UnAC: Adaptive Visual Prompting with Abstraction and Stepwise Checking for Complex Multimodal Reasoning

Anonymous ACL submission

Abstract

Recent large multimodal models (LMMs) have demonstrated impressive capabilities in image understanding, yet they still struggle to perform complex reasoning on challenging multimodal problems. In this paper, we present UnAC (Understanding, Abstracting, and Checking), a multimodal prompting method that strengthens reasoning for complex multimodal tasks in LMMs (e.g., GPT-4o, Gemini 1.5, and GPT-4V). To improve image understanding and capture fine details, we propose an adaptive visual prompting strategy that enables LMMs to focus on salient regions. We further design an image-abstraction prompt to effectively extract key information from images. In addition, we introduce a gradual self-checking scheme that improves reasoning by verifying each decomposed subquestion and its answer. Extensive experiments on three public benchmarks—MathVista, MM-Vet, and MMMU—demonstrate the effectiveness of our method.

1 Introduction

In recent years, large language models (LLMs) have advanced significantly (Brown et al., 2020; Achiam et al., 2023; Touvron et al., 2023; Bubeck et al., 2023; Chowdhery et al., 2023; Zhang et al., 2022). Models such as GPT-3 (Brown et al., 2020), PaLM (Chowdhery et al., 2023), and Llama (Touvron et al., 2023), followed by GPT-4 (Achiam et al., 2023) and PaLM 2 (Anil et al., 2023), have driven numerous breakthroughs in both industry and academia. This progress has spurred rising interest in large multimodal models (LMMs), with many approaches building powerful systems on open-source frameworks (Liu et al., 2024; Wu et al., 2023; Dai et al., 2024; Zhu et al., 2023). Recently, the releases of GPT-4V(ision) and Gemini 1.5 Flash (Team et al., 2023) have drawn substantial attention for their strong capabilities in image understanding; however, these models still struggle with complex multimodal reasoning tasks (Lu et al., 2023; Yue et al., 2023).

Since prompting approaches designed to improve the reasoning ability of LLMs in purely linguistic contexts (Yao et al., 2024; Wei et al., 2022; Yao et al., 2022; Miao et al., 2023; Zheng et al., 2023) have made significant progress, one might hope to transfer them directly to vision. However, because LMMs cannot decompose an image as easily as they can decompose a sentence, applying language-only prompts to enhance reasoning in the visual context is often ineffective. In visual question answering, major failure cases typically stem from misinterpreting the image or imprecisely summarizing its information. Such omissions or misunderstandings are closely related to limited fine-grained perception capability (Yang et al., 2023a).

Visual prompting has also been explored for various multimodal tasks, especially to enhance fine-grained recognition. Many methods encode masks (e.g., points, boxes, lines) that are fused with input features or directly overlay cues on the original image. Most recently, Yang *et al.* proposed building a visual prompting mechanism by partitioning the image into semantically meaningful regions and overlaying them to strengthen the grounding ability of GPT-4V. However, for complex questions that typically require multi-step information extraction and reasoning, merely partitioning the image is not sufficient to substantially improve reasoning performance.

In this paper, we propose a multimodal prompting method, **UnAC** (**Un**derstanding, **A**bstracting, and **C**hecking), to improve complex multimodal reasoning in LMMs. UnAC follows a three-step prompting pipeline. First, we introduce an adaptive visual prompting scheme that places image-conditioned markers to direct the model's attention toward specific regions, thereby reducing misunderstandings and missed details. By inspecting the image part by part, the model discovers more

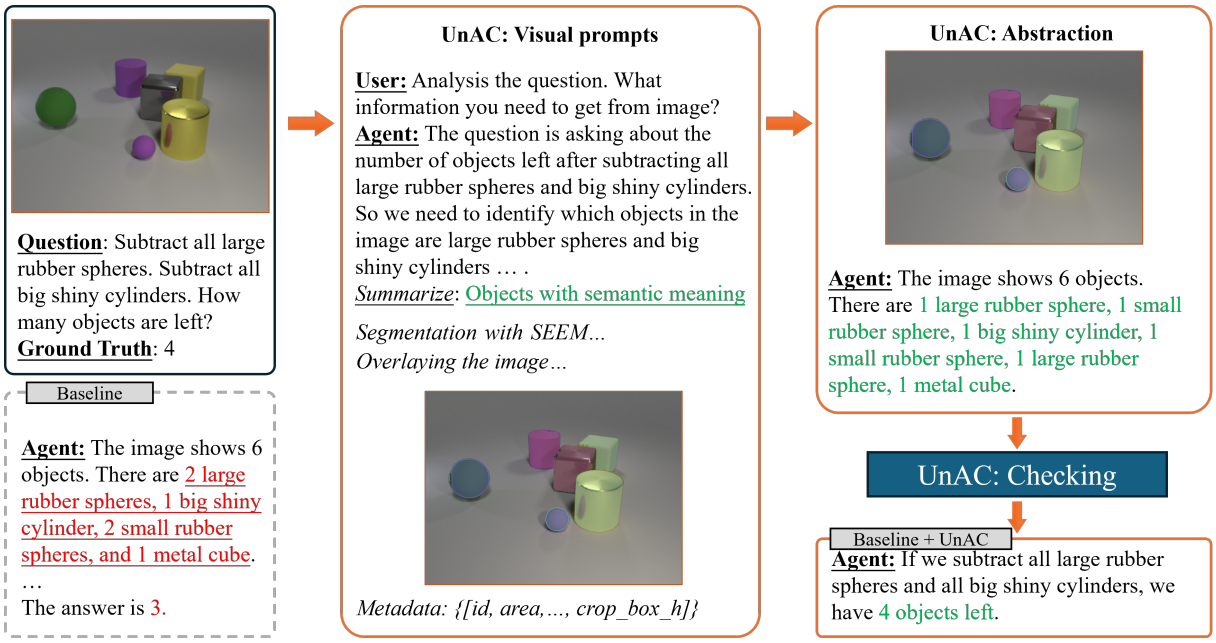

Figure 1: Example of using UnAC. In the original answer from the baseline method, the LMM incorrectly understands and describe the image which leads to the wrong answer. In UnAC which follows in the orange arrows, we first ask the LMM to analyze the question and answer what we need from the image. Then we can summarize the reply as the Objects with semantic meaning. Then employing SEEM to segment and overlay the image as visual prompts. Then abstracting the information of the image where with the markers, the LMM can correctly describe the image and abstract the right contexts. Finally, after the checking stage, we can get the right answer.

fine details and forms a more accurate global understanding. Second, to solve problems that require complex reasoning, we abstract the image into language based on the question. Inspired by how humans process visual information given a query, we extract key evidence both locally and globally by identifying question-relevant regions and converting them into textual descriptions using the established visual prompts. Third, because LMMs are prone to making errors at intermediate steps and global self-checking is often ineffective, we introduce a *gradual self-checking* scheme that verifies each decomposed subquestion and its answer individually within the visual context.

We evaluate UnAC on three datasets that assess complex problem solving in the visual domain: MathVista (Lu et al., 2023), MM-Vet (Yu et al., 2023), and MMMU (Yue et al., 2023). To demonstrate generalization, we conduct experiments on two categories of LMMs: (a) powerful large-scale models, including GPT-4V and Gemini-1.5-Flash; and (b) relatively lightweight models, including LLaVA-v1.6-7B/13B. UnAC yields improvements across all models and datasets, indicating that our method is model-agnostic. Notably, on MathVista with Gemini-1.5-Flash, our method achieves a 6.4% absolute gain.

To summarize, our main contributions are:

- We introduce **UnAC** (**Un**derstanding, **A**bstracting, and **C**hecking), a simple yet effective multimodal prompting framework that strengthens complex multimodal reasoning in LMMs.
- We design an *adaptive visual prompting* strategy that focuses the model on salient regions and reduces missed details, and couple it with question-conditioned *image abstraction* and a *gradual self-checking* procedure, yielding finer perception and more reliable step-wise reasoning.
- We validate UnAC on three benchmarks—MathVista, MM-Vet, and MMMU—achieving consistent improvements in complex visual reasoning across diverse LMMs.

2 Related Work

Prompting in LLMs. We have observed significant advancements in large language models (LLMs) (Zhang et al., 2022, 2023; Touvron et al.,

2023; Team et al., 2023; Brown et al., 2020). Although the size of LLMs has increased substantially, evoking their reasoning capabilities is still necessary with the use of more complicated designed queries, or prompting. Recently, various works have explored prompt engineering to enhance LLM capabilities. In-context learning has become a mainstream approach to instruct LLMs by providing specific examples (Brown et al., 2020; Dong et al., 2022). Building on this, techniques such as chain-of-thought and tree-of-thought (Wei et al., 2022; Yao et al., 2024) have been introduced to improve performance in arithmetic, commonsense, and symbolic reasoning tasks. Most recently, Zheng *et al.* (Zheng et al., 2023) proposed the Step-Back Prompting method which enhances the ability to retrieve information via abstracting the question. Miao *et al.* (Miao et al., 2023) introduced a general-purpose zero-shot verification schema for recognizing errors made in the reasoning process of math problems. However, their methods highly rely on that the language is easy to be decomposed. It is hard to be generalized to the question in the visual context where images are hard to decompose.

Prompting in LMMs Before the growth of large multimodal models (LMMs), visual prompting has been explored for various vision and multimodal tasks (Wang et al., 2023; Zou et al., 2024; Kirillov et al., 2023; Chen et al., 2022; Shtedritski et al., 2023). These approaches can be categorized into two main types. The first type encodes visual prompts, such as points, boxes, and strokes, into latent features, which are then used to prompt the vision models (Zou et al., 2024; Kirillov et al., 2023). The second type overlays visual marks directly onto the input images. These marks can be a red circle (Shtedritski et al., 2023), a highlighted region (Yang et al., 2023a), or multiple circles with arrows (Shtedritski et al., 2023). While these studies show the potential of pixel-level visual prompting, they are typically limited to visually referencing one or a few objects. So far, prompting LMMs has been rarely explored in academia, partly because most of the recently open-sourced models have limited capacity and are therefore unable to support such advanced capabilities. Recently, GPT-4V was released, accompanied by a comprehensive qualitative study (Yang et al., 2023b). The authors in (Yang et al., 2023b) employed a similar prompting strategy as RedCircle (Shtedritski et al., 2023) to prompt GPT-4V. Most recently, Yang *et al.* (Yang et al., 2023a) proposed to partition the image into a set of semantically meaningful regions and overlay them to enhance the grounding ability of GPT-4V. CCoT (Mitra et al., 2023) is designed as a zero-shot Chain-of-Thought prompting method to extract compositional knowledge from an LMM with utilizing scene graphs. However, both of these works can not solve the problem based on the abstract images such as geometry problem solving and math word problems.

3 UnAC: Understanding, Abstracting, and Checking

Consider a general fact when humans face a challenging problem in the visual context. To solve the problem, we first need to understand the image and the question correctly overall. Then based on the question, we will look at the image more carefully, find and abstract the useful information that can be used to solve the problem. Finally, based on the understanding and the abstraction, we infer the final answer to this challenging problem. Moreover, for a complicated question, we usually need a second look at the reasoning process and check it with the image to avoid some simple mistakes. Inspired by this common sense, we propose UnAC which means understanding, abstracting, and checking for synergizing the complicated reasoning in the visual context of large multimodal models.

3.1 Adaptive Visual Prompts.

Precisely capturing the details in the image is not straightforward for LMMs. It is hard to correct the misunderstanding of the image by itself because decomposing the image is not easy. Since LMMs are developed based on the LLMs, their abilities of language reasoning are much better than visual reasoning. It means that LMMs can perform better on analyzing the problem than analyzing the image. Therefore, we propose to build effective and adaptive multimodal prompts based on the analysis of the question. Asking the model to analyze the question and find what information we need to get from the image. We conclude the response into two kinds: Objects with semantic meaning and symbols with literal meaning. For objects with semantic meaning, we employ segmentation models to automatically segment the image. For symbols with literal meaning, we use optical character recognition (OCR) methods to detect the texts. Based on the metadata, we first denoising regions based on

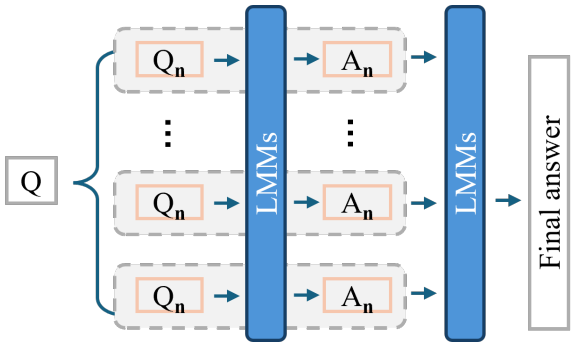

Figure 2: The workflow of gradual checking proceeds as follows: First, the input question is decomposed into several sub-questions by the LMMs. Then, the LMMs answer each sub-question. Both the sub-questions and their corresponding answers are then fed back into the LMMs to generate the final answer.

the stability score output by the segmentation/OCR methods.

In the Figure. 1, we show a successful case. For this question of subtracting the items, it requires LMMs to correctly recognizing each item in the picture which is related to objects with semantic meanings. Therefore, the visual prompts are designed as the segmentation of the image to help the LMM to better understand the image.

3.2 Image Abstraction

The visual prompts can make a better understanding of the image since the markers can catch more attentions on some local information. Partitioning the image makes it decomposable when LMMs understand the image. However, only visual prompts have limited improvements for solving complicated problems. Except for understanding the image, LMMs need to correctly abstract the image to filter the useless information to solve the problem. Without prompts of abstraction, the reasoning might be misdirected due to the markers in the image. Therefore, to fully utilize the visual prompts and get better reasoning, we need to abstract the information which is the most related to the question. Firstly, we ask LMMs to describe the picture to abstract the global information. Then based on the analysis of the question and the prompts, we ask LMMs to find the most related regions to get more details based on the markers in the image.

3.3 Gradual checking

Moreover, for some complicated questions, we usually need a second look at the image with the reasoning progressing. As discussed in (Ling et al., 2024), checking the whole reasoning process is usually ineffective for LLMs and our experiments show similar results in LMMs. However, to correct the mistake made in one step is more effective. To check individual steps of the reasoning process, the first thing we should note is that the correctness of each step is highly dependent on its context. For a question in words, the context includes the question and previous steps only. So the checking is largely dependent on the accuracy of the previous steps which is highly unstable. In the visual question answering, the information from the image becomes extra contexts which are important references for self-checking. It can be more reliable when LMMs have a good understanding of the image.

Then, we design a gradual checking prompting for better reasoning. Firstly, we let LMMs decompose the question into multi sub-questions $[Q_0, Q_1, \ldots, Q_n]$ and give the answer of each sub-questions. The answers are denoted as $[A_0, A_1, \ldots, A_n]$. In the checking stage, we check gradually. When checking Q_i and A_i, we refer the context of the previous questions and checked answers $[Q_0, Q_1, \ldots, Q_i]$ and $[A'_0, A'_1, \ldots, A'_i]$. In the last step of checking, LMMs will infer the final answer based on all questions and answers.

4 Experiments

4.1 Setup

Tasks and datasets. We experiment with the following two tasks that need complicated reasoning: (a) Mathematical reasoning in the visual context, and (b) Complicated VQA. *Mathematical reasoning:* We evaluate MathVista (Lu et al., 2023) for this task. MathVista is a consolidated mathematical reasoning benchmark within visual contexts. It contains various kinds of sub-tasks to evaluate the model's visual understanding of mathematical problems solving in different perspectives of reasoning skills. *Complicated VQA:* For this task, we evaluate two datasets called: MM-Vet (Yu et al., 2023) and MMMU (Yue et al., 2023) respectively. MM-Vet (Yu et al., 2023) is designed to evaluate large multimodal models on complex multimodal tasks that highlight six core vision-language (VL) capabilities: Recognition, Knowledge, Optical Character Recognition (OCR), Spatial Awareness, Language Generation, and Math. MMMU focuses on advanced perception and reasoning with domain-specific knowledge, challenging models to perform tasks like those faced by experts.

Table 1: Accuracy scores on the *testmini* subset of MathVista (Lu et al., 2023). ALL: overall accuracy. Task types: FQA: figure question answering, GPS: geometry problem solving, MWP: math word problem, TQA: textbook question answering, VQA: visual question answering. Mathematical reasoning types: ALG: algebraic reasoning, ARI: arithmetic reasoning, GEO: geometry reasoning, LOG: logical reasoning, NUM: numeric commonsense, SCI: scientific reasoning, STA: statistical reasoning.

Method	ALL	FQA	GPS	MWP	TQA	VQA	ALG	ARI	GEO	LOG	NUM	SCI	STA
Human performance													
Human Performance	60.3	59.7	48.4	73.0	63.2	55.9	50.9	59.2	51.4	40.7	53.8	64.9	63.9
Heuristics baselines													
Random chance	17.9	18.2	21.6	3.8	19.6	26.3	21.7	14.7	20.1	13.5	8.3	17.2	16.3
Frequent guess	26.3	22.7	34.1	20.4	31.0	24.6	33.1	18.7	31.4	24.3	19.4	32.0	20.9
Closed-sourced Large Multimodal Models (LMMs)													
Gemini-1.0-pro-vision	41.0	36.4	36.5	43.0	57.5	36.3	39.8	37.6	38.0	10.8	29.8	52.4	45.5
Gemini-1.0-pro-vision + UnAC	**47.4**(+6.4)	49.4	39.5	45.2	62.7	42.5	43.8	44.2	40.1	29.7	36.1	54.9	57.1
Gemini-1.5-flash	53.2	51.6	56.8	52.1	67.5	40.1	59.4	44.0	52.7	25.3	36.4	60.8	57.5
Gemini-1.5-flash + UnAC	**56.6**(+3.4)	57.3	59.3	54.1	71.6	41.4	62.8	46.6	59.5	30.9	37.7	65.3	65.8
GPT-4V	50.7	43.6	50.5	57.5	65.2	38.4	53.0	49.0	51.0	21.6	20.1	63.1	55.8
GPT-4V +SoM (Yang et al., 2023a)	51.2(+0.5)	50.5	52.9	49.7	64.8	37.2	53.4	44.0	51.2	18.9	32.4	62.8	57.5
GPT-4V + CCoT (Mitra et al., 2023)	51.8(+1.1)	46.2	50.2	58.2	64.2	40.4	55.0	48.2	51.2	21.6	20.1	57.1	59.2
GPT-4V + SKETCHPAD (Hu et al., 2024)	52.0(+1.3)	44.2	52.6	60.5	66.4	37.8	53.5	48.2	51.9	21.2	19.8	63.8	55.6
GPT-4V + UnAC	**57.6**(+4.9)	47.3	61.1	55.2	69.7	48.9	60.9	50.1	58.5	18.9	35.4	60.7	57.8
Open-sourced Large Multimodal Models (LMMs)													
LLaVA-OneVision-7B	34.8	43.6	21.6	27.9	43.4	37.8	26.5	32.0	23.3	19.9	24.9	49.1	44.6
LLaVA-OneVision-7B + SoM	34.6(−0.2)	43.7	19.6	29.6	43.2	37.1	26.9	31.6	20.8	19.6	25.2	50.2	43.9
LLaVA-OneVision-7B + UnAC	**36.2**(+1.4)	49.5	20.1	28.5	44.4	38.8	32.5	31.2	21.1	18.4	25.1	50.3	44.6
LLaVA-v1.6-13B	35.8	45.3	21.6	29.5	43.0	37.9	24.9	33.9	23.8	13.5	27.7	49.1	48.1
LLaVA-v1.6-13B + UnAC	**37.8**(+2.0)	37.5	31.7	30.7	53.8	38.5	33.4	34.6	32.2	10.8	25.7	53.3	44.9
InternVL2.0-8B	67.3	72.5	73.6	69.9	66.5	50.3	70.1	57.5	71.5	27.0	43.1	65.6	79.1
InternVL2.0-8B + SoM (Yang et al., 2023a)	67.2 (−0.1)	75.0	72.4	72.0	65.2	51.3	73.2	58.5	72.5	22.5	41.0	65.6	79.1
InternVL2.0-8B + CCoT (Mitra et al., 2023)	68.2 (+0.9)	75.6	76.2	72.3	65.8	49.1	69.5	60.5	70.0	25.2	42.5	68.6	75.2
InternVL2.0-8B + SKETCHPAD (Hu et al., 2024)	69.2(+1.9)	77.2	77.6	72.8	70.1	48.3	73.2	62.1	76.2	28.3	43.2	64.6	79.3
InternVL2.0-8B + UnAC	**71.6**(+4.3)	77.3	79.6	75.0	72.1	54.0	75.4	62.5	75.5	31.0	44.8	67.2	80.7

Models. To show the generalization of UnAC, we use the following state-of-the-art LLMs: close-source models including GPT4-V and Gemini-1.5-flash, relatively small LMMs including LLaVA-v1.6-7B/13B (Liu et al., 2024), LLaVA-OneVision (Li et al., 2024) and internVL2.0-8B (Chen et al., 2024). For the closed-source LMMs, we utilize the official API to make the evaluation. We use 'gpt4-turbo' and 'gemini-1.5-flash' for GPT4-V and Gemini respectively. For the open-source models, we evaluate in a single RTX 6000. We set the temperature to 0.0 for all LMMs. We use SEEM (Zou et al., 2024) for segmentation and easyOCR for building the visual prompts. Moreover, we also compare with two chain-of-thought methods including CCoT (Mitra et al., 2024) and SKETCHPAD (Hu et al., 2024) to show the superiority of UnAC as a training-free method.

Evaluation. In all datasets, they have a unique answer to each question which can be a number, a word, a phrase, or one of the choices. The accuracy (ACC) is the only metric we employed in this paper. Since the LMMs may often generate long-form answers which are hard to capture. Following Lu *et al.* (Lu et al., 2023) and Yu *et al.* (Yu et al., 2023), we instead conduct an evaluation using the GPT-4 model where we few-shot prompt the model to identify equivalence between target answers and the model predictions.

4.2 Results

Mathematical reasoning in the visual context. In Table 1, we present results on the MathVista benchmark (Lu et al., 2023), where our method consistently improves performance across all models. Specifically, we achieve a 4.9% gain on GPT-4V and 3.4% on Gemini-1.5-flash. For LLaVA-v1.6-7B/13B, the improvements are 2.6% and 2.0%, and for LLaVA-OneVision-7B, we observe a 1.4% gain—outperforming SoM (Yang et al., 2023a) on the same model. Notably, our method boosts InternVL2.0-8B by 4.3%, significantly surpassing chain-of-thought approaches like CCoT (0.9%) and SKETCHPAD (0.8%), which struggle to improve strong baselines.

Across sub-tasks, our method shows marked gains on the most challenging ones: a 8.6% im-

Table 2: Accuracy scores on the MM-Vet and the validation set of MMMU.

Method	MM-Vet	MMMU
LLaVA-v1.6-7B	47.5	36.9
LLaVA-v1.6-7B + Ours	**48.5**(+1.0)	**37.4**(+0.5)
LLaVA-OneVision-7B	57.5	48.8
LLaVA-OneVision-7B + Ours	**60.2**(+2.7)	**51.0**(+2.2)
Gemini-1.5-flash	62.2	56.1
Gemini-1.5-flash + Ours	**64.9**(+2.7)	**60.9**(+4.8)
InternVL2.0-8B	60.0	51.8
InternVL2.0-8B + UnAC	**63.3**(+3.3)	**54.7**(+2.9)
GPT4-V	67.2	57.2
GPT4-V + SoM (Yang et al., 2023a)	66.0(−1.2)	57.2
GPT4-V + CCoT	67.7(+0.5)	58.7 (+1.5)
GPT4-V + SKETCHPAD	69.3(+2.1)	59.7(+2.5)
GPT4-V + Ours	**70.3**(+3.1)	**60.7**(+3.5)

provement on Geometry Problem Solving (GPS) with GPT-4V and 4.1% on TQA with Gemini. These tasks require complex, multi-step reasoning, which benefits from better visual abstraction and our self-checking scheme. For simpler tasks like VQA and FQA, the gains confirm the effectiveness of our adaptive visual prompting. Overall, the consistent improvements demonstrate that UnAC is a model-agnostic prompting strategy. However, stronger models like GPT-4V and InternVL benefit more, as the effectiveness of both visual prompting and self-checking depends on the model's reasoning capability—further discussed in Sec. 4.3.

Complicated VQA. In Table 2, we show the results on the MM-Vet (Yu et al., 2023) and MMMU (Yue et al., 2023). In these two datasets, the questions are more generalized with a relatively simple reasoning process. Our method still makes improvements on all models. We make an improvement of 3.1% on GPT-4V with our method and make the largest increase of 4.8% on Gemini-1.5-flash on MMMU. Compared to the chain-of-thought methods, UnAC performs better. Also, it indicates the necessity of self-checking that applying SoM (Yang et al., 2023a) on GPT-4V is harmful to answer the complicated question. For LLaVA-OneVision-7B, we achieve the improvements of 2.7% on MM-Vet.

The gap between the increase on Gemini/GPT4-V and the increase of LLaVA-v1.6-7B is larger compared to that on MathVista. In these two datasets, they require more comprehensive vision-language capabilities and abundant knowledge reserve on various topics. Therefore, in those two datasets, understanding can be more important than abstracting and reasoning.

4.3 Analysis

Corrected error analysis. Comparing the original predictions of UnAC to the baseline GPT-4V model on MathVista and MM-Vet: we find that our methods correct 25.4% errors from the baseline while introducing 5.5% errors on the task of Mathematical reasoning in the visual context. For complicated VQA *i.e.* MM-Vet, UnAC corrects 20.1% errors from the baseline while introducing 6.2% errors. To further understand how UnAC corrects the errors, we annotate all the wrong predictions corrected by our method of baseline methods in the test set, and categorize them into 4 classes: (1) Misunderstanding: The error is after introducing the prompts, the LMMs misunderstand the image which is correct in the baseline method; (2) Context loss: After introducing our method, it causes the missing of some information from the image which does not happen in the baseline answers; (3) Reasoning Error: The retrieved context is relevant, but the model still fails to reason through the context to arrive at the right answer. (4) Factual Error: There is at least one factual error when the model recites its own factual knowledge.

MathVista. As shown in Figure 3 (left), about 35% of corrected errors stem from image misunderstanding, and 23% from missing context. Together, roughly 58% of errors are rectified by our adaptive visual prompts, which help LMMs better perceive image details. In contrast, some errors occur despite correct perception—due to flawed reasoning or missing factual knowledge. While SoM's visual markers aid perception, they do little to improve reasoning, limiting its effectiveness in fixing such errors. Our self-checking scheme addresses this gap, accounting for 42% of corrections through step-by-step validation.

MM-Vet. In Figure 3 (right), 49% of corrected errors are due to misunderstanding, and 18% to lost visual context—both improved by UnAC. Reasoning and factual errors account for another 23% and 10%, respectively. Since MM-Vet emphasizes fine-grained image understanding over complex reasoning, visual prompts play a larger role. As a result, both UnAC and SoM mainly correct perception-related errors. Still, UnAC demonstrates a notable 33% improvement in reasoning-related cases, thanks to its self-checking mechanism.

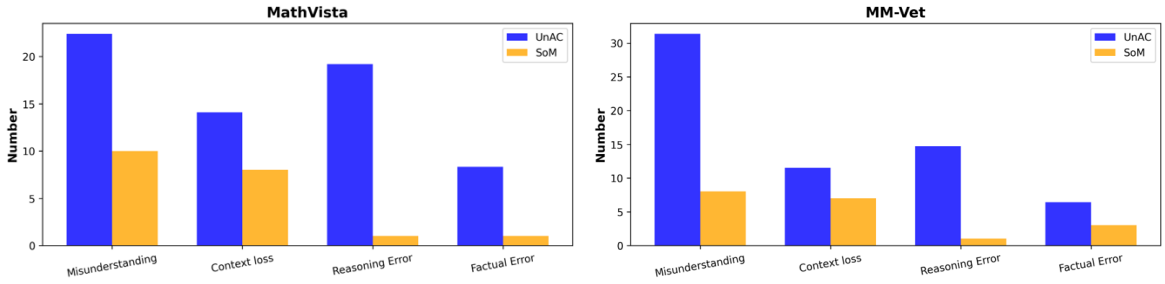

Figure 3: Corrected error analysis and comparison of UnAC and SoM. The left plot shows the comparison on MathVista while the right one represents that on MM-Vet: four classes of errors are corrected by the UnAC or SoM. The baseline model is GPT-4V for both methods.

Table 3: Accuracy scores of employing different visual prompting strategies of UnAC. The results are tested on the *testmini* subset of MathVista (Lu et al., 2023) while the baseline model is Gemini-1.5-flash.

Method	ALL	FQA	GPS	MWP	TQA	VQA
Baseline	53.2	51.6	56.8	52.1	67.5	40.1
Segmentation only	54.2	53.2	57.8	53.0	68.4	39.6
OCR only	53.3	51.6	57.3	51.1	68.7	40.1
Segmentation + OCR	54.4	53.2	57.8	53.0	68.4	39.6
Adaptive visual prompts	**56.6**	**57.3**	**59.3**	**54.1**	**71.6**	**41.4**

Discussion. Compared to the first two classes and the last two classes, the number of errors removed by correctly understanding the image and capturing more useful contexts is more than that removed by the accurate reasoning process. It indicates that the reasoning step is still a bottleneck of how well UnAC can perform for tasks such as MathVista which requires more complex reasoning.

How the abstraction and self-checking affect the final answer? As we discussed in Sec 4.2, the improvements made by UnAC are influenced by the original capability of the baseline LMMs. Although it makes sense, we want to find out how it influences our method. We conduct experiments on changing the models which is used in abstracting, checking, and final reasoning mainly with LLaVA-v1.6-7B and GPT-4V. As shown in Figure 4 (left), we replace the LLaVA-v1.6-7B with GPT-4V on different roles in our prompting process. Comparing the first Four rows, the final conclusion performs much better when replacing LLaVA-v1.6-7B with GPT-4V for performing abstracting, and checking respectively. The best performance is contributed by using GPT-4V to make both abstracting and checking among these three ablations. It indicates that better abstracting and checking are helpful for increasing the overall performance. However, comparing the four rows and the bottom row, although GPT-4V may provide the accurate answer to the question in the checking stage, the LLaVA-v1.6-7B still infers bad reasoning in the last step. Moreover, comparing the fourth row and fifth row, we can find that even LLaVA-v1.6-7B provides the bad prompts, GPT-4V still has the ability of self-correction in conclusion. Although improving the abstracting and checking can lead to better performance, the reasoning abilities of LMMs are still the bottleneck of how well UnAC can perform in solving complicated questions.

Why do visual prompts need to be adaptive? In this ablation, we want to show the effect of making the visual prompts adaptive. As shown in Table 3, we conduct experiments on applying different types of visual prompts. Comparing the first two lines, the improvements when employing the segmentation or OCR only are very limited. Although partitions can help the LMMs to focus on a certain part of the image, they also increase the risk of focusing on the wrong regions on the image. Since the whole picture has been overlayed everywhere, it may confuse the attention of LLMs. Adding boxes on the image to let LMMs focus on certain parts, it also increase the risk of incorrect regions which are useless for the question answering. Moreover, for some tasks, markers of segmentation or boxes from OCR is not helpful such as solving a geome-

Table 4: Accuracy scores using GPT-4V on the *testmini* subset of MathVista (Lu et al., 2023) under different checking. For the global checking, we use a simple prompt of 'Please check your answer if there are any errors.'

Method	ALL	FQA	GPS	MWP	TQA	VQA
w/o Checking	52.7	55.5	53.4	49.2	65.8	37.7
Global Checking	53.4	55.5	53.4	49.2	65.0	42.7
Gradual Checking	**57.6**	47.3	61.1	55.2	69.7	48.9

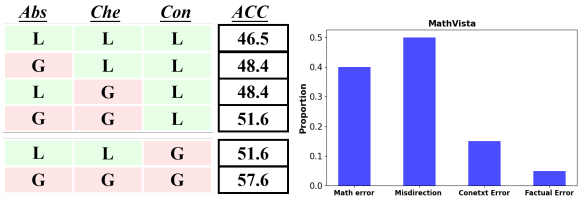

Figure 4: **Left**: The overall accuracy of changing different part of UnAC on the textmini dataset of MethVista (Lu et al., 2023). L means LLaVA-v1.6-7B and G means GPT-4V. *Abs*, *Che* and *Con* represent the abstracting, checking and conclusion stages respectively. **Right**: The error analysis on MethVista with Gemini-1.5-flash using global checking.

try problem or understanding a function plot. Both prompts can not provide much useful information.

Global checking or gradual checking. To prove that LMMs can not perform the global checking in an effective way like LLMs (Ling et al., 2024), we conduct experiments on comparing the performance of global checking prompting with the proposed one-step checking method. As shown in Table 4, compared to UnAC without checking, the performance of using the global checking shows very limited improvement overall. Although it increase the accuracy of the textbook question answering and visual question answering tasks, it makes the qualities on tasks of math word problem and geometry problem solving worse. We also conduct experiments on analyzing the errors made by the global checking. As shown in Figure. 4 (right), we define another set of errors which are related to the reasoning process only. The classes of errors are (1) Math error: The additional mathematical errors like computation and mathematical inference; (2) Misdirection: Leading to focusing the wrong regions of the images. (3) Context error: Incorrectly understanding the images or solutions in the previous steps. Misdirection and Math errors are the most frequent errors occurring which have 50% and 39%. It indicates that the global checking easily makes the reasoning process into the wrong direction due to the limitation of the reasoning ability of LMMs.

5 Limitations

Nevertheless, visual prompts are neither necessary nor possible to work in all scenarios. For instance, when facing highly abstract problems like geometry problem solving, the understanding of the image mostly depends on the original capability or the trained dataset of the LMMs since even a simple shape like a heptagon might be misidentified. How to effectively develop visual prompts for such problems is still a challenging topic and that's one of the future works we will target on.

6 Conclusion

In this paper, we propose a novel multimodal prompting method, namely UnAC (Understanding, Abstracting, and Checking), to synergize reasoning for complicated problems in visual context of LMMs. UnAC consists of an adaptive visual prompting building, the prompts of image abstraction and a gradual checking scheme. Suffecient experiments show the effectiveness of UnAC on improving the ability of complicated multimodal reasoning.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, and 1 others. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind

Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, and 1 others. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, and Hengshuang Zhao. 2022. Focalclick: Towards practical interactive image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1300–1309.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. 2024. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. 2024. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *arXiv preprint arXiv:2406.09403*.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, and 1 others. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and 1 others. 2024. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2024. Deductive verification of chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 36.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.

Ning Miao, Yee Whye Teh, and Tom Rainforth. 2023. Selfcheck: Using llms to zero-shot check their own step-by-step reasoning. *arXiv preprint arXiv:2308.00436*.

Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. 2023. Compositional chain-of-thought prompting for large multimodal models. *arXiv preprint arXiv:2311.17076*.

Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. 2024. Compositional chain-of-thought prompting for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14420–14431.

Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. 2023. What does clip know about a red circle? visual prompt engineering for vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11987–11997.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. 2023. Seggpt: Segmenting everything in context. *arXiv preprint arXiv:2304.03284*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. 2023. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. 2023a. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023b. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1):1.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, and 1 others. 2023. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arXiv preprint arXiv:2311.16502*.

Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. 2023. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, and 1 others. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H Chi, Quoc V Le, and Denny Zhou. 2023. Take a step back: Evoking reasoning via abstraction in large language models. *arXiv preprint arXiv:2310.06117*.

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

Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. 2024. Segment everything everywhere all at once. *Advances in Neural Information Processing Systems*, 36.