2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Toward Grounded Commonsense Reasoning

Minae Kwon, Hengyuan Hu, Vivek Myers†, Siddharth Karamcheti, Anca Dragan†, Dorsa Sadigh
Stanford University, UC Berkeley†
{mnkwon, hengyuan, skaramcheti, dorsa}@cs.stanford.edu,
{vmyers, anca}@berkeley.edu†

***Abstract*— Consider a robot tasked with tidying a desk with a meticulously constructed Lego sports car. A human may recognize that it is not appropriate to disassemble the sports car and put it away as part of the "tidying." How can a robot reach that conclusion? Although large language models (LLMs) have recently been used to enable commonsense reasoning, grounding this reasoning in the real world has been challenging. To reason in the real world, robots must go beyond passively querying LLMs and *actively gather information from the environment* that is required to make the right decision. For instance, after detecting that there is an occluded car, the robot may need to actively perceive the car to know whether it is an advanced model car made out of Legos or a toy car built by a toddler. We propose an approach that leverages an LLM and vision language model (VLM) to help a robot actively perceive its environment to perform grounded commonsense reasoning. To evaluate our framework at scale, we release the MESSYSURFACES dataset which contains images of 70 real-world surfaces that need to be cleaned. We additionally illustrate our approach with a robot on 2 carefully designed surfaces. We find an average 12.9% improvement on the MESSYSURFACES benchmark and an average 15% improvement on the robot experiments over baselines that do not use active perception. The dataset, code, and videos of our approach can be found at https://minaek.github.io/grounded_commonsense_reasoning/.**

I. INTRODUCTION

Imagine you are asked to clean up a desk and you see a meticulously constructed Lego sports car on it. You might immediately recognize that the normative behavior is to leave the car be, rather than taking it apart and putting it away as part of the "cleaning". But how would a robot in that same position know that's the right thing to do? Traditionally, we would expect this information to be specified in the robot's objective – either learned from demonstrations [1], [2], [3] or from human feedback [4], [5], [6], [7]. While a robot could expensively query a human for their preferences on how to clean the car, we explore a different question in this work: how can we equip robots with the commonsense reasoning necessary to follow normative behavior *in the absence of personalized input from the human*? The ability to behave in a commonsense, normative manner can be an effective prior over robot behavior when personalized feedback is not present. When feedback is present, having a good prior can reduce the amount of human specification needed.

Recent work has demonstrated that large language models (LLMs) trained on internet data have enough context for commonsense reasoning [8], making moral judgements [9], [10], or acting as a proxy reward function capturing human preferences [11]. Rather than explicitly asking a human for the answer, the robot could instead ask an LLM whether it would be appropriate to clean up the car. But in real-world environments, this is easier said than done. Tapping into an LLM's commonsense reasoning skills in the real-world requires the ability to *ground language in the robot's perception of the world* – an ability that might be afforded by powerful vision-and-language models (VLMs). Unfortunately, we find that today's VLMs cannot reliably provide all the relevant information for commonsense reasoning. For instance, a VLM may not describe that the sports car is constructed from Legos, or that it contains over 1000 pieces – details that are key to making decisions. While advanced multi-modal models might alleviate this problem, a fundamental limitation is the image itself might not contain all the relevant information. If the sports car is partially occluded by a bag (as in Fig. 1), no VLM could provide the necessary context for reasoning over what actions to take. Such a system would instead need the ability to move the bag – or move *itself* – to actively gather the necessary information. Thus, in order to perform "grounded commonsense reasoning" robots must go beyond passively querying LLMs and VLMs to obtain action plans and instead *directly interact with the environment.* Our insight is that robots must reason about what additional information they need to make appropriate decisions, *and then actively perceive the environment to gather that information*.

Acting on this insight, we propose a framework to enable a robot to perform grounded commonsense reasoning by iteratively identifying details it still needs to clarify about the scene before it can make a decision (e.g. is the model car made out of intricate Lego pieces or MEGA Bloks?) and actively gathering new observations to help answer those questions (e.g. getting a close up of the car from a better angle). In this paper, we focus on the task of cleaning up real-world surfaces through commonsense reasoning. Our framework is shown in Fig. 1. Given a textual description of the desk, an LLM asks follow-up questions about the state of each object that it needs in order to make a decision of what the robot should do with that object. The robot actively perceives the scene by taking close-up photos of each object from angles suggested by the LLM. The follow-up questions and close-up photos are then given to a VLM so that it can provide more information about the scene.

This process can be repeated multiple times. The LLM then decides on an action the robot should take to clean

979-8-3503-8457-4/24/$31.00 ©2024 IEEE

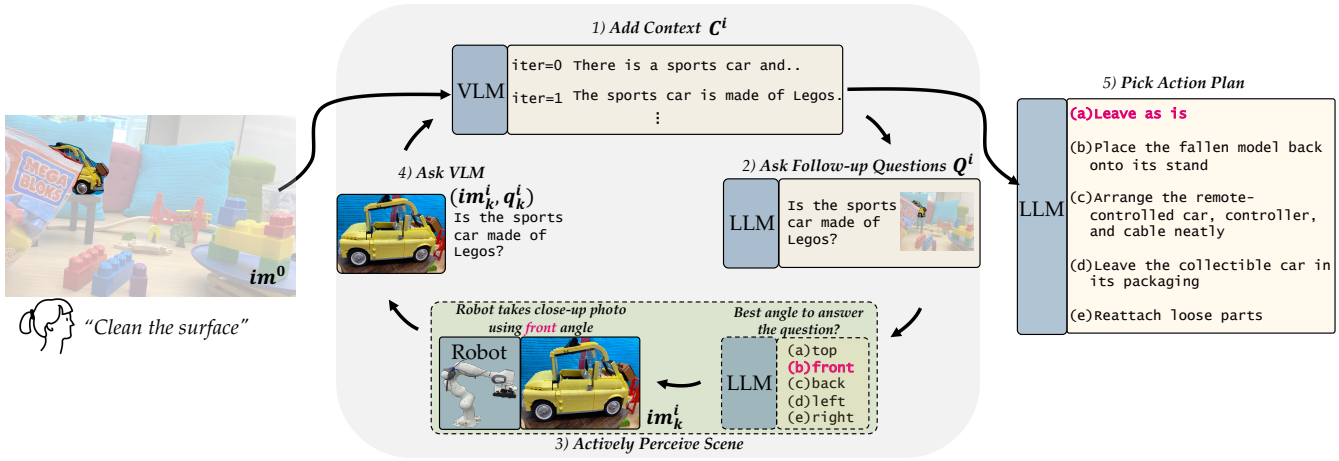

Fig. 1. **Grounded Commonsense Reasoning Framework.** We demonstrate our framework using the sports car. Blue boxes indicate the model and yellow boxes indicate its output. Our framework takes an image of the scene and an instruction as input. 1) The VLM outputs an initial description of the scene $\mathscr{C}^0$ from the initial image im^0. 2) The LLM asks follow-up questions about each object in the scene, $\mathscr{Q}^i$. 3) The robot takes a close-up image im_k^i of each object k. It is guided by an LLM that chooses the best angle that would help answer the question. 4) We pair the close-up images with the follow-up questions and ask the VLM to answer them. Answers are appended to the context. We repeat steps 1-4 to gather more information. 5) We query an LLM to choose the most appropriate way to tidy the object.

the object in an appropriate manner. For example, our robot leaves the Lego sports car intact, throws a browning half-eaten banana in the trash, but keeps an unopened can of Yerba Mate on the desk. Furthermore, we release the MESSYSURFACES dataset containing images of 70 surfaces as well an evaluation benchmark that assesses how well a robot can clean up each surface in an appropriate manner. The dataset is available here.

We evaluate our framework on our benchmark dataset as well as on a real-world robotic system. We examine each component of our framework, asking whether the robot asks useful follow-up questions, whether the robot chooses informative close-up images, and whether the images actually help a VLM more accurately answer questions. We find an average 12.9% improvement on the MESSYSURFACES benchmark and an average 15% improvement on the robot experiments over baselines that do not use active perception.

II. RELATED WORK

a) Commonsense Reasoning: Large language models are trained on internet-scale data, making them effective commonsense reasoners [12], [13], [14], [15]. Prior works have studied whether LLMs' commonsense reasoning aligns with human values [9], [10], [16], [11]. There is evidence that when LLMs make moral or social judgements, they align with the normative beliefs of the population that generated their training data [17]. In addition, prior work show commonsense reasoning models can align with conventional beliefs [18], [19], [20], [21]. *Our approach is in line with commonsense reasoning; instead of adapting to individual preferences, we show we can take commonsense actions to clean up a scene.*

b) Learning Human Preferences: Past work on aligning with human preferences has focused on using human feedback to infer rewards and policies by designing queries for active preference learning [22], [4], [6], [23], performing inverse reinforcement learning [24], [25], or recovering reward signals from language feedback [11], [26], [27], [28], [29]. Policies defined via LLMs have also been directly tuned with language feedback by approaches like RLHF [30]. Instead of querying humans, we leverage normative values from pre-trained models. While some works use normative values from LLMs in negotiations and games [31], these are not grounded in the real world. *In this work, we do not focus on particular human preferences, though the normative responses of LLMs could be fine-tuned for particular applications.*

c) Active Perception: When robots must use commonsense reasoning like humans, active information gathering may be important [32]. Approaches like TidyBot actively zoom-in on objects to better categorize them [33]. Other approaches such as Inner Monologue seek out additional environment information, but need aid from a human annotator or assume access to simulators [34], [35]. VLMs have also been used for active perception in navigation [36], [37], [38]. *In this work, we show that active perception is necessary for grounded commonsense reasoning, enabled by the semantic knowledge in an LLM.*

d) LLMs for Robotics: Past work uses semantic knowledge in LLMs for task planning. Methods like SayCan decompose natural language tasks into primitive action plans [39], [40], [41]. In addition, approaches such as Code as Policies [42], [43] use LLMs to write Python programs that plan with executable robot policy code. Other approaches use multimodal sequence models to reason about language-conditioned manipulation [44], [45], [46], [47]. *We use the semantic awareness of an LLM to reason about action plans. Unlike the above works, an LLM interactively queries an off-the-shelf VLM to ground the scene.*

III. Grounding Commonsense Reasoning

We propose a framework that combines existing foundation models in a novel way to enable active information gathering, shown in Fig. 1. Our framework makes multiple calls to an LLM and VLM to gather information. The LLM plays a number of distinct roles in our framework that we distinguish below: generating informative follow-up questions, guiding active perception, and choosing an action plan. In every call, the LLM takes in and outputs a string $\texttt{LLM}: A^* \to A^*$, and the VLM takes in an image, string pair and outputs a string $\texttt{VLM}: \mathscr{I} \times A^* \to A^*$, where A^* is the set of all strings and $\mathscr{I}$ is the set of all images. The context $\mathscr{C}^i \in A^*$ contains information about the scene that the robot has gathered up to iteration i of the framework. Initially, the inputs to our framework are an image of the scene $im^0 \in \mathscr{I}$ (i.e., an unblurred image from Fig. 1) and an instruction (e.g., "clean the surface").

VLM Describes the Scene. Our framework starts with the VLM producing an initial description $\mathscr{C}^0$ of the scene from the scene image im^0. The description can contain varying amounts of information — in the most uninformative case, it may simply list the objects that are present. In our experiments, this is the description that we use.

LLM Generates Follow-Up Questions. To identify what information is missing from $\mathscr{C}^0$, we use an LLM to generate informative follow-up questions as shown in stage (2) of Fig. 1. We prompt an LLM with $\mathscr{C}^0$ and ask the LLM to produce a set of follow-up questions $\mathscr{Q}^i = \{q_1^i, \ldots, q_K^i\}$ for the K objects. LLMs are apt for this task because of their commonsense reasoning abilities. We use Chain-of-Thought prompting [48] where we first ask the LLM to reason about the appropriate way to tidy each object before producing a follow-up question (see examples in the supplementary). For example, the LLM could reason that the sports car should be put away if it is a toy but left on display if someone built it. The resulting follow-up question asks whether the sports car is built with Lego blocks. We assume that the information in $\mathscr{C}^0$ is accurate (i.e., correctly lists the names of all the objects) to prevent the LLM from generating questions based on inaccurate information.

Robot Actively Perceives the Scene. At this stage, one might normally query the VLM with the original scene image im^0. However if the object-in-question is obstructed or too small to see, the scene image might not provide enough information for the VLM to answer the follow-up question accurately (e.g., the sports car is obstructed in Fig. 1). Instead, we would like to provide an unobstructed close-up image $im_k^i \in \mathscr{I}$ of the object k to "help" the VLM accurately answer the generated questions. Taking informative close-up images requires interaction with the environment — something we can use a robot for.

To actively gather information, the robot should proceed based on some notion of "informativeness" of camera angles. To determine "informativeness", we can again rely on the commonsense knowledge of LLMs. Although LLMs don't have detailed visual information about the object, they can suggest reasonable angles that will be, on average, informative. For instance, an LLM will choose to take a photo from the top of an opaque mug, instead of its sides, to see its content. In practice, we find that this approach works well and can improve the informativeness of an image by 8%. We query an LLM to choose a close-up angle of the object from a set of angles {<FRONT>, <BACK>, <LEFT>, <RIGHT>, <TOP>} that would give an unobstructed view. We then pair the close-up images with their questions $\{(im_1^i, q_1^i), \ldots, (im_k^i, q_K^i)\}$ and query the VLM for answers to these questions in step (4) of our framework. We concatenate the VLM's answers for each object and append them to our context $\mathscr{C}^i$ to complete the iteration. To gather more information about each object, steps $1-4$ can be repeated where the number of iterations is a tunable parameter.

LLM Chooses an Action Plan. In the final step, for each object, we prompt the LLM with the context $\mathscr{C}^i$ and a multiple choice question that lists different ways to tidy an object. The LLM is then instructed to choose the most appropriate option. The multiple choice options come from the MessySurfaces benchmark questions, a bank of 308 multiple-choice questions about how to clean up real-life objects found on messy surfaces. For example, in Fig. 1, the LLM chooses to leave the sports car as is because it infers that the sports car must be on display. To map the natural language action to robot behavior, we implement a series of hand-coded programmatic skill primitives that define an API the LLM can call into. See §V for more details.

IV. The MessySurfaces Dataset

To assess a robot's ability to perform commonsense reasoning in grounded environments, we introduce the MessySurfaces dataset. The dataset consists of images of 308 objects across 70 real-world surfaces that need to be cleaned. An average of 68% of objects are occluded in scene-level images[1], so we also provide 5 close-up images as a way for the robot to "actively perceive" the object, see Fig. 2 for an example. MessySurfaces also includes a benchmark evaluation of multiple choice questions for each object where each option corresponds to different ways to tidy the object. Through a consensus of 5 human annotators, we determine which one of the choices is the most appropriate. To do well, a robot must reason about the appropriate way to clean each object from the images alone. Since no human preferences are given, the robot must identify relevant attributes of each object from the images (e.g., is the sports car built out of Legos or MEGA Bloks?) and then reason about how to tidy the object using this information. MessySurfaces contains 45 office desks, 4 bathroom counters, 5 bedroom tables, 8 kitchen counters, 4 living room tables and 4 dining tables.

a) Data Collection Process: We recruited 51 participants to provide images of cluttered surfaces. Each participant was asked to pick 4 – 6 objects on a surface. They were

[1]Computed as the average number of times annotators indicated a question cannot be answered by the scene image.

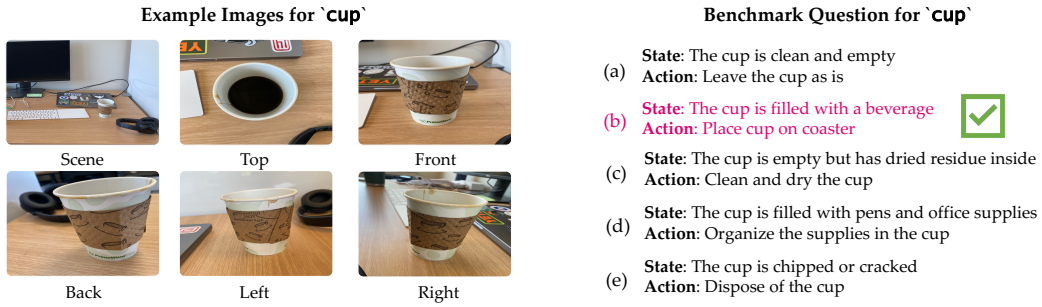

Fig. 2. **MessySurfaces Example.** Each object in MessySurfaces is represented by a scene image and 5 close-up images. Each object also has a benchmark question that presents 5 options to tidy the object; each option is constructed by producing a cleaning action conditioned on a hypothetical object state.

then asked to take a photo of the scene-level view as well as close-up photos of each object from the top, right, left, front, and back angles – the offline equivalent of having a robot actively navigate a scene. The task took approximately 15 – 30 minutes. After receiving the photos, we post-processed each image and cropped out any identifiable information.

b) Benchmark Evaluation: The benchmark questions consist of 5 LLM-generated multiple choice options about how to manipulate each object to clean the surface in an appropriate manner. To make the options diverse, we asked the LLM to first identify 5 states the object could be in and then queried it to come up with a cleaning action for each of those states (see Fig. 2 for an example). For each question, we recruited 5 annotators to choose the correct state-action pair based on the scene and close-up images of the object. Annotators were also given an option to indicate if none of the choices were a good fit. We used the majority label as our answer and omitted 16 questions (out of 324) where a majority thought none of the choices were a good fit. For questions that had two equally popular answers, we counted both as correct. Our annotators agreed on average 67% of the time. To evaluate the quality of our multiple choice options, we asked annotators to rate how appropriate each cleaning action is for each object state. Annotators gave each option an average rating of 4.1 out of 5. The average rating for the correct option was 4.4 out of 5. *Annotators.* In total, we recruited 350 English-speaking annotators from Prolific that were based in the U.S. or U.K. with an approval rating of at least 98%. Our study is IRB-approved.

V. Experiments

We examine how well our approach can perform grounded commonsense reasoning on the MessySurfaces dataset as well as a real-world robotic system.

Primary Metric. We use accuracy on the benchmark questions as our primary metric. Each benchmark question presents 5 options on how to tidy the object, with accuracy defined as the percentage by which our framework selects the most appropriate option (as indicated by our annotators).

Baselines. Key to our approach (■ **Ours-LLM**) is the ability to supplement missing information by asking questions and actively perceiving the environment. To evaluate this, we compare the following:

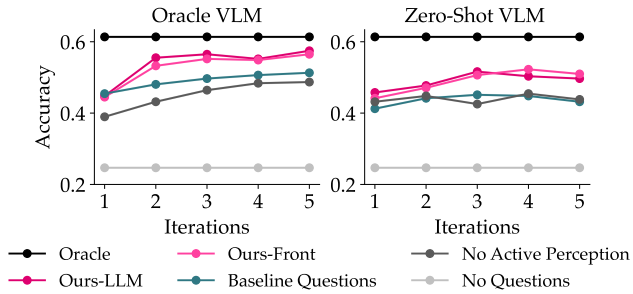

Fig. 3. **MessySurfaces Benchmark Accuracy.** For both the Oracle VLM and InstructBLIP, on average, our approach outperforms all baselines on the MessySurfaces benchmark. Accuracy is given by the percentage by which our framework selects the most appropriate (as indicated by our annotators) way to tidy each object.

- ■ **Oracle.** We ask a human annotator to answer the benchmark questions where they can actively perceive the scene using all angles.
- ■ **Ours-LLM.** Our approach as described in §III.
- ■ **Ours - Front.** Inspired by TidyBot [33], this is a variant of our approach wherein we simulate "zooming" into the image, using the "front" angle image as input to the VLM. The "front" angles can be the most informative angle in many cases, making it an effective heuristic.
- ■ **Baseline Questions.** This baseline evaluates the need for normative commonsense reasoning by asking more factual questions (e.g., "What color is the cup?").
- ■ **No Active Perception.** This baseline evaluates the need for active perception in our framework by allowing the robot to ask questions *that are answered solely from the scene image*.
- ■ **No Questions.** This baseline requires the robot to perform grounded commonsense reasoning from an initial description of the scene. The robot does not ask questions or actively perceive the environment, instead operating in an open-loop fashion akin to methods like SayCan [39].

Implementation Details. We use GPT-4 with temperature 0 as our LLM and InstructBLIP [49] (Flan-T5-XXL) as our VLM. We also report "oracle" results where a human

answers questions instead of the VLM to simulate results our approach could achieve if the VLM were near-perfect (denoted as the "Oracle VLM"). Further implementation details (e.g., prompts, model usage) are in the supplementary.

A. *Evaluation on* MESSYSURFACES

We evaluate our method on the 308 benchmark questions across 5 iterations of our framework. After each iteration, the robot is evaluated on the information it has accumulated up until that point. We measure accuracy on each question and report results using both the Oracle VLM and zero-shot performance on InstructBLIP. Although **No Question** and **Oracle** are "open-loop" methods that do not require iteration, we plot their results as a constant for comparison.

After 5 iterations, for both the Oracle VLM and InstructBLIP, our approaches outperform all baselines: **No Question**, **No Active Perception**, and **Baseline Questions**. Notably, **Ours-LLM** significantly outperforms **No Question** by an average of 27.7% across the two VLM types, $p < 0.01$. **Ours-LLM** also outperforms **Baseline Questions** by an average of 5% across the VLM types, $p > 0.05$ and outperforms **No Active Perception** by an average of 6%, $p > 0.05$. Using an Oracle VLM allows **Ours-LLM** to close the gap with the **Oracle** by an average of 5% more than using InstructBLIP. Although our approach outperforms baselines using both VLMs, we suspect that InstructBLIP gives lower accuracies because the MESSYSURFACES images – especially the close-up images – are out of distribution. For this reason, we presume that our approach gives a smaller advantage over other baseline methods when using InstructBLIP.

These results suggest that asking questions and actively perceiving the environment can enable grounded commonsense reasoning; with better VLMs, we can reach close to human-level performance. However, we were puzzled why the human **Oracle** was not more accurate. We hypothesize that in some situations, it is unclear what the most appropriate way to clean an object would be – our annotators agreed 67% of the time. To obtain higher accuracy, commonsense reasoning may sometimes not be enough and we must query user preferences to personalize the cleaning action; we explore this further in §VI and the supplementary. We now analyze each component of our framework.

Does the LLM Ask Good Follow-Up Questions? We first evaluate the LLM's follow-up questions and the reasoning used to produce those questions. On average, 82% of users agreed that the reasoning was valid and 87% agreed that the reasoning was appropriate. To evaluate the follow-up questions, we asked users to rate each question's usefulness and relevance for tidying the surface on a 5-point Likert scale. We compared against **Baseline Questions**, where we removed the constraint that LLM-generated questions must be relevant for commonsense reasoning about normative values. An example baseline question is, "Does the cup have a logo?" All prompts and example questions are in the supplementary. **Users rated our questions to be significantly more useful and relevant for tidying surfaces compared to the baseline** ($p < 0.01$, Fig. 4). However, across iterations, the average usefulness and relevance of our questions decreased. This result may be because there are not many useful and relevant questions to ask about simple objects such as a keyboard without interacting with them or people in the room.

Does the LLM Suggest Informative Close-Up Angles? We next focus on whether the close-up angles suggested by the LLM are informative. For each object, we asked users whether the object's follow-up question is answerable from the close-up angle chosen by the LLM by showing them the corresponding close-up image. We also do this for the "front" angle. As our main baseline, we ask users whether questions are answerable from the scene-level view. Additionally, we compare against angles that the LLM did not choose ("Non-LLM Angles"), as well as non-front angles. **Across 5 iterations we find that, on average, 35.5% more questions are answerable by LLM-chosen angles and 31% more questions are answerable by the front angles compared to the scene, $p < 0.01$. The LLM-chosen angles and front angle are also significantly more informative than the non-LLM-chosen angles and non-front angles respectively.** This trend holds for each iteration (Fig. 5).

Do Our Close-Up Angles Improve VLM Accuracy? Using VLMs for grounded commonsense reasoning pose challenges when there are obstructions in the image (e.g., a bag blocking the sports car) or when they are not able to describe relevant details. We hypothesized that providing a close-up image would "help" a VLM answer follow-up questions more accurately. We evaluate whether close-up images can actually improve VLM accuracy on follow-up questions. From the results in Table I, we see that **having access to close-up angles greatly improves the zero-shot prediction accuracy for both VLM variants.** More importantly, the front angles and the LLM proposed angles generally outperform other angles. These results show that it is beneficial to have both active perception and correct angles for our tasks.

B. *Evaluation on Real-World Robotic System*

To assess the performance of our system on a real-world robot, we assemble 2 surfaces with 11 objects that require complex commonsense reasoning to tidy up. Importantly, we design these surfaces so that the commonsense way to tidy each object would be unambiguous. The first surface resembles a child's play area, with toys of ranging complexities (e.g., a MEGA Bloks structure, a partially built toy train set, and a to-scale Lego model of an Italian sports car). The robot must understand which toys to clean up and which toys should be left on display. The second surface, shown in §C, consists of trash that a robot must sort through and decide whether to recycle, put in landfill, or keep on the desk.

Grounding Language in Robot Behavior. Following the active perception component of our framework, we use a robot arm (equipped with a wrist camera) to servo to angles produced by the LLM and take photos. To map the LLM-produced angles and natural-language action plans to robot behavior, we implement a series of programmatic skill primitives (e.g., `relocate(''block'')`). In this work, each "view" and "action" primitive is defined assuming

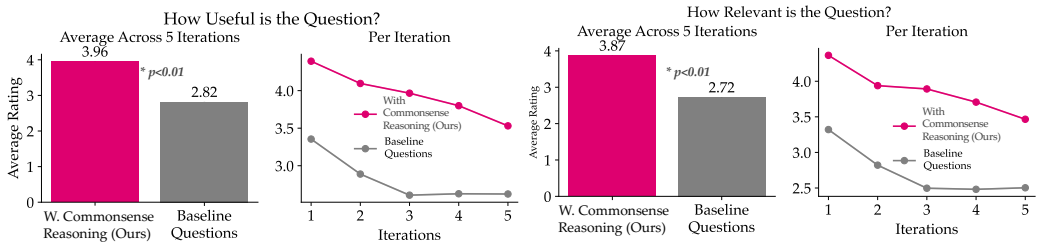

Fig. 4. **How Good are the Follow-Up Questions?** Users rated our questions to be significantly more useful and relevant compared to baseline questions, $p < 0.01$. However, the average usefulness and relevance of questions decreased over iterations.

TABLE I
VLM MULTIPLE-CHOICE PREDICTION ACCURACY (ZERO-SHOT) UNDER DIFFERENT ANGLES OVER 5 ITERATIONS

	Scene	Non-front Angles	Front Angle	Non-LLM Angles	LLM Angle
InstructBLIP (Vicuna)	47.98	51.06	**52.64**	50.94	**53.21**
InstructBLIP (Flan-T5)	51.95	53.99	**56.74**	54.08	**56.30**

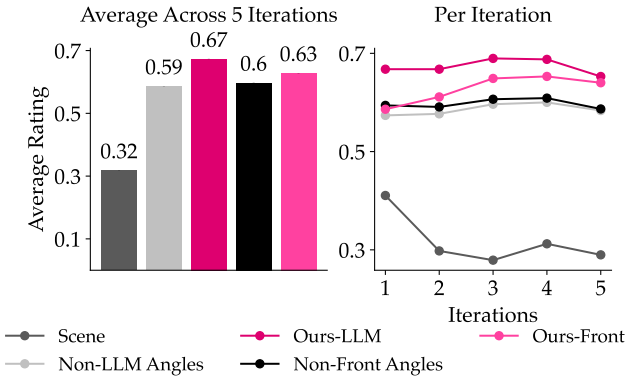

Fig. 5. **Do We Choose Informative Close-Up Angles?** An average of 33.25% more questions are answerable by the LLM-chosen angles and front angles compared to the scene, $p < 0.01$. The LLM-chosen angles and front angle are also significantly more informative than the non-LLM-chosen angles and non-front angles respectively.

access to the ground-truth object class and position. These programmatic skill primitives define an API that the LLM can call into, similar to the process introduced by [42]. Each action plan is translated to a sequence of these programmatic skills, which are then executed in an open loop (further implementation details are in the supplementary).

Benchmark Evaluation Results. To evaluate our method, we designed benchmark questions for each of the 11 objects in a similar manner to that outlined in §IV. We recruited 5 annotators on Prolific to choose the correct answer and took the majority label. We report results for both the Oracle VLM and InstructBLIP after running 5 iterations of our framework (see Figure in the supplementary). **Across both types of VLMs, Ours-LLM beats Baseline Questions by an average of** 13.5%**, beats No Active Perception by an average of** 18%**, and beats No Questions by an average of** 13.5%**.** With the Oracle VLM, we achieve **Oracle** performance. With InstructBLIP, our method produces a smaller advantage over baselines.

VI. Discussion

The purpose of this work is to equip robots with basic grounded commonsense reasoning skills to reduce the need for human specification. These reasoning skills can later be personalized towards an individual's preferences. To this end, we conduct a preliminary study to explore how we can add personalization on top of our framework. We analyzed questions that the human **Oracle** got incorrect in §V and found that object attributes such as "dirtiness" can indeed be subjective. This may have caused the **Oracle** to incorrectly answer some questions. We experimented with adding personalization information to 8 questions where both the **Oracle** and our framework chose the same incorrect answer. **We found an average** 86% **improvement in accuracy, suggesting that preference information helps further enable grounded commonsense reasoning.** See the supplementary for more details.

Limitations and Future Work. While our work presents a first step towards actively grounded commonsense reasoning, there are some limitations to address. One limitation is our reliance on heuristics to guide our active perception pipeline – while the five specified angles are enough for most of the questions in the MessySurfaces dataset, there are cases where objects may be occluded, or otherwise require more granular views to answer questions; future work might explore learned approaches for guiding perception based on uncertainty, or developing multi-view, queryable scene representations [50], [51]. Similarly, we are limited by an inability to *interact with objects dynamically* – e.g., opening boxes, removing clutter. Finally, while we focus on commonsense behaviors, there are times where the "right" thing to do is to ask for human preferences.

Acknowledgements. This work was supported by DARPA YFA, NSF Award #2006388, #2125511, #2218760, AFOSR YIP, JP Morgan, ONR, and TRI.

REFERENCES

[1] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[2] D. S. Brown, W. Goo, and S. Niekum, "Better-than-demonstrator imitation learning via automatically-ranked demonstrations," Oct. 2019.

[3] M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh, "Learning Reward Functions by Integrating Human Demonstrations and Preferences," June 2019.

[4] D. Sadigh, A. D. Dragan, S. S. Sastry, and S. A. Seshia, "Active preference-based learning of reward functions," in *Proceedings of Robotics: Science and Systems (RSS)*, July 2017.

[5] M. Li, A. Canberk, D. P. Losey, and D. Sadigh, "Learning human objectives from sequences of physical corrections," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 2877–2883.

[6] E. Biyik, D. P. Losey, M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh, "Learning Reward Functions from Diverse Sources of Human Feedback: Optimally Integrating Demonstrations and Preferences," 2021.

[7] T. Fitzgerald, P. Koppol, P. Callaghan, R. Q. Wong, R. Simmons, O. Kroemer, and H. Admoni, "INQUIRE: INteractive Querying for User-aware Informative REasoning."

[8] A. Talmor, O. Yoran, R. L. Bras, C. Bhagavatula, Y. Goldberg, Y. Choi, and J. Berant, "CommonsenseQA 2.0: Exposing the limits of AI through gamification," *arXiv preprint arXiv:2201.05320*, 2022.

[9] L. Jiang, J. D. Hwang, C. Bhagavatula, R. L. Bras, J. Liang, J. Dodge, K. Sakaguchi, M. Forbes, J. Borchardt, S. Gabriel, *et al.*, "Can Machines Learn Morality? The Delphi Experiment," July 2022.

[10] D. Hendrycks, C. Burns, S. Basart, A. Critch, J. Li, D. Song, and J. Steinhardt, "Aligning ai with shared human values," *arXiv preprint arXiv:2008.02275*, 2020.

[11] M. Kwon, S. M. Xie, K. Bullard, and D. Sadigh, "Reward design with language models," *arXiv preprint arXiv:2303.00001*, 2023.

[12] 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," 2020.

[13] C. M. Rytting and D. Wingate, "Leveraging the Inductive Bias of Large Language Models for Abstract Textual Reasoning."

[14] B. Zhang and H. Soh, "Large Language Models as Zero-Shot Human Models for Human-Robot Interaction," Mar. 2023.

[15] X. Zhou, Y. Zhang, L. Cui, and D. Huang, "Evaluating Commonsense in Pre-Trained Language Models," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 9733–9740, Apr. 2020.

[16] Z. Jin, S. Levine, F. Gonzalez Adauto, O. Kamal, M. Sap, M. Sachan, R. Mihalcea, J. Tenenbaum, and B. Schölkopf, "When to Make Exceptions: Exploring Language Models as Accounts of Human Moral Judgment," *Advances in Neural Information Processing Systems*, vol. 35, pp. 28 458–28 473, Dec. 2022.

[17] K. C. Fraser, S. Kiritchenko, and E. Balkir, "Does Moral Code Have a Moral Code? Probing Delphi's Moral Philosophy," May 2022.

[18] P. Ammanabrolu, L. Jiang, M. Sap, H. Hajishirzi, and Y. Choi, "Aligning to Social Norms and Values in Interactive Narratives," May 2022.

[19] D. Hendrycks, M. Mazeika, A. Zou, S. Patel, C. Zhu, J. Navarro, D. Song, B. Li, and J. Steinhardt, "What Would Jiminy Cricket Do? Towards Agents That Behave Morally," Feb. 2022.

[20] D. Hendrycks, C. Burns, S. Basart, A. Critch, J. Li, D. Song, and J. Steinhardt, "Aligning AI With Shared Human Values," Feb. 2023.

[21] R. Zellers, Y. Bisk, A. Farhadi, and Y. Choi, "From Recognition to Cognition: Visual Commonsense Reasoning," Mar. 2019.

[22] R. Akrour, M. Schoenauer, and M. Sebag, "April: Active preference learning-based reinforcement learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2012, pp. 116–131.

[23] M. Cakmak, S. S. Srinivasa, M. K. Lee, J. Forlizzi, and S. Kiesler, "Human preferences for robot-human hand-over configurations," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 1986–1993.

[24] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning." in *Aaai*, vol. 8, 2008, pp. 1433–1438.

[25] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich, "Maximum margin planning," Pittsburgh, Pennsylvania, 2006.

[26] L. Fan, G. Wang, Y. Jiang, A. Mandlekar, Y. Yang, H. Zhu, A. Tang, D.-A. Huang, Y. Zhu, and A. Anandkumar, "MineDojo: Building Open-Ended Embodied Agents with Internet-Scale Knowledge," June 2022.

[27] S. Singh and J. H. Liao, "Concept2Robot 2.0: Improving Learning of Manipulation Concepts Using Enhanced Representations."

[28] L. Shao, T. Migimatsu, Q. Zhang, K. Yang, and J. Bohg, "Concept2Robot: Learning manipulation concepts from instructions and human demonstrations," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1419–1434, Dec. 2021.

[29] S. Mirchandani, S. Karamcheti, and D. Sadigh, "Ella: Exploration through learned language abstraction," Oct. 2021.

[30] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, "Fine-Tuning Language Models from Human Preferences," Jan. 2020.

[31] H. Hu and D. Sadigh, "Language instructed reinforcement learning for human-ai coordination," in *40th International Conference on Machine Learning (ICML)*, 2023.

[32] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. Sukhatme, "Interactive Perception: Leveraging Action in Perception and Perception in Action," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1273–1291, Dec. 2017.

[33] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "TidyBot: Personalized Robot Assistance with Large Language Models," May 2023.

[34] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, *et al.*, "Inner Monologue: Embodied Reasoning through Planning with Language Models," July 2022.

[35] X. Zhao, M. Li, C. Weber, M. B. Hafez, and S. Wermter, "Chat with the Environment: Interactive Multimodal Perception using Large Language Models," Mar. 2023.

[36] J. Yu, Y. Xu, J. Y. Koh, T. Luong, G. Baid, Z. Wang, V. Vasudevan, A. Ku, Y. Yang, B. K. Ayan, *et al.*, "Scaling autoregressive models for content-rich text-to-image generation," *arXiv*, 2022.

[37] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual Language Maps for Robot Navigation," Mar. 2023.

[38] D. Shah, B. Osinski, B. Ichter, and S. Levine, "LM-Nav: Robotic Navigation with Large Pre-Trained Models of Language, Vision, and Action," July 2022.

[39] 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," Aug. 2022.

[40] M. Attarian, A. Gupta, Z. Zhou, W. Yu, I. Gilitschenski, and A. Garg, "See, Plan, Predict: Language-guided Cognitive Planning with Video Prediction," Oct. 2022.

[41] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents," Mar. 2022.

[42] 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," May 2023.

[43] D. Surís, S. Menon, and C. Vondrick, "ViperGPT: Visual Inference via Python Execution for Reasoning," Mar. 2023.

[44] 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," Dec. 2022.

[45] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, *et al.*, "PaLM-E: An Embodied Multimodal Language Model," Mar. 2023.

[46] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-maron, M. Giménez, Y. Sulsky, J. Kay, J. T. Springenberg, *et al.*, "A Generalist Agent," *Transactions on Machine Learning Research*, Nov. 2022.

[47] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan, "VIMA: General Robot Manipulation with Multimodal Prompts," Oct. 2022.

[48] 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," *arXiv*, 2022.

[49] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. Li, P. Fung, and S. Hoi, "Instructblip: Towards general-purpose vision-language models with instruction tuning," *arXiv preprint arXiv:2305.06500*, 2023.

[50] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[51] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "LERF: Language Embedded Radiance Fields," Mar. 2023.

[52] M. A. Research, "Polymetis: A real-time pytorch controller manager," https://github.com/facebookresearch/fairo/tree/main/polymetis, 2021–2023.

[53] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The pinocchio c++ library – a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *IEEE International Symposium on System Integrations (SII)*, 2019.

[54] J. Carpentier, F. Valenza, N. Mansard, *et al.*, "Pinocchio: fast forward and inverse dynamics for poly-articulated systems," https://stack-of-tasks.github.io/pinocchio, 2015–2023.