 120 evaluations. When disturbances are added (red), only the Inner Monologue variants consistently complete the instructions.

Analysis. Without adversarial disturbances, the baseline SayCan performs reasonably on all tasks, yet incorporating LLM-informed feedback in Inner Monologue allows further improvement by effectively retrying or replanning under natural failures. The most notable difference is in the cases with adversarial disturbances. Without any LLM-informed feedback SayCan has success rate close to 0% since LLM always assume successful execution of previous skills. Inner Monologue significantly outperforms SayCan because of its ability to invoke appropriate recovery modes depending on the environment feedback. Analysis on the failure causes indicates that *Success* and *Object* feedback can reduce LLM planning failures and thus overall failure rate, albeit at the cost of introducing new failure modes to the system.

4.4 Plan Generalization Capabilities

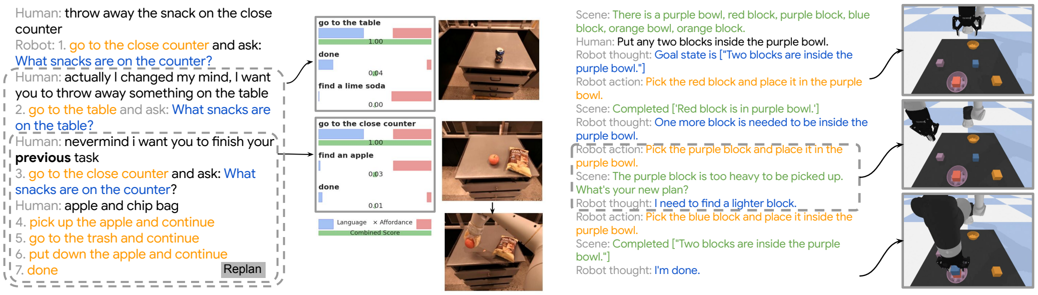
Although LLMs can generate fluent continuation from the prompted examples, we surprisingly find that, Inner Monologue demonstrates many impressive reasoning and replanning behaviors beyond the examples given in the prompt. Using a pre-trained LLM as the backbone, the method also inherits many of the appealing properties from its versatility and general-purpose language understanding. In this section, we demonstrate a few of these capabilities; additional capabilities are shown in Appendix (Fig **??** and Fig **??**).

Continued Adaptation to New Instructions. Although not explicitly prompted, the LLM planner can react to human interaction that changes the high-level goal mid-task. Fig 5a demonstrates a challenging case, where *Human* feedback changes the goal during the plan execution, and then changes the goal yet again by saying “finish the previous task”. In another instance, despite not being explicitly prompted to terminate after a human says “please stop”, the LLM planner generalizes to this scenario and predicts a “done” action.

Self-Proposing Goals under Infeasibility. Instead of mindlessly following human-given instructions, Inner Monologue can also propose alternative goals to achieve when the previous goal becomes infeasible. In Fig 5b, to solve the task “put any two blocks inside the purple bowl”, while the first attempted block is intentionally made too heavy for the robot, Inner Monologue proposes to “find a lighter block” and successfully solves the task.

Multilingual Interaction. Pre-trained LLMs are known to be able to translate from one language to another, without any finetuning. We observe that such multilingual understanding also transfers to the embodied settings. Fig 5c shows a case when an instruction is in Chinese, the LLM planner can still correctly interpret it, re-narrate it as a concrete goal to execute in English, and accordingly replan its future actions. Occasionally, we find that this capability even extends to symbols and emojis.

Retrospective Scene Understanding. We also observe that Inner Monologue demonstrates retrospective scene understanding based on past actions and environment feedback, which requires temporal and embodied reasoning. In Fig 5d, after series of actions, we can turn to ask questions about the resulting scene, again a structure that has not appeared in the prompt.

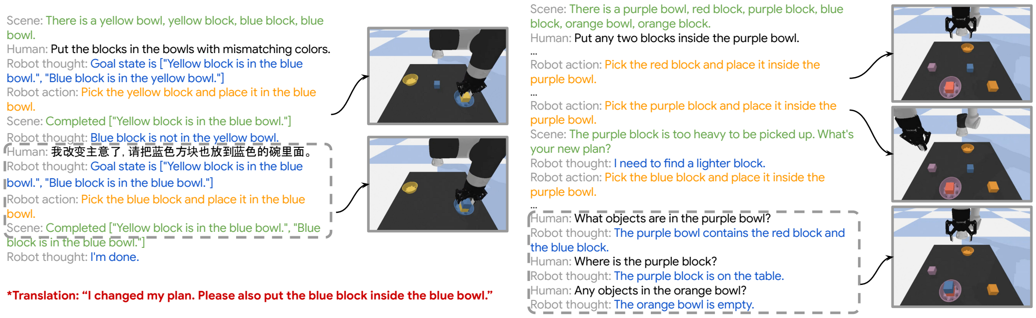


Human: throw away the snack on the close counter
Robot: 1. go to the close counter and ask: What snacks are on the counter?
Human: actually I changed my mind, I want you to throw away something on the table
2. go to the table and ask: What snacks are on the table?
Human: nevermind i want you to finish your **previous** task
3. go to the close counter and ask: What snacks are on the counter?
Human: apple and chip bag
4. pick up the apple and continue
5. go to the trash and continue
6. put down the apple and continue
7. done

Scene: There is a purple bowl, red block, purple block, blue block, orange bowl, orange block.
Human: Put any two blocks inside the purple bowl.
Robot thought: Goal state is ["Two blocks are inside the purple bowl."]
Robot action: Pick the red block and place it in the purple bowl.
Scene: Completed ['Red block is in purple bowl.']
Robot thought: One more block is needed to be inside the purple bowl.
Robot action: Pick the purple block and place it in the purple bowl.
Scene: The purple block is too heavy to be picked up. What's your new plan?
Robot thought: I need to find a lighter block.
Robot action: Pick the blue block and place it inside the purple bowl.
Scene: Completed ["Two blocks are inside the purple bowl."]
Robot thought: I'm done.

(a) Continued Adaptation to New Instructions

(b) Self-Proposing Goals under Infeasibility



Scene: There is a yellow bowl, yellow block, blue block, blue bowl.
Human: Put the blocks in the bowls with mismatching colors.
Robot thought: Goal state is ["Yellow block is in the blue bowl.", "Blue block is in the yellow bowl."]
Robot action: Pick the yellow block and place it in the blue bowl.
Scene: Completed ["Yellow block is in the blue bowl."]
Robot thought: Blue block is not in the yellow bowl.
Human: 我改变主意了，请把蓝色方块也放到蓝色的碗里面。
Robot thought: Goal state is ["Yellow block is in the blue bowl.", "Blue block is in the blue bowl."]
Robot action: Pick the blue block and place it in the blue bowl.
Scene: Completed ["Yellow block is in the blue bowl.", "Blue block is in the blue bowl."]
Robot thought: I'm done.

*Translation: "I changed my plan. Please also put the blue block inside the blue bowl."

Scene: There is a purple bowl, red block, purple block, blue block, orange bowl, orange block.
Human: Put any two blocks inside the purple bowl.
...
Robot action: Pick the red block and place it inside the purple bowl.
...
Robot action: Pick the purple block and place it inside the purple bowl.
Scene: The purple block is too heavy to be picked up. What's your new plan?
Robot thought: I need to find a lighter block.
Robot action: Pick the blue block and place it inside the purple bowl.
...
Human: What objects are in the purple bowl?
Robot thought: The purple bowl contains the red block and the blue block.
Human: Where is the purple block?
Robot thought: The purple block is on the table.
Human: Any objects in the orange bowl?
Robot thought: The orange bowl is empty.

(c) Multilingual Interaction

(d) Interactive Scene Understanding

Figure 5: Informing LLM with embodied feedback enables many generalization capabilities, all of which are achieved without similar prompted examples. For instance, Inner Monologue can continually adapt to new instructions given by humans, propose new goals to achieve when faced with infeasibility for the previous plan, interact with humans in different natural languages, and answer questions about the current scene given past actions and feedback.

Despite the appealing findings about these generalization capabilities, we observe that they are of varying levels of consistency when no similar examples have been provided in the prompt, likely limited by the current capabilities of the language models. However, we believe that further investigations into these behaviors and addressing their limitations would each lead to exciting future directions.

5 Conclusions, Limitations & Future Works

In this work, we investigated the role that environment feedback plays for LLMs reasoning in tasks involving embodied robotic planning and interaction. We presented a general formulation Inner Monologue that combines different sources of environment feedback with methods fusing LLM planning with robotic control policies and studied its instantiations in three distinct domains. We found that environment feedback significantly improves high-level instruction completion, especially in challenging scenarios with adversarial disturbances. Finally, we analyze generalization capabilities of Inner Monologue that highlight how closed-loop language feedback enables replanning even in complex unseen settings.

Limitations. In Sec 4.1 and Sec 4.3, we assume access to oracle scene descriptors in the form of human observers or scripted systems to provide textual description back to the LLM planner. We study the viability of learned systems scene description and object recognition in Appendix Table **??**. As for failure modes, Inner Monologue may fail due to several sources of errors: (1) success detections, (2) LLM planning errors, and (3) control errors. False negative predictions from the success detector lead to additional retry attempts, while false positive predictions add adversarial partial observability to the environment. In some instances, we found that the LLM planners ignored the environment feedback and still proposed policy skills involving objects not present in the scene. Additionally, the performance of low-level control policies limits not only overall high-level instruction completion performance, but also limits the scope of tasks that the LLM is able to plan actions for.

Future Works. Several fronts can be improved by future works. First, with advances in image/video captioning and visual-question answering, a fully automated system of Inner Monologue can be implemented without a human in the loop as an oracle. Second, improvements can be made on how to aggregate potentially inaccurate sources of information, such as using text to describe the uncertainty of the feedback modules, or including additional feedback modules for safety and ethics for the proposed plans. Finally, enabling low-level control policies to take as input the textual feedback by LLM also leads to exciting future directions.

Acknowledgments

The authors would like to thank Kanishka Rao and Vincent Vanhoucke for valuable feedback and discussions. In addition, the authors would like to acknowledge the large team who built [21], upon which we construct our Kitchen Mobile Manipulation experiments.

References

[1] L. P. Kaelbling and T. Lozano-Pérez. Integrated task and motion planning in belief space. *The International Journal of Robotics Research*, 32(9-10):1194–1227, 2013.

[2] A. G. Barto and S. Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete event dynamic systems*, 13(1):41–77, 2003.

[3] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, and S. Riedel. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019.

[4] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.

[5] J. Davison, J. Feldman, and A. M. Rush. Commonsense knowledge mining from pretrained models. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 1173–1178, 2019.

[6] A. Talmor, Y. Elazar, Y. Goldberg, and J. Berant. olmpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758, 2020.

[7] A. Roberts, C. Raffel, and N. Shazeer. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*, 2020.

[8] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[9] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[10] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. Chi, Q. Le, and D. Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

[11] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*, 2022.

[12] A. K. Lampinen, I. Dasgupta, S. C. Chan, K. Matthewson, M. H. Tessler, A. Creswell, J. L. McClelland, J. X. Wang, and F. Hill. Can language models learn from explanations in context? *arXiv preprint arXiv:2204.02329*, 2022.

[13] M. Nye, A. J. Andreassen, G. Gur-Ari, H. Michalewski, J. Austin, D. Bieber, D. Dohan, A. Lewkowycz, M. Bosma, D. Luan, et al. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*, 2021.

[14] L. S. Vygotsky. *Thought and language*. MIT press, 2012.

[15] P. Carruthers. Thinking in language?: evolution and a modularist possibility. Cambridge University Press, 1998.

[16] L. Vygotsky. Tool and symbol in child development. *The vygotsky reader*, 1994.

[17] L. S. Vygotsky. Play and its role in the mental development of the child. *Soviet psychology*, 5(3): 6–18, 1967.

[18] C. Colas, T. Karch, C. Moulin-Frier, and P.-Y. Oudeyer. Vygotskian autotelic artificial intelligence: Language and culture internalization for human-like ai. *arXiv preprint arXiv:2206.01134*, 2022.

[19] A. Zeng, A. Wong, S. Welker, K. Choromanski, F. Tombari, A. Purohit, M. Ryoo, V. Sindhwani, J. Lee, V. Vanhoucke, et al. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv preprint arXiv:2204.00598*, 2022.

[20] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*. PMLR, 2022.

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

[22] L. P. Kaelbling and T. Lozano-Pérez. Hierarchical planning in the now. In *Workshops at the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[23] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel. Combined task and motion planning through an extensible planner-independent interface layer. In *2014 IEEE international conference on robotics and automation (ICRA)*, 2014.

[24] R. E. Fikes and N. J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 1971.

[25] E. D. Sacerdoti. A structure for plans and behavior. Technical report, SRI International, Menlo Park California Artificial Intelligence Center, 1975.

[26] D. Nau, Y. Cao, A. Lotem, and H. Munoz-Avila. Shop: Simple hierarchical ordered planner. In *Proceedings of the 16th international joint conference on Artificial intelligence*, 1999.

[27] S. M. LaValle. *Planning algorithms*. Cambridge university press, 2006.

[28] M. Toussaint. Logic-geometric programming: An optimization-based approach to combined task and motion planning. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[29] M. A. Toussaint, K. R. Allen, K. A. Smith, and J. B. Tenenbaum. Differentiable physics and stable modes for tool-use and manipulation planning. *Robotics: Science and Systems Foundation*, 2018.

[30] B. Eysenbach, R. R. Salakhutdinov, and S. Levine. Search on the replay buffer: Bridging planning and reinforcement learning. *Advances in Neural Information Processing Systems*, 2019.

[31] D. Xu, S. Nair, Y. Zhu, J. Gao, A. Garg, L. Fei-Fei, and S. Savarese. Neural task programming: Learning to generalize across hierarchical tasks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[32] D. Xu, R. Martín-Martín, D.-A. Huang, Y. Zhu, S. Savarese, and L. F. Fei-Fei. Regression planning networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[33] T. Silver, R. Chitnis, N. Kumar, W. McClinton, T. Lozano-Perez, L. P. Kaelbling, and J. Tenenbaum. Inventing relational state and action abstractions for effective and efficient bilevel planning. *arXiv preprint arXiv:2203.09634*, 2022.

[34] D. Shah, P. Xu, Y. Lu, T. Xiao, A. Toshev, S. Levine, and B. Ichter. Value function spaces: Skill-centric state abstractions for long-horizon reasoning. *ICLR*, 2022. URL `https://openreview.net/pdf?id=vgqS1vkkCbE`.

[35] A. Srinivas, A. Jabri, P. Abbeel, S. Levine, and C. Finn. Universal planning networks: Learning generalizable representations for visuomotor control. In *International Conference on Machine Learning*, pages 4732–4741. PMLR, 2018.

[36] T. Kurutach, A. Tamar, G. Yang, S. J. Russell, and P. Abbeel. Learning plannable representations with causal infogan. *Advances in Neural Information Processing Systems*, 31, 2018.

[37] A. Akakzia, C. Colas, P.-Y. Oudeyer, M. Chetouani, and O. Sigaud. Grounding language to autonomously-acquired skills via goal generation. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=chPj_I5KMHG`.

[38] S. Pirk, K. Hausman, A. Toshev, and M. Khansari. Modeling long-horizon tasks as sequential interaction landscapes. *arXiv preprint arXiv:2006.04843*, 2020.

[39] T. Kollar, S. Tellex, D. Roy, and N. Roy. Toward understanding natural language directions. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 259–266. IEEE, 2010.

[40] S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, and N. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, pages 1507–1514, 2011.

[41] M. Bollini, S. Tellex, T. Thompson, N. Roy, and D. Rus. Interpreting and executing recipes with a cooking robot. In *Experimental Robotics*, pages 481–495. Springer, 2013.

[42] S. Tellex, R. Knepper, A. Li, D. Rus, and N. Roy. Asking for help using inverse semantics. 2014.

[43] T. Kollar, S. Tellex, D. Roy, and N. Roy. Grounding verbs of motion in natural language commands to robots. In *Experimental robotics*, pages 31–47. Springer, 2014.

[44] V. Blukis, Y. Terme, E. Niklasson, R. A. Knepper, and Y. Artzi. Learning to map natural language instructions to physical quadcopter control using simulated flight. *arXiv preprint arXiv:1910.09664*, 2019.

[45] S. Nair and C. Finn. Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation. *ArXiv*, abs/1909.05829, 2020.

[46] F. Xia, C. Li, R. Martín-Martín, O. Litany, A. Toshev, and S. Savarese. Relmogen: Integrating motion generation in reinforcement learning for mobile manipulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[47] C. Li, F. Xia, R. Martin-Martin, and S. Savarese. Hrl4in: Hierarchical reinforcement learning for interactive navigation with mobile manipulators. In *Conference on Robot Learning*, 2020.

[48] Y. Jiang, S. Gu, K. Murphy, and C. Finn. Language as an abstraction for hierarchical deep reinforcement learning. In *NeurIPS*, 2019.

[49] D. Hafner, K.-H. Lee, I. Fischer, and P. Abbeel. Deep hierarchical planning from pixels. *arXiv preprint arXiv:2206.04114*, 2022.

[50] S. Mirchandani, S. Karamcheti, and D. Sadigh. Ella: Exploration through learned language abstraction. *Advances in Neural Information Processing Systems*, 34:29529–29540, 2021.

[51] P. A. Jansen. Visually-grounded planning without vision: Language models infer detailed plans from high-level instructions. *arXiv preprint arXiv:2009.14259*, 2020.

[52] P. Sharma, A. Torralba, and J. Andreas. Skill induction and planning with latent language. *arXiv preprint arXiv:2110.01517*, 2021.

[53] V. Micheli and F. Fleuret. Language models are few-shot butlers. *arXiv preprint arXiv:2104.07972*, 2021.

[54] S. Li, X. Puig, Y. Du, C. Wang, E. Akyurek, A. Torralba, J. Andreas, and I. Mordatch. Pre-trained language models for interactive decision-making. *arXiv preprint arXiv:2202.01771*, 2022.

[55] V. Shwartz, P. West, R. L. Bras, C. Bhagavatula, and Y. Choi. Unsupervised commonsense question answering with self-talk. *arXiv preprint arXiv:2004.05483*, 2020.

[56] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[57] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[58] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[59] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[60] C. Paxton, Y. Bisk, J. Thomason, A. Byravan, and D. Foxl. Prospection: Interpretable plans from language by predicting the future. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6942–6948. IEEE, 2019.

[61] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. *Advances in Neural Information Processing Systems*, 33:13139–13150, 2020.

[62] V. Blukis, R. A. Knepper, and Y. Artzi. Few-shot object grounding and mapping for natural language robot instruction following. *arXiv preprint arXiv:2011.07384*, 2020.

[63] C. Lynch and P. Sermanet. Language conditioned imitation learning over unstructured data. *Robotics: Science and Systems*, 2021. URL `https://arxiv.org/abs/2005.07648`.

[64] Y. Chen, R. Xu, Y. Lin, and P. A. Vela. A joint network for grasp detection conditioned on natural language commands. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4576–4582. IEEE, 2021.

[65] O. Mees, L. Hermann, and W. Burgard. What matters in language conditioned robotic imitation learning. *arXiv preprint arXiv:2204.06252*, 2022.

[66] C. Yan, F. Carnevale, P. Georgiev, A. Santoro, A. Guy, A. Muldal, C.-C. Hung, J. Abramson, T. Lillicrap, and G. Wayne. Intra-agent speech permits zero-shot task acquisition. *arXiv preprint arXiv:2206.03139*, 2022.

[67] G. Kuhlmann, P. Stone, R. Mooney, and J. Shavlik. Guiding a reinforcement learner with natural language advice: Initial results in robocup soccer. In *The AAAI-2004 workshop on supervisory control of learning and adaptive systems*. San Jose, CA, 2004.

[68] A. Najar, O. Sigaud, and M. Chetouani. Interactively shaping robot behaviour with unlabeled human instructions. *Autonomous Agents and Multi-Agent Systems*, 34(2):1–35, 2020.

[69] P. Sharma, B. Sundaralingam, V. Blukis, C. Paxton, T. Hermans, A. Torralba, J. Andreas, and D. Fox. Correcting robot plans with natural language feedback. *arXiv preprint arXiv:2204.05186*, 2022.

[70] R. Loftin, B. Peng, J. MacGlashan, M. L. Littman, M. E. Taylor, J. Huang, and D. L. Roberts. Learning behaviors via human-delivered discrete feedback: modeling implicit feedback strategies to speed up learning. *Autonomous agents and multi-agent systems*, 30(1):30–59, 2016.

[71] 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*, pages 8748–8763. PMLR, 2021.

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

[73] J. Lu, D. Batra, D. Parikh, and S. Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

[74] Z. Wang, J. Yu, A. W. Yu, Z. Dai, Y. Tsvetkov, and Y. Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021.

[75] A. Suglia, Q. Gao, J. Thomason, G. Thattai, and G. Sukhatme. Embodied bert: A transformer model for embodied, language-guided visual task completion. *arXiv preprint arXiv:2108.04927*, 2021.

[76] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33: 22243–22255, 2020.

[77] A. Jain, M. Guo, K. Srinivasan, T. Chen, S. Kudugunta, C. Jia, Y. Yang, and J. Baldridge. Mural: multimodal, multitask retrieval across languages. *arXiv preprint arXiv:2109.05125*, 2021.

[78] J. Sun, D.-A. Huang, B. Lu, Y.-H. Liu, B. Zhou, and A. Garg. Plate: Visually-grounded planning with transformers in procedural tasks. *IEEE Robotics and Automation Letters*, 7(2):4924–4930, 2022.

[79] F. Sener and A. Yao. Zero-shot anticipation for instructional activities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 862–871, 2019.

[80] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14829–14838, 2022.

[81] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, and J. Lee. Transporter networks: Rearranging the visual world for robotic manipulation. *Conference on Robot Learning (CoRL)*, 2020.

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

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

[84] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015.

[85] F.-J. Chu, R. Xu, and P. A. Vela. Real-world multiobject, multigrasp detection. *IEEE Robotics and Automation Letters*, 3(4):3355–3362, 2018.

[86] D. Kalashnikov, J. Varley, Y. Chebotar, B. Swanson, R. Jonschkowski, C. Finn, S. Levine, and K. Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv preprint arXiv:2104.08212*, 2021.

[87] T. Migimatsu and J. Bohg. Grounding predicates through actions. *arXiv preprint arXiv:2109.14718*, 2021.

[88] Y. Cui, S. Niekum, A. Gupta, V. Kumar, and A. Rajeswaran. Can foundation models perform zero-shot task specification for robot manipulation? In *Learning for Dynamics and Control Conference*, pages 893–905. PMLR, 2022.

[89] M. Liang and X. Hu. Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3367–3375, 2015.

[90] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[91] Z. Zou, Z. Shi, Y. Guo, and J. Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019.

[92] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[93] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[94] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. Corso, and J. Gao. Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13041–13049, 2020.

[95] H. Cai, C. Gan, T. Wang, Z. Zhang, and S. Han. Once for all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations*, 2020. URL `https://arxiv.org/pdf/1908.09791.pdf`.

[96] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.

[97] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

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

[99] T. Xiao, E. Jang, D. Kalashnikov, S. Levine, J. Ibarz, K. Hausman, and A. Herzog. Thinking while moving: Deep reinforcement learning with concurrent control. *arXiv preprint arXiv:2004.06089*, 2020.