End-to-End Embodied Decision Making in the Perception-Cognition-Action Chain

Anonymous ACL submission

Abstract

Recently, agents based on Multimodal Large Language Models (MLLMs) have emerged as a promising area of research. However, effective benchmarks for MLLM-based agents are absent. To this end, in this paper, we present **PCA-Bench**, an end-to-end embodied decision-making benchmark, comprising 1) *PCA-Eval*, a novel automatic evaluation metric inspired by the perception-action loop in cognitive science, assessing the decision-making ability of MLLMs from the perspectives of **P**erception, **C**ognition, and **A**ction. 2) *Embodied-Instruction-Evolution (EIE)*, an automatic framework for synthesizing instruction tuning examples in various multi-modal embodied environments, including autonomous driving, domestic robotics, and open-world gaming. Our experiments on PCA-Bench demonstrate that visual perception and reasoning with world knowledge are two core abilities for an agent to make correct actions. Advanced MLLMs like GPT-4 Vision exhibit superior performance than their open-source counterpart. Additionally, our EIE method substantially enhances open-source MLLMs' performance, at times even surpassing GPT-4 Vision in certain sub-scores. We believe PCA-Bench serves as an effective bridge between MLLMs and their application in embodied agents. The benchmark will be made open-source.

1 Introduction

Recently, we have witnessed a remarkable surge in the development of Large Language Model (LLM) based agents (Xi et al., 2023a), unlocking a plethora of downstream applications in interactive environments such as autonomous driving (Hu et al., 2023; Wayve, 2023), domestic assistance (Huang et al., 2022b), and game playing (Fan et al., 2022; Wang et al., 2023a; Zhu et al., 2023b).

Nevertheless, LLMs encounter a modality gap when tackling embodied tasks, since their training

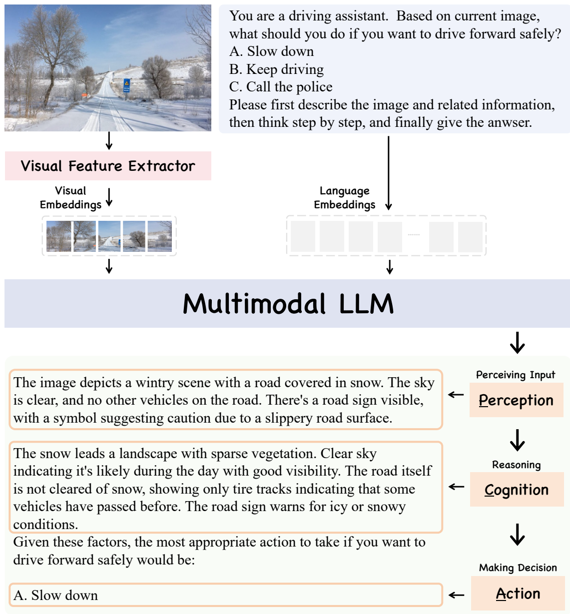


Figure 1: Example of End-to-End embodied decision making with multimodal Large Langauge Models in the Perception-Cognition-Action Chain.

is exclusively based on textual data, in contrast to the multimodal observations that arise from embodied environments. A prevalent approach is transforming these multimodal observations into text via various APIs (Wu et al., 2023; Yang et al., 2023). However, such a non-end-to-end process can be complex and may lose information. Therefore, we are interested in whether current state-of-the-art MLLMs (Zhu et al., 2023a; Dai et al., 2023a; Liu et al., 2023a; Li et al., 2023c; Zhao et al., 2023) are capable of performing various embodied decision-making tasks in an end-to-end manner. However, at present, there are no established benchmarks that connect academically influential domains in embodied decision-making with MLLMs.

To address the challenges of the insufficient benchmarking problem, we introduce **PCA-Bench**, an end-to-end embodied decision-making benchmark for MLLM-based agents. It consists of *PCA-Eval*, a novel automatic evaluation metric, and *Embodied-Instruction-Evolution (EIE)*, an automatic framework for synthesizing instruction tuning examples in various multi-modal embodied environments, including autonomous driving, domestic robotics, and open-world gaming.

PCA-Eval is designed to evaluate the embodied decision-making capabilities of agents from three key perspectives: Perception, Cognition, and Action. This is inspired by the Perception-Action loop (Fuster, 2004) in Cognitive Science, a fundamental concept that describes how organisms process sensory information (Perception) to interact with their environment through actions, offering a comprehensive framework for assessment. Figure 1 shows how MLLMs are prompted to make decisions in the PCA chain. Adopting this approach offers two major advantages: **(1)** It enables a more comprehensive evaluation of the decision-making process, with each decision step being assessed in terms of perception, cognition, and action. **(2)** The evaluation can be conducted outside complex simulation environments, simplifying the process of evaluating different agents.

From a data-centric perspective, using LLM to synthesize training examples is an increasingly popular method for enhancing the capabilities of LLM itself without the need for additional human input. We aim to expand this approach to enhance the embodied decision-making skills of MLLMs. Unlike conventional text-based instruction generation methods like Self-Instruct (Wang et al., 2023d), generating instructions for embodied environments poses distinct challenges. It demands not just the creation of textual instructions but also the generation of corresponding accurate observations. To overcome this, we propose **Embodied Instruction Evolution (EIE)**, which integrates external environments with LLMs, thereby extending the LLM's data synthesizing ability to various embodied environments.

We conducted extensive experiments and analysis on PCA-Bench. Our findings are summarized as:

1. Visual perception and reasoning with world knowledge are two core abilities for an agent to make correct decisions. GPT4-Vision shows strong zero-shot cross-modal reasoning ability for embodied decision-making tasks, significantly surpassing open-source counterparts in all sub-scores.

2. EIE could significantly enhance the performance of open-source MLLMs such as Llava-1.5 and Qwen-VL-Chat (surpassing GPT-4V at some scores), validating the effectiveness of the method.

3. GPT4-Vision could surpass Tool-Using LLM agents in a PCA-Bench subset with diverse API return annotations. We analyze each method's strengths and weaknesses and suggest future directions, such as improving cross-modal Chain-of-Thought reasoning ability and aligning agents' decisions with human values.

Figure 2: Instances of PCA-Bench in 3 domains.

2 PCA-Bench

2.1 Problem Definition

Embodied Decision-making problems are commonly formalized with a partially observable Markov decision process (POMDP).

$$\mathcal{M} := \langle S, A, T, R, \Omega, O, \gamma \rangle \tag{1}$$

For an End-to-End embodied decision making model $\mathcal{F}$, we care about given the multi-modal observation $o \in O$, the goal description g, a subset of candidates actions $A_C \subseteq A$, whether the agent could make correct action $a \in A_C$ and give proper reasoning process r.

$$\mathcal{F}(g, o, A_C) = (a, r) \tag{2}$$

As shown in Figure 2, each instance in the benchmark is a 6-element tuple: <*image*, *question*, *action candidates*, *answer*, *reason*, *key concept*>. The image is collected from various embodied environments, like transportation scenes, housekeeper environments, and Minecraft. Questions, action candidates, and answers are derived from real tasks within the corresponding environment. The reasoning explains why the answer is the best choice for the current image, while the key concept highlights the most question-related aspect in the image.

Unlike traditional visual question-answering datasets that emphasize visual perception (e.g., VQA (Goyal et al., 2017)) or visual reasoning (e.g., NLVR (Suhr et al., 2017)),the most distinctive characteristic of PCA-Bench is its grounding in embodied actions. Compared to embodied simulation environments like ALFRED (Shridhar et al., 2020) and Minedojo (Fan et al., 2022), PCA-Bench proves to be more effective in evaluating various LLM-based agents. This is primarily due to the provision of high-level actions that can be readily implemented or programmed using the low-level actions in the corresponding domains. The high-level actions are more comprehensible for LLMs than the direct low-level actions like robotic movements in the simulation environments because (1) the high-level actions are in the form of natural languages, making it easier for LLMs to understand the meaning and connect with world knowledge. (2) LLMs are not grounded with low-level actions during the pretraining or finetuning stage, making it hard for LLMs to understand the consequences of executing an action.

To answer a question in PCA-Bench, the agent must possess the following abilities: (1) Perception: accurately identify the concept related to the question within the image; (2) Cognition: engage in reasoning based on image perception and worldly knowledge; (3) Action: comprehend the potential actions, selecting the one that best aligns with the outcome of the reasoning process. A deficiency in any of these abilities would possibly result in an incorrect answer, posing a significant challenge to the more complex capabilities of embodied agents.

2.2 PCA-Eval

For each instance, we prompt the agent to deliver an answer comprising a reasoning process r, and a final action a, represented as $< r, a >$. By comparing the model prediction with the ground truth answer, we can obtain a fine-grained diagnosis of the decision making process as following:

Perception Score (P-Score) measures the model's accuracy in perceiving the observation. It is computed based on whether the agent's reasoning process r includes the key concept of the instance. A score of 1 is assigned if at least one question-related key concept is described by the agent; otherwise, it is 0. For the top example in Figure 2, the agent should output "clear road" or "no car visible" or other semantically equivalent concepts in its description of the image to get the perception score.

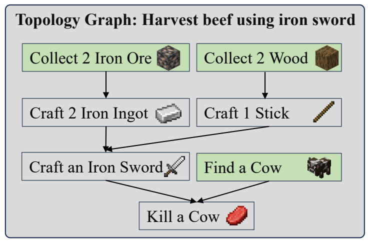


Figure 3: Illustration of task topology graph. Events in green represent the leaf nodes of the graph.

Cognition Score (C-Score) assesses the model's ability to reason, comprehend, and make informed decisions based on the perceived input data and world knowledge. The score is 1 if the reasoning process is correct, otherwise the score is 0. For the instance in Figure 2, the agent should link the "clear road" to the action "keep driving" based on transportation commonsense to get the score.

Action Score (A-Score) measures the model's ability to generate appropriate and effective responses or actions based on the perceived input data and the cognitive understanding of the context. The score is assigned a value of 1 if the agent selects the correct action; otherwise, the score is set to 0.

2.3 Automatic Evaluation

Recent advancements have seen researchers harnessing powerful LLMs for the evaluation of output of language models. Studies have revealed that the outcomes from LLMs could exhibit remarkable alignment with human judgments (Zheng et al., 2023; Wang et al., 2023c,b). In our investigation, we employed GPT-4 to automatically evaluate perception, cognition, and action scores based on the model's outputs. Our findings underscore a significant agreement between GPT-4 annotations and human annotator results. This is substantiated by Pearson correlation coefficients of 0.8, 0.9, and 0.95 for perception, cognition, and action evaluations, respectively. For a detailed description of our evaluation tool, we utilize the template as shown in Table 3 to query GPT-4, aiming to evaluate its responses and assign scores for perception, cognition, and action.

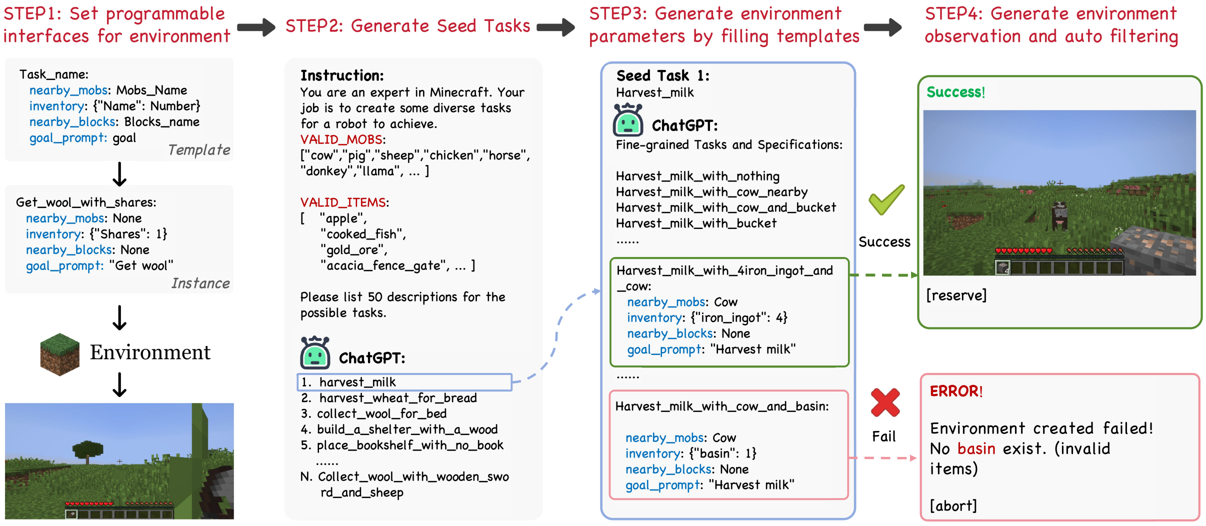


Figure 4: Pipeline of the Embodied Instruction Evolution method.

2.4 Benchmark Dataset Overview and Embodied Instruction Evolution

The PCA-Bench benchmark now includes a training set consisting of 7,510 examples and a test set comprising 813 examples. The entire training set is autonomously generated through the Embodied Instruction Evolution (EIE) method, entirely without human intervention. For the test set, 313 examples are exclusively written by 3 human experts for each domain, while the remaining 500 examples are initially produced using the EIE method and subsequently undergo manual checking and filtering by human experts. We ensured that there are no shared environmental observations between the training and test sets. Moreover, every test case has been verified by at least three authors of this paper. The details of human annotation pipeline could be found in Appendix B. We introduce the three domains encompassed by our dataset as follows:

Autonomous Driving. In the autonomous driving domain, instances are derived from real-world transportation scenes, which requires the agent to have particular abilities such as traffic sign recognition, obstacle detection, and decision-making at intersections. The dataset aims to evaluate an agent's ability to perceive and interpret visual information while making safe and efficient driving decisions. The images are collected from TT100K (Zhu et al., 2016) dataset and annotators are instructed to propose an image-conditioned question that is grounded with real actions of vehicles.

Domestic Robot. The domestic assistance domain features instances from the ALFRED (Shridhar et al., 2020; Kolve et al., 2017) environment, which simulates a housekeeper robot performing tasks within a household setting. These tasks may include object manipulation, navigation, and interaction with various appliances. The environment assesses an agent's ability to understand and execute complex instructions while navigating and interacting with a dynamic environment. Annotators are asked to select one image from the randomly generated scenes in the environment, propose a question related to the items on the scene, and annotate the full information of the instance.

Open-World Game. In the open-world game domain, instances are sourced from the Minecraft environment, where agents are tasked with exploring, crafting, and surviving in a procedurally generated world. This dataset evaluates an agent's ability to reason and plan actions within a complex, open-ended environment, which often requires long-term strategizing and adaptability. Annotators receive predefined tasks from MineDojo (Fan et al., 2022) as a reference during the task generation phase. For each task, we instruct the annotator to sketch a task topology graph, exemplified in Figure 3. The task should be completed in accordance with the topological order of the graph, where the event located in the leaf nodes should be finished first. Each node in the task topology graph can be viewed as a step in the sequential decision. We list the in-domain task distribution in Appendix A.

The annotation of PCA-Eval examples is a laborintensive task. As illustrated in Figure 4, we introduce Embodied Instruction Evolution (EIE), a method for automatically augmenting examples in the PCA-Eval format using Large Language Models, such as ChatGPT. This process involves four key steps: **1) Setup of Programmable Interface:** Establish a programmable interface with a corresponding template, ensuring that observations in the embodied environment can be generated based on specific parameters. **2) Generation of Seed Tasks:** Create initial seed tasks for each environment. These tasks are representative of the general challenges an agent might encounter. We provide ChatGPT with sample tasks and enable it to generate additional seed tasks. **3) Task Specification and Template Filling:** For each seed task, we instruct ChatGPT to break down the task into multiple subtasks, following its event topology graph (as seen in Figure 3). This approach mimics the multi-step decision-making process. After determining the subtask names, we use the LLM to populate the environment parameter templates created in Step 1 for each subtask. **4) Observation Generation and Filtering:** Generate observations for the environment and implement an automatic process to filter out invalid instances. For domains without programmable environments (autonomous driving), step 1 and step 4 are not needed, we collect real traffic images and utilize GPT4-Vision to generate seed task based on the image content.

EIE leverages the capabilities of Large Language Models to reduce manual labor and improve the diversity and scalability of PCA-Bench examples.

3 Experiments

3.1 Tracks

Zero Shot End-to-End. The test set of PCA-Bench serves as an effective tool for comparing the embodied decision-making and cross-modal reasoning capabilities of various Multimodal Language Learning Models (MLLMs). In this evaluation, the same images and prompts are provided to each model under test. Additionally, to address the challenge of perceiving certain non-visual information from images, details such as “items in hand” and “items in inventory”, particularly relevant in domestic and gaming domains, are directly included in the question prompts.

In our analysis, we benchmark the performance of the most recently open-sourced models, including LLaVA1.5 and Qwen-VL-Chat, as well as the API-only GPT4-V model. All models are evaluated using their default inference configurations to ensure a fair and standardized comparison.

Finetuning with EIE. In this track, we extend the capabilities of open-source MLLMs by fine-tuning them with the training set generated through our Embodied Instruction Evolution (EIE) method. After the fine-tuning process, these trained models are subjected to the test set of PCA-Bench. We finetune the LLaVA-7b/13b, MMICL and Qwen-VL-Chat models on the training set for 5 epochs. The training details are in Appendix E.

Zero Shot Modality Conversion. In this track, we introduce and compare a new baseline, termed HOLMES, which utilizes LLM without multi-modal perception capabilities. Instead, HOLMES relies on modality conversion APIs for embodied decision-making processes. Within the HOLMES framework, the LLM must continuously invoke various APIs, retrieving and processing return information about the environment. The HOLMES method is illustrated in Figure 9 from Appendix.

We evaluate two LLMs in this track: ChatGPT-3.5-Turbo and GPT-4-0613, comparing their performances against the advanced GPT-4-Vision. Implementation details of the HOLMES framework and the APIs are provided in Appendix C.

3.2 Evaluation and Metrics

We use our PCA-Eval evaluation tool proposed in Section 2.3 to automatically assess the output of different models through three lenses: perception (P-Score), cognition (C-Score), and action (A-Score).

3.3 Main Results

Zero Shot Results. The results of the zero-shot end-to-end track is shown in Table 1. Among all MLLMs, GPT4-V, outperforms existing open-source models by achieving the highest scores of 0.86, 0.7, and 0.68 in the perception, cognition, and action dimensions respectively. This performance represents a 15% action score improvement over its strongest open-source counterpart, LLaVA1.5-13B. The impressive performance of GPT4-V is primarily attributed to its exceptional ability to perceive visual information across different domains and the world knowledge in the language model, particularly in the challenging game domain.

Impact of Finetuning with EIE. The results of the fine-tuning track are illustrated in Figure 5. Our

Model	Size	Traffic			Domestic			Game			Average		
		P	C	A	P	C	A	P	C	A	P	C	A
MiniGPT4 (Zhu et al., 2023a)†	7B	0.45	0.37	0.48	0.81	0.38	0.38	0.38	0.14	0.27	0.55	0.30	0.38
LLaVA1.5 (Liu et al., 2023a)†	7B	0.44	0.44	0.53	0.92	0.48	0.44	0.8	0.35	0.39	0.72	0.42	0.45
Qwen-VL-Chat (Bai et al., 2023)†	7B	0.53	0.36	0.62	0.77	0.41	0.44	0.39	0.18	0.25	0.56	0.33	0.44
MiniGPT4 (Zhu et al., 2023a)†	13B	0.41	0.37	0.5	0.85	0.35	0.33	0.41	0.22	0.33	0.56	0.31	0.39
InstructBLIP (Dai et al., 2023b)†	13B	0.36	0.41	0.42	0.90	0.44	0.39	0.33	0.25	0.24	0.53	0.37	0.35
MMICL (Zhao et al., 2023)†	13B	0.31	0.49	0.47	0.81	0.3	0.33	0.41	0.18	0.27	0.51	0.32	0.36
SPHINX-v1 (Lin et al., 2023)†	13B	0.46	0.48	0.61	0.95	0.55	0.31	0.71	0.35	0.43	0.71	0.46	0.45
LLaVA1.5 (Liu et al., 2023a)†	13B	0.49	0.56	0.61	0.95	0.62	0.46	0.74	0.45	0.51	0.73	0.54	0.53
GPT-4V (OpenAI, 2023)‡	UNK	**0.73**	**0.72**	**0.74**	**0.96**	**0.66**	**0.62**	**0.88**	**0.72**	**0.69**	**0.86**	**0.7**	**0.68**

Table 1: Zero Shot results on the test set of PCA-Bench. Highest scores in each line are **bold** while second highest scores are underlined. Models with † are fully open-source. Models with ‡ only provide API to access. P, C, and A represent Perception, Cognition, and Action Scores, respectively.

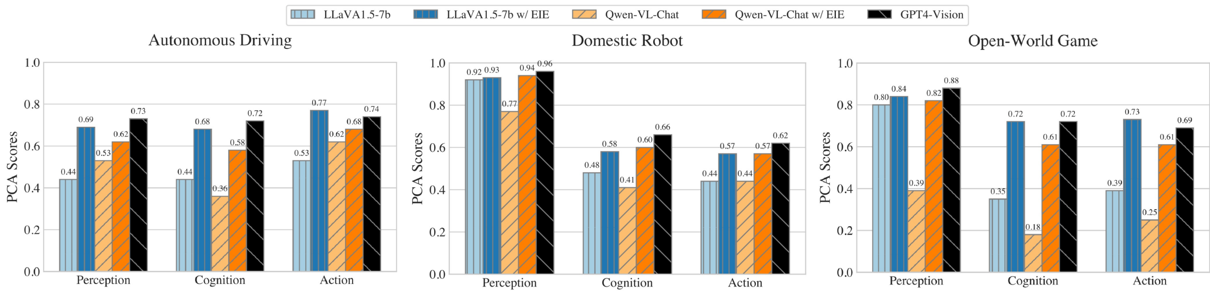


Figure 5: Performance comparsion between models' zero-shot results and models' finetuned results with the data generated by Embodied-Instruct-Evolution (EIE) method. EIE improves the performance on all domains for both LLaVA1.5-7b and Qwen-VL-Chat models. Results of LLavA1.5-13B and MMICL are in Figure 14 from appendix.

EIE method has been found to significantly enhance the general decision-making abilities of various models, encompassing perception, cognition, and action. Notably, it has led to an average increase of 0.24 and 0.19 in action scores for the LLaVA1.5-7b and Qwen-VL-Chat models, respectively. Results for LLaVA1.5-13b and MMICL are illustrated in Figure 14, also showing improved performance when trained with EIE. In some cases, these sub-scores have matched or even surpassed those of the GPT4-V model, thereby demonstrating the effectiveness of the EIE method.

Comparison Between End-to-End and Modality Conversion Method In the zero-shot modality conversion track, we conduct an analysis and comparison of the outputs generated by the End2End method with GPT4-V, as well as the HOLMES method with GPT4 and ChatGPT-3.5. The results are listed in Table 2.

The results show that the HOLMES system based on GPT4 achieves 0.71 Action Score, which is on par with GPT4-V's performance (0.74). This indicates that, overall, the HOLMES system is able to accurately understand the task goal, split the larger goal into multiple smaller steps, and correctly invoke the relevant APIs to accomplish each step. Specifically, the HOLMES system based on GPT4 can recognize the key concepts in a task, and perceive the state and environment of these concepts through the results returned by APIs. Consequently, the system achieves an average Perception Score of 0.88, which even outperforms GPT4-V's 0.84. However, compared End2End methods, HOLMES relies on multi-step reasoning for the final decision, in which reasoning errors tend to accumulate, and thus achieves a lower Cognition Score in both Domestic and Game domains.

On the other hand we also find that the End2End method effectively mitigates information loss during the modality conversion process. As illustrated in Figure 6, an image depicts a road with several nearby cars. GPT4-V is capable of discerning that the street is not crowded, thereby suggesting that the driver can continue driving.

Conversely, GPT4, while aware of the number of cars, lacks information about their spatial relation, leading it to recommend slowing down. This suggests that the End2End method is superior in perceiving certain visual features that are not cap-

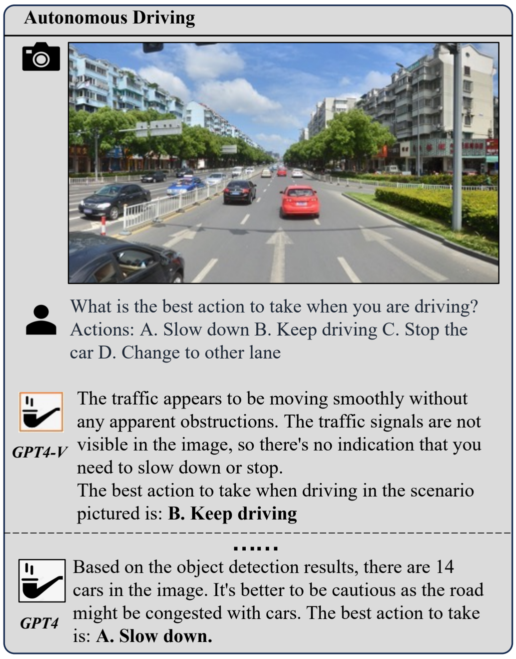


Figure 6: A Comparison between GPT4-V and GPT4-HOLMES

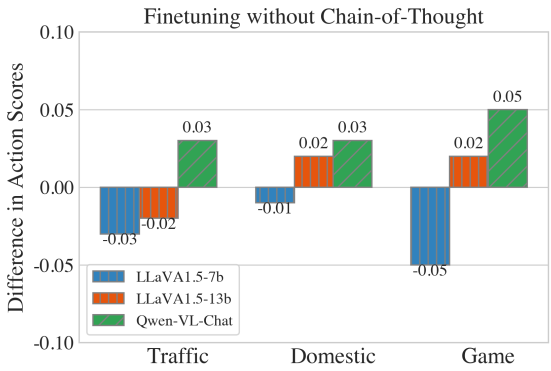


Figure 7: Action scores changes when training without reasoning process for different models. The benefit of CoT finetuning is not consistent among models.

tured by the APIs. Conversely, some specialized APIs, such as traffic sign detection, outperform GPT4-V in tasks like traffic sign detection, as they are specifically trained for this task. This could enable the HOLMES method to gather more accurate information than the End2End model.

4 Discussion

4.1 Does Chain-of-Thought Finetuning Improve Cross-modal Reasoning?

Unlike vanilla finetuning, which solely focuses on delivering direct answers, Chain-of-Thought Finetuning necessitates the model to first articulate its reasoning before presenting the answer. This approach has been demonstrated to be a highly effective instruction tuning paradigm for LLMs (Chung et al., 2022; Kim et al., 2023). We have incorporated this methodology in our previous finetuning experiments.

To further evaluate its impact, we conducted an ablation study where the reasoning process was omitted from the target output during the training of MLLMs. We then assessed the variations in action scores on the test set. As depicted in Figure 7, to our surprise, the figures suggest that Chain-of-Thought finetuning exerts a relatively minor influence when compared to conventional label finetuning. We have noticed that similar phenomena has been identified by Zhang et al. (2023) that standard CoT finetuning does not work for MLLMs in their explorations.

We think there are two potential explanations: 1) Task Variation: Contrary to mathematics datasets like GSM8K, the current task doesn't require multi-step complex reasoning to arrive at the final answer. 2) Modality Discrepancy: The CoT capability, inherent in LLMs, is only moderately adjusted for visual input for current open-source MLLMs. This adaptation process could potentially impair the reasoning ability. We defer to future research how to effectively harness the CoT capabilities of LLMs to enhance embodied decision-making processes.

4.2 Alignment between Agent Decisions and Human Values

We have observed instances where the decisions made by the agent contradict human values. For instance, consider the scenario depicted in Figure 10. The image illustrates a crosswalk devoid of pedestrians. The appropriate response in this situation would be to slow down, as caution is paramount when approaching a crosswalk, regardless of the presence or absence of pedestrians. However, upon processing the information that the crosswalk is unoccupied, ChatGPT suggests that maintaining the current speed is the optimal action, arguing that the absence of pedestrians eliminates the need to slow down. The rationale provided by ChatGPT is logical, yet it does not align with human values.

5 Related Work

Embodied Decision Making. Research on embodied decision-making is an emerging trend for artificial intelligent agents to interact with their sur-

Method	Model	Traffic			Domestic			Game			Average		
		P	C	A	P	C	A	P	C	A	P	C	A
End-to-End	GPT-4V	0.75	0.73	0.78	0.81	**0.69**	**0.67**	**0.95**	**0.79**	**0.77**	0.84	**0.74**	**0.74**
HOLMES	ChatGPT	0.75	0.68	0.66	**0.88**	0.52	0.50	0.78	0.40	0.36	0.80	0.53	0.51
	GPT4	**0.87**	**0.82**	**0.82**	0.85	0.61	0.56	0.91	0.77	0.74	**0.88**	0.73	0.71

Table 2: Comparison between End-to-End (MLLM) and HOLMES (LLM+API) methods on a subset of PCA-Bench with API annotation.

roundings and accomplish numerous tasks. This necessitates proficiency in vision perception, world knowledge, and commonsense reasoning, areas where a large language model can provide some level of expertise. We group prior work on embodied decision-making with LLM into two main trends. The first trend is to transform multimodal information, including object and scenery identification, the current states of AI agents, and the feedback from the environments, to texts. Text-based LLMs can then reason over the textual clues to determine the next action towards completing a designated task (Huang et al., 2022a; Li et al., 2022; Huang et al., 2022b; Chen et al., 2023). This line of research divides the entire decision-making process into two phases: (1) information seeking, usually involving MLLMs to verbalize the current status of AI agents in the vision-based environment with natural language; (2) reasoning and planning with text-based LLMs to decide what the AI agent should do in the next step with textual clues. The other line of research uses multimodal LLMs directly for end-to-end decision making, such as PALM-E (Driess et al., 2023b). The end-to-end decision making poses greater challenges to multimodal LLMs as it requires the combination of different functionalities including perception, cognition, and action, whereas decision making without explicit multiple steps mitigates the error propagation between information seeking and reasoning.

LLM-Powered Agents. LLMs pre-trained on large-scale multimodal (including text, image, video, etc.) corpus demonstrate impressive emergent abilities and immense popularity (Brown et al., 2020; Wei et al., 2022), and have seen tremendous success across various domains covering various NLP and CV tasks (Radford et al., 2019; Chowdhery et al., 2022; Touvron et al., 2023; Alayrac et al., 2022; Zhu et al., 2023a; Li et al., 2023b). Consequently, using LLMs to empower the AI agents (Xi et al., 2023b; Liu et al., 2023b; Park et al., 2023; Wang et al., 2023e) becomes more and more promising. Specifically, we can employ LLMs to enhance the decision making ability of the agents (Nakano et al., 2022; Yao et al., 2022; Li et al., 2023d; Song et al., 2023; Li et al., 2023a), expanding their perception and action space through strategies like tool utilization (Schick et al., 2023; Qin et al., 2023; Lu et al., 2023). Although LLM-based agents demonstrate reasoning and planning abilities through techniques like Chain of Thought or problem decomposition (Wei et al., 2023; Yao et al., 2023; Kojima et al., 2022), they inherently lack visual perception, and are limited to the discrete textual content. Therefore, integrating multimodal information can offer agents a broader context and a more precise understanding (Driess et al., 2023a), enhancing their environmental perception. However, no evaluation protocol or benchmark is currently available to evaluate decision making within the multimodal context.

6 Conclusion

In this paper, we introduce PCA-Bench, a multimodal benchmark designed to assess the embodied decision-making capabilities of Multimodal Large Language Models (MLLMs). This benchmark features PCA-EVAL, a novel fine-grained automatic evaluation tool that diagnoses decision-making processes from three critical perspectives: perception, cognition, and action. To enhance the decision making ability from data perspective, we propose Embodied Instruction Evolution method to automatically synthesize instruction tuning examples in various multi-modal embodied environments, which has been proved effective in our main experiments. We believe that powerful MLLMs pave a new and promising way toward decision making in embodied environments and we hope PCA-Bench could be serve as a good benchmark in bridging MLLMs and embodied artificial intelligence.

7 Limitation

The current scope of PCA-Bench is confined to merely three domains in static environments. One of our future work aims to broaden this scope to encompass more domains and dynamic embodied environments where MLLMs could keep getting feedback. We do not apply different reasoning enhancement method like Reflection in the decision making process of MLLMs. We just use the simplest prompting method and leave the exploration of better cross-modal Chain-of-Thought method for future studies.

References

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

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. ArXiv, abs/2308.12966.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901.

Xiaoyu Chen, Shenao Zhang, Pushi Zhang, Li Zhao, and Jianyu Chen. 2023. Asking before action: Gather information in embodied decision making with language models. arXiv preprint arXiv:2305.15695.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023a. Instructblip: Towards general-purpose vision-language models with instruction tuning.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Albert Li, Pascale Fung, and Steven C. H. Hoi. 2023b. Instructblip: Towards general-purpose vision-language models with instruction tuning. ArXiv, abs/2305.06500.

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. 2023a. Palm-e: An embodied multimodal language model.

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

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. 2022. Minedojo: Building open-ended embodied agents with internet-scale knowledge. In Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track.

Joaquın M. Fuster. 2004. Upper processing stages of the perception–action cycle. Trends in Cognitive Sciences, 8(4):143–145.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pages 6325–6334.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. 2023. Planning-oriented autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

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

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. 2022b. Inner monologue: Embodied reasoning through planning with language models. In arXiv preprint arXiv:2207.05608.

Seungone Kim, Se June Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. 2023. The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. Advances in neural information processing systems, 35:22199–22213.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. 2017. AI2-THOR: An Interactive 3D Environment for Visual AI. arXiv.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023a. Camel: Communicative agents for "mind" exploration of large language model society.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023b. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597.

Lei Li, Yuwei Yin, Shicheng Li, Liang Chen, Peiyi Wang, Shuhuai Ren, Mukai Li, Yazheng Yang, Jingjing Xu, Xu Sun, Lingpeng Kong, and Qi Liu. 2023c. M^3it: A large-scale dataset towards multi-modal multilingual instruction tuning. arXiv preprint arXiv:2306.04387.

Minghao Li, Feifan Song, Bowen Yu, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023d. Api-bank: A benchmark for tool-augmented llms.

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.

Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, Jiaming Han, Siyuan Huang, Yichi Zhang, Xuming He, Hongsheng Li, and Yu Qiao. 2023. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning.

Ruibo Liu, Ruixin Yang, Chenyan Jia, Ge Zhang, Denny Zhou, Andrew M Dai, Diyi Yang, and Soroush Vosoughi. 2023b. Training socially aligned language models in simulated human society. arXiv preprint arXiv:2305.16960.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023. Chameleon: Plug-and-play compositional reasoning with large language models. arXiv preprint arXiv:2304.09842.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback.

OpenAI. 2023. Gpt-4v(ision) system card.

Joon Sung Park, Joseph C O'Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. arXiv preprint arXiv:2304.03442.

Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Yufei Huang, Chaojun Xiao, Chi Han, et al. 2023. Tool learning with foundation models. arXiv preprint arXiv:2304.08354.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9.

Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. arXiv.

Shuhuai Ren, Aston Zhang, Yi Zhu, Shuai Zhang, Shuai Zheng, Mu Li, Alex Smola, and Xu Sun. 2023. Prompt pre-training with twenty-thousand classes for open-vocabulary visual recognition. arXiv preprint arXiv:2304.04704.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Yifan Song, Weimin Xiong, Dawei Zhu, Wenhao Wu, Han Qian, Mingbo Song, Hailiang Huang, Cheng Li, Ke Wang, Rong Yao, Ye Tian, and Sujian Li. 2023. Restgpt: Connecting large language models with real-world restful apis.

Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. 2017. A corpus of natural language for visual reasoning. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 217–223.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

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

Peiyi Wang, Lei Li, Liang Chen, Feifan Song, Binghuai Lin, Yunbo Cao, Tianyu Liu, and Zhifang Sui. 2023b. Making large language models better reasoners with alignment. arXiv preprint arXiv:2309.02144.

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023c. Large language models are not fair evaluators. arXiv preprint arXiv:2305.17926.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023d. Self-instruct: Aligning language models with self-generated instructions.

Zihao Wang, Shaofei Cai, Anji Liu, Xiaojian Ma, and Yitao Liang. 2023e. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. ArXiv, abs/2302.01560.

Wayve. 2023. Lingo.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Chenfei Wu, Sheng-Kai Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. 2023. Visual chatgpt: Talking, drawing and editing with visual foundation models. ArXiv, abs/2303.04671.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Qin Liu, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huan, and Tao Gui. 2023a. The rise and potential of large language model based agents: A survey. ArXiv, abs/2309.07864.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, and Tao Gui. 2023b. The rise and potential of large language model based agents: A survey.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. 2023. Mm-react: Prompting chatgpt for multimodal reasoning and action. ArXiv, abs/2303.11381.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In The Eleventh International Conference on Learning Representations.

Weiran Yao, Shelby Heinecke, Juan Carlos Niebles, Zhiwei Liu, Yihao Feng, Le Xue, Rithesh Murthy, Zeyuan Chen, Jianguo Zhang, Devansh Arpit, et al. 2023. Retroformer: Retrospective large language agents with policy gradient optimization. arXiv preprint arXiv:2308.02151.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023. Multimodal chain-of-thought reasoning in language models. arXiv preprint arXiv:2302.00923.

Haozhe Zhao, Zefan Cai, Shuzheng Si, Xiaojian Ma, Kaikai An, Liang Chen, Zixuan Liu, Sheng Wang, Wenjuan Han, and Baobao Chang. 2023. Mmicl: Empowering vision-language model with multi-modal in-context learning. arXiv preprint arXiv:2309.07915.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023a. Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv preprint arXiv:2304.10592.

Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, Yu Qiao, Zhaoxiang Zhang, and Jifeng Dai. 2023b. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. arXiv preprint arXiv:2305.17144.

Zhe Zhu, Dun Liang, Songhai Zhang, Xiaolei Huang, Baoli Li, and Shimin Hu. 2016. Traffic-sign detection and classification in the wild. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

A Examples of PCA-Bench

A.1 Data Distribution

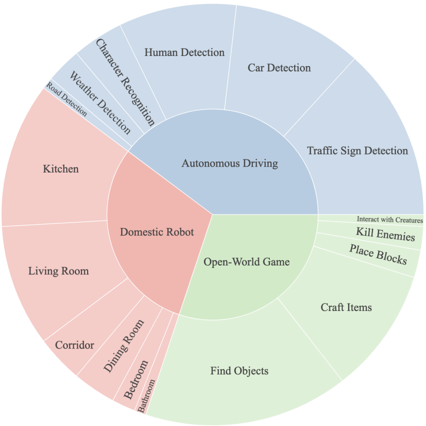


Figure 8: Domain and required ability distribution of PCA-Bench.

The PCA-Bench's data distribution across various domains is outlined in Figure 8. For the Autonomous Driving domain, instances are grouped by their respective task types. In the Domestic Robot domain, instances are grouped by their locations. In the Open-World Game domain, instances are grouped by the tasks they aim to accomplish.

B Human Annotation Pipelines

The annotation process consists of two stages: (1) Dataset Annotation, and (2) Dataset Refinement. During the initial stage, three annotators are assigned to each domain, adhering strictly to the respective annotation guidelines. They first pinpoint the source images from each domain that are informative and meaningful so that they can write questions for each image. All annotators are from the author list of this paper. The annotators have the responsibility to ensure every question has only one correct answer and accurate rationales. In the subsequent stage, annotators are instructed to scrutinize the output actions and rationales presented by ChatGPT and check the annotations. This process aims to address the challenge of multiple correct answers, as ChatGPT can furnish comprehensive explanations for its actions. These explanations assist annotators in assessing the acceptability of ChatGPT's response, particularly when it deviates from the established ground truth answer. This enables annotators to refine annotations to ensure the presence of a single correct answer.

B.1 PCA-EVAL Examples

We list three examples of each domain from PCA-EVAL, as shown in Figure 11, 12, and 13.

C Zero Shot Modality Conversion: HOLMES

To optimize the evaluation process of HOLMES method, we pre-execute all relevant APIs for each instance within a selected subset of 300 instances from the PCA-Bench test set, recording the results for individual instances. This method enables immediate access to specific API results, eliminating the need to rerun the model for each evaluation instance.

Traffic Domain. Below is the API description for the traffic domain.

```
# API Description for Traffic Domain:
def detect_traffic_sign():
    """
    Detects traffic signs in the image.
    :return: list of detected traffic
   signs and coordinates, e.g. ['stop
   ','max speed limit']
    """
    pass

def object_detection():
    """
    Detects objects in the image.
    :return: dict of detected objects
   and number of the objects, e.g. {'
   car':10, 'person':1}
    """
    pass

def ocr():
    """
    Performs OCR on the image.
    :return: list of detected text, e.g.
     ['Changjiang road', 'Right lane
   closure']
    """
    pass

def image_caption():
    """
    Generates a caption for the image.
    :return: caption, e.g. 'A red car
   driving down the street'
    """
    pass

def weather_detection():
    """
    Detect current weather.
    :return: weather, e.g. 'rainy' or '
   clear'
    """
    pass
```

- *detect_traffic_sign()*: The detection of road traffic signs model utilize YOLO (Redmon and

Figure 9: Three examples of HOLMES solving questions from different domains of PCA-Bench.

Farhadi, 2018) which trained on the Tsinghua-Tencent 100K dataset (Zhu et al., 2016). TT100K comprises 100,000 images encompassing 30,000 instances of traffic signs. The end-to-end YOLO enables simultaneous detection and classification of traffic signs.

- *object_detection()*: Objects demanding attention during vehicle operation primarily encompass cars, pedestrians, and bicycles. A surfeit of vehicles can lead to traffic congestion, while the presence of pedestrians or bicycles ahead necessitates cars to decelerate and proceed cautiously. Hence, the *object_detection()* API predominantly identifies three key object categories: cars, pedestrians, and bicycles. We utilize PMOP (Ren et al., 2023), a model trained on vision-language models through the prompt pre-training method, which enables the detection and counting of the three mentioned objectives by modifying specific class names.

- *ocr()*: We employ PaddleOCR[1] to extract textual information from images, providing crucial road data for real-time navigation.

- *image_caption()*: To initially streamline the road information within the image, we employ the BLIP2-flan-t5-xl to generate an initial caption for the picture. This caption, derived from basic image data, is then utilized as input for the model to facilitate decision-making.

- *weather_detection()*: Weather detection leverages a pre-trained ResNet50 model[2], derived from a dataset of more than 70,000 weather records. This model extracts weather information from provided images to inform decision-making.

Domestic Robot Domain. Below is the API description for the Domestic Robot domain.

```
#API Description for Domestic Robot
    Domain
def object\_detection():
    """
    Detects objects in current view,
    which you don't need do find.
    :return: list of detected objects, e
    .g. ['chair','table']
    """
    pass

def list_items_in_hands():
    """
    Lists items in your hand, which you
    don't need to pick up
    :return: list of items in hand, e.g.
     ['coffee cup','milk']
    """
    pass
```

Game Domain. Below is the API description for the Game domain (Minedojo).

[1]https://github.com/PaddlePaddle/PaddleOCR/tree/release/2.7

[2]https://github.com/mengxianglong123/weather-recognition

```python
#API Description for Game Domain
def list_nearby_mobs_in_minecraft():
    """
    Lists nearby mobs in Minecraft.
    :return: list of nearby mobs, e.g.
   ['creeper', 'pig']
    """
    pass

def list_inventory_information():
    """
    Lists inventory information of the
   player in Minecraft.
    :return: list of inventory
   information with number, e.g. [('
   diamond', 64), ('iron', 32)]
    """
    pass
```

Note that within the Domestic Robot Domain and Game Domain, APIs can be directly accessed within the virtual environment, allowing for the perception of the surrounding objects and the current picture context.

Figure 10: An case showing the value mis-alignment between of agent and human’s decision.

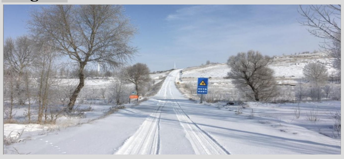
D Automatic Evaluation

We utilize the template as shown in Table 3.

E Training Details

Table 4 shows the specific parameters used for fine-tuning in different models. The PCA results on the three domains of PCA bench before and after fine-tuning different models are shown in Figure 14.

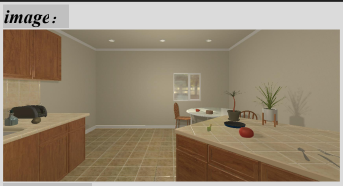
Quetsion:
Based on current image, what is the best action to take when you are driving on road?
Action candidates:
(A) Drive close to the right side
(B) Drive close to the left side (C) Call 911
Answer:
(B) Drive close to the left side
Reason:
Based on the image, The right side of the road is closed, so you should drive close to the left.
Key Concept:
right lane closure



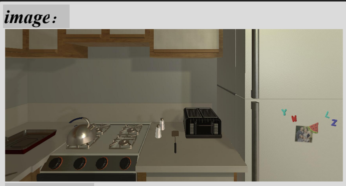
Quetsion:
Based on current image, What should you do if you want to drive forward safely?
Action candidates:
(A) Slow down (B) Keep driving (C) Call the police
Answer:
(A) Slow down
Reason:
The current heavy snowfall is causing slippery roads. So you should slow down.
Key Concept:
Snowy weather

Quetsion:
You are at 20km/h now. What should you do if you want to overtake the car in front of you?
Action candidates:
(A) Speed up to 50km/h (B) You can't overtake now (C) Turn Left
Answer:
(B) You can't overtake now
Reason:
You did not exceed the speed limit. But road is congested. So you can't overtake now.
Key Concept:
congested road

Figure 11: Three examples of PCA-EVAL in the autonomous driving domain.



Quetsion:
You are in a room, the goal is to water the plant. What is the best action to take next?
Action candidates:
(A) Find a sprinkler (B) Pick up the sprinkler
(C) Use sprinkler water the plant
Answer:
(C) Use sprinkler water the plant
Reason:
There is a house plant in front of us. We have a sprinkler in hands. Therefore, our next step is to water the plant using the sprinkler.
Key Concept:
plant, sprinkler



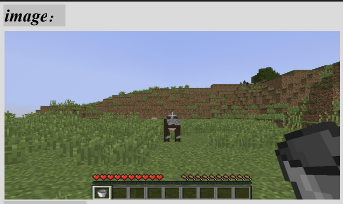
Quetsion:
You are in a room, the goal is to wash the cup. What is the best action to take next?
Action candidates:
(A) Find the cup (B) Find the dish sponge
(C) Find a sink (D) Open the faucet
Answer:
(A) Find the cup
Reason:
There is no cup in front of us. Therefore, our next step is to find a sink.
Key Concept:
cup, dish sponge, sink

Quetsion:
You are in a room, the goal is to cook rice. What is the best action to take next?
Action candidates:
(A) Go to the kitchen (B) Pick up rice
(C) Pick up pot (D) Boil water
Answer:
(A) Go to the kitchen
Reason:
We can infer from the objects around us that we are in the living room so our next step is to go to the kitchen
Key Concept:
rice, kitchen, living room

Figure 12: Three examples of PCA-EVAL in the domestic robot domain.

Quetsion:
To harvest beef using an iron sword, based on the image, which is your next action?
Action candidates:
(A) find cow (B) kill cow (C) craft iron sword
(D) find iron ore (E) craft iron ingot
Answer:
(C) craft iron sword
Reason:
You need a cow and kill it with an iron sword for beef. You're near a cow, but lack an iron sword. So, craft one using two iron ingots and a stick. You have two iron ingots, and two sticks so you can craft an iron sword.
Key Concept:
Cow nearby, No iron sword, 2 sticks,2 iron ingots

Quetsion:
To place a crafting table in front of you, based on the image, which is your next action?
Action candidates:
(A) find planks (B) craft crafting table (C) place crafting table
Answer:
(A) find planks
Reason:
To place a crafting table in front of you, your inventory must have a crafting table. To get a crafting table in your inventory, you need to make one. You need 4 planks to make a crafting table. Since you have 3, find one more first.
Key Concept:
No crafting table, 3 planks



Quetsion:
To harvest milk, which is your next action?
Action candidates:
(A) find cow (B) milk cow (C) craft bucket (D) find iron ore (E) craft iron ingot
Answer:
(B) milk cow
Reason:
To get milk, you need to find a cow, and milk it with a bucket. There is a cow nearby, and you have a bucket, so you can milk the cow.
Key Concept:
Cow nearby, Have bucket

Figure 13: Three examples of PCA-EVAL in the open-world game domain.

[Question]: {question}
[Action Choices]: {actions}
[Agent Answer]: {model_output}
[Correct Action]: {true_action}
[Key Concepts]: {key_concept}
[Reference Reasoning Process]: {reason}
[System]
We would like you to access the agent's performance in the multimodal reasoning task about domain. In this task, the agent is given an image, a [Question], and several candidate [Action Choices], and is asked to give an [Agent Answer] for the [Question]. The [Agent Answer] encapsulates the agent's perception of the image's [Key Concepts], the agent's cognition reasoning process and the final selected action.
We request you to give three types of scores for the agent's [Agent Answer] in comparison to the given [Key Concepts], [Reference Reasoning Process] and [Correct Action]:
1. action score: If the selected action in the [Agent Answer] matches that of the [Correct Action], the action score is 1; otherwise, it is 0.
2. perception score: This score evaluates the model's capability to perceive and interpret observations. It is contingent on whether the [Agent Answer] includes any of the [Key Concepts] of the instance. If it accurately describes any one of the [Key Concepts], the score is 1; otherwise, it is 0.
3. cognition score: This score gauges the model's ability to reason, comprehend, and make informed decisions based on perceived input data and world knowledge. If the reasoning process in the [Agent Answer] aligns with the [Reference Reasoning Process], the score is 1; otherwise, it is 0.
Please note that there are only scores of 0 and 1.
You should carefully compare the [Agent Answer] with the [Correct Action], [Key Concepts] and [Reference Reasoning Process] to give your assessment.
You need first to give your assessment evidence and then the scores.
Your output MUST contain 6 lines with the following format:
action assessment evidence: (assessment evidence here)
action score: (score here)
perception assessment evidence: (assessment evidence here)
perception score: (score here)
cognition assessment evidence: (assessment evidence here)
cognition score: (score here)

Table 3: The template of querying GPT-4.

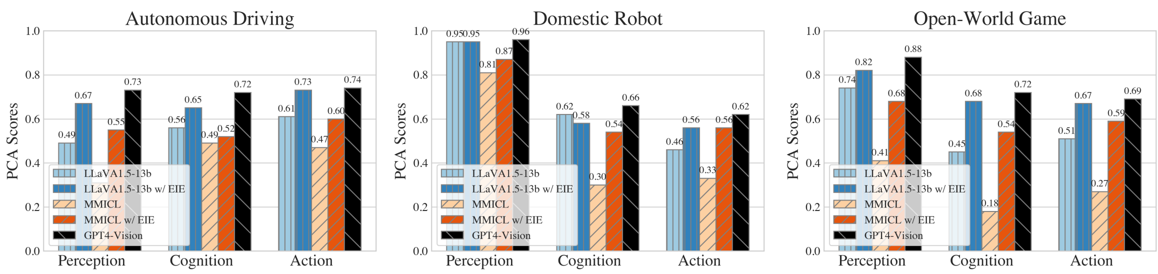


Figure 14: Performance comparsion between models' zero-shot results and models' finetuned results with the data generated by Embodied-Instruct-Evolution (EIE) method. EIE improves the performance on all domains for both LLaVA1.5-13b and MMICL models.

Model	Parameter	Value
Qwen-VL-Chat/LLaVA1.5-7/13b	Learning Rate	2e-4
	Use Lora Finetuning?	Yes
	Lora Rank	8
	Lora Alpha	32
	Global Batchsize	20
	Weight Decay	0
	Train Epochs	5
	Lr Scheduler Type	Cosine
	Warmup Ratio	0.03
MMICL	Learning Rate	5e-4
	Use Lora Finetuning?	No
	Global Batchsize	20
	Weight Decay	5e-4
	Train Epochs	5
	Lr Scheduler Type	Linear
	Warmup Ratio	0.2

Table 4: Training details for different models with EIE.