Physically Grounded Vision-Language Models for Robotic Manipulation

Jensen Gao[1], Bidipta Sarkar[1], Fei Xia[2], Ted Xiao[2], Jiajun Wu[1],
Brian Ichter[2], Anirudha Majumdar[2,3], Dorsa Sadigh[1,2]

***Abstract*— Recent advances in vision-language models (VLMs) have led to improved performance on tasks such as visual question answering and image captioning. Consequently, these models are now well-positioned to reason about the physical world, particularly within domains such as robotic manipulation. However, current VLMs are limited in their understanding of the physical concepts (e.g., material, fragility) of common objects, which restricts their usefulness for robotic manipulation tasks that involve interaction and physical reasoning about such objects. To address this limitation, we propose PHYSOBJECTS, an object-centric dataset of 39.6K crowd-sourced and 417K automated physical concept annotations of common household objects. We demonstrate that fine-tuning a VLM on PHYSOBJECTS improves its understanding of physical object concepts, including generalization to held-out concepts, by capturing human priors of these concepts from visual appearance. We incorporate this physically grounded VLM in an interactive framework with a large language model-based robotic planner, and show improved planning performance on tasks that require reasoning about physical object concepts, compared to baselines that do not leverage physically grounded VLMs. We additionally illustrate the benefits of our physically grounded VLM on a real robot, where it improves task success rates. We release our dataset and provide further details and visualizations of our results at https://iliad.stanford.edu/pg-vlm/.**

I. INTRODUCTION

Large language models (LLMs) have shown great promise for converting language instructions into task plans for embodied agents [1], [2]. The fundamental challenge in applying LLMs for this is grounding them to the physical world, through sensory input such as vision. Prior work has made progress towards grounding LLMs by using vision-language models (VLMs) to indicate the presence of objects in a scene, or to provide feedback about occurrences in a scene [3]–[7]. However, vision could be used to further improve grounding by extracting more detailed scene information. For robotic manipulation, understanding physical concepts of objects, such as their material composition or their fragility, would help planners identify relevant objects to interact with, and affordances based on physical or safety constraints. For example, if a human wants a robot to get a cup of water, the robot should be able to determine if a cup already has water or something else in it. Also, the robot should handle the cup with greater caution if it is more fragile.

How can we use vision to reason about physical object concepts? Prior work has studied this problem using more traditional vision techniques, such as self-supervised learning on object interaction data. However, object interaction data can be challenging to collect when scaling up beyond a small set of objects in well-defined settings. While precise estimation of physical properties may sometimes be impossible without interaction data, humans can use their visual perception to reason at a high level about physical concepts without object interactions. For example, humans can reason that a glass cup is more fragile than a plastic bottle, and that it would be easier to use a bowl to hold water than a shallow plate. This reasoning is often based on prior semantic knowledge of visually similar objects, and can be done from static visual appearance alone.

Similarly, VLMs pre-trained using large-scale data have demonstrated broad visual reasoning abilities and generalization [8]–[13], and thus have the potential to physically reason about objects in a similar fashion as humans. Therefore, we propose to leverage VLMs as a scalable way of providing the kind of high-level physical reasoning that humans use to interact with the world, which can benefit a robotic planner, without the need for interaction data. The general and flexible nature of VLMs also removes the need to use separate task-specific vision models for physical reasoning. VLMs have already been commonly incorporated into robotic planning systems [3]–[7], [13], making them a natural solution for endowing physical reasoning into robotic planning.

However, while modern VLMs have improved significantly on tasks such as visual question answering (VQA), and there has been evidence of their potential for object-centric physical reasoning [14], we show in this work that their out-of-the-box performance for this still leaves much to be desired. Although VLMs have been trained on broad internet-scale data, this data does not contain many examples of object-centric physical reasoning. This motivates incorporating a greater variety and amount of such data when training VLMs. Unfortunately, prior visual datasets for physical reasoning are not well-suited for understanding common real-world objects, which is desirable for robotics. To address this, we propose PHYSOBJECTS, an object-centric dataset with human physical concept annotations of common household objects. Our annotations include categorical labels (e.g., object X is made of plastic) and preference pairs (e.g., object X is heavier than object Y).

Our main contributions are PHYSOBJECTS, a dataset of 39.6K crowd-sourced and 417K automated physical concept annotations of real household objects, and demonstrating that using it to fine-tune a VLM significantly improves physical reasoning. We show that our physically grounded VLM achieves improved test accuracy on our dataset, including on held-out physical concepts. Furthermore, to illustrate

[1]Stanford University, [2]Google DeepMind, [3]Princeton University. Contact: jenseng@stanford.edu.

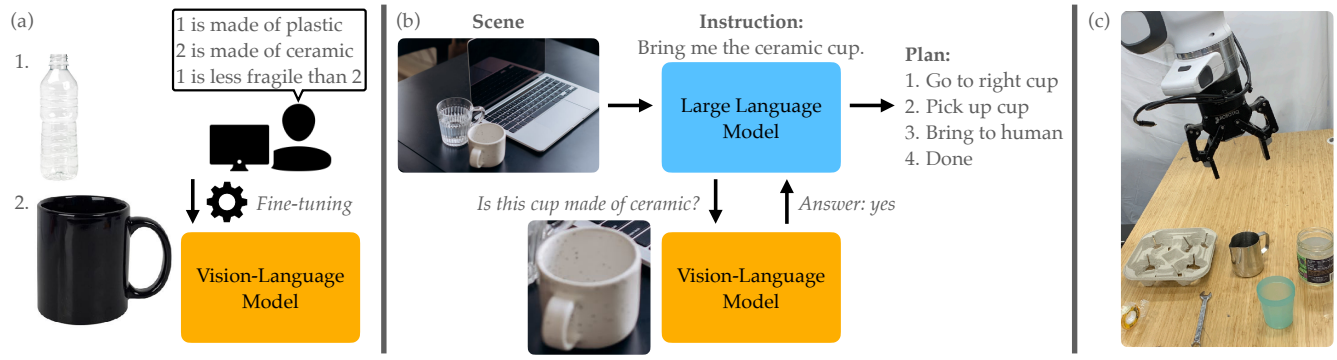


Fig. 1: (a) We collect physical concept annotations of common household objects for fine-tuning VLMs. (b) We use the fine-tuned VLM in an LLM-based robotic planning framework, where the LLM queries the VLM about physical concepts of objects in the scene, before producing a plan. (c) We evaluate LLM-generated plans on a real Franka Emika Panda robot.

the utility of improved physical reasoning for robotics, we incorporate our physically grounded VLM with an LLM-based robotic planner, where the LLM queries the VLM about physical concepts of objects in its scene. Our system achieves improved planning performance on tasks that require physical reasoning, compared to baselines that do not use physically grounded VLMs. Finally, we demonstrate the benefits of our physically grounded VLM for planning with a real robot, where its usage improves task success rates.

II. Related Work

We review prior work on physical reasoning, object attribute datasets, VLMs, using LLMs for robotic planning, and using LLMs and VLMs together in an interactive system.

Physical Reasoning. Prior works have studied estimating physical object properties from vision by learning from interaction data [15]–[17]. Other works focus on learning representations that capture physical concepts, rather than direct estimation [18], [19]. Unlike these works, we use pre-trained VLMs and human annotations as a more scalable alternative to learning from interaction. Mind's Eye investigates physical reasoning using LLMs [20], but relies on grounding using a simulator, which would be difficult to scale to the real world. VEC investigates physical reasoning with LLMs and VLMs [21], but reasons from text descriptions, while we reason from real images. OpenScene uses CLIP [22] to identify objects in scenes using properties such as material and fragility, but these results are only qualitative in nature [14]. In our work, we propose PhysObjects to better quantify and improve object-centric physical reasoning, and leverage this reasoning for robotic manipulation.

Object Attribute Datasets. There have been prior visual object attribute datasets with concepts included in PhysObjects, such as material and transparency [23]–[26]. However, they focus more on visual attributes such as color, while we focus on physical concepts. Physics 101 provides a dataset of object interaction videos and property measurements [16], but PhysObjects includes a greater variety of objects that are more relevant for household robotics.

Vision-Language Models. VLMs have made large improvements on multi-modal tasks such as VQA, by leveraging internet-scale image and text data [8]–[10], [12]. In our experiments, we use InstructBLIP [11] as our base VLM for fine-tuning and comparison, as it was the state-of-the-art open-source VLM at the time of our experiments. PaLM-E has shown strong performance on general visual-language tasks and robotic planning [13], but there has not been focused evaluation of it for physical reasoning. SuccessVQA fine-tunes VLMs on human data for success detection by treating it as a VQA task, and achieves better generalization than models designed specifically for success detection [27]. We similarly fine-tune VLMs on human data for physical reasoning by casting it as a VQA problem, to benefit from the generalization abilities and versatility of VLMs.

LLMs for Robotic Planning. Many recent works have used LLMs as robotic planners. SayCan uses visual value functions to provide affordances for grounding [2], but does not benefit from VLMs. Follow-up works have used VLMs for grounding LLM planners through object detection, or providing feedback about what has happened (e.g., success detection) [3]–[7]. Our work focuses on expanding the use of VLMs for grounding through physical reasoning, to let LLM-based planners perform tasks that require a deeper physical understanding of the world.

LLM/VLM Interaction. Our planning evaluation falls in the framework of Socratic Models [28], where large models interact with each other through text to perform tasks such as VQA [29], [30] and image captioning [31]. Most similar to our evaluation is Matcha, where an LLM receives a task instruction, obtains object-centric feedback from its environment, and uses this for task planning [32]. However, this work does not focus on visual feedback, as their evaluation is in a simulated environment where physical concepts are not visually observable. In contrast, we focus on physical reasoning from vision in real-world scenes.

III. PhysObjects Dataset

To benchmark and improve VLMs for object-centric physical reasoning, we propose PhysObjects, a dataset of 39.6K crowd-sourced and 417K automated physical concept annotations for images of real household objects.

Image Source. We use the publicly released challenge version of the EgoObjects dataset [33] as our image source. To our knowledge, this was the largest object-centric dataset of real images that was publicly released when constructing PHYSOBJECTS. The dataset consists of frames from egocentric videos in realistic household settings, which makes it particularly relevant for household robotics. It includes 117,424 images, 225,466 object bounding boxes with corresponding category labels from 277 object categories, and 4,203 object instance IDs. PHYSOBJECTS consists of physical concept annotations for a large subset of this image data. [1]

We construct random training, validation, and test sets based on object instance IDs. We split the dataset per object category to ensure each object category is represented in each set when possible. Our training, validation, and test sets consist of 73.0%, 14.8%, and 12.2% of objects, respectively.

Concept	Description
Mass	how heavy an object is
Fragility	how easily an object can be broken/damaged
Deformability	how easily an object can change shape without breaking
Material	what an object is primarily made of
Transparency	how much can be seen through an object
Contents	what is inside a container
Can Contain Liquid	if a container can be used to easily carry liquid
Is Sealed	if a container will not spill if rotated
Density (*held-out*)	how much mass per unit of volume of an object
Liquid Capacity (*held-out*)	how much liquid a container can contain

TABLE I: Our physical concepts and brief descriptions

Physical Concepts. We collect annotations for eight main physical concepts and two additional concepts reserved for held-out evaluation. We select concepts based on prior work and what we believe to be useful for robotic manipulation, but do not consider all such concepts. For example, we do not include *friction* because this can be challenging to estimate without interaction, and we do not include *volume* because this requires geometric reasoning, which we do not focus on.

Of our main concepts, three are continuous-valued and applicable to all objects: *mass*, *fragility*, and *deformability*. Two are also applicable to all objects, but are categorical: *material* and *transparency*. *Transparency* could be considered continuous, but we use discrete values of *transparent*, *translucent*, and *opaque*. The other three are categorical and applicable only to container objects: *contents*, *can contain liquid*, and *is sealed*. We define which object categories are containers, resulting in 956 container object instances.

Our two held-out concepts are *density*, which is continuous and applicable to all objects, and *liquid capacity*, which is continuous and applicable only to containers. We only collect test data for these held-out concepts. We list all concepts and their brief descriptions in Table I.

For categorical concepts, we define a set of labels for each concept. Annotations consist of a label specified for a given object and concept. For the concepts *material* and *contents*, when crowd-sourcing, we allow for open-ended labels if none of the pre-defined labels are applicable.

[1]We publicly release our dataset on our website. Because the EgoObjects license does not permit incorporating it into another dataset, we release our annotations separately from the image data.

For continuous concepts, annotations are preference pairs, where given two objects, an annotation indicates that either one object has a higher level of a concept, the objects have roughly *equal* levels, or the relationship is *unclear*. We use preferences because it is generally more intuitive for humans to provide comparisons than continuous values [34], [35]. This is especially true when annotating static images with physical concepts, where it is difficult to specify precise grounded values. For example, it would be difficult to specify the *deformability* of a sponge as a value out of 10. Comparisons have also been used to evaluate LLMs and VLMs for physical reasoning in prior work [21]. Therefore, the kind of grounding studied in PHYSOBJECTS for continuous concepts is only relational in nature.

Automatic Annotations. Before crowd-sourcing, we first attempt to automate as many annotations as possible, so that crowd-workers only annotate examples that cannot be easily automated. For categorical concepts, we assign concept values to some of the defined object categories in EgoObjects, such that all objects in a category are labeled with that value. For continuous concepts, we define *high* and *low* tiers for each concept, such that all objects from a *high* tier category have a higher level of that concept than all objects from a *low* tier category. Then, we automate preference annotations for all object pairs between the two tiers.

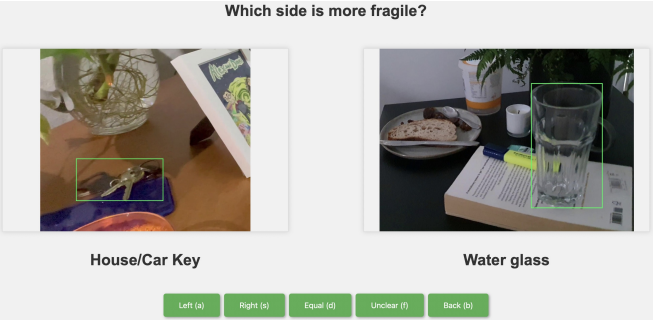


Fig. 2: Annotation UI for *fragility*. Here, the label is *right*, i.e., the *water glass* is more fragile than the *house/car key*.

Crowd-Sourcing Annotations. We obtain additional annotations via crowd-sourcing, using 573 crowd-workers on the Prolific platform. Crowd-workers use a web-based user interface (example for *fragility* shown in Fig. 2) where they are presented with object bounding boxes in the context of their overall image, and provide annotations using on-screen buttons or their keyboard. For categorical concepts, we collect annotations for the majority of objects that were not automatically annotated. For continuous concepts, because it is impractical to annotate every pair of objects in the dataset, we randomly sample pairs to annotate. We enforce that 20% of the sampled pairs are between objects of the same category, to prioritize understanding differences between objects of the same category. We collect annotations from three crowd-workers for each example. To promote high-quality data, we include attention checks as 10% of provided examples, which have known labels, and only keep data from annotators that achieve 80% accuracy on these.

	Most Common	Text Only	InstructBLIP	Single Concept FT (ours)	PG-InstructBLIP (ours)
Mass	42.2	73.3	62.2	**80.0**	**80.0**
Fragility	64.9	64.9	78.4	91.2	**94.6**
Deformability	46.5	62.8	67.4	**95.3**	93.0
Material	37.1	73.9	67.1	83.7	**84.6**
Transparency	77.6	82.2	85.8	89.4	**90.1**
Contents	39.5	50.9	35.1	81.6	**83.3**
Can Contain Liquid	56.3	**92.2**	59.4	84.4	87.5
Is Sealed	80.6	80.6	74.2	80.6	**87.1**
Average	55.6	72.6	66.2	85.8	**87.5**

TABLE II: Test accuracy for main concepts on crowd-sourced PHYSOBJECTS

Dataset Statistics. We crowd-source 39.6K annotations for 13.2K examples, and automate annotations for 417K additional examples. For crowd-sourced annotations, 93.7% of examples have at least 2/3 annotator label agreement, and 58.1% have unanimous agreement.

IV. PHYSICALLY GROUNDING VISION-LANGUAGE MODELS

Fine-Tuning VLMs. We work with the FlanT5-XXL [36] version of InstructBLIP [11]. InstructBLIP takes as input a single RGB image and text prompt, and predicts text as output. In our setup, we choose the model inputs to be a single bounding box of an object, and a question text prompt corresponding to each concept.

Learning From Preferences. Learning for categorical concepts amounts to maximum likelihood of annotated labels. However, it is not as straightforward to train a VLM on preferences for continuous concepts, because preference learning requires a continuous score. To do this with VLMs, which naturally have discrete text outputs, we prompt the VLM with questions that can be answered with *yes* or *no* for continuous concepts. Then, we extract the following score function:

$$s(o, c) = \frac{p(\text{yes} \mid o, c)}{p(\text{no} \mid o, c)}$$

where o is an object bounding box image, c is a concept, and $p(\cdot|o, c)$ is the likelihood under the VLM of text, conditioned on the object image and concept. We use this as our score function because it can take any non-negative value, and $\log s(o, c)$ has the intuitive interpretation as the difference of log-likelihoods between *yes* and *no*.[2] We then use the Bradley-Terry model [37] to estimate the probability of a human indicating that object o_1 has a higher value than object o_2 for concept c as:

$$P(o_1 > o_2 \mid c) = \frac{s(o_1, c)}{s(o_1, c) + s(o_2, c)}.$$

We assume a dataset $\mathcal{D}$ of preference annotations (o_1, o_2, c, y), where $y \in \{[1, 0], [0, 1], [0.5, 0.5]\}$ corresponds to if o_1 is preferred, o_2 is preferred, or if they are indicated to be equal. We then fine-tune the VLM by minimizing the following objective:

$$\mathcal{L}(\mathcal{D}) = -\,\mathbb{E}_{(o_1,o_2,c,y)\sim\mathcal{D}}[y_1 \log P(o_1 > o_2 \mid c) + y_2 \log(1 - P(o_1 > o_2 \mid c)].$$

In practice, this is the binary cross-entropy objective where the logits for each object image o is the difference of log-likelihoods $\log s(o, c) = \log p(\text{yes} \mid o, c) - \log p(\text{no} \mid o, c)$.

V. EXPERIMENTAL RESULTS

We evaluate VLMs for physical reasoning using 1) test accuracy on PHYSOBJECTS, 2) planning accuracy on real scenes for physical reasoning tasks, and 3) task success rate on a real robot.

A. Dataset Evaluation

We refer to InstructBLIP fine-tuned on all main concepts in PHYSOBJECTS as Physically Grounded InstructBLIP, or PG-InstructBLIP. [3] We focus our evaluation on crowd-sourced examples, because as described in Section III, these were collected with the intent for their labels to not be discernible from object category information alone, and thus they are generally more challenging. We report test accuracy on these examples in Table II. Our baselines include *Most Common*, where the most common label in the training data is predicted, *Text Only*, where an LLM makes predictions using in-context examples from PHYSOBJECTS, but using object category labels instead of images, and InstructBLIP. We also compare to versions of InstructBLIP fine-tuned on single concept data. We find that PG-InstructBLIP outperforms InstructBLIP on all concepts, with the largest improvement on *contents*, which InstructBLIP has the most difficulty with. We also find that PG-InstructBLIP performs slightly better than the single concept models, suggesting possible positive transfer from using a single general-purpose model compared to separate task-specific models, although we acknowledge the improvement here is not extremely significant. PG-InstructBLIP also generally outperforms *Most Common* and *Text Only*, suggesting that our evaluation benefits from reasoning beyond dataset statistics, and from using vision.

[2]We experimented with other choices of score functions, and found that while all performed similarly with respect to test accuracy on PHYSOBJECTS, we found this score function to produce the most interpretable range of likelihoods for different responses, which we hypothesize to be beneficial for downstream planning.

[3]We release the model weights for PG-InstructBLIP on our website.

	Instruct-BLIP	PG-InstructBLIP (ours)
Density	54.2	**70.3**
Liquid Capacity	65.4	**73.0**
Average	59.8	**71.7**

TABLE III: Test accuracy for held-out concepts on crowd-sourced PHYSOBJECTS

Generalization Results. We additionally evaluate both InstructBLIP and PG-InstructBLIP on test data for our held-out concepts, which we report in Table III. We find that PG-InstructBLIP improves upon InstructBLIP by **11.9%**, despite having never seen these evaluated concepts nor object instances during fine-tuning. We believe this suggests that fine-tuning VLMs can offer possible generalization benefits to concepts that are related to those seen during fine-tuning.

	Instruct-BLIP	PG-InstructBLIP (ours)
Mass	55.6	**82.2**
Fragility	70.3	**83.8**
Deformability	76.7	**88.4**
Material	67.7	**83.4**
Transparency	81.5	**83.8**
Contents	32.5	**81.6**
Can Contain Liquid	56.3	**89.1**
Is Sealed	71.0	**80.6**
Average	64.0	**84.1**

TABLE IV: Test accuracy for main concepts with paraphrased prompts

In Table IV, we report results for main concepts on unseen paraphrased question prompts. We find that PG-InstructBLIP still outperforms InstructBLIP, with limited degradation from the original prompts, suggesting robustness to question variety from using a large pre-trained VLM.

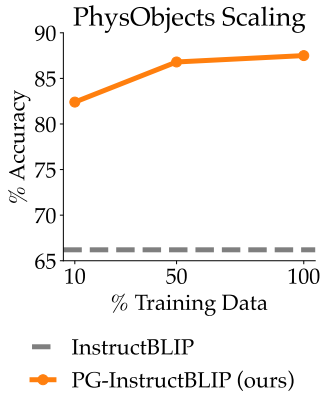


Fig. 3: Performance scaling with dataset size

Dataset Scaling. In Fig. 3, we illustrate how average performance scales with dataset size, by fine-tuning on different fractions of data from PHYSOBJECTS. Performance scales positively, but the models still benefit significantly from only 10% of our dataset, suggesting that the physical reasoning of VLMs can be improved with relatively small amounts of annotated data.

Additional Results. We include additional results in our Appendix (found on our website). These include showing that PG-InstructBLIP has limited degradation on general VQA benchmarks compared to InstructBLIP, suggesting that existing systems using VLMs can benefit from PHYSOBJECTS for physical reasoning, without sacrificing other reasoning abilities. We also include results using different question prompts, using a smaller version of InstructBLIP, evaluating on automatically annotated data, transfer to held-out concepts, and ablations on our fine-tuning process.

B. Real Scene Planning Evaluation

Next, we evaluate the efficacy of PG-InstructBLIP for robotic planning on unseen images of real scenes. We provide an example scene in Fig. 4. We evaluate on tasks with language instructions, and assume a library of primitive robotic operations with language descriptions.

Planning Framework. The LLM used in our planning framework is GPT-4 [38]. It is first given object detections in the scene, a list of primitives, and the task instruction, and then asks a VLM questions about objects in the scene. There are no constraints on the questions. Afterwards, the LLM either indicates the task is not possible, or produces a plan consisting of primitives to execute.

Fig. 4: Example scene in our planning evaluation

Task Category	No VLM	Instruct-BLIP	PG-InstructBLIP (ours)
Single Concept	36.8	68.4	**84.1**
Multi-Concept	27.8	27.8	**94.4**
Common Knowledge	35.7	78.6	**85.7**
Overall	33.3	56.9	**88.2**

TABLE V: Task plan accuracy on 51 real scenarios

Results. We report task planning accuracy using InstructBLIP and PG-InstructBLIP in Table V. We also compare to a planner that does not use VLM interaction for grounding. We evaluate on 51 task scenarios across 8 scenes, using a non-author human to evaluate task plans. We divide our task scenarios into three categories. *Single Concept* requires identifying objects using one physical concept, e.g., finding the heaviest object. *Multi-Concept* requires reasoning about multiple physical concepts, e.g., asking for a metal container that can hold water. This may include concepts outside of PHYSOBJECTS. *Common Knowledge* requires additional reasoning about common knowledge of objects, e.g., understanding the label of a container. While our tasks focus on physical concepts in PHYSOBJECTS, the LLM can ask questions about other concepts that may also be useful, particularly for *Common Knowledge* tasks.

PG-InstructBLIP outperforms InstructBLIP on all task categories, especially *Multi-Concept*. It does slightly better on *Common Knowledge*, suggesting that it can reason about non-PHYSOBJECTS concepts at a similar level as InstructBLIP. Using no VLM performs substantially worse than using VLM interaction, indicating that our tasks require additional grounding beyond object detection. We provide further details of results on our website.

C. Real Robot Evaluation

Lastly, we evaluate plans on real scenes using a Franka Emika Panda robot. We use a similar planner as in the previous section, but with different prompts and primitives. We assume a library of primitives for pick-and-place tasks. We evaluate on two scenes, with five tasks per scene, which we provide in Table VI. We report success rates using InstructBLIP and PG-InstructBLIP in Table VII. We ensure the primitives execute successfully, so our success rates only reflect plan quality.

Scene Image	Task Instructions
	1) Move all objects that are not plastic to the side. 2) Find a container that has metals. Move all metal objects into that container. 3) Move all containers that can be used to carry water to the side. 4) Put the two objects with the least mass into the least deformable container. 5) Move the most fragile object to the side.
	1) Put all containers that can hold water to the side. 2) Put all objects that are not plastic to the side. 3) Put all objects that are translucent to the side. 4) Put the three heaviest objects to the side. 5) Put a plastic object that is not a container into a plastic container. Choose the container that you are most certain is plastic.

TABLE VI: Scene images and task instructions for our real robot evaluation

We find that using PG-InstructBLIP leads to successful robot executions more often than InstructBLIP. For example, when asked "Is this object not plastic?" about the ceramic bowl in Fig. 5a, InstructBLIP incorrectly assigns a likelihood of 0.89 to *yes*, while PG-InstructBLIP only assigns 0.18. However, when asked "Is this object translucent?" about the glass jar in Fig. 5b, both InstructBLIP and PG-InstructBLIP incorrectly assign likelihoods of 0.95 and 0.91 to *yes*, respectively. We note that while these questions relate to physical concepts in PHYSOBJECTS, neither are formatted like the training questions for PG-InstructBLIP. For example, the training prompt for *transparency* was "Is this object transparent, translucent, or opaque?". This suggests that despite using a large pre-trained VLM, PG-InstructBLIP may sometimes still fail due to out-of-distribution questions. We provide more results and visualizations on our website.

	Instruct-BLIP	PG-InstructBLIP (ours)
Scene 1	2/5	**5/5**
Scene 2	2/5	**4/5**
Overall	4/10	**9/10**

TABLE VII: Success rates for real robot evaluation

(a) Ceramic bowl

(b) Glass jar

Fig. 5: Objects from our real robot evaluation

VI. DISCUSSION

Summary. In this work, we propose PHYSOBJECTS, the first large-scale dataset of physical concept annotations of real household object images, and demonstrate that fine-tuning a VLM on it significantly improves its physical reasoning abilities, including on held-out physical concepts. We find that using the fine-tuned VLM for real-world robotic planning improves performance on tasks that require physical reasoning. We believe our work makes progress toward expanding the applicability of VLMs for robotics.

Limitations and Future Work. While we show PHYSOBJECTS can improve the physical reasoning of a VLM, it still makes errors relative to human judgment. Also, while our proposed methodology for continuous concepts improves relational grounding, which we show can be useful for robotic planning, the model outputs are not grounded in real physical quantities, which would be needed for some applications, e.g., identifying if an object is too heavy to be picked up. Future work can investigate incorporating data with real physical measurements to improve grounding.

While we believe the physical concepts in this work to have broad relevance for robotics, future work can expand on these for greater downstream applications. This could include expanding beyond physical reasoning, such as geometric reasoning (e.g., whether an object can fit inside a container), or social reasoning (e.g., what is acceptable to move off a table for cleaning). We believe our dataset is a first step towards this direction of using VLMs for more sophisticated reasoning in robotics.

ACKNOWLEDGMENTS

This work was supported by NSF Awards 2132847, 1941722, and 2338203, ONR N00014-23-1-2355 and YIP, DARPA YFA, and Ford. We thank Minae Kwon, Siddharth Karamcheti, Suvir Mirchandani, and other ILIAD lab members for helpful discussions and feedback, and Siddharth Karamcheti for helping to set up the real robot evaluation.

REFERENCES

[1] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR, 2022.

[2] Brian Ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander T Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Kyle Jeffrey, Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and Chuyuan Kelly Fu. Do as i can, not as i say: Grounding language in robotic affordances. In *6th Annual Conference on Robot Learning*, 2022.

[3] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Tomas Jackson, Noah Brown, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models. In *6th Annual Conference on Robot Learning*, 2022.

[4] Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S Ryoo, Austin Stone, and Daniel Kappler. Open-vocabulary queryable scene representations for real world planning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11509–11522. IEEE, 2023.

[5] Wenlong Huang, Fei Xia, Dhruv Shah, Danny Driess, Andy Zeng, Yao Lu, Pete Florence, Igor Mordatch, Sergey Levine, Karol Hausman, et al. Grounded decoding: Guiding text generation with grounded models for embodied agents. *Advances in Neural Information Processing Systems*, 36, 2023.

[6] Satvik Sharma, Huang Huang, Kaushik Shivakumar, Lawrence Yunliang Chen, Ryan Hoque, brian ichter, and Ken Goldberg. Semantic mechanical search with large vision and language models. In *7th Annual Conference on Robot Learning*, 2023.

[7] Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas Funkhouser. Tidybot: Personalized robot assistance with large language models. *Autonomous Robots*, 2023.

[8] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

[9] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish V Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme Ruiz, Andreas Peter Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. PaLI: A jointly-scaled multilingual language-image model. In *The Eleventh International Conference on Learning Representations*, 2023.

[10] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 19730–19742, 2023.

[11] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[12] Shikun Liu, Linxi Fan, Edward Johns, Zhiding Yu, Chaowei Xiao, and Anima Anandkumar. Prismer: A vision-language model with multi-task experts. *Transactions on Machine Learning Research*, 2024.

[13] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-e: An embodied multimodal language model. In *Proceedings of the 40th International Conference on Machine Learning*, pages 8469–8488, 2023.

[14] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Openscene: 3d scene understanding with open vocabularies. In *CVPR*, 2023.

[15] Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. *Advances in neural information processing systems*, 28, 2015.

[16] Jiajun Wu, Joseph J Lim, Hongyi Zhang, Joshua B Tenenbaum, and William T Freeman. Physics 101: Learning physical object properties from unlabeled videos. In *BMVC*, volume 2, page 7, 2016.

[17] Yunzhu Li, Toru Lin, Kexin Yi, Daniel Bear, Daniel L.K. Yamins, Jiajun Wu, Joshua B. Tenenbaum, and Antonio Torralba. Visual grounding of learned physical models. In *ICML*, 2020.

[18] Michael Janner, Sergey Levine, William T. Freeman, Joshua B. Tenenbaum, Chelsea Finn, and Jiajun Wu. Reasoning about physical interactions with object-oriented prediction and planning. In *International Conference on Learning Representations*, 2019.

[19] Zhenjia Xu, Jiajun Wu, Andy Zeng, Joshua B Tenenbaum, and Shuran Song. Densephysnet: Learning dense physical object representations via multi-step dynamic interactions. In *Robotics: Science and Systems (RSS)*, 2019.

[20] Ruibo Liu, Jason Wei, Shixiang Shane Gu, Te-Yen Wu, Soroush Vosoughi, Claire Cui, Denny Zhou, and Andrew M. Dai. Mind's eye: Grounded language model reasoning through simulation. In *The Eleventh International Conference on Learning Representations*, 2023.

[21] Lei Li, Jingjing Xu, Qingxiu Dong, Ce Zheng, Xu Sun, Lingpeng Kong, and Qi Liu. Can language models understand physical concepts? In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[23] Genevieve Patterson and James Hays. Coco attributes: Attributes for people, animals, and objects. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*, pages 85–100. Springer, 2016.

[24] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017.

[25] Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Learning to predict visual attributes in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13018–13028, 2021.

[26] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, Amir Mousavi, Yiwen Song, Abhimanyu Dubey, and Dhruv Mahajan. Paco: Parts and attributes of common objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7141–7151, June 2023.

[27] Yuqing Du, Ksenia Konyushkova, Misha Denil, Akhil Raju, Jessica Landon, Felix Hill, Nando de Freitas, and Serkan Cabi. Vision-language models as success detectors. In *Proceedings of The 2nd Conference on Lifelong Learning Agents*, pages 120–136, 2023.

[28] Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Marcin Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael S Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. Socratic models: Composing zero-shot multimodal reasoning with language. In *The Eleventh International Conference on Learning Representations*, 2023.

[29] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3081–3089, 2022.

[30] Zhenwei Shao, Zhou Yu, Meng Wang, and Jun Yu. Prompting large language models with answer heuristics for knowledge-based visual question answering. In *Computer Vision and Pattern Recognition (CVPR)*, pages 14974–14983, 2023.

[31] Deyao Zhu, Jun Chen, Kilichbek Haydarov, Xiaoqian Shen, Wenxuan Zhang, and Mohamed Elhoseiny. Chatgpt asks, blip-2 answers:

Automatic questioning towards enriched visual descriptions. *arXiv preprint arXiv:2303.06594*, 2023.

[32] Xufeng Zhao, Mengdi Li, Cornelius Weber, Muhammad Burhan Hafez, and Stefan Wermter. Chat with the environment: Interactive multimodal perception using large language models. *arXiv preprint arXiv:2303.08268*, 2023.

[33] Meta. Egoobjects dataset. https://ai.facebook.com/datasets/egoobjects-dataset/, Last accessed on 2023-05-28.

[34] Dorsa Sadigh, Anca D. Dragan, S. Shankar Sastry, and Sanjit A. Seshia. Active preference-based learning of reward functions. In *Proceedings of Robotics: Science and Systems (RSS)*, July 2017.

[35] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[36] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022.

[37] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

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

[39] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip. July 2021.

[40] Dongxu Li, Junnan Li, Hung Le, Guangsen Wang, Silvio Savarese, and Steven C. H. Hoi. Lavis: A library for language-vision intelligence, 2022.

[41] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[42] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.

[43] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019.

[44] Matthias Minderer, Alexey Gritsenko, Maxim Neumann Austin Stone, Dirk Weissenborn, Aravindh Mahendran Alexey Dosovitskiy, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple open-vocabulary object detection with vision transformers. *ECCV*, 2022.

[45] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[46] Yixin Lin, Austin S. Wang, Giovanni Sutanto, Akshara Rai, and Franziska Meier. Polymetis. https://facebookresearch.github.io/fairo/polymetis/, 2021.

APPENDIX

A. Physical Concepts Details

In this section, we provide details on how we define each of our ten physical concepts, which we communicate to crowd-workers before annotation. We also list the pre-defined options for categorical concepts.

Continuous-Valued, Applicable to All Objects

Mass: This refers to how heavy an object is. If an object has contents inside, this includes how heavy both the object and its contents are combined.

Fragility: This refers to how easily an object can be broken or damaged. An object has higher *fragility* than another if a person would handle it more carefully to avoid breaking it.

Deformability: This refers to how easily an object can change shape without breaking. An object has more *deformability* than another if less force is needed to change its shape without breaking it.

Density (*held-out*): This refers to the amount of mass per unit of volume of the object. If an object has contents inside, this only refers to the object, not the contents.

Continuous-Valued, Applicable to Containers

Liquid Capacity (*held-out*): This refers to the volume of liquid a container can contain without spilling.

Categorical-Valued, Applicable to All Objects

Material: This refers to what an object is made of. If an object is made of multiple materials, it refers to what material makes up the largest portion of the object that is visible. This does not refer to the contents of a container. The pre-defined options we include are *plastic*, *glass*, *ceramic*, *metal*, *wood*, *paper*, *fabric*, *food*, *unknown*, and *other* (annotator provides an open-ended response if this option is chosen).

Transparency: This refers to how much can be seen through an object. The pre-defined options we include are *transparent*, *translucent*, *opaque*, and *unknown*. *Transparent* refers to an object that can be seen clearly through, almost as if it was not there. *Translucent* refers to an object where some details can be seen through the object, but the details are not as clear as if it was *transparent*. *Opaque* refers to an object that cannot be seen through at all. This concept only refers to the object itself, and not the contents of a container. If different parts of an object have different levels of *transparency*, it refers what level applies to the largest visible portion of the object.

Categorical-Valued, Applicable to Containers

Contents: This refers to the contents of a container that are clearly visible and identifiable. The pre-defined options we include are *nothing*, *water*, *food*, *oil*, *soap*, *unknown*, and *other* (annotator provides an open-ended response if this option is chosen).

Can Contain Liquid: This refers to if a container can be used to transport a liquid across a room without a person needing to be particularly careful about not spilling it. The pre-defined options we include are *yes*, *no*, and *unknown*.

Is Sealed: This refers to if a container can be rotated by any amount in any direction without spilling its contents. The pre-defined options we include are *yes*, *no*, and *unknown*.

Container Categories. We define the following object categories from EgoObjects as containers: *bottle*, *container*, *plate*, *bowl*, *mug*, *water glass*, *measuring cup*, *wine glass*, *tea cup*, *frying pan*, *flowerpot*, *tin can*, *kettle*, *vase*, *coffee cup*, *mixing bowl*, *saucer*, *jug*, *serving tray*, *pitcher (container)*, and *picnic basket*.

B. Automatic Annotation Details

We list the object categories we assign to *high* and *low* tiers for automating preference pair annotations for continuous concepts in Table VIII. We list the object categories for which we assign a concept label in Table IX. If a concept is not listed in these tables, we do not provide automatic annotations for that concept.

We originally assigned the label *metal* for *material* to the object category *house/car key*, but realized after crowdsourcing that not all instances of this category should have been given this assignment. Therefore, we manually labeled these examples for *material*, but still considered these to be automatic annotations for dataset purposes.

C. Crowd-Sourcing Details

Choosing Annotation Images. There are multiple bounding box images in EgoObjects for each object instance. To determine which to present for annotating an object, we choose the bounding box with the highest CLIP [22] similarity with the object's category label, as a heuristic for the object's visibility. We use the CLIP-ViT-H-14-laion2B-s32B-b79K model from OpenCLIP [39]. In Fig. 6, we show an example of randomly sampled bounding boxes for an instance of the object category *guitar*, arranged from left-to-right in decreasing order of CLIP similarity. The objects in bounding boxes with lower CLIP similarities tend to be less visible.

Fig. 6: Bounding boxes for an instance of *guitar*, in decreasing order of CLIP similarity

Attention Checks. We generate attention checks for crowdworkers by randomly sampling from the automatic annotations, which have known labels. For the concepts *contents*, *density*, and *liquid capacity*, for which there are no automatic annotations, we manually label a small set of objects for attention checks.

Other Details. Each annotation job on Prolific consisted of 250 annotations for a single concept, of which 25 are attention checks. Participants were paid an average of 15.50 US dollars per hour, and each annotation job took on average 20-30 minutes to complete, depending on the concept.

Concept	High	Low
Mass	television, microwave oven, table, nightstand, chest of drawers	pen, paper, spoon, fork, glasses, sunglasses, scissors, watch, necklace, house/car key, pencil, earrings, ring, screwdriver, book, container, plate, bowl, pillow, remote control, clothing, mug, laptop, knife, mobile phone, toy, computer mouse, water glass, towel, headphones, spatula, frying pan, measuring cup, banana, wallet, blanket, candle, apple, wine glass, picture frame, computer keyboard, game controller/pad, tea cup, tin can, handbag, whisk, orange, belt, plastic bag, salt and pepper shakers, cutting board, perfume, stapler, footwear, tablet coputer, teddy bear, cookie, scarf, coffee cup, ball, mixing bowl, pear, alarm clock, light switch, bread, jacket, tennis ball, sandal, saucer, laptop charger, camera, yoga mat, power plugs and sockets, cream, shirt, baseball bat, sun hat, paper towel, kitchen knife, doll, can opener, sock, facial tissue holder, boot, hair dryer
Fragility	water glass, television	house/car key, dumbbell, screwdriver, kitchen knife
Deformability	pillow, clothing, towel, blanket, belt, plastic bag, scarf, jacket, yoga mat, shirt, paper towel, sock	remote control, mug, mobile phone, computer mouse, water glass, frying pan, flowerpot, scissors, wine glass, house/car key, dumbbell, cutting board, microwave oven, toaster, blender, pressure cooker, kitchen knife, table, spoon, laptop, knife, fork, glasses, spatula, sunglasses, chair, measuring cup, pencil, picture frame, computer keyboard, game controller/pad, tea cup, tin can, salt and pepper shakers, television, coffeemaker, stapler, tablet computer, kettle, vase, coffee cup, mixing bowl, computer monitor, stool, ring, alarm clock, light switch, saucer, printer, screwdriver, guitar, camera, jug, gas stove, baseball bat, humidifier, chest of drawers, sink, can opener, nightstand, hair dryer

TABLE VIII: Object category assignments to *high* and *low* tiers for continuous concepts

Concept	Label	Categories
Material	Plastic	remote control, computer mouse, computer keyboard, game controller/pad, plastic bag
	Glass	water glass, wine glass
	Metal	tin can, kitchen knife, can opener
	Paper	book, paper, paper towel
	Fabric	clothing, towel, blanket, scarf, sock
	Food	banana, apple, orange, cookie, pear, bread
Transparency	Transparent	wine glass
	Opaque	book, pillow, remote control, clothing, laptop, mobile phone, towel, headphones, spatula, chair, frying pan, banana, wallet, flowerpot, scissors, apple, houseplant, house/car key, pencil, computer keyboard, tin can, whisk, dumbbell, orange, belt, cutting board, toaster, teddy bear, tablet computer, cookie, pear, computer monitor, stool, light switch, bread, pressure cooker, scarf, laptop charger, guitar, camera, yoga mat, shirt, baseball bat, paper towel, kitchen knife, sink, chest of drawers, can opener, boot, nightstand, hair dryer
Can Contain Liquid	Yes	bottle, mug, water glass, measuring cup, wine glass, tea cup, kettle, coffee cup, mixing bowl, jug, pitcher (container), tin can
	No	picnic basket, serving tray
Is Sealed	No	plate, bowl, mug, water glass, measuring cup, wine glass, tea cup, frying pan, flowerpot, kettle, vase, coffee cup, mixing bowl, saucer, jug, serving tray, pitcher (container), picnic basket

TABLE IX: Concept label assignments of object categories for categorical concepts

In the annotation user interface, for each object example, the object is shown in the context of its surrounding scene, with the object indicated by its bounding box. We also provide the object's category label to help clarify which object is to be annotated. Crowd-workers can choose an annotation label by clicking on an associated button, or typing an associated keyboard key. We also provide a *back* option to go to the previous example to correct mistakes. For the concepts *material* and *contents*, the user may choose *other* as an option, whereupon they are presented with a text box to type an open-ended label. We do not annotate objects from the categories *pet*, *cat*, and *dog*, to omit objects that are living.

We provide instructions to annotators that are specific to each concept, to encourage annotations that agree with our concept definitions. We provide an image of the instruction page provided to annotators for the *fragility* concept, which also includes an example of the annotation user interface, in Fig. 7. The instructions for how to annotate each property are also repeated at the bottom of the annotation user interface.

We detail the number of examples per concept and dataset split for PHYSOBJECTS in Table X. This is before any preprocessing of the data for annotator agreement or labels. For the crowd-sourced data, the count refers to the number of examples, not the number of annotations, for which there are three times as many. We also provide the percent of crowd-sourced examples with majority agreement (at least 2/3) and unanimous agreement per concept in Table XI.

D. Training Details

Hyperparameters. We provide hyperparameters used for fine-tuning InstructBLIP in Table XII. These hyperparameters are largely derived from those proposed for fine-tuning BLIP-2 [10]. When fine-tuning, we only update the Q-Former parameters, as done during instruction tuning for InstructBLIP. We use a linear warmup of the learning rate, followed by a linear decay with a minimum learning rate of 0. We fine-tune using mixed precision bfloat16 training. We use a prompt template for questions, which is used both during training and inference. We load the InstructBLIP model using the LAVIS library [40]. We train and evaluate using the evaluation image processor provided by LAVIS, as we do not use image data augmentation.

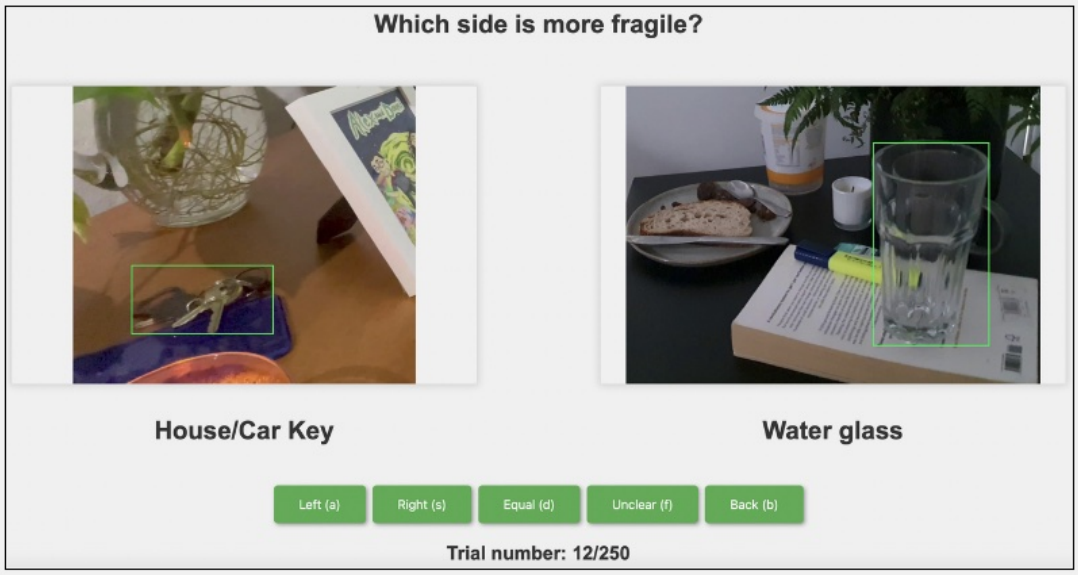
Validation & Data Filtering. For most experiments, we evaluate on validation data every 250 gradient steps and choose the checkpoint with the lowest validation loss. For experiments fine-tuning for a single concept, we validate every 100 gradient steps. Our validation set consists of all validation data for all concepts without balancing, except we limit the number of automatically generated examples for *mass* and *deformability* to 100. For validation data, we only use the bounding box image with the highest CLIP object category similarity score for each object, which for

Physical Properties Annotations

Instructions

Thank you for participating in this user study! In this task, you will classify 250 pairs of images based on how **fragile** the objects are.

When you begin this study, you will see an interface similar to the one below:



You will be presented with a pair of images, each of which contains an object boxed in green. There is a label below the images to help confirm what specific object we want you to analyze. **Only** use the label to identify the object, **not** to inform your annotation. If there are multiple objects of the given label in the green box, choose the object that best matches the box.

Below these are options for you to click. You may also use your keyboard to indicate an option, using the letter in parenthesis for each button. If you make a mistake, you can click the green **Back** button (or press 'b'), which allows you to go back to the previous example.

If you consider one of the objects to be more fragile than the other, you should indicate which one is more fragile using either **Left** or **Right**. If you think both objects are roughly the same fragile, choose **Equal**. If you cannot determine from the images due to low resolution, bad lighting, or other reasons, choose **Unclear**.

We define fragility as **how easily an object can be broken or damaged**. You should consider an object more fragile than another if you would handle it more carefully to avoid breaking it.

In general, please try to give some annotation instead of **Unclear** if you are somewhat confident in your responses. Even if you may be wrong, try your best to give an answer.

Occasionally, you will randomly be given an example that has a known correct answer. **You must achieve at least 80% accuracy on these examples to be compensated for the study.**

Once you complete all 250 examples, you are done, and you will be given a link to return back to Prolific to confirm your submission.

Ready to start?

We see that your Prolific ID is **EXAMPLE**. Please contact the researchers if this is not the case.

Consent Confirmation:

Do you consent to the collection of your responses and a temporary collection of your Prolific ID?

○ Yes

Start

Fig. 7: Instruction page for the *fragility* concept

Concept	Source	Train	Validation	Test
Mass	Crowd-sourced	2108	86	56
	Automatic	87269	4536	2688
Fragility	Crowd-sourced	2096	99	57
	Automatic	2397	110	80
Deformability	Crowd-sourced	2101	84	65
	Automatic	293540	13384	9888
Material	Crowd-sourced	2316	460	374
	Automatic	612	130	119
Transparency	Crowd-sourced	1993	394	313
	Automatic	1046	224	194
Contents	Crowd-sourced	641	134	125
	Automatic	0	0	0
Can Contain Liquid	Crowd-sourced	318	68	64
	Automatic	342	70	67
Is Sealed	Crowd-sourced	164	30	31
	Automatic	444	91	86
Density (*held-out*)	Crowd-sourced	0	0	500
Liquid Capacity (*held-out*)	Crowd-sourced	0	0	500

TABLE X: Number of examples per concept and dataset split

Concept	% Majority Agreement	% Unanimous Agreement
Mass	94.2	58.8
Fragility	93.6	53.1
Deformability	90.5	48.1
Material	93.7	59.4
Transparency	97.0	72.5
Contents	90.4	49.8
Can Contain Liquid	99.3	64.2
Is Sealed	98.2	74.7
Density (*held-out*)	93.3	50.7
Liquid Capacity (*held-out*)	89.1	46.0

TABLE XI: Agreement among crowd-workers per concept

crowd-sourced data is also the bounding box image presented for annotation. For crowd-sourced validation data, we filter our data to only include examples with at least 2/3 majority agreement among annotators, and only use the majority label. We do not apply this filtering for training data. For preference pair annotations, we remove data annotated with *unclear*.

Dataset Balancing. We construct sub-datasets for dataset balancing purposes. For the categorical concepts except *is sealed*, we combine the crowd-sourced and automatically annotated data for each concept into one sub-dataset per concept. For the other concepts, we keep separate sub-datasets for crowd-sourced and automatically annotated data. We keep separate sub-datasets for *is sealed* because for its crowd-sourced data, we only train using the bounding box image for the object that was presented for annotation, rather than randomly sampling one of its bounding box images (as described in the below sub-section), as values for this concept may change for the same object instance. We keep separate datasets for the continuous concepts because there is a large imbalance between the number of crowd-sourced and automatically annotated examples for these concepts. To balance these sub-datasets, we sample from each of them during training at a rate proportional to the square root of the number of annotations in the sub-dataset, as proposed in InstructBLIP for instruction tuning.

Additional Training Details. For most objects, each time we sample one for training, we randomly sample one of its bounding box images as input to the model, as a form of data augmentation. We do not do this with crowd-sourced data for the *contents* and *is sealed* concepts, because labels for these concepts may vary across different images of the same object. Instead, we only use the bounding box image that was presented for annotation.

To promote robustness to different queries to the VLM, we include object category labels in the question prompt for half of the training examples (e.g., asking "Is this *bottle* heavy?"),

Hyperparameter	Value
Max fine-tuning steps	10000
Warmup steps	1000
Learning rate	1e-5
Batch size	128
AdamW β	(0.9, 0.999)
Weight decay	0.05
Image resolution	224
Prompt template	Question: {} Respond unknown if you are not sure. Short answer:

TABLE XII: Hyperparameters for fine-tuning InstructBLIP

Concept	Question Prompt
Mass	Is this object heavy?
Fragility	Is this object fragile?
Deformability	Is this object deformable?
Material	What material is this object made of?
Transparency	Is this object transparent, translucent, or opaque?
Contents	What is inside this container?
Can Contain Liquid	Can this container hold a liquid inside easily?
Is Sealed	Is this container sealed?
Density (*held-out*)	Is this object dense?
Liquid Capacity (*held-out*)	Can this object hold a lot of liquid?

TABLE XIII: Question prompts for each concept, without object category labels

and omit this information in the other half (e.g., asking "Is this *object* heavy?"). We experimented with training on one or multiple question prompts per concept, and found this to not significantly affect performance, so we only use one prompt per concept for simplicity. We include the question prompts for each concept in Table XIII. These are versions of the prompts without object category labels. When including category labels, we replace either the word "object" or "container" with the object's category label from EgoObjects. We also pluralize the prompt to have correct grammar if the category label is plural.

We experimented with removing Q-Former text conditioning in InstructBLIP while fine-tuning, and found this to improve results on general VQA evaluation and evaluation with held-out paraphrased question prompts, so we report results using models trained without this text conditioning. In our ablation results in Table XXI, we find that this does not significantly change performance for our main crowd-sourced evaluation.

E. Evaluation Details

Further PHYSOBJECTS Evaluation Details. For crowd-sourced test evaluation data, we only include examples with at least 2/3 annotator agreement, and use the majority label as ground-truth. For categorical concepts, we predict by choosing the label with the highest likelihood out of all labels in PHYSOBJECTS for the concept. For continuous concepts, we predict the object in a pair with the higher score from Section IV as the one with higher concept value. We only evaluate on preference examples with a definite, non-equal preference label. For the *Most Common* baseline with continuous concepts, we also only include examples with a definite, non-equal preference when determining the most common label in the training data. We note that that *Most Common* baseline is not particularly meaningful for continuous concepts, because the preference labels and predictions are invariant to ordering in each preference pair. Therefore, a more natural baseline for these concepts would be random guessing, which would achieve 50% accuracy.

Similarly as with validation data, for test data we only evaluate using the bounding box image with the highest CLIP object category similarity per object, which for crowd-sourced data is also the bounding box image presented for annotation. We evaluate using the same question prompts per concept as during training, which are listed in Table XIII. Unless stated otherwise, we report evaluation results without object category labels in the question prompt, because this gives slightly better results for the base InstructBLIP model.

Text Only Baseline. For this baseline, we use ground truth object category labels from EgoObjects. We use the 'text-davinci-003' InstructGPT model [41] as our LLM. For each concept, we use 128 in-context examples randomly sampled from the training data in PHYSOBJECTS for that concept. Because in-context learning is limited by context length, and therefore it is desirable to use the best quality in-context examples when possible, we first apply to the training data the same majority filtering process used on crowd-sourced test data as described in the previous subsection. We also remove preference annotations with the label *unclear*, as

Concept	Question Prompt
Mass	Does this object weigh a lot?
Fragility	Is this object easily breakable?
Deformability	Is this object easily bendable?
Material	What is this object made of?
Transparency	Would you describe this object as opaque, transparent, or translucent?
Contents	What does this container contain?
Can Contain Liquid	Is this container able to hold water inside easily?
Is Sealed	Is this container sealed shut?

TABLE XIV: Paraphrased question prompts for main concepts, without object category labels

done in our VLM fine-tuning setup. We treat each example as a question answering task, using question prompts for each concept similar to those in Table XIII, but modified to refer to general object classes, rather than specific instances. We make predictions by selecting the most likely completion of the LLM conditioned on the in-context examples and test example. For categorical concepts, we first include in the LLM context all possible labels in PHYSOBJECTS for the concept. For continuous concepts, because we only evaluate on examples with definite preferences, we restrict predictions to only definite preferences using logit bias, although the in-context examples may include *equal* as a possible answer.

Paraphrased Question Prompts. In Table XIV, we list the paraphrased prompts used in the evaluation for Table IV.

	InstructBLIP	PG-InstructBLIP (ours)
VQAv2	71.4	67.5
OK-VQA	52.4	48.7

TABLE XV: Accuracy on existing VQA benchmarks

Limited VQA Degradation. Ideally, training on PHYSOBJECTS should be done while co-training on other vision and language datasets to preserve general reasoning abilities. In this work, we do not do this because we focus primarily on physical reasoning. However, we show that fine-tuning on only PHYSOBJECTS does not significantly degrade general VQA performance. In Table XV, we compare InstructBLIP to PG-InstructBLIP on VQAv2 [42] and OK-VQA [43]. These results suggest that existing systems using VLMs can benefit from PHYSOBJECTS for physical reasoning, without sacrificing other reasoning abilities.

We perform VQA evaluation using the LAVIS library, using their configurations for evaluation of BLIP-2. Although PG-InstructBLIP is fine-tuned without Q-Former text conditioning, we found that Q-Former text conditioning during VQA evaluation improved performance, so we report these results. We believe this is because InstructBLIP was instruction tuned with this text conditioning. We also experimented with VQA evaluation on PG-InstructBLIP fine-tuned with Q-Former text conditioning, but found this to have worse results, possibly due to overfitting on our limited variety of question prompts. We believe these issues can be mitigated by co-training on PHYSOBJECTS in combination with other vision and language datasets, which we leave for future work.

Motivated by these VQA results, for our planning evaluations we also evaluate PG-InstructBLIP using Q-Former text conditioning, to avoid possible degradation when answering questions that do not pertain concepts in PHYSOBJECTS. We verified that evaluating PG-InstructBLIP using Q-Former text conditioning did not significantly affect test accuracy on PHYSOBJECTS.

Including Object Category Labels in Question Prompts. We generally report evaluation results without ground-truth object category labels in the question prompt. In Table XVI, we compare including object category labels or not, and find that all models are not extremely sensitive to this.

Including Concept Definitions in Question Prompts. While we did not spend extensive effort designing the question prompts for each concept (shown in Table XIII), we aimed for them to be concise while still eliciting the desired concept. As seen in Table XVIII, the base InstructBLIP model achieves above chance performance on all concepts, suggesting that these prompts do elicit the desired concept to some extent. However, these prompts do not contain our definitions for each concept provided to annotators, as described in Appendix A. We analyze whether including concept definitions in the question prompt would improve base VLM performance in Table XVIII, which contains our original crowd-sourced test accuracy results, with additional evaluation of the base InstructBLIP model using modified prompts that contain concept definitions, which we provide in Table XVII. We find that while including concept definitions improves performance for some concepts (*mass*, *deformability*, *contents*, *can contain liquid*), this still does not match PG-InstructBLIP on these concepts, and overall performance in fact *decreases* compared to the original prompts. We believe this could be because InstructBLIP does not have strong enough language understanding to properly incorporate the concept definitions when providing responses. For this reason, and for simplicity, we use prompts without concept definitions in the rest of our experiments.

Using a Smaller VLM. To analyze the effect of VLM size on physical reasoning, in Table XIX we provide evaluation results using the InstructBLIP version with the smaller Flan-T5 XL as its base LLM, compared to the Flan-T5 XXL version used in all other experiments. We find that while the smaller Flan-T5 XL version generally has worse base

	InstructBLIP		Single Concept FT (ours)		PG-InstructBLIP (ours)	
Category Labels	Yes	No	Yes	No	Yes	No
Mass	60.0	62.2	**84.4**	80.0	80.0	80.0
Fragility	75.7	78.4	91.2	91.2	**97.3**	94.6
Deformability	69.8	67.4	88.4	**95.3**	90.7	93.0
Material	73.3	67.1	**86.8**	83.7	85.7	84.6
Transparency	84.5	85.8	89.1	89.4	89.8	**90.1**
Contents	34.2	35.1	80.7	81.6	82.5	**83.3**
Can Contain Liquid	57.8	59.4	84.4	84.4	82.8	**87.5**
Is Sealed	71.0	74.2	80.6	80.6	**87.1**	**87.1**
Average	65.8	66.2	85.7	85.8	87.0	**87.5**

TABLE XVI: Test accuracy for main concepts on crowd-sourced PHYSOBJECTS, with and without object category labels

Concept	**Question Prompt**
Mass	The heaviness of an object refers to its mass. It includes the contents of the object if it has something inside it. Is this object heavy?
Fragility	Fragility refers to how easily an object can be broken or damaged. Is this object fragile?
Deformability	Deformability refers to how easily an object can change shape without breaking. Is this object deformable?
Material	The material of an object refers to what material makes up the largest portion of the object that is visible. It does not refer to the contents of a container. What material is this object made of?
Transparency	Transparency refers to how much can be seen through an object. A transparent object can be clearly seen through, almost as if it was not there. A translucent object can be seen through with some details, but not as clearly as if it was transparent. An opaque object cannot be seen through at all. The transparency of an object does not refer to the transparency of its contents if it has anything inside it. Is this object transparent, translucent, or opaque? If different portions of the object have different levels of transparency, respond with the level that applies to the largest visible portion of the object.
Contents	What is inside this container? Only respond with contents that are clearly visible and identifiable.
Can Contain Liquid	A container can contain liquid if it can be used to transport a liquid across a room without a person needing to be particularly careful about not spilling it. Can this container contain liquid?
Is Sealed	A container is sealed if it can be rotated by any amount in any direction without spilling its contents if it has anything inside. Is this container sealed?

TABLE XVII: Question prompts with definitions for each main concept, without object category labels

	InstructBLIP		PG-InstructBLIP (ours)
Prompt Type	Original	w/ Concept Definitions	Original
Mass	62.2	71.1	**80.0**
Fragility	78.4	67.6	**94.6**
Deformability	67.4	69.8	**93.0**
Material	67.1	66.0	**84.6**
Transparency	85.8	65.3	**90.1**
Contents	35.1	36.0	**83.3**
Can Contain Liquid	59.4	64.1	**87.5**
Is Sealed	74.2	64.5	**87.1**
Average	66.2	63.0	**87.5**

TABLE XVIII: Test accuracy for main concepts on crowd-sourced PHYSOBJECTS, with additional base InstructBLIP evaluation on prompts with definitions for each concept

	InstructBLIP		PG-InstructBLIP (ours)	
LLM Version	Flan-T5 XL	Flan-T5 XXL	Flan-T5 XL	Flan-T5 XXL
Mass	62.2	62.2	**82.2**	80.0
Fragility	64.9	78.4	**97.3**	94.6
Deformability	48.8	67.4	**97.7**	93.0
Material	69.1	67.1	82.6	**84.6**
Transparency	74.0	85.8	87.5	**90.1**
Contents	18.4	35.1	**86.8**	83.3
Can Contain Liquid	68.8	59.4	**87.5**	**87.5**
Is Sealed	67.7	74.2	80.6	**87.1**
Average	59.2	66.2	**87.8**	87.5

TABLE XIX: Test accuracy for main concepts on crowd-sourced PHYSOBJECTS, using different VLM versions

performance, after fine-tuning on PHYSOBJECTS, we see that performance is comparable between the two model sizes. This suggests that for physical reasoning, fine-tuning on human data such as PHYSOBJECTS could reduce the need for larger model sizes. While fine-tuned evaluation performance is similar across model sizes, for simplicity of comparison, we only report results using the larger Flan-T5 XXL models in all other experiments.

	InstructBLIP	PG-InstructBLIP (ours)
Mass	72.8	**99.9**
Fragility	95.0	**100**
Deformability	96.0	**98.8**
Material	89.1	**98.3**
Transparency	97.4	**100**
Can Contain Liquid	98.5	**100**
Is Sealed	**100**	**100**
Average	92.7	**99.6**

TABLE XX: Test accuracy for main concepts on automatically annotated PHYSOBJECTS

Results on Automatically Annotated Data. We report evaluation results on automatically annotated data in Table XX. Performance is generally much higher on this data compared to the crowd-sourced data, because these are easier examples that can be determined from object categories alone.

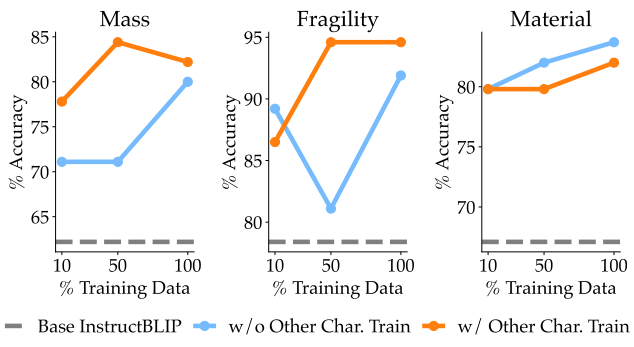


Fig. 8: Performance scaling on held-out concepts

Held-Out Concept Scaling. In these experiments, we evaluate the transfer abilities of InstructBLIP across different concepts when fine-tuning on PHYSOBJECTS. We fine-tune models on data from PHYSOBJECTS for all concepts except one, and then report results of additional fine-tuning on the held-out concept. We compare to fine-tuning base InstructBLIP without training on the other concepts, and base InstructBLIP without any fine-tuning. Results for three concepts are shown in Fig. 8. We chose these concepts because we believed they had the most generalization potential from the other concepts. We find that there are some signs of positive transfer on *mass* and *fragility*, although we see slight negative transfer on *material*. We believe that more positive transfer could be attained by co-training with other vision and language datasets.

Ablations. We report additional ablation results on crowd-sourced PHYSOBJECTS examples in Table XXI. We list each ablation below:

1) *No Auto Data*: Instead of training on both crowd-sourced and automatically annotated data, we train on only crowd-sourced data.
2) *Filtered*: Instead of training on all annotations for crowd-sourced data, we filter the data similarly as during evaluation: we only include examples with at least 2/3 annotator agreement, and use the majority label as ground-truth.
3) *Q-Former Text*: Instead of removing Q-Former text conditioning during fine-tuning, we include it, as done for the original InstructBLIP model.
4) *No Category Info*: Instead of training on both question prompts with and without object category information, we only train on question prompts *without* it.
5) *Only Category Info*: Instead of training on both question prompts with and without object category information, we only train on question prompts *with* it. Here, unlike the rest of the evaluations, we evaluate with object category information to match the training setup.

We find that overall performance for each ablated version of our model does not change significantly, suggesting some robustness of our fine-tuning process to different design decisions. In particular, we find that including automatically

	PG-InstructBLIP (ours)	No Auto Data	Filtered	Q-Former Text	No Category Info	Only Category Info
Mass	80.0	**84.4**	75.6	80.0	75.6	77.8
Fragility	94.6	94.6	97.3	97.3	**100**	**100**
Deformability	**93.0**	**93.0**	90.7	90.7	**93.0**	88.4
Material	84.6	83.4	83.4	85.4	84.6	**86.5**
Transparency	90.1	89.4	89.1	**92.1**	89.8	91.7
Contents	83.3	81.6	85.1	**87.7**	84.2	86.8
Can Contain Liquid	87.5	89.1	85.9	84.4	**90.6**	84.4
Is Sealed	**87.1**	83.9	83.9	71.0	80.6	**87.1**
Average	87.5	87.4	86.4	86.1	87.3	**87.8**

TABLE XXI: Ablation results for main concepts on crowd-sourced PHYSOBJECTS

annotated data does not significantly impact performance on crowd-sourced data, which perhaps is not surprising because base InstructBLIP already performs well on automatically annotated examples, as seen in Table XX. *Only Category Info* very slightly improves upon PG-InstructBLIP, but uses privileged object category information at evaluation time.

F. Real Scene Planning Evaluation Details

Planning Framework. Our planning framework consists of first providing the scene image to an OWL-ViT ViT-L/14 open-vocabulary object detector [44], which produces object bounding boxes and category labels from the EgoObjects categories. We then provide the list of detected objects and the task instruction to our LLM, which is GPT-4 [38] with temperature 0. The LLM is additionally provided with the robotic primitives, and a few-shot chain-of-thought prompt [45] with instructions to ask questions about objects in the scene to determine how to complete the task, and then produce a plan using the primitives. There is no constraint on the questions that the LLM can ask, except for encouragement in the prompt to ask questions that can be answered with *yes* or *no*. The same prompt is used for all scenes and tasks, which we provide in Listing 1.

After the LLM asks a set of object-centric questions, a VLM answers each question prompted with the bounding box of the object indicated by the LLM, and then provides the LLM with its highest likelihood responses and their associated likelihoods/confidence scores, as done in prior work for VQA [30]. This continues until the LLM decides it has enough information, whereupon it either indicates that the task is not possible, or produces a plan consisting of a list of primitives to execute for the task. The few-shot examples in Listing 1 illustrate how interaction between the LLM and VLM for planning is structured.

Primitives. We list the primitives for our real scene planning evaluation below:

- go to object [X]
- pick up object [X]
- bring to human object [X]
- put down object [X]
- done

The primitives (except *done*) are parameterized by a letter (in place of [X]) that identifies each detected object in the scene. The assignment of letters is provided in the list of object detections given to the LLM planner.

Scenes and Tasks. In Table XXII, we provide the scene images in our evaluation, and the detected objects and task instructions for each scene. We also indicate the task type for each instruction.

Prompts. We provide the prompts used by our LLM-based planning framework for our scene planning evaluation. The version with VLM interaction is in Listing 1 and the version without VLM interaction is in Listing 2. The parts of the prompts in square brackets are replaced with the corresponding information specific to the task, in the same format as the prompt example.

Listing 1: Prompt for LLM planner with VLM interaction.

```
You are a household robot. You are able to move most
     household objects, but not large or heavy furniture.
     You are to be safe and not break anything.
You will be given a list of objects in the scene.
A human will give you a task instruction to perform.
First, ask questions about the objects to learn more about
      them to determine how to properly complete the task.
Indicate the object letters before asking the question,
     and do not provide the answer.
Format each question like "Question about object [A, B]:
     Is this object heavy?".
Only ask informative questions that help you understand
     how to properly complete the task.
Some of the object descriptions in the list may be
     inaccurate, so it is good to ask questions to confirm
      information about them.
Ask questions that can be answered with yes or no when
     possible.
Only ask one question at a time.
After each question, you will be given a response
     containing the 3 most likely possible answers for
     each object, with corresponding confidence scores.
An example response for one object would be "A: yes (0.8),
      no (0.1), unknown (0.1)".
Otherwise, use the information you have obtained to
     produce a numbered list of actions to perform the
     task, starting with "Plan:".
You are only able to perform the following actions, where
     "X" is a placeholder for an object letter from the
     given list of objects:
1. go to object X
2. pick up object X
3. bring to human object X
4. put down object X
5. done

Only perform actions that are possible. You are able to
     move most household objects, including most
     electronic appliances, but not large or heavy
     furniture or parts of a room (like a door or heater).
      You are to be safe and not break anything. Choose
     the "done" action when the plan is complete and then
     stop. If no plan can satisfy the task, tell the human
     , choose the "done" action and stop.
```

```
Only respond with the plan list and do not include any
    other text.

Scene 1:
The following objects are in the scene: A (bottle), B (
    bowl), C (countertop), D (shirt), E (lock)

Instruction: Bring me the heaviest object.

Thought: I can ask which object is heavy out of all the
    objects, and I can find the one with the largest
    weight that I can carry.
Question about [A, B, C, D, E]: Is this object heavy?
Answer:
A: No (0.50), Yes (0.24), Unknown (0.21)
B: No (0.90), Unknown (0.05), Yes (0.04)
C: Yes (0.80), Unknown (0.10), No (0.05)
D: No (0.73), Unknown (0.20), Yes (0.06)
E: No (0.41), Yes (0.35), Unknown (0.19)
Thought: The heaviest object is the one with the highest
    Yes value or the lowest No value (if none are Yes).
    The heaviest object is C. However, I cannot carry a
    countertop since it is a heavy piece of furniture.
    Therefore, I will to choose the next heaviest, which
    would be E, a lock that I can carry easily.
Plan:
1. Go to object E
2. Pick up object E
3. Bring to human object E
4. Done

Instruction: Bring me the most deformable container.
Thought: I can first ask which objects are containers out
    of all the objects I can carry. Of these containers,
    I can ask which is deformable.
Question about [A, B, C, D, E]: Is this object a container
    ?
Answer:
A: Yes (0.55), No (0.20), Unknown (0.19)
B: Yes (0.90), Unknown (0.05), No (0.04)
C: No (0.85), Unknown (0.06), Yes (0.05)
D: No (0.62), Unknown (0.20), Yes (0.06)
E: No (0.41), Yes (0.35), Unknown (0.19)
Thought: The only objects that are confidently Yes are B
    and A. I should ask which is deformable next.
Question about [A, B]: Is this object deformable?
Answer:
A: Yes (0.80), Unknown (0.15), No (0.04)
B: No (0.55), Unknown (0.26), Yes (0.05)
Thought: The most deformable object is the one with the
    highest Yes value or the lowest No value (if none are
     Yes). Since some of the answers are Yes, the answer
    is A (the highest yes value).
Plan:
1. Go to object A
2. Pick up object A
3. Bring to human object A
4. Done

Scene 2:
The following objects are in the scene: [list of objects
    in the scene]

Instruction: [instruction specified here]
```

Listing 2: Prompt for LLM planner without VLM interaction.

```
You are a household robot. You are able to move most
    household objects, but not large or heavy furniture.
    You are to be safe and not break anything.
You will be given a list of objects in the scene.
A human will give you a task instruction to perform.
Use the object information to produce a numbered list of
    actions to perform the task, starting with "Plan:".
You are only able to perform the following actions, where
    "X" is a placeholder for an object letter from the
    given list of objects:
1. go to object X
2. pick up object X
3. bring to human object X
4. put down object X
5. done

Only perform actions that are possible. You are able to
    move most household objects, including most
    electronic appliances, but not large or heavy
    furniture or parts of a room (like a door or heater).
     You are to be safe and not break anything. Choose
    the "done" action when the plan is complete and then
    stop. If no plan can satify the task, tell the human,
     choose the "done" action and stop.
Only respond with the plan list and do not include any
    other text.

Scene 1:
The following objects are in the scene: A (bottle), B (
    bowl), C (countertop), D (shirt), E (lock)

Instruction: Bring me the heaviest object.

Thought: I cannot carry a countertop since it is a heavy
    piece of furniture. Out of the rest, a good guess
    would be the lock.
Plan:
1. Go to object E
2. Pick up object E
3. Bring to human object E
4. Done

Instruction: Bring me the most deformable container.
Thought: Typically shirts are easy to fold, so a good
    choice for the most deformable object would be the
    shirt.
Plan:
1. Go to object D
2. Pick up object D
3. Bring to human object D
4. Done

Scene 2:
The following objects are in the scene: [list of objects
    in the scene]

Instruction: [instruction specified here]
```

Evaluation Procedure. We evaluate task planning accuracy using a non-author human evaluator. For each evaluation, the evaluator is given the task instruction, the image of the scene, the list of detected objects in the scene and their bounding boxes, and the generated task plan, and they are asked to evaluate whether the task plan successfully performed the task instruction for the given scene. We provide the following instructions to the evaluator on what to consider when evaluating whether a plan was correct:

> Instructions: For each scene, there is a list of objects (under 'Options:'). Below that is a table of tasks for that scene. The instruction given to a robot is on the left. On the right are the choices from 3 different robots. You need to mark which ones are correct or incorrect. It may be possible that multiple robots got it right or none of them got it right. Be aware that in tasks that involve moving objects, the robot should not plan to move an object that is very heavy, like large furniture.

While the planner usually creates plans using only the provided primitives, it sometimes specifies primitives that were not provided. Because the purpose of this evaluation is on assessing if a LLM planner can benefit from physical reasoning using a VLM, and not on creating a functional planning system, we do not do anything to handle these cases. We the evaluator to judge if these plans satisfy the task instruction like the others. We provide example executions for different versions of our planning framework on our website.

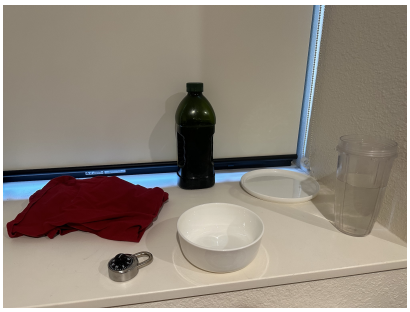
Scene Image	Object Detections	Task Instructions
	1) bottle 2) pitcher (container) 3) bowl [flatter bowl] 4) towel [shirt] 5) countertop 6) bowl [taller ceramic bowl] 7) measuring cup [lock]	1) Bring me the heaviest object. [S] 2) Bring me the most deformable object. [S] 3) Bring me the most fragile object. [S] 4) Bring me all containers that you can confidently determine have water. [M] 5) Bring me the container with oil. [M] 6) Among all empty containers, bring me the ones that cannot be used to carry water. [M] 7) Bring me the metal object. [S]
	1) suitcase [blue crate] 2) stool 3) hair dryer [mirror] 4) chair [chair that the mirror is on] 5) dishwasher [metal cabinet in top right] 6) chair [blue chair] 7) bottle [Elmer glue container] 8) bottle [Mod Podge container] 9) container [paint thinner container] 10) desk 11) mug [mug with paintbrushes] 12) facial tissue holder [container with glitter] 13) pencil	1) Bring me the heaviest object. [S] 2) Bring me a metal container. [M] 3) Bring me a small, empty cup that I can fill with water to clean my paintbrushes. If there are none, tell me that there are no small empty cups. [M] 4) Bring me the clear container with art supplies. [C] 5) Bring me the metal object that is reflective. [M] 6) Bring me paint thinner. [C] 7) Bring me a wooden object. [S]

TABLE XXII: Scene images, object detections, and task instructions for our real scene planning evaluation (scenes 1 and 2). The object category labels given by OWL-ViT are sometimes inaccurate or ambiguous, in which case we provide more precise labels in square brackets. Note that the planner only has access to the original OWL-ViT labels. Tasks are labeled with S, M, or C for *Single Concept*, *Multi-Concept*, or *Common Knowledge*, respectively.

Scene Image	Object Detections	Task Instructions
	1) clothing [green hoodie] 2) towel 3) clothing [striped shirt] 4) bottle [sunscreen bottle] 5) towel [socks] 6) mouse [ear thermometer] 7) suitcase 8) bottle [hand sanitizer] 9) hair dryer [dumbbell] 10) clothing [blue shirt]	1) Bring me the heaviest object. [S] 2) Bring me all clear containers. [M] 3) Bring me the hard plastic object. [M] 4) Bring me the lightest piece of clothing. [S] 5) Bring me the object I can pack my clothes into. [C] 6) It is cold outside. Bring me something that can keep me warm. [C] 7) It is sunny outside. Bring me the container of sunscreen. [C]
	1) facial tissue holder [paper towel dispenser] 2) light switch [left electric outlet] 3) light switch [right electric outlet] 4) mixer 5) toaster 6) kettle 7) paper towel 8) water glass [plastic cup] 9) salt and pepper shakers [salt] 10) bottle [jam container] 11) frying pan [baking pan] 12) container [salmon-colored container] 13) salt and pepper shakers [pepper] 14) countertop	1) Bring me the heaviest object. [S] 2) Bring me the heavier glass container. [M] 3) Bring me something that is easy to tear. [C] 4) Bring me the lightest container that is empty but can be filled with water. [M] 5) Bring me the most deformable container with a lid. [M] 6) Bring me all metal containers that can be used to carry water. [M] 7) Bring me the object that can be used in an oven. [C]

TABLE XXII: Scene images, object detections, and task instructions for our real scene planning evaluation (scenes 3 and 4). The object category labels given by OWL-ViT are sometimes inaccurate or ambiguous, in which case we provide more precise labels in square brackets. Note that the planner only has access to the original OWL-ViT labels. Tasks are labeled with S, M, or C for *Single Concept*, *Multi-Concept*, or *Common Knowledge*, respectively.

Scene Image	Object Detections	Task Instructions
	1) toaster 2) light switch [electric outlet] 3) envelope [napkin on microwave] 4) light switch 5) microwave oven [microwave] 6) door [microwave door] 7) bottle [glass sauce bottle] 8) picnic basket [drying rack] 9) soap dispenser 10) bottle [plastic bottle with blue vanilla flavor] 11) mug [dry mug] 12) sink 13) frying pan [dirty pan in sink] 14) mug [dirty mug in sink] 15) countertop 16) waste container 17) cupboard 18) plastic bag [trashbag]	1) Bring me the heaviest object that you can carry. [S] 2) Bring me an empty mug that I can use to make tea. [C] 3) Bring me the most deformable object. [S] 4) Bring me the glass object. [S] 5) Bring me a metal pan that is in the sink. [C] 6) Bring me the container that stores trash. [C]
	1) envelope [sign on napkin dispenser] 2) humidifier [napkin dispenser] 3) ladle [metal tongs] 4) food [two salad containers on the right] 5) bottle [red wine vinegar bottle] 6) frying pan [closer salad tray] 7) paper [napkin coming out of dispenser] 8) countertop 9) bottle [olive oil bottle] 10) bottle [black container on the right] 11) bottle [black container on the left] 12) juice [olive oil inside bottle] 13) cabinetry 14) countertop [more cropped in view of countertop] 15) bowl [paper plate under the counter]	1) Bring me the most deformable object. [S] 2) Bring me the lightest metal object. [M] 3) Bring me the heaviest glass container. [M] 4) Serve some food on a plate using objects in the scene. [C] 5) Bring me an empty container that you can confidently use to contain liquids, if one exists. Otherwise, tell me that no suitable containers exist. [M] 6) Bring me the container of olive oil. [C]

TABLE XXII: Scene images, object detections, and task instructions for our real scene planning evaluation (scenes 5 and 6). The object category labels given by OWL-ViT are sometimes inaccurate or ambiguous, in which case we provide more precise labels in square brackets. Note that the planner only has access to the original OWL-ViT labels. Tasks are labeled with S, M, or C for *Single Concept*, *Multi-Concept*, or *Common Knowledge*, respectively.

Scene Image	Object Detections	Task Instructions
	1) whiteboard 2) door [leftmost door] 3) paper 4) window [window of left door of rightmost pair] 5) door [left door of rightmost pair] 6) table [taller table] 7) chair [leftmost short chair facing towards the camera] 8) chair [tall chair] 9) chair [short chair behind pillar] 10) chair [rightmost short chair, facing towards the camera] 11) table [long wooden table] 12) door [rightmost door] 13) couch 14) chair [left side, facing away from camera] 15) coffee table	1) Go to the piece of furniture that is the softest. [C] 2) Go to the glass object that is not part of a window or door. [S] 3) Bring me the lightest object. [S] 4) Among all pieces of furniture, go to the one that is lightest. [C] 5) Go to the table that does not have a wooden surface. [S]
	1) box [binder] 2) bottle [large plastic tub] 3) bottle [plastic bottle] 4) box [algorithms textbook] 5) pitcher (container) [blue metal cup] 6) water glass [small glass cup] 7) headphones [phone cable] 8) dumbbell [power brick] 9) adhesive tape [ruler]	1) Bring me the container that is most likely to be metal. [M] 2) Bring me the heaviest container. Only consider the container itself, not the contents inside. [S] 3) Bring me the sealed container with juice. [M] 4) Bring me the most bendable object. [S] 5) Bring me the most fragile object. [S] 6) Bring me all containers that are made of plastic (with very high confidence). [M]

TABLE XXII: Scene images, object detections, and task instructions for our real scene planning evaluation (scenes 7 and 8). The object category labels given by OWL-ViT are sometimes inaccurate or ambiguous, in which case we provide more precise labels in square brackets. Note that the planner only has access to the original OWL-ViT labels. Tasks are labeled with S, M, or C for *Single Concept*, *Multi-Concept*, or *Common Knowledge*, respectively.

G. Real Robot Evaluation Details

Primitives. We list the primitives for our real robot evaluation below:

- move [X] to the side
- move [X] into [Y]
- done

Similarly as with our planning-only evaluation, our primitives are parameterized by a letter (in place of [X] or [Y]) that identifies each detected object in the scene. The assignment of letters is provided in the list of object detections given to the LLM planner.

Scenes and Tasks. In Table XXIII, we provide the scene images in our real robot evaluation, the detected objects in each scene, and the task instructions for each scene.

Prompts. We use the same prompt structure from the real scene planning evaluation. For tasks that involve moving an object to a side, we add "In your plan, you may only use the following primitive: move X to the side (where X is an object). Do not move furniture." For tasks that involve moving an object into a container, we add "In your plan, you may only use the following primitive: move X into Y (where X is an object and Y is a container). Do not move furniture."

Evaluation Procedure. We run real robot experiments using a 7-DoF Franka Emika Panda robot with a Robotiq 2F-85 gripper, using Polymetis [46] for real-time control. We obtain our pick-and-place primitives by collecting a kinesthetic demonstration for each primitive, and replaying the demonstration to execute it. The objects in the images provided to the object detector are not in the exact same positions as when the robot is acting, because the objects have to be rearranged when collecting demonstrations for each primitive. However, this does not affect the planner because we do not provide it object positions, as our planning framework does not make use of them.

Because our evaluation focuses on planning quality and not successful execution of primitives, we retry execution of each plan until all of the primitives are successfully executed. Therefore, our success rates are only reflective of planning quality, and not that of the primitives.

We evaluate the success rate of robot executions using a non-author human evaluator. For each evaluation, the evaluator is given the task instruction and a video of the robot executing the generated plan, and they are asked to evaluate whether the robot successfully performed the task instruction. We provide visualizations of robot executions on our website.

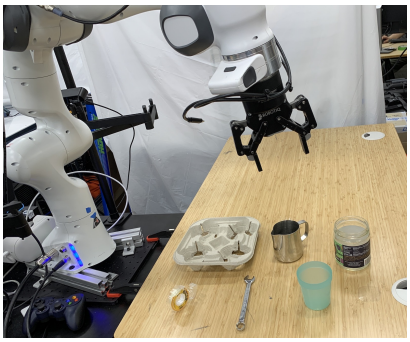
Scene Image	Object Detections	Task Instructions
	1) towel [handbag] 2) bottle [paint bottle] 3) bowl [yellow plastic bowl] 4) tool [container of metals] 5) desk [full table] 6) saucer 7) bowl [ceramic bowl] 8) spoon 9) pencil [pen] 10) pencil 11) milk [snack packet]	1) Move all objects that are not plastic to the side. 2) Find a container that has metals. Move all metal objects into that container. 3) Move all containers that can be used to carry water to the side. 4) Put the two objects with the least mass into the least deformable container. 5) Move the most fragile object to the side.
	1) bottle [glass jar] 2) mug [metal mug] 3) scale [cardboard cupholder] 4) mug [plastic cup] 5) adhesive tape 6) tool [wrench]	1) Put all containers that can hold water to the side. 2) Put all objects that are not plastic to the side. 3) Put all objects that are translucent to the side. 4) Put the three heaviest objects to the side. 5) Put a plastic object that is not a container into a plastic container. Choose the container that you are most certain is plastic.

TABLE XXIII: Scene images, object detections, and task instructions for our real robot evaluation. The object category labels given by OWL-ViT are sometimes inaccurate or ambiguous, in which case we provide more precise labels in square brackets. Note that the planner only has access to the original OWL-ViT labels.