ActPlan-1K: Benchmarking the Procedural Planning Ability of Visual Language Models in Household Activities

Ying Su[1], Zhan Ling[2], Haochen Shi[1], Jiayang Cheng[1], Yauwai Yim[1], Yangqiu Song[1]

[1]HKUST, [2]University of California, San Diego

ysuay@connect.ust.hk, z6ling@ucsd.edu

hshiah@connect.ust.hk, {jchengaj,ywyimaa,yqsong}@cse.ust.hk

Abstract

Large language models (LLMs) have been adopted to process textual task description and accomplish procedural planning in embodied AI tasks because of their powerful reasoning ability. However, there is still lack of study on how vision language models (VLMs) behave when multi-modal task inputs are considered. Counterfactual planning that evaluates the model's reasoning ability over alternative task situations are also under exploited. In order to evaluate the planning ability of both multi-modal and counterfactual aspects, we propose ActPlan-1K. ActPlan-1K is a multi-modal planning benchmark constructed based on ChatGPT and household activity simulator iGibson2. The benchmark consists of 153 activities and 1,187 instances. Each instance describing one activity has a natural language task description and multiple environment images from the simulator. The gold plan of each instance is action sequences over the objects in provided scenes. Both the correctness and commonsense satisfaction are evaluated on typical VLMs. It turns out that current VLMs are still struggling at generating human-level procedural plans for both normal activities and counterfactual activities. We further provide automatic evaluation metrics by finetuning over BLEURT model to facilitate future research on our benchmark.

1 Introduction

Recent researches adopt large language model (LLM) or multi-modal large language model (MLLM) as agents to accomplish embodied AI tasks (Ahn et al., 2022; Wang et al., 2023; Shah et al., 2022; Li et al., 2022; Driess et al., 2023; Huang et al., 2023; Zhang and Kordjamshidi, 2023; Xi et al., 2023). The LLMs generate procedural plan according to textual descriptions or embodied observations. The procedural plan consists of action sequences that needs to achieve in order to finish the task.

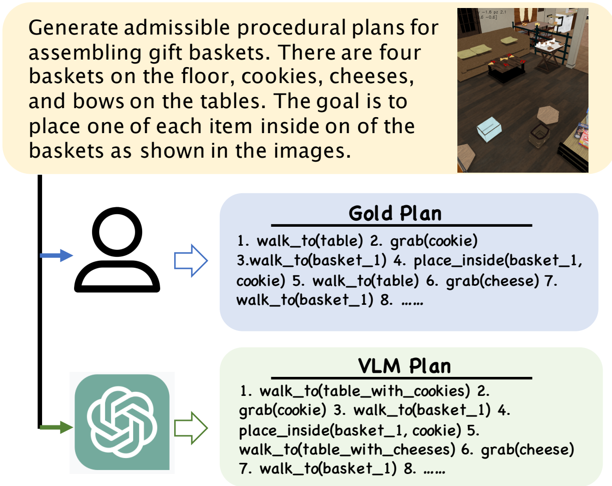


Figure 1: Generating procedural plan for household activities with VLMs via prompting with task description and environment images.

While LLMs can generate plausible action plans over current embodied AI tasks with their powerful reasoning ability on texts, it is unclear of how VLM agents behave on multi-modal embodied AI tasks. Previous works typically evaluate activity completeness in simulator environments (Shridhar et al., 2019; Puig et al., 2020; Srivastava et al., 2021), lacking of evaluation on procedural plans over multi-modal task descriptions.

In this paper, we address the rare evaluation on the procedural planning ability of VLMs for human activities by proposing a new benchmark ActPlan-1K. The multi-modal benchmark consists of human activity descriptions in natural language and environment images, mimicking the real-world household scenarios. An example is presented in Figure 1. By taking in the multi-modal inputs, VLMs generate admissible procedural plan composed by actions and objects.

To collect the multi-modal household activity dataset, we adopt iGibson2 (Li et al., 2021) to generate environment scenes where the activities happen. iGibson2 is a household activity simulator providing customized activity definition and cor-

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 14953–14965
November 12-16, 2024 ©2024 Association for Computational Linguistics

	ActPlan-1K	EgoPlan-Bench	COPLAN	Behavior-100	AI2Thor Room Rearr	VirtualHome	ALFRED	Watch-And-Help
overall procedural plan	✓	✗	✓	✗	✗	✗	✗	✗
activity reflects human behavior	✓	✓	✓	✓	✗	✗	✗	✗
counterfactual	✓	✗	✓	✗	✗	✗	✗	✗
multi-modality	✓	✓	✗	✗	✗	✗	✗	✗
activities	153	3,269	-	100	1	549	7	5
scenes/rooms	15/100	419	-	15/100	-/120	6/24	-/120	7/29
object categories	391	558	-	391	118	308	84	117
source	iGibson2,ChatGPT	GPT-4	GPT-3	iGibson2	AI2THOR	Virtualhome	AI2THOR	Virtualhome

Table 1: Comparison of ActPlan-1K to previous benchmarks related to household activities. Multi-modality means if the dataset provides both textual and visual task descriptions.

responding scene sampling. For activities defined in Behavior-100 (Srivastava et al., 2021), we translate the activity definition in BDDL symbolic form to natural language task description. Environment images are sampled in the simulator by loading the BDDL activity definition. Both the natural language task description and environment images are collected as the multi-modal instance.

To further evaluate the counterfactual planning ability under constrained situations in real-world applications (Brahman et al., 2023), we design counterfactual activities based on normal activities defined in Behavior-100. Specifically, we request ChatGPT to generate situated conditions on the normal activities. Human annotators select the the situated conditions which can be defined and sampled in the household simulator. Counterfactual activity instances are then collected in the same way as normal activities.

Our benchmark covers 153 activities, 1,187 activity instances in total. Each activity instance contains 2˜5 images. The gold procedure length is 23.95 and 31.93 for normal and situated activities respectively. We evaluate typical VLMs (i.e., Claude (Anthropic, 2024), Gemini-Pro (Team et al., 2023), and GPT-4V (Achiam et al., 2023)) that are capable of conducting the task. Evaluation results on human score (correctness and commonsense satisfaction) show that current VLMs are still struggling to generate procedural plans of high-quality. To enhance future study with the proposed benchmark, we also propose automatic evaluation metrics including least common subsequence (LCS) score and finetuned BLEURT (Sellam et al., 2020) accuracy score. The dataset is available[1].

2 Related Work

2.1 Household Embodied AI benchmarks

A substantial body of work constructs embodied environments or simulators, such as AI2THOR (Kolve et al., 2017), Gibson (Xia et al., 2018), iGibson (Shen et al., 2021; Li et al., 2021), VirtualHome (Puig et al., 2018), AIHabitat (Savva et al., 2019). Based on them, VirtualHome (Puig et al., 2018) crowdsources commonsense information about typical activities in people's homes and collect programs that formalize the activity in natural language. Watch-And-Help (Puig et al., 2020) focuses on social perception and human-AI collaboration. ALFREAD (Shridhar et al., 2019) consists of expert demonstrations in interactive visual environments for 25k natural language directives from AI2THOR2.0 (Kolve et al., 2017). Behavior-100 (Srivastava et al., 2021) defines the various household activities to enrich physical and semantic properties with iGibson2.0 (Li et al., 2021).

Recent researches leverage the emergent reasoning capabilities of LLM and collect datasets through few-shot prompting. COPLAN (Brahman et al., 2023) iteratively expands the planning dataset in natural language by prompting GPT-3 with seed examples. EgoPlan-Bench (Chen et al., 2023) is an instruction-tuning dataset from videos with human-object interactions, evaluating the VLMs' ability in predicting next-step actions in question-answering form. Our ActPlan-1k collects multimodal inputs based on both the simulator iGibson2 and ChatGPT. Comparison of our work to previous works is listed in Table 1.

2.2 Planning with LLMs and VLMs

The use of large language models or vision language models in embodied AI tasks have become an increasingly popular research topic, by making procedural plans for high-level tasks with actionable and commonsense knowledge (Ahn et al., 2022; Raman et al., 2022; Li et al., 2022; Huang et al., 2022; Du et al., 2023; Song et al., 2023).

LM-Nav (Shah et al., 2022) adopts LLMs to parse free-form instructions into landmarks for a vision-language model to infer a joint probability over the landmarks and images. GPT-3 and Codex

[1]https://github.com/HKUST-KnowComp/ActPlan-1K

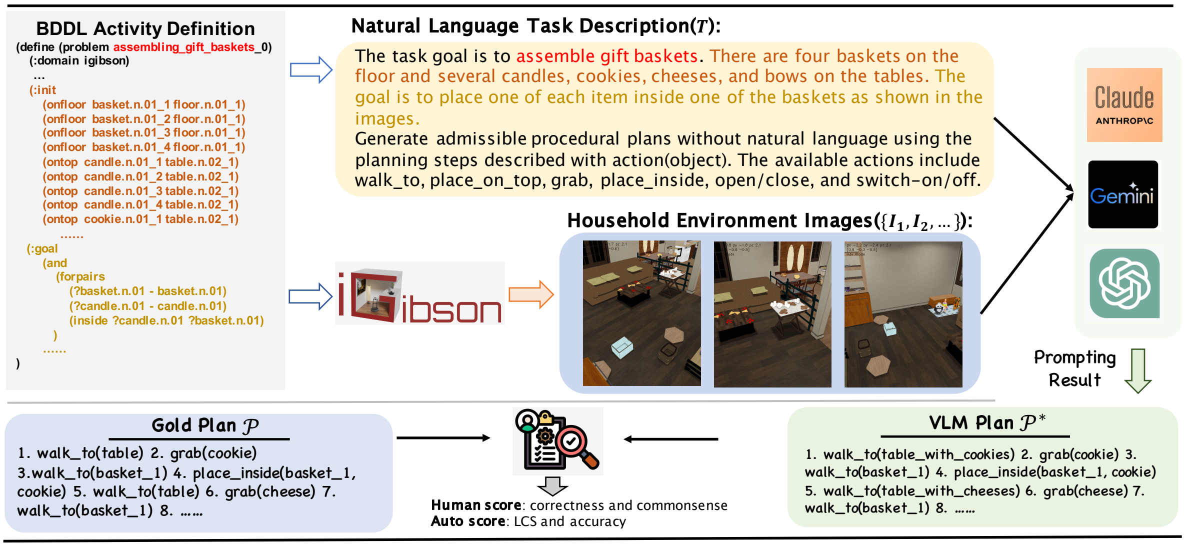


Figure 2: Overview of ActPlan-1K dataset collection and evaluation. The BDDL activity definition is converted into natural language description. Environment images are sampled in the simulator after loading the BDDL definition. After prompting VLMs, VLM plan is evaluated by both human score and automatic metrics compared to gold plan.

can produce plausible action plans with example of task description and its associated action sequences (Huang et al., 2022). LLM-Planner (Song et al., 2023) design strategies for LLMs to generate high-level plans dynamically to accomplish final goal in interactive manner with few-shot prompting. The multi-modal LLM PaLM-E (Driess et al., 2023) takes in images, state estimates, or other sensor modalities to generate executable sequence plans. In Text2Motion (Lin et al., 2023), a library of learned skills are adopted to interface with an LLM for generating feasible plans prior to execution. AutoRT (Ahn et al., 2024) utilizes both the vision-language models (VLMs) and large language models (LLMs) to generate and determine executable high-level objectives. While LLMs have been exploited in the embodied AI tasks mostly with textual description, we target at benchmarking the planning ability of VLMs with both textual and visual descriptions.

3 ActPlan-1K Benchmark

ActPlan-1K contains multi-modal planning instances both of normal activities and counterfactual activities, with same instance form in Section 3.1. The instance collection process of them is also same with BDDL definition and illustrated in Section 3.2. Details of constructing counterfactual activities are introduced in Section 3.3. The benchmark statistics is presented in Section 3.4.

3.1 Problem Definition

Given household environment $\mathcal{E}$, there are manipulable objects set $\mathcal{O}$. For each household activity $\mathcal{T}$, an VLM agent $\mathcal{A}$ takes in the task description T and environment images $\{I_1, I_2, ...\}$ as input, generates procedural plan $\mathcal{P}^*$ that can accomplish the task. The household environments have multiple interior spaces therefore there are multiple images to ensure that necessary spatial information is provided. The overall framework is presented in Figure 2.

The gold procedural plan $\mathcal{P}$ consists of multi-step action sequences which are conducted on the objects. To facilitate the evaluation of high-level plans in real-world scenarios, we design symbolic action plans followed the design in VirtualHome (Puig et al., 2018). Specifically, we use 8 main actions, namely, *walk_to*, *place_on_top*, *grab*, *place_inside*, *open/close*, *switch-on/off*. Additional actions such as *wipe* and *pour* are permitted if they satisfy real-life scenario commonsense.

3.2 ActPlan-1K Instance Collection

To construct multi-modal input for describing the household activity, we use an household activity simulator iGibson2 (Li et al., 2021). The iGibson2 simulator[2] provides visual components for image collection, including renderer and viewer. In the simulator, 15 household environments with 342 types of objects are provided for activity scene

[2]https://svl.stanford.edu/igibson/

initialization.

iGibson2 takes in BDDL definitions in BDDL and loads household environment according to the BDDL activity definition. BDDL provides predicate-logic descriptions for object-centric activity design. New activities with customized initialization of objects and scene can be defined with BDDL language[3] and then loaded in the simulator[4].

We use seed BDDL activities from Behavior-100 (Srivastava et al., 2021) and load them in selected household environment $\mathcal{E}$ in iGibson2. In viewer mode, we record the touring video around the indoor scenes. Images covering the main information in BDDL activity definition are selected as visual information $\{I_1, I_2, ...\}$. Generally, 2 to 5 images are selected to cover the main contents. As for natural language description, we transform the BDDL activity description into natural language form T by prompting ChatGPT with few-shot examples. As for gold plan $\mathcal{P}$, we ask human annotators to write in action sequences over object subset from $\mathcal{O}$. The actions are mostly from pre-defined 8 actions and the objects are mostly from the activity definition. Actions can also be conducted on the objects in the images which are not in the definition if necessary. An example of multi-modal input and output is presented in Figure 2.

3.3 Counterfactual Activity

Counterfactual planning necessitates agents to engage in reasoning about alternative constrained scenarios and formulate corresponding plans (Brahman et al., 2023). The ability to engage in counterfactual planning holds significant importance for VLM agents, as constrained situations frequently arise in real-world applications. Enhancing this capability enables agents to effectively navigate and respond to complex and dynamic environments.

We define multi-modal counterfactual activities by adjusting the task descriptions and gold plans from normal activities in Behavior-100. Behavior-100 activity definitions are common ones and require normal procedures to accomplish the goal. To acquire counterfactual activities based on normal activities in Behavior-100, there are three steps as follows:

- Step 1: Request ChatGPT for common procedures to accomplish the given activities, and request for what might happen during the procedure of common activities;

- Step 2: Human selection of the circumstances that are commonsense plausible, and suitable for BDDL definition;

- Step 3: Load the new counterfactual BDDL activity definition in iGibson2 simulator and collect the visual inputs.

[3]https://behavior.stanford.edu/activity-annotation

[4]https://stanfordvl.github.io/iGibson/sampling.html

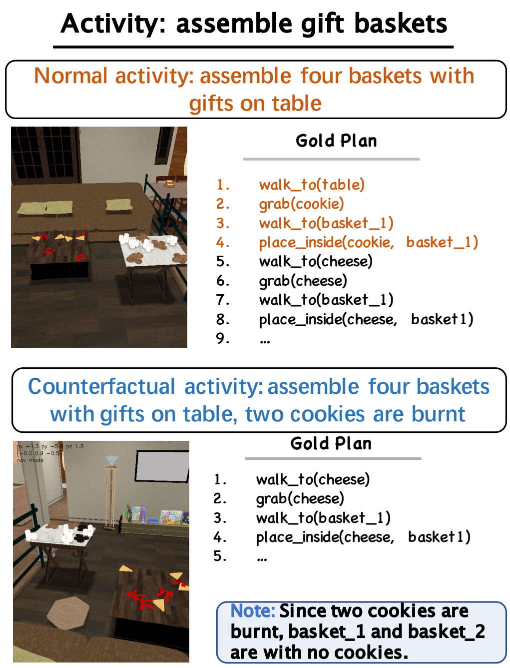


Figure 3: Example of normal activity and counterfactual activity. The gold plans are different because of the counterfactual situation.

The detailed pipeline of construction is presented in Appendix A. An prompting example for Step 1 is presented in Figure 10.

To ensure the counterfactual activity BDDL definition can be loaded for image collection, human annotation is required in Step 2. The selected counterfactual circumstances should also be commonsense plausible, which means reasonable and acceptable based on human understanding and knowledge of the world.

After loading the counterfactual definition in the simulator, environment images are collected in Step 3. An detailed example of normal activity and counterfactual activity is in Figure 3. In the example, the task inputs and gold plans are different between the two activities. Gold procedural plans for "assemble gift baskets" will distribute each cookie into each basket. However when two cookies are burnt, they will not be suitable as gifts so they should not

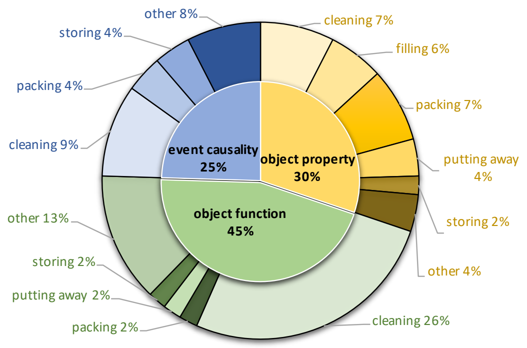


Figure 4: Distribution of counterfactual activities.

be assembled in the gift baskets. Therefore, this situation will lead to a different procedural plan.

3.4 Activity Statistics

The counterfactual activities can be broadly classified into three categories:

Object Property. Under this situation, the plan need to be adjusted in order to consider the special properties of object to adjust the plans. An example is shown in Figure 3. In the example, burnt cookies are not suitable as gifts, so two baskets (basket_1 and basket_2) will not be assembled cookies. The property of cookie (burnt and unburnt) is considered to adjust the plan.

Object Function. The functionalists of different objects are considered and designed to solve a special problem. An example is shown in Appendix B Figure 8.

Event Causality. To accomplish the activity goal, additional steps are required to handle unexpected events in the middle of the normal procedures. An example is shown in Appendix B Figure 9.

In detail, we group activities in each category them by activity types: *cleaning*, *filling*, *packing*, *putting away*, *storing*, and *other*. The distribution is presented in Figure 4.

For both normal activities and counterfactual activities, each activity is loaded in three randomly selected household environments in the simulator. To diversify the visual content, each activity in each household environment is sampled up to three times in each household environment. The overall instance distribution is listed in Table 2. In general, the counterfactual activities have longer plan sequences than normal activities since additional steps are required to handle counterfactual situations.

Type	Num	Instances	Avg len
Normal	100	825	23.95
Counterfactual	53	362	31.93

Table 2: Distribution of activity instances. Avg len represents the average length of plan sequences.

4 Evaluation Metrics

Evaluation metrics over VLM generated procedural plan $\mathcal{P}^*$ and gold plan $\mathcal{P}$ have two types: human evaluation and automatic evaluation.

4.1 Human Evaluation

The activity plans in ActPlan-1K are defined to have complex and long procedural steps, and can not be directly evaluated in the simulator. Therefore, we ask human annotators to annotate the correctness and commonsense satisfaction of $\mathcal{P}^*$.

Correctness. The correctness is 1 if the entire plan can achieve the final activity goal. The plan steps does not strictly follows the step orders of the gold plan. Otherwise the correctness is 0.

Commonsense Satisfaction. The commonsense satisfaction is 1 if every procedural step is commonsense plausible. The final goal is not necessarily achieved. If one of the procedural step does not satisfy commonsense constraints, the commonsense satisfaction is 0.

4.2 Automatic Evaluation

Automatic evaluation is conducted based on the N-gram based metric: longest common subsequence (LCS), and BLEURT-based accuracy score.

4.2.1 LCS

Similar to (Puig et al., 2018), we calculate the LCS between two plans $\mathcal{P}^*$ and $\mathcal{P}$. The normalized LCS score is also calculated by normalizing LCS length with the maximum length of the two plans. To judge if two steps are semantically equivalent, we use a sentence transformer *all-MiniLM-L6-v2*[5] to compare the similarity of two plan steps. Two plan steps are semantic similarly if the score is greater than 0.8.

[5] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Model	Normal activity		Counterfactual activity		Total	
	Correctness	Commonsense	Correctness	Commonsense	Correctness	Commonsense
Claude-3-haiku	8.0	13.8	3.9	9.4	6.7	12.4
Claude-3-sonnet	22.2	34.9	10.2	29.9	18.4	25.9
Gimini-Pro-1.5	32.0	44.0	22.8	41.7	29.1	43.3
GPT-4V	29.1	39.6	9.4	25.2	22.3	39.6

Table 3: Correctness score (%) and commonsense satisfaction score (%) of VLMs on ActPlan-1K.

4.2.2 Accuracy

By converting the plan sequences into sentence pairs, we design a classification task to automatically evaluate the quality of generated plan. We follow the task-specific finetuning schema in BLEURT (Sellam et al., 2020). The BLEURT is a learned evaluation metric based on BERT (Kenton and Toutanova, 2019) that can model human judgements with small task-specific data on text generation tasks. We finetune BLEURT models on ActPlan-1K plans to build automatic evaluation on generated plans.

Given sentence pair $\mathbf{x}$ and $\tilde{\mathbf{x}}$, $\mathbf{x}$ is a sentence for gold plan $\mathcal{P}$ and $\tilde{\mathbf{x}}$ is a sentence for generated plan $\mathcal{P}^*$. $\{(\mathbf{x}, \tilde{\mathbf{x}}, y_i)\}_{n=1}^N$ is a training dataset of size N, where y_i is a classification label that indicates if $\tilde{\mathbf{x}}$ is semantically equivalent to $\mathbf{x}$.

$\mathbf{x}$ and $\tilde{\mathbf{x}}$ are as input to a BLEURT transformer, which has been finetuned on large amount of synthetic data to model human rating. A sequence of contextualized vectors are returned as:

$$\mathbf{v}_{[CLS]}, \mathbf{v}_1, ... = \text{BLEURT}([\mathbf{x}, \tilde{\mathbf{x}}]), \quad (1)$$

where $[\mathbf{x}, \tilde{\mathbf{x}}]$ stands for the concatenation of $\mathbf{x}$ and $\tilde{\mathbf{x}}$ concatenated with a $[SEP]$ token. $\mathbf{v}_{[CLS]}$ is the representation for the special $[CLS]$ token. With a linear layer on top of the $[CLS]$ vector, the classification label is predicted as:

$$\tilde{y} = \mathbf{W}\mathbf{v}_{[CLS]} + \mathbf{b}, \quad (2)$$

With a few thousand examples as supervised data, a supervised classification loss:

$$loss = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1-y_i)\log(1-p_i)], \quad (3)$$

where $y_i \in [0, 1]$ and p_i is the softmax probability.

To finetune the model, we construct synthetic training sequences from gold plans and generated plans from VLMs (i.e., Gemini-Pro-1.5 (Team et al., 2023) and GPT-4V (Achiam et al., 2023)). Specifically, the plan sequences are split into train/val/test with 60/20/20 normal activities, and the corresponding counterfactual activities. For each activity instance, more than two gold sequences are labeled. For generated plans, the correctness scores are human labeled.

The gold plans and correct plans are mixed to form positive sentence pairs. To form negative sentence pairs with the gold plan, wrong plan from VLMs or randomly shuffled gold plan with same activity is selected. The synthetic data are used to finetune `BLEURT`(based on BERT-large) and `BLEURT-base`(based on BERT-base) with a classification layer on top of them. Learning rate is 5e-6 for `BLEURT` and 1e-6 for `BLEURT-base`. Batch size is set as 12. Each model is finetuned for 70 epochs. The model achieves best score on val set is selected as the best model.

5 Experiment Result

5.1 Experiment Setup

By adopting VLMs as agent $\mathcal{A}$, the input are task description in natural language form T and sampled environment images $\{I_1, I_2, ...\}$, forming as multi-modal input:

$$\mathcal{P}^* = \text{VLM}([T, I_1, I_2, ...]) \quad (4)$$

The VLMs are Claude-3-haiku and Cluade-3-sonnet[6], Gemini-Pro-1.5 (Team et al., 2023), and GPT-4V (Achiam et al., 2023). The decoding temperature is 0 for all the models in the evaluation. Please see Appendix C for additional details.

5.2 Correctness and Commonsense Satisfaction

Human evaluation results on VLM generated plans are shown in Table 3. On both the correctness and commonsense metrics, the scores are low and this shows that procedural planning is still challenging for VLMs.

[6]https://www.anthropic.com/news/claude-3-family

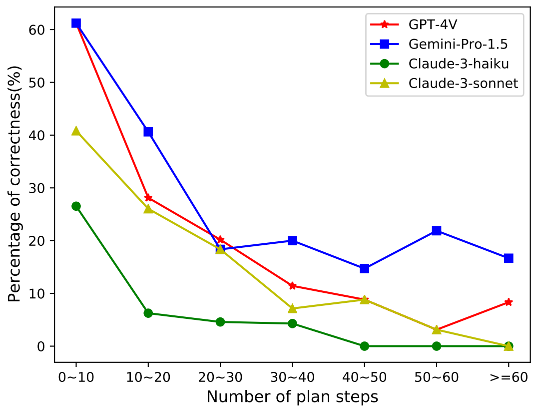


Figure 5: Correctness (%) with plan sequence length.

Counterfactual activities are harder to accomplish. VLMs achieves much higher scores on normal activities than counterfactual activities. It demonstrates that counterfactual reasoning ability over embodied household activities still requires enhancement. Comparing correctness and commonsense satisfaction, VLMs are more capable to provide the commonsense knowledge required for both type of activities, while not as good as designing accurate procedural steps.

Gemini-Pro achieves better performance than other VLMs. Comparing the VLMs, we can find that Gemini-Pro achieves better performances on all the human evaluation metrics. Compared to GPT-4V, the strength mainly lies on the ability to accomplish counterfactual activities. Combined with Figure 5, we can find that the strength also lies on plans with longer sequence plans.

Correctness is degrading with sequence length increases. As presented in Figure 5. The correctness of sequence length drops greatly with the sequence length increases. Long-sequence plan is hard as it needs to take consideration of more objects and events. The phenomenon is same for all the four VLMs.

5.3 Automatic Evaluation

5.3.1 LCS

Evaluation results of the LCS and Norm LCS scores are shown in Table 4. Results show Gemini-Pro-1.5 achieves best performances on LCS and Norm.LCS among the VLMs. About half of the sequences from GPT-4V match the correct ones. Claude-3-haiku and Claude-3-sonnet achieves much lower scores than GPT-4V and Gemini-Pro-1.5, which is consistent with the correctness and commonsense score in Table 3. LCS

Model	LCS	Norm. LCS
Claude-3-haiku	8.62	0.38
Claude-3-sonnet	11.87	0.48
Gemini-Pro-1.5	18.00	0.67
GPT-4V	14.23	0.55

Table 4: LCS and Norm LCS Score (%) of VLMs on ActPlan-1K.

Model	VLM	MCC	Val(Acc)	Test(Acc)
BLEURT-base	Gemini-Pro-1.5	0.258	65.4	65.4
	GPT-4V	0.258	79.0	70.4
BLEURT	Gemini-Pro-1.5	0.321	75.3	80.2
	GPT-4V	0.367	81.5	76.5

Table 5: Accuracy (%) and MCC of VLMs on ActPlan-1K val and test splits.

and Norm LCS scores provide an easy and approximate evaluation metric for the generated plans.

5.4 Accuracy

The results of accuracy on val and test splits from Gemini-Pro and GPT-4V for finetuning BLEURT models are presented in Table 5 separately. On both BLEURT-base and BLEURT, the accuracy scores are all above 60%, achieving as high as 81.5%. This shows that automatic classification task has promising results in modeling human evaluation on the correctness metric.

Since the correctness labels are significantly imbalanced as many are 0, we further calculate the Matthews Correlation Coefficient(MCC) between the original correctness labels and predicted classification results on val split. The MCC score measures the correlation of human label and automatic predicted labels. Results in Table 4 show that on BLEURT, the MCC achieves 0.321 and 0.367 on Gemini-Pro and GPT-4V, close to moderate standard. To further improve the correlation, increasing the training data amount would help make the evaluation results more close to human ratings.

BLEURT achieves better performance than BLEURT-base due to its larger model size. The performance gain is obvious on MCC and accuracy score for both of the VLMs. Results with BLEURT would make the automatic evaluation on accuracy more robust to use.

5.5 Ablation Study

To see if the images are necessary or provide key information for procedural planning, we prompt

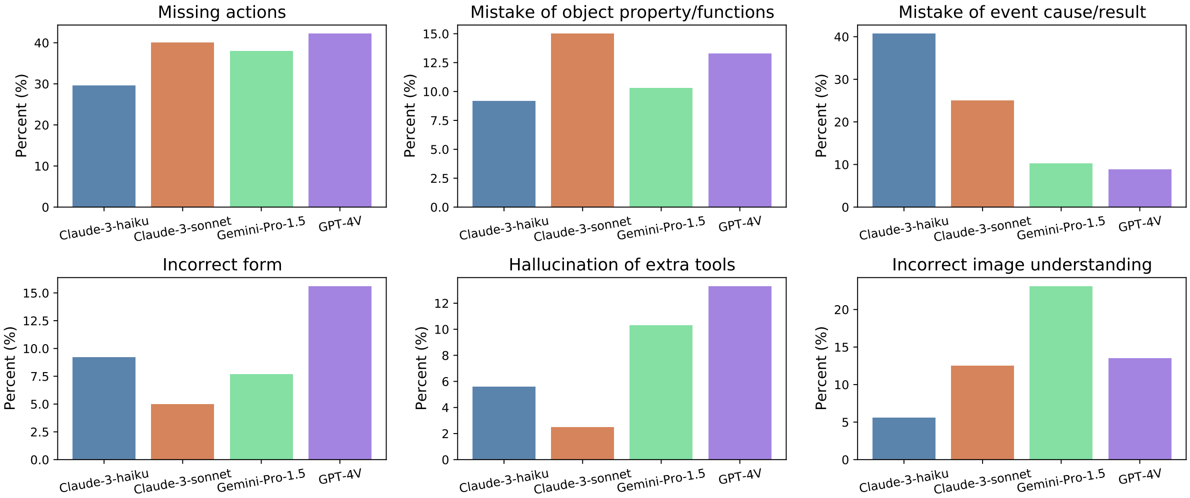


Figure 6: Error distribution (%) of VLMs on six types of errors.

Model(w or w/o image)	Correctnetss	Commonsense	Avg.len
ChatGPT(w/o)	13.3	25.0	18.76
Gemini-Pro-1.5(w/o)	26.7	35.0	52.63
Gemini-Pro-1.5(w)	36.7	43.3	26.93

Table 6: Ablation study results of no images as prompting input to ChatGPT and Gemini-Pro-1.5.

ChatGPT and Gemini-pro with text description only. Results of 40 normal activities with corresponding 20 counterfactual activities are presented in Table 6. Further detailed results in normal and counterfactual ones are in Appendix E. Results show that without images, the correctness or commonsense quality drops while with longer steps. The reasons are:

ActPlan-1K is challenging for LLM with no visual information considered. ChatGPT generates much shorter responses than Gemini-Pro-1.5 on average, and the correctness and commonsense scores are much lower than Gemini-Pro-1.5.

Inaccurate understanding of objects and events. The textual input is lacking of background information, leading to inaccurate plan generation. For example, there is usually sink in bathroom and kitchen which can be used as water source and wash items. If not specifically pointed out, Gemini-pro would generate plans without these utilities and lead to plan failure.

Hallucination leads to longer steps. Since there is lack of visual input of the spatial information, Gemini-pro generates repetitive steps (i.e., *walk_to(kitchen), walk_to(table)* before grab items in kitchen), or hallucinates on the specific states of objects that need to be handled.

5.6 Error Analysis

We conduct error analysis by sampling 40 activity plans (including both normal and counterfactual activities) from Claude-3-haiku, Claude-3-sonnet, Gemini-Pro-1.5 and GPT-4V. The results are presented in Figure 6. Details of example instances are presented in Appendix D.

In both normal and counterfactual activities, the error analysis are broadly categorized as:

- **Missing actions**. Between consecutive subsequences, there are lack of actions to complete. For example, after grabbing food in refrigerator, there is lack of "walk_to(sink)" before placing the food into sink.

- **Mistake of object property/function**. Mistakes caused by misunderstanding the properties or functions of the objects, which are mostly from situated circumstances, such as failure to recognize that burnt cookies is not suitable as gifts, salt can help ice melting instead of vinegar, etc.

- **Mistake of event cause/effect**. Mistakes caused by misunderstanding the normal procedures required for an activity. For example, by adopting soap and rag to clean shoes, soap and rag should be soaked before cleaning, not directly to do cleaning in dry state.

- **Incorrect form**. Mistakes caused by generating sub-sequences not in the required "action(object)" form.

- **Hallucination of extra tools**. Error in generating sub-sequences by utilizing extra tools which are not shown in the images or task descriptions.
- **Incorrect image understanding**. Error in understanding accurate contents in the images, such as the distance between refrigerator, number of clean plates in the images.

From Figure 6, VLMs have different error distributions on the various error types. Missing actions is the mostly occurred error and mistake of object property/functions is a common occurred error for all VLMs.

6 Conclusion

We present ActPlan-1K, a multi-modal counterfactual household planning dataset, to evaluate both the procedural planning and counterfactual planning ability of vision language models. Our dataset constructed based on ChatGPT and household simulators consists of 1.2k instances covering normal and counterfactual activities. Evaluation on typical VLMs shows that ActPlan-1K is still challenging for the models.

Limitations

The images collected from iGibson2 simulator are of low-resolution which may have influence on image understanding of VLMs. The limited types of object states and scene sampling constraints in the simulator restrict the definition of counterfactual activities. While VLMs as agents for embodied AI tasks is an important and promising research direction, we believe that future development of household simulators will benefit better construction of evaluation benchmarks of our kind.

Acknowledgements

The authors of this paper were supported by the NSFC Fund (U20B2053) from the NSFC of China, the RIF (R6020-19 and R6021-20) and the GRF (16211520 and 16205322) from RGC of Hong Kong.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Michael Ahn, Debidatta Dwibedi, Chelsea Finn, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Karol Hausman, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, et al. 2024. Autort: Embodied foundation models for large scale orchestration of robotic agents. *arXiv preprint arXiv:2401.12963*.

AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*.

Faeze Brahman, Chandra Bhagavatula, Valentina Pyatkin, Jena D Hwang, Xiang Lorraine Li, Hirona J Arai, Soumya Sanyal, Keisuke Sakaguchi, Xiang Ren, and Yejin Choi. 2023. Plasma: Making small language models better procedural knowledge models for (counterfactual) planning. *arXiv preprint arXiv:2305.19472*.

Yi Chen, Yuying Ge, Yixiao Ge, Mingyu Ding, Bohao Li, Rui Wang, Ruifeng Xu, Ying Shan, and Xihui Liu. 2023. Egoplan-bench: Benchmarking egocentric embodied planning with multimodal large language models. *arXiv preprint arXiv:2312.06722*.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. 2023. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*.

Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. 2023. Guiding pretraining in reinforcement learning with large language models. *arXiv preprint arXiv:2302.06692*.

Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. 2023. Visual language maps for robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK.

Wenlong Huang, P. Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *ArXiv*, abs/2201.07207.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. 2017. Ai2-thor: An interactive 3d environment for visual ai. *arXiv e-prints*, pages arXiv–1712.

Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Vainio, Cem Gokmen, Gokul Dharan, and Tanish Jain. 2021. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks.

Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, et al. 2022. Pre-trained language models for interactive decision-making. *Advances in Neural Information Processing Systems*, 35:31199–31212.

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. 2024. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26689–26699.

Kevin Lin, Christopher Agia, Toki Migimatsu, Marco Pavone, and Jeannette Bohg. 2023. Text2motion: From natural language instructions to feasible plans. *Autonomous Robots*, 47(8):1345–1365.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. 2018. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8494–8502.

Xavier Puig, Tianmin Shu, Shuang Li, Zilin Wang, Yuan-Hong Liao, Joshua B Tenenbaum, Sanja Fidler, and Antonio Torralba. 2020. Watch-and-help: A challenge for social perception and human-ai collaboration. In *International Conference on Learning Representations*.

Shreyas Sundara Raman, Vanya Cohen, Eric Rosen, Ifrah Idrees, David Paulius, and Stefanie Tellex. 2022. Planning with large language models via corrective re-prompting.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. 2019. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892.

Dhruv Shah, Blazej Osinski, Brian Ichter, and Sergey Levine. 2022. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action.

Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D'Arpino, Shyamal Buch, Sanjana Srivastava, Lyne Tchapmi, et al. 2021. igibson 1.0: A simulation environment for interactive tasks in large realistic scenes. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7520–7527. IEEE.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2019. Alfred: A benchmark for interpreting grounded instructions for everyday tasks.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M. Sadler, Wei-Lun Chao, and Yu Su. 2023. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, and C. Karen Liu. 2021. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. 2018. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9068–9079.

Yue Zhang and Parisa Kordjamshidi. 2023. Vln-trans, translator for the vision and language navigation agent. In *The 61st Annual Meeting of the Association for Computational Linguistics (ACL-2023)*.

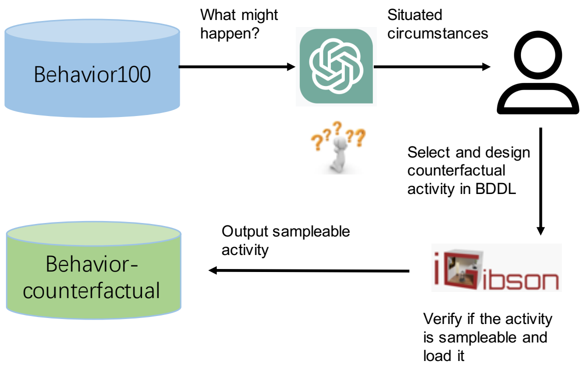


Figure 7: Example of prompting LLM for counterfactual activity circumstances.

A Annotation Pipeline of Counterfactual Activities

The prompting example of acquiring situated circumstances is presented in Figure 10. After getting the result, human check on the results and select circumstances that are suitable for defining in the BDDL form and loaded in the simulator. In this example, for circumstance 3, cookies or cheese might gone bad. The simulator can not model states of cookie as "going bad" but state of "burnt". Therefore, the counterfactual activity is defined with burnt cookies. A new BDDL counterfactual activity definition is constructed. The following natural task description from ChatGPT and image collecting process from iGibson2 are same as normal activities.

As for the counterfactual plan, since burnt cookies are also not safe or suitable to be as gifts, two of the baskets will not be assembled with cookies finally. Therefore, the object state results in a plan change. The full pipeline is presented in Figure 7.

B Examples of Counterfactual Activities

Examples of counterfactual activities designing on object functions and event causality are shown in Figure 8 and 9.

In Figure 8, when water supply is off, the watering source is bottled water on the countertop. In this activity, the function of bottled water is considered to replace original water source from kitchen sink faucet.

In Figure 9, before cleaning the microwave oven as common procedures, the agent should first clear the burnt cookies in the microwave oven. In this activity, the clearing burnt cookie event is supposed to happen before cleaning the microwave oven.

Figure 8: Example of normal activity and counterfactual activity.

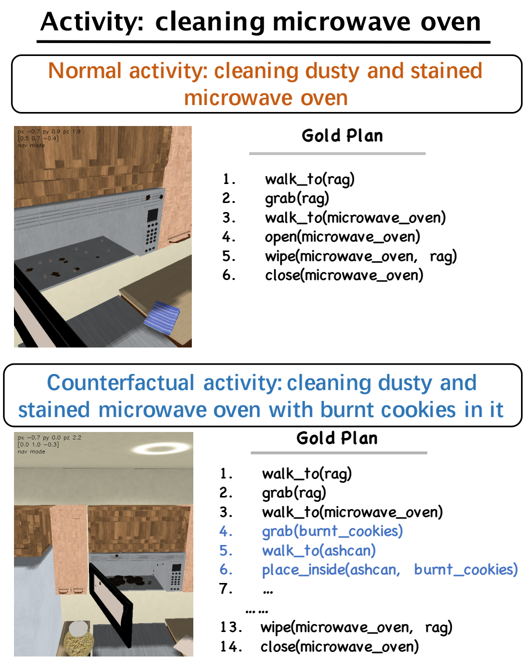


Figure 9: Example of normal activity and situated activity.

C VLMs

Gemini-Pro is from a family of hightly capable multimodal models developed at Google (Team et al., 2023), which also contains Gemini-Ultra and Gemini-Nano.

Claude 3 is a new family of large multimodal

Activity: assembling_gift_baskets

😃 **You:** What are common procedures for assembling four gift baskests with 4 baskets on floor, 4 candles, 4 cookies, 4 cheeses, 4 bows on table? What unexpected situations might happen during the common procedure? List them seperately.

🤖 **AI:** Common Procedures:
1. Lay out the four baskets on the floor.
2. Place a candle, a cookie, and a cheese in each basket.
3. Tie a bow around each basket to complete the gift.
4. Arrange the four baskets on the table for presentation.

Unexpected Situations:
1. Running out of supplies: If you run out of candles, cookies, cheeses, or bows, you may not be able to complete all of the gift baskets. Make sure to check your inventory before starting to assemble the baskets.
2. Breaking a candle, cookie, or cheese: Dropping or mishandling the items can cause them to break or become damaged, which can affect the quality of the gift. Be careful when handling the items and avoid placing them on an unstable surface.
3. Discovering mold or spoilage: If the cookies or cheeses have gone bad, they may not be safe to include in the gift baskets. Always check the quality of the items before using them.
4. Misplacing a basket or bow: If you misplace one of the baskets or bows, it may be difficult to complete all of the gift baskets. Make sure to keep track of all of the items as you assemble the baskets.
5. Not having enough space: If the table is not large enough to accommodate all of the gift baskets, it may be difficult to present them in an attractive manner. Consider using multiple tables or finding a larger space to display the baskets.
6. Not having enough time: Assembling gift baskets can be time-consuming, and you may not have enough time to complete all of them before the deadline. Make sure to plan ahead and allocate enough time for the assembly process.

Figure 10: Example of prompting LLM for counterfactual activities.

Gemini-pro	Normal activity			counterfactual activity		
	Correctness	Commonsense	Avg Len	Correctness	Commonsense	Avg Len
w image	45.0	50.0	23.75	20.0	30.0	33.30
w/o image	32.5	40.0	48.48	15.0	25.0	60.95

Table 7: Details of ablation study of normal and counterfactual activities with Gemini-Pro-1.5.

models from Anthropic AI, including Haiku, Sonnet, and Opus[7].

GPT-4V (Achiam et al., 2023) is a large multimodal model capable of processing image and text inputs and producing text outputs, built by OpenAI.

D Example of Errors

For each of the six error types, one example sampled from the VLM prompting results is presented in Table 8.

E Ablation Study

As presented in Table 7, counterfactual activities have the same trend with normal activity, while the scores of normal activities drop greater than counterfactual activities.

F Evaluation of open-source VLMs on ActPlan-1K

We have conducted experiments on typical VLMs like VILA (Lin et al., 2024). However, current open-source VLMs can not generate structured results as required(e.g., walk_to(table). One example output for activating assembling basket is as follows:

1. Walk to the table. 2. Place a cookie on top of the bowl. 3. Grab a candle from the floor. 4. Place the candle next to the cookie on the table. 5. Open the cheese container and place a cheese inside. 6. Close the cheese container. 7. Walk to the floor. 8. Place a cookie on top of the bowl. 9. Grab a candle from the floor. ... ... 38. Walk to the floor. 39. Place a cookie on top of the bowl. 40. Grab a candle from the.

Therefore, we have not reported the results on open-source VLMs.

G Human Annotators

For the human annotation in gold plans and human scoring of correctness and commonsense satisfaction, we train three PhD students from CSE department with the requirements illustrated in the paper. By checking annotation results from the students, the overall annotation quality is good on the above tasks.

[7]https://www.anthropic.com/news/claude-3-family

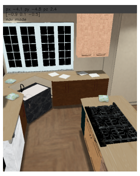
Error Type	Activity	Image Input	VLM Plan	Gold Plan
missing actions	cleaning freezer		... walk_to(refrigerator) open(refrigerator) grab(food_item1) place_inside(sink, food_item1) ...	... walk_to(refrigerator) open(refrigerator) grab(food_item1) walk_to(sink) place_inside(sink, food_item1) ...
mistake of object property/ function	packing food for work		... walk_to(countertop) grab(chives) place_inside(carton, chives) grab(cinnamon) place_inside(carton, cinnamon) ...	... walk_to(countertop) grab(chives) place_inside(carton, chives) grab(chili) place_inside(carton, chili) ...
mistake of event cause/result	cleaning sneakers		... walk_to(sink) switch_on(faucet) place_inside(sink, soap) switch_off(faucet) walk_to(living_room) walk_to(sneaker1) clean(sneaker1, soap, brush) ...	... walk_to(sink) switch_on(faucet) place_inside(sink, soap) switch_off(faucet) walk_to(living_room) walk_to(sneaker1) grab(sneaker_1) walk_to(sink) place_inside(sink, sneaker1) clean(sneaker1) ...
incorrect form	brush lint off clothing		... walk_to(scrub_brush) grab(scrub_brush) grab(sweater1) (brush sweater1) ...	... grab(scrub_brush) grab(sweater1_on_floor) brush(sweater1_on_floor, scrub_brush) ...
hallucination of extra tools	watering houseplants		... walk_to(dining_room_plant_1) grab(watering_can) pour(watering_can, dining_room_plant_1) place(watering_can) walk_to(dining_room_plant_2) ...	... walk_to(kitchen) walk_to(sink) place_inside(sink, dining_room_plant1) place_inside(sink, dining_room_plant2) soak(dining_room_plant1) soak(dining_room_plant2) ...
incorrect image understanding	putting dishes away after cleaning		walk_to(countertop_1) grab(plate_1) walk_to(cabinet) open(cabinet) place_inside(cabinet, plate_1) walk_to(countertop_2) grab(plate_2) walk_to(cabinet) place_inside(cabinet, plate_2) close(cabinet)	... walk_to(kitchen) walk_to(countertop_1) grab(plate_1) grab(plate_2) grab(plate_3) grab(plate_4) walk_to(countertop_2) grab(plate_5) grab(plate_6) grab(plate_7) grab(plate_8) ...

Table 8: Error examples.