# CogCoM: Train Large Vision-Language Models Diving into Details through Chain of Manipulations

**Anonymous Author(s)**
Affiliation
Address
email

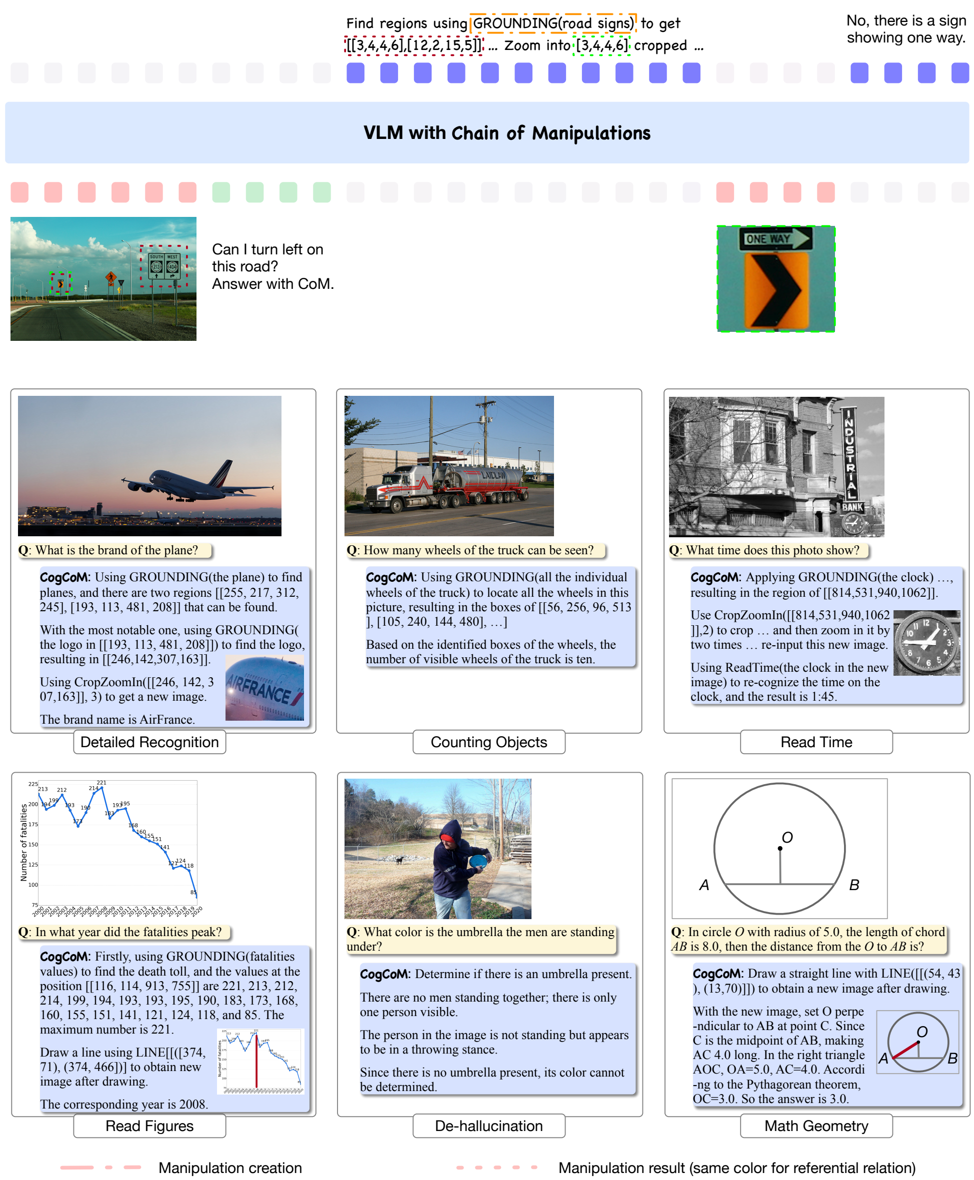


Figure 1: CogCoM solves various visual problems with Chain of Manipulations mechanism. Note that the CoM reasoning generates evidential and explainable steps, without involving external tools.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

## Abstract

Vision-Language Models (VLMs) have demonstrated their broad effectiveness thanks to extensive training in aligning visual instructions to responses. However, such training of conclusive alignment leads models to ignore essential visual reasoning, further resulting in failures in meticulous visual problems and unfaithful responses. Drawing inspiration from human cognition in solving visual problems (*e.g.*, *marking*, *zoom in*), this paper introduces **Chain of Manipulations**, a mechanism that enables VLMs to solve problems step-by-step with evidence. After training, models can solve various visual problems by eliciting intrinsic manipulations (*e.g., grounding*, *zoom in*) with results (*e.g., boxes*, *image*) actively without involving external tools, while also allowing users to trace error causes. We study the roadmap to implement this mechanism, including (1) a flexible design of manipulations upon extensive analysis, (2) an efficient automated data generation pipeline, (3) a compatible VLM architecture capable of multi-turn multi-image, and (4) a model training process for versatile capabilities. With the design, we also manually annotate 6K high-quality samples for the challenging graphical mathematical problems. Our trained model, **CogCoM**, equipped with this mechanism with 17B parameters achieves state-of-the-art performance across 9 benchmarks from 4 categories, demonstrating the effectiveness while preserving the interpretability. Our code, model weights, and collected data will be publicly available.


# 1 Introduction

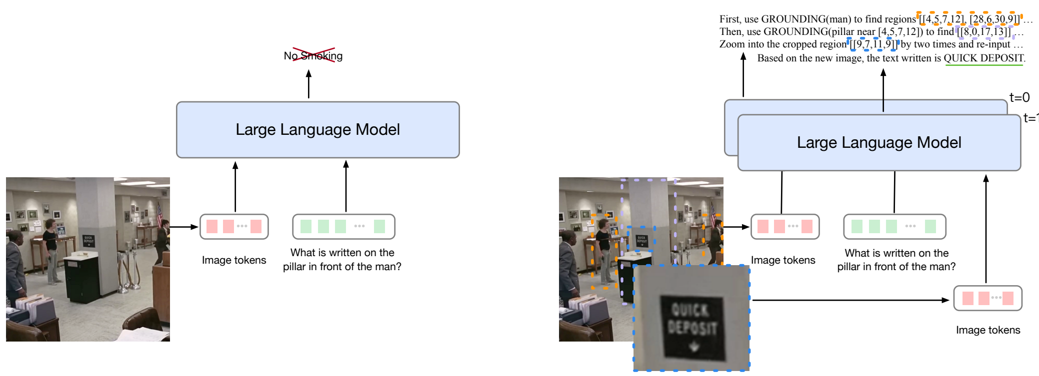


Figure 2: In comparison with existing VLMs, CogCoM performs the multiple steps of evidential reasoning with chain of manipulations (CoM) to achieve the faithful answer to visual scene.

Benefiting from the advantage of Large Language Models (LLMs) in broad world knowledge, large Vision Language Models (VLMs) (Alayrac et al., 2022; Wang et al., 2023b) that are further trained to understand visual inputs have demonstrated viabilities on broad multimodal scenarios, such as visual question answering (Liu et al., 2023b), visual grounding (Peng et al., 2023), optical character recognition (Zhang et al., 2023b). The research employing VLMs as foundation models (Bai et al., 2023; Sun et al., 2023b; Wang et al., 2023b) usually involves two main stages of training, where the first stage develops intrinsic visual understanding ability through exposure to massive image-caption pairs, and the second stage endows the models with problem-solving capabilities through the instruction tuning.

However, existing tuning methods train models to respond to instructions with conclusive language responses upon visual inputs, which leads models to ignore the essential intermediate visual reasoning and further results in failures in meticulous visual problems, unfaithful responses, and even hallucinations. For example in the left subplot of Figure 2, we test the top-performing model CogVLM (Wang et al., 2023b) about the details in the image (*i.e., texts written on a pillar*), and it directly responds an incorrect answer (*i.e., NO SMOKING*), most likely from bias to visual or linguistic priors (*i.e., typical scenes with a pillar in office*). The absence of the essential reasoning on the visual scene may lead to a rash response (Hwang et al., 2023).

Humans solve problems regarding visual details by marking or processing the given images for convenience and rigor, which we refer to as manipulations. For example, we find targets by sequentially locating references, and concentrate on subtle details by zooming into a corresponding region. Most of VLMs have developed numerous intrinsic capabilities (*e.g.,* grounding boxes, recognizing texts) during the first stage of training. By further imitating the fundamental human behaviours (*e.g.,* cropping, zoom in), models have the potential to perform this cognitive reasoning process. Three major obstacles in eliciting VLMs with such reasoning are (1) flexible definitions of manipulations covering most visual problems, (2) an efficient data collection pipeline capable of producing abundant training data, and (3) a multi-turn multi-image VLM structure compatible with existing models.

Inspired by the human cognition in solving visual problems, we introduce **Chain of Manipulations (CoM)**, a mechanism that enables VLMs to solve problems step-by-step with evidence, with each step potentially involving a manipulation on the visual input and its corresponding result, both generated by the model to facilitate the success and fidelity. This paper studies a complete roadmap with manipulations design, data collection, model architecture and training process for training general VLMs with this mechanism. We first formally design 6 basic manipulations upon the pilot experiments, which are capable of handling diverse visual problems. Next, we propose a cascading data generation pipeline based on reliable large language models (*e.g.,* LLMs, the linguistic annotators) and visual foundational models (*e.g.,* VFMs, the visual annotators), which can automatically produce abundant error-free training data. We collect 70K CoM samples with this pipeline. We then devise a multi-turn multi-image model architecture compatible with typical VLMs structures. Based on a data recipe incorporating the curated corpus, we finally train a general VLM equipped with CoM reasoning mechanism, named CogCoM, which possesses capabilities of chat, captioning, grounding and reasoning. Additionally, benefiting from the expressive capability of the proposed mechanism, we further manually annotated 6K high-quality samples of graphical mathematical problems, each accompanied by a CoM reasoning process, to advance the research of VLMs in solving challenging mathematical problems.

We conduct extensive experiments on 9 benchmarks from 4 categories, including TextVQA (Singh et al., 2019), ST-VQA (Biten et al., 2019), TallyVQA (Acharya et al., 2019), and GQA Hudson & Manning (2019) for detailed visual question answering, RefCOCO (Yu et al., 2016), RefCOCO+(Yu et al., 2016), and RefCOCOg (Mao et al., 2016) for visual grounding, POPE (Li et al., 2023c) for hallucination validation, and MM-Vet (Yu et al., 2023b) for general multimodal ability. Our model achieves up to 9.0 and 1.09 accuracy improvement on the detailed VQA and grounding benchmarks, respectively, and the superior performance on the general multimodal benchmark. The results demonstrate the effectiveness of the mechanism while maintaining the interpretability of outputs.

## 2 Terminology

We first conduct pilot experiments to investigate the possible manipulations capable of handling diverse visual problems.

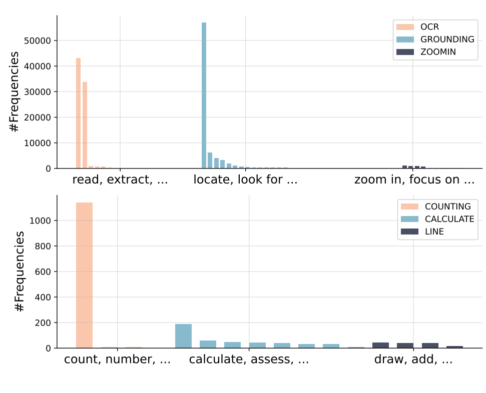


Figure 3: Distribution of the generated 465 actions base on GPT-4, mapped into 6 manipulations.

Specifically, given a question about an image, we prompt the advanced large language model, GPT-4, to generate solving steps by optionally utilizing possible actions on the image that facilitate problem-solving. We conduct this experiment on 170K questions from TextVQA, a dataset requiring detailed reasoning and recognition on images. To ensure the stability, we manually write 4 demonstrations as priors, The detailed statistics are available at Appendix C.3.

We utilize the StanfordCoreNLP toolkit to extract verb phrases referring to the actions, and the distribution of frequencies is shown in Figure 3. Through result analysis, we find that most of the actions can be mapped to 6 fundamental manipulations on images: *OCR*, *Grounding*, *CropZoomIn*, *Counting*, *Calculate*, and *Line*.

Based on the observation, we formally predefine a set of 6 manipulations, which can either be developed from pre-training or be learned from fine-tuning with the imitation to human behaviors: $\mathcal{M} \subseteq \{\textit{OCR}(tgt) \rightarrow txt,\ \textit{Grounding}(tgt) \rightarrow bbx,\ \textit{Counting}(tgt) \rightarrow num,\ \textit{Calculate}(tgt) \rightarrow num,\ \textit{CropZoomIn}(bbx, x) \rightarrow img,\ \textit{Line}(pts) \rightarrow img\}$, where the parameters or results $tgt, txt, bbx, num, x, img, pts$ refer to the bounding boxes, zoom ratio, image, target description, numbers, texts, and points, respectively. In addition to the predefined manipulations, we also allow trained models to create new manipulations during inference to facilitate problem-solving. We empirically find that more complicated goals can be derived from these fundamental manipulations.

We then define the **standard CoM data structure** to streamline the subsequent data construction and validation process. Given a question $Q$ about an initial input image $I_0$, a VLM equipped with chain of manipulations mechanism solves the problem to achieve final answer as $\textit{VLM}_{\varsigma}(A, C|I_0, Q)$, where $\varsigma$ refers to the reasoning chain with evidence,

$$\begin{aligned} \varsigma &= (step_1, step_2, ...) \\ step_i &= (f_i, c_i), \quad f_i \in \mathcal{M} \end{aligned} \tag{1}$$

where $C = (c_i, c_2, ..., c_{|C|})$ refers to the free-form textual descriptions incorporating manipulation names $f_i$ and corresponding results from utilizing $f_i$. This definition explicitly declares the symbolic execution process, while also being compatible with linguistic reasoning steps. Based on this definition, we can clearly construct standard CoM samples that incorporating the manipulation executions and linguistic steps with evidence. After the data construction, we can utilize a simple method to convert the standard CoM samples to the **compatible VQA samples**.

## 3 Data Collection

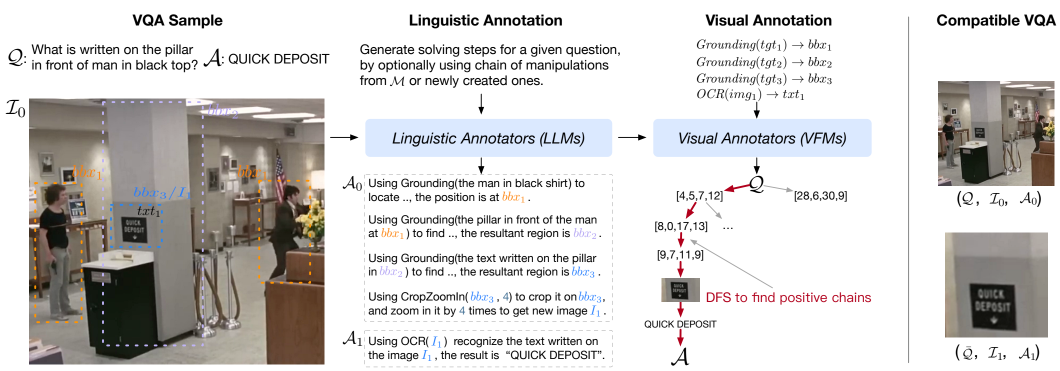


Figure 4: A cascading data generation pipeline that automatically produces standard CoM samples. Given an original VQA sample, the linguistic annotator (LLMs) taught with usage of manipulations (prompt) is first asked to provide solving steps for the question $\mathcal{Q}$, and the visual foundational models (VFMs) are then engaged to replace the manipulations results, followed by a final traversal on the tree branched by the possible manipulation results to find positive paths terminating to the answer $\mathcal{A}$.

In this section, we first introduces the automated data generation pipeline (illustrated in Figure 4), that employs reliable LLMs as linguistic annotators and VFMs as the visual annotators to produce error-free CoM samples upon prevalent VQA corpus, and then present the manual annotation of high-quality CoM samples for the challenging graphical mathematical problems.

### 3.1 Automated Data Generation

Given a general corpus $\mathcal{D} = \{(I, Q, A)\}$ consisting of triplet samples of images with corresponding visual question-answer pairs, our automated data generation pipeline consists of a linguistic annotator and several visual annotators according to the manipulations. For a question $Q$ in each sample, we first engage the linguistic annotator to generate manipulations-assisted solving steps with the CoM format $(f_i, c_i)$, where the corresponding results of the instantiated manipulation executions are set with variables as placeholders. In this paper, we adopt GPT-4 (OpenAI, 2023a), a large language model with reliable language understanding and generation abilities as the linguistic annotator. We design a comprehensive prompt including the task requirements, usage of manipulations, and output data format, and further manually annotate 5 demonstrations for a stable generation. The detailed implementations are available at Appendix C.4.

We then employ essential visual annotators to supply the results of manipulations requested in the solving steps by exactly performing the corresponding manipulations. By empirically analyzing the manipulations from both predefined set and newly created ones (refers to Appendix C.3 for a detailed statistics), we reveal the *Grounding* and *OCR* are two fundamental manipulations, and most of the others can be consequently derived (*e.g., CropZoomIn* along a region of box, *Counting* upon recognized boxes, and *Calculate* for the recognized formula). Therefore, we employ two visual foundational models, GroundingDINO (Liu et al., 2023c) and PaddleOCR (Du et al., 2020), and develop the implementations of these manipulations[1]. The execution of the manipulations will transform the sequential reasoning steps into a **tree** $\mathcal{T}$, as the input of current manipulation $f_1(x_a)$ may rely on one of the multiple results of previous manipulation $f_2 \rightarrow (x_b, x_c)$, *i.e.*, $x_a$ rely on $x_b$ (*e.g.,* step 2 for finding pillars in Figure 5). We then perform a traversal on each produced tree with Depth First Search (DFS) to find all positive paths $\{\mathcal{P}_i | \mathcal{P}_i \in \mathcal{T}, i = 1, 2, ...\}$ that can terminate with the final answer $A$ from the result of the last manipulation. Based on this method, the generated CoM samples with positive paths are guaranteed to be error-free. We implement this pipeline on 3 existing datasets that require detailed recognition or objects counting, TextVQA (Singh et al., 2019), ST-VQA (Biten et al., 2019), and TDIUC (Shrestha et al., 2019), to build 70K CoM samples [2]. The designed prompt, a generated example with linguistic and visual results, and detailed algorithm illustration are available at AppendixC.1.

### 3.2 Human Annotation

The analysis from Fig.1 of AlphaGeometry (Trinh et al., 2024) shows that outputting auxiliary lines in linguistic reasoning process helps LLMs to solve complex geometry problems. Benefiting from the expressive capability of CoM structure, we have also manually annotated high-quality CoM samples for the graphical mathematical problems to facilitate VLMs in solving this challenging scenario. Similar to the automated pipeline, we engage 10 human experts as the linguistic annotators and visual annotators, where each expert is asked to annotate the linguistic solving steps and the use of manipulations, as well as the results of manipulations on images. We perform this annotation on the MathVista (Lu et al., 2023) and ChartQA (Masry et al., 2022), which include geometric and chart math problems, resulting in the collection of 6K high-quality CoM math samples.

Finally, we adapt the CoM samples to be compatible with VQA-style training samples. For each CoM sample including $n$ images from manipulations outputs $(I_0, Q, C_0, I_1, C_1, ..., I_n, A)$, we convert it into a multi-turn VQA sample segmented by the images $[(I_0, Q, C_0), (I_1, \bar{Q}, C_1), ..., (I_n, \bar{Q}, A)]$, where $C_i$ represents the intermediate steps between $I_i$ and $I_{i+1}$, and $\bar{Q}$ is a simple prompt asking model to answer question based on history. This transformation converts CoM samples into multi-turn VQA samples that are compatible with existing VLMs training data. The detailed statistics of the data generation are available at Appendix C.3.

## 4 Model Training

### 4.1 Architecture

We use the same model architecture as CogVLM (Wang et al., 2023b), a general VLM approach that involves four fundamental components: (1) a Visual Encoder, (2) an MLP Adapter, (3) an LLM Backbone, and (4) a Visual Expert Module, for a reliable multimodal understanding. Concretely, the pre-trained EVA2-CLIP-E (Sun et al., 2023a) with 4B parameters and Vicuna-7B-v1.5 (Chiang et al., 2023) are adopted as the visual encoder and LLM backbone, respectively. A two-layer MLP (SwiGLU (Shazeer, 2020)) is further engaged to map the output of the visual encoder into the linguistic space of the LLM backbone. The visual expert module adds the vision-specific weights into the attention layer and feed-forward layer of each block in the LLM backbone, resulting in a total of 6.5B additional parameters for the deep fusion of modalities.

[1] We simply implement the *CropZoomIn* referring to human behaviors with a local code interpreter.

[2] The success rate of GPT-4 to achieve the positive paths is 0.3555.

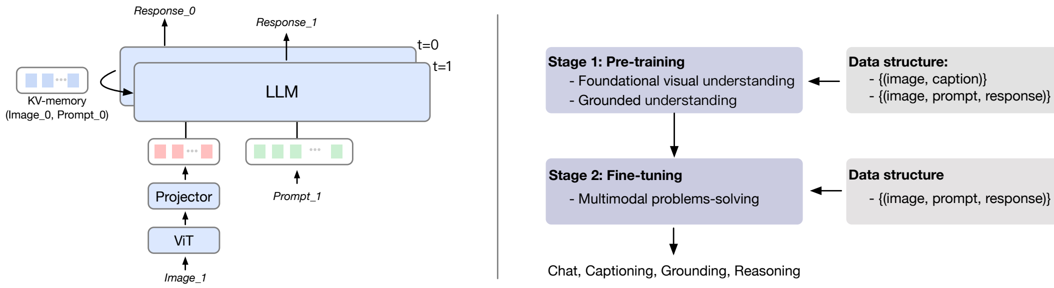


Figure 5: **Left**: A compatible VLM architecture capable of multi-turn multi-image understanding. **Right**: An effective training process to develop a general VLM with versatile capabilities.

Based on this general architecture, we develop a memory-based multi-turn multi-image VLM approach. Specifically, for a multi-turn VQA sample $[(I_t, Q_t, A_t)|t = 1, 2, ...]$, where $A_t$ refers to $C_t$ in CoM, we keep the accumulated KV memories of each layer in the LLM backbone throughout these turns. And at each turn $t$ in training and inference, we calculate the attention function $att$ as:

$$\begin{aligned} att(\boldsymbol{X}) &= softmax(\frac{\boldsymbol{Q}_t \boldsymbol{K}_t'^T}{\sqrt{d}})\boldsymbol{V}_t' \\ \boldsymbol{K}_t' &= \text{trunc}(\text{concat}(\boldsymbol{K}_0, \boldsymbol{K}_1, ..., \boldsymbol{K}_t)) \\ \boldsymbol{V}_t' &= \text{trunc}(\text{concat}(\boldsymbol{V}_0, \boldsymbol{V}_1, ..., \boldsymbol{V}_t)) \end{aligned} \tag{2}$$

where $\boldsymbol{Q}_t \in \mathbb{R}^{s \times d}$ is query representation of current layer, and the $\boldsymbol{K}_t', \boldsymbol{V}_t' \in \mathbb{R}^{(s \times t) \times d}$ refer to the concatenation of accumulated representations and will be further truncated if the sequence length $s \times t$ is greater than a predefined threshold. At $t > 0$, the new image $I_t$ will be cropped from $I_{t-1}$ and amplified with the Bicubic Interpolation (Keys, 1981).

## 4.2 Training

The proposed CogCoM-17B relies on two main stages of training, to develop the capabilities of general multimodal task-solving as well as the visual reasoning.

**First Stage Pre-Training** This stage consists of two ordinal sub-phases of training for foundational visual understanding and grounded generation. Following the pre-training of CogVLM (Wang et al., 2023b), we first train model on 1.5B image-text pairs cleaned from the LAION-2B (Schuhmann et al., 2022) and COYO-700M (Byeon et al., 2022) with 120,000 iterations and batch size of 8,192. We then train model on 40M grounded image-question-answer triples cleaned from LAION-115M (Li et al., 2023b) with 60,000 iterations and batch size of 1,024, where each noun phrase in the answer is followed by a list of coordinates $[[x_0, y_0, x_1, y_1], ...]$[3] referring the phrase to the grounded objects in the image. Both phases adopt the next token prediction objective, and train the 6.5B parameters of visual experts.

**Second Stage Alignment** This stage further trains the model to align with human preferences on solving practical visual problems. We fuse the produced CoM data with 3 types of corpus, including MultiInstruct (Xu et al., 2022), LLaVAR (Zhang et al., 2023b), and ShareGPT4V (Chen et al., 2023c), referring the abilities of instruction-following, texts-recognizing, and detailed-captioning. This fusion results in a total of 570K $(I, Q, A)$ samples, where the answer $A$ in CoM data consists of multiple turns. For the training data of CoM, we randomly prepend a lunching prompt[4] $P^{\mathcal{M}}$ to questions $Q = P^{\mathcal{M}} + Q$ asking models to optionally use manipulations for the adaption of explicitly eliciting. We empirically show that the model can effectively learn the evidential visual reasoning by ingesting this portion of CoM data. We train model with 14,000 iterations and a batch size of 160, where the learning rate reaches $10^{-5}$ after 280 steps of warm-up and then decays linearly. The parameters of 6.5B visual experts are trained with the objective of next token prediction. These two stages of training result in our standard version of CogCoM involving both chat and reasoning capabilities. More training details are available at Appendix D.2.

[3] $x_i, y_i \in [000, 999]$ refer to the normalized pixel coordinates.
[4] See Appendix D.1 for examples.

# 5 Experiment

To quantitatively validate the suitability and efficiency of the proposed method, we conduct experiments on 9 benchmarks corresponding to 4 categories of multimodal capabilities, as well as on a newly constructed testbed that includes the evidential reasoning paths with a keypoints-aware metric. Following previous works, we train two generalist versions of CogCoM for adapting to the different scenarios of Visual Question Answering and Visual Grounding, and evaluate the standard version with a qualitative analysis (Hwang et al., 2023). We also evaluate the time complexity.

- **Detailed Visual Question Answering.** This task involves models to perform detailed reasoning or recognition on images. We use 4 prominent benchmarks including, GQA (Hudson & Manning, 2019), TextVQA (Singh et al., 2019), ST-VQA (Biten et al., 2019), and TallyVQA (Acharya et al., 2019).
- **Visual Grounding.** Visual grounding evaluates the crucial abilities of VLMs on meticulous position understanding. We evaluate our model on 3 standard benchmarks, RefCOCO (Yu et al., 2016), RefCOCO+ (Yu et al., 2016), and RefCOCOg (Mao et al., 2016).
- **General Multimodal Capabilities & Hallucination.** We also evaluate on a general multimodal benchmark, MM-Vet (Yu et al., 2023b), and a hallucination detection benchmark POPE (Li et al., 2023c), to investigate the helpfulness of visual reasoning.

## 5.1 Experiments on Detailed VQA

VLMs have demonstrated the well-known superiority in visual scenes with salient content understanding. We evaluate the effectiveness of CogCoM on VQAs on detailed understanding, which typically require models to perform multiple actions (*find, read*) or multiple reasoning steps (*recognizing and then calculating*). Following previous studies (Wang et al., 2023b), we train our model obtained from the first-phase of stage-1 on a mixture of data, including an instruction corpus of MultiInstruct, 13 publicly available VQA datasets (only using training set), a newly created VQA dataset built through promoting GPT-4V (OpenAI, 2023b) for image-oriented question-answer generation, and the automatically generated 70K CoM corpus. This training results in a generalist VQA model incorporating CoM reasoning. For all existing VQA tasks, we directly prompt CogCoM with given questions and examine the correctness of outputted answers.

| Type | Model | GQA test-balanced | TallyVQA simple | TallyVQA complex | TextVQA test | ST-VQA test |
|---|---|---|---|---|---|---|
| Generalist | Flamingo (Alayrac et al., 2022) | - | - | - | 54.1 | - |
| | GIT (Wang et al., 2022a) | - | - | - | 59.8 | - |
| | GI2 (Wang et al., 2022a) | - | - | - | 67.3 | - |
| | BLIP-2 (Li et al., 2023b) | 44.7$^{\dagger}$ | - | - | - | 21.7 |
| | InstructBLIP (Dai et al., 2023) | 49.5$^{\dagger}$ | - | - | - | 50.7$^{\dagger}$ |
| | Qwen-VL (Bai et al., 2023) | 59.3 | - | - | 63.8 | - |
| | CogVLM (Wang et al., 2023b) | 65.2 | 79.8 | 68.0 | 69.7 | 61.0 |
| | **CogCoM** | **71.7** | **84.0** | **70.1** | **71.1** | **70.0** |
| Specialist SOTAs | | 72.1 (CFR) | 86.0 ( PaLI-X) | 75.6 (PaLI-X) | 71.4 (PaLI-X) | 86.0 (SMoLA) |

Table 1: Performance on Visual Question Answering benchmarks, where the results labeled with $^{\dagger}$ refer to the few-shot setting. CogCoM achieves SOTA across the board, and demonstrates the effectiveness on the visual reasoning and scene texts recognition benchmarks.

### 5.1.1 GQA, TextVQA, ST-VQA, TallyVQA

**Settings** GQA is a compositional VQA benchmark with diverse reasoning questions coming from semantic functional programs. TallyVQA is an objects counting benchmark with human-annotated complex counting questions involving challenging non-zero counterparts. TextVQA and ST-VQA are two texts understanding benchmarks requiring models to answer questions through textual cues on images. We use the official evaluation scripts for GQA and TallyVQA, which calculate the accuracy score by the Exact Matching (EM) between model predictions and answers. For TextVQA and ST-VQA, we submit our model predictions to the official online websites for calculating the accuracy with VQA Score metric (Antol et al., 2015).

**Results** As the results shown in Table 2, CogCoM achieves the state-of-the-art performance in comparison with all generalist models, and achieves significant improvements over the baseline model. Specifically, compared to the baseline model, our model achieves up to 5.97 and 9.0 percentage points improvement on the benchmarks that requires complex reasoning and detailed recognition, respectively. On GQA and TextVQA, CogCoM also obtains comparable results with the large-scale specialist SOTAs. This result demonstrates the effectiveness of the proposed approach in solving details recognition problem.

### 5.1.2 Experiments for Reasoning Accuracy and Time Complexity

Due to the lack of resource, we build CoM-test, a benchmark with evidential reasoning chains on the TextVQA test set based on the proposed data generation pipeline, and also introduce a keypoints-aware metric to validate the correctness of reasoning paths (see Appendix C.3 for detailed statistics). We also evaluate the time complexity for model generation on a held-out benchmark, MM-Vet.

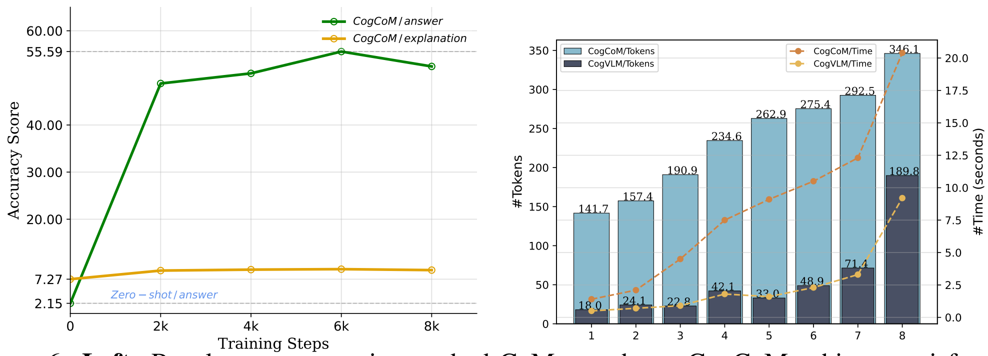


Figure 6: **Left**: Results on a reasoning testbed CoM-test shows CogCoM achieves satisfactory performance with only 70K training data and 2K steps. **Right**: Results on MM-Vet shows that CogCoM produces comprehensive reasoning content without incurring excessive time overhead.

**Reasoning Accuracy** To validate the correctness of execution and results of manipulations in reasoning paths, we introduce a keypoints-aware evaluation metric that concentrates on these contents and their order. Concretely, given a predicted chain-answer pair $(C', A')$ and the ground truth pair $(C, A)$, we first extract the keypoints (*i.e.,* the name, parameters, and results of manipulations) in $A', A$ to form two lists, and then discretize these two lists into $K'$ and $K$ based on a bag-of-words composed of all keypoints. Then, we calculate the normalized Levenshtein Distance $s_K = Levenshtein(K', K)/N$ as the manipulation score. We also compute the BLEU (Papineni et al., 2002) score $s_C = \text{BLEU}(C', C)$ as the paragraph score. Finally, a weighted average of these two scores serves as the ultimate reasoning score s $acc = (0.6 \times s_K + 0.4 \times s_C)/2$.

We train our first-stage model only using the 70K automated CoM data without other supervision for qualitatively evaluate the effectiveness of chains, and the results are shown in the left subplot of Figure 6. We find that by training with the CoM chains, our model can swiftly achieve the satisfactory performance of 48.41 accuracy score with 2k training steps, and obtain the optimal result of 55.59 with 8K steps. Additionally, the explanation scores gradually improve along with the model performance, indicating that successful reasoning steps contribute to the achieving of final answer.

**Time Complexity** We also evaluate the time complexity and average length of tokens during model reasoning on a held-out test set, MM-Vet. Specifically, we run CogCoM and the baseline model on all 218 questions, and record the time overhead as well as the average number of outputted tokens (using the Vicuna-7B-v1.5 tokenizer). We divide the 218 samples into 8 intervals based on the time expenditure for each sample and calculate the average values of the time complexity and the number of tokens for each interval, with the results presented in the right subplot of Figure 6.

From the results we find that compared to baseline model, CogCoM produces information-intensive reasoning content (*e.g.,* detection boxes, auxiliary lines) without incurring infeasible time overhead. For example, without quantitive optimization, CogCoM outputs 262.9 informative tokens in approximately 9 seconds. With the advantages in long-context optimization techniques (Hooper et al., 2024), we believe that it is crucial for models to produce informative content and accurate responses.

### 5.2 Experiments on Visual Grounding

The task of visual grounding requires models to precisely provide the corresponding coordinates of regions in an image based on the given target description. Following the existing work (Wang et al., 2023b), we train our model obtained by the first stage on a mixture of datasets, including an instruction corpus MultiInstruct, a high-quality grounded VQA corpus introduced in CogVLM, and the 70K CoM data. This training results in a generalist grounding model that is excelling at visual grounding while capable of reasoning. For all benchmarks, we prompt CogOM in a chat manner to ask the model to provide grounded coordinates, such as "*Where is* $\langle$*expr*$\rangle$ *answer in [x0,y0,x1,y1] format.*", where the $\langle$*expr*$\rangle$ refers to the target expression. We use the standard metric, that considers a prediction as correct when the intersection-over-union (IoU) between boxes is greater than 0.5.

| **Type** | **Model** | **RefCOCO** | | | **RefCOCO+** | | | **RefCOCOg** | |
|---|---|---|---|---|---|---|---|---|---|
| | | val | test-A | test-B | val | test-A | test-B | val | test |
| Generalist | OFA-L* (Wang et al., 2022b) | 79.96 | 83.67 | 76.39 | 68.29 | 76.00 | 61.75 | 67.57 | 67.58 |
| | Shikra-7B (Chen et al., 2023b) | 87.01 | 90.61 | 80.24 | 81.60 | 87.36 | 72.12 | 82.27 | 82.19 |
| | Shikra-13B (Chen et al., 2023b) | 87.83 | 91.11 | 81.81 | 82.89 | 87.79 | 74.41 | 82.64 | 83.16 |
| | Qwen-VL (Bai et al., 2023) | 89.36 | 92.26 | 85.34 | 83.12 | 88.25 | 77.21 | 85.58 | 85.48 |
| | CogVLM (Wang et al., 2023b) | **92.51** | 93.95 | 88.73 | 87.52 | 91.81 | 81.43 | **89.46** | 90.09 |
| | **CogCoM** | 92.34 | **94.57** | **89.15** | **88.19** | **92.80** | **82.08** | 89.32 | **90.45** |
| Specialist SOTAs | | 92.64 (UNINEXT) | 94.33 (UNINEXT) | 91.46 (UNINEXT) | 88.77 (ONE-PEACE) | 92.21 (ONE-PEACE) | 83.23 (ONE-PEACE) | 89.22 (ONE-PEACE) | 89.37 (UNINEXT-H) |

Table 2: Results on VG benchmarks, where the specialist SOTAs are quoted from (Bai et al., 2023).

**Results** As shown in Figure 2, CogCoM achieves the best performance in 6 out of all 8 sub-sets. Based on the training with a mixture of broad capabilities, this result indicates that our model exhibits a superior grounding abilities while offers potential to solve a variety of tasks.

### 5.3 Experiments on General Multimodal Evaluation and Hallucination Examination

We further examine the general multimodal capabilities, and the hallucination issue. We use the generalist VQA model and obtain model predictions by directly asking the original questions in benchmarks. We use the challenging adversarial version and official evaluation scripts for POPE.

| **Method** | **LLM** | **MM-Vet** | **POPE**$_{adv}$ |
|---|---|---|---|
| InstructBLIP (Dai et al., 2023) | Vicuna-13B | 25.6 | 77.3 |
| LLaVA (Liu et al., 2023b) | LLaMA2-7B | 28.1 | 66.3 |
| DreamLLM (Dong et al., 2023) | Vicuna-7B | 35.9 | 76.5 |
| LLaVA-1.5 (Liu et al., 2023a) | Vicuna-13B | 36.3 | 84.5 |
| CogVLM (Wang et al., 2023b) | Vicuna-7B | 45.5$^{\dagger}$ | 87.2 |
| **CogCoM** | Vicuna-7B | **46.1** | **87.8** |

Table 3: Evaluation results on the general and hallucination assessment benchmarks.

**Results** As shown in Table 3, we can see that CogCoM improves the performance by 0.6 points compared to the baseline model on MM-Vet, and achieves the superior performance on POPE which is in consistent with the baseline model. This result suggests that out model maintains superior reasoning capabilities while preserving effectiveness in general multimodal tasks, and simultaneously exhibits lower hallucination.

## 6 Conclusion

This paper studies the problems presented by the conclusive alignment training of VLMs, and proposes a mechanism, Chain of Manipulations (CoM), that enables VLMs to solve problems step-by-step by actively manipulating visual inputs as evidence. We realize this methodology by proposing (1) a flexible data structure, (2) an efficient data generation framework capable of producing abundant samples, (3) a memory-based architecture compatible with existing VLMs, and (4) a training process for versatile capabilities. We also annotate 6K graphical math samples with reasoning chains to facilitate the advancement of VLMs in solving mathematical problems. Experiments on 9 public benchmarks show that our trained 17B general VLM can produce informative reasoning content while achieving superior performance on diverse multimodal problems.

# References


Acharya, M., Kafle, K., and Kanan, C. Tallyqa: Answering complex counting questions. In *Proceedings of the AAAI conference on artificial intelligence*, 2019.

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 2022.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., and Parikh, D. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, 2015.

Awadalla, A., Gao, I., Gardner, J., Hessel, J., Hanafy, Y., Zhu, W., Marathe, K., Bitton, Y., Gadre, S., Sagawa, S., et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint*, 2023.

Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., and Zhou, J. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint*, 2023.

Biten, A. F., Tito, R., Mafla, A., Gomez, L., Rusinol, M., Valveny, E., Jawahar, C., and Karatzas, D. Scene text visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2019.

Byeon, M., Park, B., Kim, H., Lee, S., Baek, W., and Kim, S. Coyo-700m: Image-text pair dataset, 2022.

Changpinyo, S., Sharma, P., Ding, N., and Soricut, R. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Chen, D., Liu, J., Dai, W., and Wang, B. Visual instruction tuning with polite flamingo. *arXiv preprint arXiv:2307.01003*, 2023a.

Chen, K., Zhang, Z., Zeng, W., Zhang, R., Zhu, F., and Zhao, R. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint*, 2023b.

Chen, L., Li, J., Dong, X., Zhang, P., He, C., Wang, J., Zhao, F., and Lin, D. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*, 2023c.

Chen, X., Wang, X., Changpinyo, S., Piergiovanni, A., Padlewski, P., Salz, D., Goodman, S., Grycner, A., Mustafa, B., Beyer, L., et al. Pali: A jointly-scaled multilingual language-image model. In *The Eleventh International Conference on Learning Representations*, 2022.

Chen, X., Djolonga, J., Padlewski, P., Mustafa, B., Changpinyo, S., Wu, J., Ruiz, C. R., Goodman, S., Wang, X., Tay, Y., et al. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint*, 2023d.

Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023.

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

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

Dong, R., Han, C., Peng, Y., Qi, Z., Ge, Z., Yang, J., Zhao, L., Sun, J., Zhou, H., Wei, H., et al. Dreamllm: Synergistic multimodal comprehension and creation. *arXiv preprint arXiv:2309.11499*, 2023.

Du, Y., Li, C., Guo, R., Yin, X., Liu, W., Zhou, J., Bai, Y., Yu, Z., Yang, Y., Dang, Q., et al. Pp-ocr: A practical ultra lightweight ocr system. *arXiv preprint arXiv:2009.09941*, 2020.

Hong, W., Wang, W., Lv, Q., Xu, J., Yu, W., Ji, J., Wang, Y., Wang, Z., Dong, Y., Ding, M., et al. Cogagent: A visual language model for gui agents. *arXiv preprint arXiv:2312.08914*, 2023.

Hooper, C., Kim, S., Mohammadzadeh, H., Mahoney, M. W., Shao, Y. S., Keutzer, K., and Gholami, A. Kvquant: Towards 10 million context length llm inference with kv cache quantization. *arXiv preprint arXiv:2401.18079*, 2024.

Huang, K.-H., Zhou, M., Chan, H. P., Fung, Y. R., Wang, Z., Zhang, L., Chang, S.-F., and Ji, H. Do lvlms understand charts? analyzing and correcting factual errors in chart captioning. *arXiv preprint arXiv:2312.10160*, 2023a.

Huang, S., Dong, L., Wang, W., Hao, Y., Singhal, S., Ma, S., Lv, T., Cui, L., Mohammed, O. K., Liu, Q., et al. Language is not all you need: Aligning perception with language models. *arXiv preprint*, 2023b.

Hudson, D. A. and Manning, C. D. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

Hwang, A., Head, A., and Callison-Burch, C. Grounded intuition of gpt-vision's abilities with scientific images. *arXiv preprint*, 2023.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, 2021.

Keys, R. Cubic convolution interpolation for digital image processing. *IEEE transactions on acoustics, speech, and signal processing*, 1981.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 2017.

Li, B., Zhang, Y., Chen, L., Wang, J., Yang, J., and Liu, Z. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*, 2023a.

Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint*, 2023b.

Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W. X., and Wen, J.-R. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023c.

Li, Y., Zhang, C., Yu, G., Wang, Z., Fu, B., Lin, G., Shen, C., Chen, L., and Wei, Y. Stablellava: Enhanced visual instruction tuning with synthesized image-dialogue data. *arXiv preprint arXiv:2308.10253*, 2023d.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 2014.

Liu, H., Li, C., Li, Y., and Lee, Y. J. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023a.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. *arXiv preprint*, 2023b.

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023c.

Lu, P., Bansal, H., Xia, T., Liu, J., Li, C., Hajishirzi, H., Cheng, H., Chang, K.-W., Galley, M., and Gao, J. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A. L., and Murphy, K. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

Masry, A., Long, D. X., Tan, J. Q., Joty, S., and Hoque, E. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022.

OpenAI, R. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2023a.

OpenAI, R. Gpt-4v(ision) system card. *Citekey: gptvision.*, 2023b.

Ordonez, V., Kulkarni, G., and Berg, T. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 2011.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002.

Peng, Z., Wang, W., Dong, L., Hao, Y., Huang, S., Ma, S., and Wei, F. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint*, 2023.

Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., and Komatsuzaki, A. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 2022.

Sharma, P., Ding, N., Goodman, S., and Soricut, R. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018.

Shazeer, N. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Shrestha, R., Kafle, K., and Kanan, C. Answer them all! toward universal visual question answering models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., and Rohrbach, M. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

Sun, Q., Fang, Y., Wu, L., Wang, X., and Cao, Y. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023a.

Sun, Q., Yu, Q., Cui, Y., Zhang, F., Zhang, X., Wang, Y., Gao, H., Liu, J., Huang, T., and Wang, X. Generative pretraining in multimodality. *arXiv preprint*, 2023b.

Trinh, T. H., Wu, Y., Le, Q. V., He, H., and Luong, T. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

Wang, J., Yang, Z., Hu, X., Li, L., Lin, K., Gan, Z., Liu, Z., Liu, C., and Wang, L. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022a.

Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., Ma, J., Zhou, C., Zhou, J., and Yang, H. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. 2022b.

Wang, W., Chen, Z., Chen, X., Wu, J., Zhu, X., Zeng, G., Luo, P., Lu, T., Zhou, J., Qiao, Y., et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *arXiv preprint*, 2023a.

Wang, W., Lv, Q., Yu, W., Hong, W., Qi, J., Wang, Y., Ji, J., Yang, Z., Zhao, L., Song, X., et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint*, 2023b.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022c.

Wu, P. and Xie, S. V*: Guided visual search as a core mechanism in multimodal llms. *arXiv preprint arXiv:2312.14135*, 2023.

Xu, Z., Shen, Y., and Huang, L. Multiinstruct: Improving multi-modal zero-shot learning via instruction tuning. *arXiv preprint arXiv:2212.10773*, 2022.

Yin, S., Fu, C., Zhao, S., Xu, T., Wang, H., Sui, D., Shen, Y., Li, K., Sun, X., and Chen, E. Woodpecker: Hallucination correction for multimodal large language models. *arXiv preprint arXiv:2310.16045*, 2023.

Yu, L., Poirson, P., Yang, S., Berg, A. C., and Berg, T. L. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, 2016.

Yu, Q., Li, J., Wei, L., Pang, L., Ye, W., Qin, B., Tang, S., Tian, Q., and Zhuang, Y. Hallucidoctor: Mitigating hallucinatory toxicity in visual instruction data. *arXiv preprint arXiv:2311.13614*, 2023a.

Yu, W., Yang, Z., Li, L., Wang, J., Lin, K., Liu, Z., Wang, X., and Wang, L. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023b.

Zeng, Y., Zhang, H., Zheng, J., Xia, J., Wei, G., Wei, Y., Zhang, Y., and Kong, T. What matters in training a gpt4-style language model with multimodal inputs? *arXiv preprint arXiv:2307.02469*, 2023.

Zhang, S., Sun, P., Chen, S., Xiao, M., Shao, W., Zhang, W., Chen, K., and Luo, P. Gpt4roi: Instruction tuning large language model on region-of-interest. *arXiv preprint arXiv:2307.03601*, 2023a.

Zhang, Y., Zhang, R., Gu, J., Zhou, Y., Lipka, N., Yang, D., and Sun, T. Llavar: Enhanced visual instruction tuning for text-rich image understanding. *arXiv preprint*, 2023b.

# A Related Works

## A.1 Large Vision-Langauge Models as Foundations

Most of LVLMs rely on the training on publicly available image-caption pairs, including ALIGN (Jia et al., 2021), MSCOCO (Lin et al., 2014), VG Krishna et al. (2017), CC3M Sharma et al. (2018), CC12M (Changpinyo et al., 2021), SBU (Ordonez et al., 2011), LAION2B (Schuhmann et al., 2022), LAION400M Schuhmann et al. (2021). Starting from Flamingo (Alayrac et al., 2022), a series of LVLMs have focused on training the adaptation layers to align the visual representation to the frozen LLMs on a mixture of image-text pairs with the above corpus, including BLIP2 Li et al. (2023b), KOSMOS Huang et al. (2023b), and OpenFlamingo (Awadalla et al., 2023). Inspired by success of instruction tuning in LLMs (Wang et al., 2022c), a line of works have devoted efforts to build vision-oriented instruction-answer pairs through GPT4 and train models for imitation, such as LLAVA (Liu et al., 2023b), Otter (Li et al., 2023a), VisionLLM (Wang et al., 2023a), MultiInstruct (Xu et al., 2022), Lynx (Zeng et al., 2023), InstructBLIP (Dai et al.), CleverFlamingo (Chen et al., 2023a) and StableLLaVA (Li et al., 2023d). Recently, researchers have proven the efficiency of developing LVLMs with two stages of training, the first stage of abundant pretraining on image-caption pairs and the second stage of alignment on image-question-answer triples, such as PALI (Chen et al., 2022), PaLI-X (Chen et al., 2023d), Qwen-VL (Bai et al., 2023), and CogVLM Wang et al. (2023b).

## A.2 Large Vision-Language Models with Reasoning

To further enhance the ability of LVLMs in solving high-level visual problems, research focusing on various aspects of reasoning is attracting broad attention. We simply divide existing studies into tree broad categories. The first line of research focus on enhance train models with a mastery of cross-modal grounded reasoning, where grounded instruction-following supervision is build through public visual grounding dataset or GPT4-V for training, including KOSMOS-2 (Peng et al., 2023), Shikra (Chen et al., 2023b), and GPT4ROI (Zhang et al., 2023a). The second aspect of efforts have been devoted into promoting models to understand artificial visual scenes, such as figures, charts, and receipts. These studies includes CogAgent (Hong et al., 2023) and CHARTVE (Huang et al., 2023a). Some other studies address the crucial problem of hallucination in LVLMs with counterfactual or interpretable reasoning (Yu et al., 2023a; Yin et al., 2023). V* (Wu & Xie, 2023) also contributes efforts to enhance the details recognition of VLMs based the LLM-guided searching process.

# B Limitation and Impact

Though we try to develop an accurate and robust framework that engages remarkable LLM to provide basic solving steps, adopts reliable visual tools to obtain visual contents, and then acquires feasible paths based on traversal, there are still limitations in our methodology that we hope to improve in the future. First, We find that the diversity of linguistic solving steps is insufficient, and the inaccuracy of visual tools (*e.g.,* the rough granularity of grounding boxes, OCR failures on slant letters) will lead to a large amount of negative paths (effectively utilizing these paths would beneficial). We suggest to promote these limitations with dedicate prompts and improved visual tools. Second, our current model re-input the manipulated images with a set of hard prompts, which may bring speed losses. This is expected to be improved by implementing the physical manipuations into the calculations in vector space. This work presents a general visual reasoning mechanism that alleviate the problems caused by existing conclusion-alignment training for VLMs, introduces a data production framework involving LLMs and visual tools as reliable annotators, and devises a memory-based compatible VLM architecture. We expect this work to bring three benefits to the community. First, the proposed visual reasoning mechanism may push the progress of VLMs in solving complex visual problems. Second, the introduced data production framework may be applied to widespread training scenarios to promote the development of current data-driven machine learning. Third, we hope that the memory-based architecture will be helpful for VLMs in multi-turn long contexts.

# C Details of Data Production

In this section, we further introduce the details of CoM data production, with the overall algorithm of a pseudo code, an example of the solving steps generation with LLM and corresponding guideline, an example of the reasoning chains completion with visual tools. We also list the details of data statistics for the synthesised training data as well as the evaluation data of CoM-test, followed by a limitation analysis for the current data production method.

## C.1 Algorithm for the Automate Data Generation Pipeline

We provide the pseudocode of the CoM synthesis algorithm to clearly explain the process of data generation, thereby facilitating understanding and reproduction 1.

**Algorithm 1** Synthesising Chain of Manipulations

1: **Define:** $\begin{cases} Manipulations: \{f_i : x \to y \mid f_i \in \mathcal{M}\} \\ Linguistic\ Annotator: \Psi_L \quad //We\ use\ GPT4\ in\ this\ work \\ Visual\ Annotator: \Psi_V \quad //We\ use\ PaddleOCR\ and\ GroundingDINO\ in\ this\ work \end{cases}$

2: **Input:** Image $I$, Question $Q$, Answer $A$

3: // Linguistic Annotation

4: Prompt $\Psi_L$ with guideline $P^L$ to generate reasoning steps:

$$\varsigma = \Psi_L(Q|P^L), \quad where \begin{cases} \varsigma = (steps_1, steps_2, ...) \\ steps_i = (f_i, desc_i) \end{cases} \tag{3}$$

5: Define tree $\mathcal{T}$

6: **for** $i = 1$ **to** $|\varsigma|$ **do**

7: Extract $x_i, y_i$ instantiated with $f_i$ in $step_i$

8: Extract referential boxes $B$ from $x_i$

9: **for** $b$ in $B$ **do**

10: Leverage $\Psi_V$ to acquire corresponding visual content $y'_i = \Psi(x_i|I, b)$, and apply $y_i$ to tree

$$\mathcal{T}.level[i].append(y_i) \tag{4}$$

11: **end for**

12: **end for**

13: Traverse $\mathcal{T}$ to obtain positive chains that leads to given answer with terminal return

$$[\varsigma_1, \varsigma_2, ...] = DFS(\mathcal{T}|A) \tag{5}$$

14: Return $[\varsigma_1, \varsigma_2, ...]$

## C.2 The CoM-test Benchmark and Evaluation Metric

To measure the correctness of CoM chains, we introduce a **keypoints-aware metric**. The intuition is that we care about the key elements including actions (*i.e.,* manipulation name), targets (*i.e.,* manipulation input), and visual contents (*i.e.,* manipulation returns) of each step in the path, as well as the logical execution order of manipulations. Given a pair of chain-answer annotation $(c, a)$ and corresponding model prediction $(c', a')$, we first sequentially extract the key elements from $c$ and $c'$ to construct two ordered lists, and then replace the elements in the lists with their fixed indices in a Bag-of-Elements $\mathcal{E} = c \cup c'$ to result in lists of $k$ and $k'$. We thus calculate the score as the normalized Levenshtein Distance $s_c = Levenshtein(k, k')/N$ between the two lists, where $N$ is the maximum length between $k$ and $k'$. We adopt this simple discretization strategy with low time complexity to concentrate on the key points as well as the solving order. We further consider the linguistic matching of paragraphs by calculating the BLEU (Papineni et al., 2002) score between two chains $s_p = \text{BLEU}(c, c')$, and the final sore is a weighted combination as $acc = (0.6 \times s_c + 0.4 \times s_p)/2$.

### C.3 Data Statistics

We develop a strategy to extract predicate phrases based constituency parsing with StandordCoreNLP, in which we extract verb, conjunction-connected verb phrase, preposition-connected verb phrase.

Besides the standard CoM data incorporating manipulations with explicit visual evidences, the proposed data synthesising framework is compatible of producing implicit visual reasoning steps $step'_i = (desc_i)$ without involving the manipulations. We thereby also build this partial CoM data on the corpus consisting of absurd visual questions (*i.e.,* asking unanswerable questions based on the given image) to further resist the toxic hallucinations. Specifically, given an image $I$ with a question $Q$,we prompt GPT-4V (OpenAI, 2023b) to solve the question step-by-step to acquire the reasoning chains.

| **Data Source** | **#QAs** | **#Chains** | **#Steps/Chain** | **#Manipulations Types/Chain** |
|---|---|---|---|---|
| TextVQA (Biten et al., 2019) | 10782 | 13766 | 2.93 | 2.41 |
| ST-VQA (Singh et al., 2019) | 4814 | 3959 | 2.88 | 2.43 |
| TDIUC-count (Shrestha et al., 2019) | 53547 | 54523 | 2.35 | 0.74 |
| TDIUC-absurd (Shrestha et al., 2019) | 11677 | 11677 | 4.09 | - |
| CoM-test | 4609 | 8612 | 3.26 | 2.18 |

Table 4: Detailed statistics the the training data and evaluation data synthesised with CoM production.

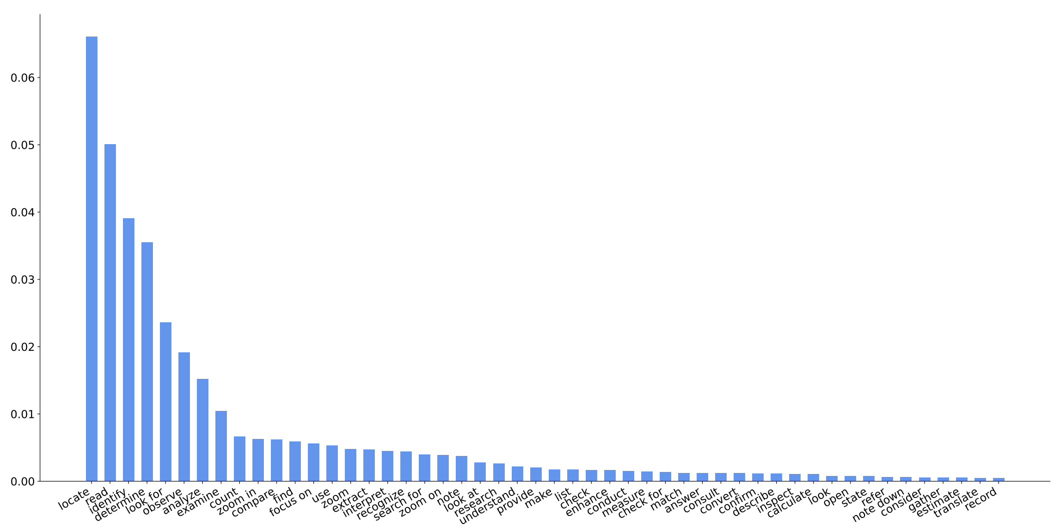


Figure 7: Distribution of the top-50 generated manipulations out of total $465$ based on 4-shot prompting, where the *first three bars* are scaled with $20\%$ for a smooth visualization of all data.

| | | |
|---|---|---|
| MANIPULATIONS | *OCR_i(tgt)->txt_i:* | *i-th OCR manipulation, that recognize the natural texts written on the target `tgt`, and return the recognized texts `txt_i`.* |
| | *calculate(tgt)->res_i:* | *i-th calculate manipulation, that calculate the formula specified by the target `tgt` in current image, and return the calculation result `res_i`.* |
| | *grounding_i(tgt)->bbx_i:* | *i-th grounding manipulation, that locates the object(s) specified by the target noun phrase `tgt` in current image, and return the resulting bounding box(es) as `bbx_i` where each box is represented by the top-left and bottom-right coordinates.* |
| | *crop_and_zoomin_i(bbx)->img_i:* | *i-th crop_and_zoomin manipulation which is useful to identify small and subtle objects in image, that first crops the current image using the box `bbx` defined by the top-left and bottom-right coordinates, and then zoom in the cropped image by two times and finally return the resulting image `img_i`.* |
| REQUIREMENTS (PROMPT) | *You are a visual assistant capable of generating solving steps for image-oriented visual questions. In each step, you can optionally use a manipulation to operate the image, which can be used to acquire specific information from the image or to acquire the processed new image (please be aware that these manipulations will not actually be performed when you generate the solving steps). The manipulation can be one from the predefined ones, or can be a new one you create yourself (should there indeed be a need), where the predefined manipulations with their descriptions are listed below:\n\n{MANIPULATIONS}.\n\nGiven a question Q abount an image, please generate a series of essential solving steps, where the output of each step is a tuple consisting of a Manipulation (leave it to None if the current step doesn't involve any manipulation) and a Description: (1) Manipulation f(x)->y, that is the manipulation `f` targeting `x` to obtain specific information or image `y`; (2) Description, which is a sentence describing the current solving step.\n Please adhere the following format: given an input of 'Q: xxx. The essential solving steps are: ', the output should like 'Step 1: (Manipulation, Description); Step 2: (Manipulation, Description); ...'.\n\n There are several examples:\n {DEMONSTRATIONS}\n\n Q: {QUESTION} The essential solving steps are:* | |

Manipulations Definition and Linguistic Annotation Guideline

| | | |
|---|---|---|
| $INPUTS$ | **Question**: What number police station is on the building? | |
| $OUTPUTS_L$ | Step 1: grounding_1(the building)->bbx_1, | Locate the building in the image and return the bounding box of the building as `bbx_1`. |
| | Step 2: (grounding_2(number police station on the building in box `bbx_1`)->bbx_2, | Identify the number of the police station on the building in box `bbx_1` and return the bounding box of the number as `bbx_2`. |
| | Step 3: (OCR_1(number in region `bbx_2`)->txt_1, | Recognize the number in the region `bbx_2` and return the recognized number as `txt_1`. |

*Linguistic Annotation*

$INPUTS$

**Question**: What number police station is on the building?

**Answer**: 43

$OUTPUTS_L$

*Visual Annotation*

$OUTPUTS_{VL}$


Locate the building in the image and return the bounding box of the building as `bbx_1`.


Identify the number of the police station on the building in box `bbx_1` and return the bounding box of the number as `bbx_2`.


Recognize the number in the region `bbx_2` and return the recognized number as `txt_1`.

An Example to show the linguistic annotation results and Visual annotation results

Figure 8: An example shows the configuration, inputs, outputs of the linguistic annotation and visual annotation.

### C.4 Details of the Linguistic/Visual Annotations

In this work, we adopt the GPT4-turbo as the linguistic annotator for generating problems solving steps, and the API call was conducted during the period of 2023.9 - 2023.12. For the visual annotators, we leverage the the currently best-performing tools, GroundingDINO and PaddleOCR, to acquire all visual contents requested by the manipulations. For a clear description to the production setting and results, we illustrate the guiding prompt, and an example-based linguistic annotation results as well as the visual annotation results in Figure 8.

### C.5 Limitation Analysis for the Data Production

For the implemented data framework, we engage the remarkable LLM to provide basic solving steps, adopt two reliable visual tools (*i.e.,* GroundingDINO and PaddleOCR) to acquire corresponding visual contents, and then perform the traversal to achieve feasible reasoning paths, which ensures the correctness and robustness of data synthesizing. However, we also find that there are three major limitations caused by the employed models and could be improved in future:

- The lack of diversity in linguistic reasoning steps. The 5-shot prompting to the GPT-4 gains a stable solving steps, but it also results in the descriptions for executing manipulations or general thinking are similar. We suggest that this can be addressed by employing diversified prompts or requirements.
- The inaccuracy of visual tools. We find that there are a considerable amount of negative paths caused by the failures of visual tools, such as the rough granularity of bounding boxes and the error recognition of slated letters or long sentences. This issue can be relieved by improving the semantic understanding capabilities of visual tools.

# D Details of Training

## D.1 Launching Prompts

- Please solve the problem gradually via a chain of manipulations, where in each step you can selectively adopt one of the following manipulations GROUNDING(a phrase)→boxes, OCR(an image or a region)→texts, CROP_AND_ZOOMIN(a region on given image)→new_image, CALCULATE(a computable target)→numbers, or invent a new manipulation, if that seems helpful. {QUESTION}
- Please tackle a given question in a stepbystep manner. For each step one of the following manipulations (depicted as Name(Input)→Retrun) can be optionally used: GROUNDING(a phrase)→boxes, OCR(an image or a region)→texts, CROP_AND_ZOOMIN(a region on given image)→new_image, CALCULATE(a computable target)→numbers, or develop a new manipulation yourself (if it is indeed required). {QUESTION}
- Please go through the question incrementally with chain of manipulations (optionally use manipulation when needed) such as GROUNDING(a phrase)→boxes, OCR(an image or a region)→texts, CROP_AND_ZOOMIN(a region on given image)→new_image, CALCULATE(a computable target)→numbers, and create a new manipulation if necessary. {QUESTION}

## D.2 Training settings

| **Parameters** | **Stage1-1** | **State1-2** | **Stage-2** |
|---|---|---|---|
| Hardware Environment | 3,840 A100xdays | 256 A100xdays | 160 A100xdays |
| Objective | next token prediction | next token prediction | next token prediction |
| Images | 1.5B | 40M | 576K |
| Batch size | 8192 | 1024 | 160 |
| Iterations | 120,000 | 60000 | 14000 |
| Optimizer | AdamW | AdamW | AdamW |
| Learning rate | 1e-4 | 1e-5 | 1e-5 |
| Warm up steps | 7200 | 1200 | 280 |
| Trainable weights | 6.5B visual expert | 6.5B visual expert | 6.5B visual expert |

Table 5: Training details of all stages.

# E Details of Qualitative Analysis

## E.1 Qualitative Analysis

We investigate the evidential reasoning capability of CogCoM on scenarios that requires different types of meticulous reasoning, including recognizing textual details, reading time, understanding charts and counting objects. The results are shown in Figure 1. The first case demonstrates that CogCoM finds the region corresponding to the plane logo through two steps of grounding and then achieves the answer based on zooming in the cropped region. The second case illustrates the ability of CogCoM in reading time, by locating the device that displays time and then transforming the time into words based on the read_timne manipulation. In the forth example, CogCoM first identifies all visible truck wheels through grounding, and then leverages the counting manipulation to consolidate the total number.

### E.2 Representative Cases


***CoM*:**

Given a question, please solve the question step-by-step with a chain of manipulations, where in each step you can optionally use one of the following manipulations (defined as Name(Input)->Return) on the image: GROUNDING(a phrase)->boxes, OCR(an image or a region)->texts, CROP_AND_ZOOMIN(a region on given image)->new_image, CALCULATE(a computable target)->numbers, or a new manipulation created by yourself (if it is indeed needed).

*Question:* What is written on the traffic sign near the bus?
*Model:* Using GROUNDING(the traffic sign near the bus) to locate the traffic sign near the bus in current image, which is *[[084,157,926,703;676,146,740,273]]*. Utilizing OCR(texts in region *[[084,157,926,703]]*) to read the texts written on the traffic sign within region [[084,157,926,703]], resulting Taxi. So ultimately, the conclusive answer to the question in discussion is Taxi.


***Visual Grounding:***
*Question:* Find the region in image that "a traffic sign" describes.
*Model:* [[678,145,739,270]]

***Grounded Captioning:***
*Question:* The goal of this task is to generate description for one part of the image. The part is specified by [[678,145,739,270]].
*Model:* ~~A no tax sign~~

***Image Captioning:***
*Question:* In this task, you are given an image and you will need to generate some text to describe it. Please give me a concise answer.
*Answer:* Taxi bus with rainbow colors on side.

***Multi Choice:***
*Question:* Select the color that does not appear in the rainbow sign of this bus. (1) red. (2) yellow. (3) pink. (4) black. Please give me a direct answer.
*Answer:* black

***Detailed Captioning:***
*Question:* Explain the visual content of the image in great detail.
*Answer:* The image captures a scene on a city street, where a white bus with red and blue accents is parked. The bus bears the text "Trafalgar" in bold red letters across its side, indicating it's likely part of a tourist or travel company. The bus is positioned next to a traffic sign that reads "TAXI", suggesting it might be waiting for passengers or ready to embark on a journey. In the background, there are buildings and trees, adding depth to the urban landscape. The overall impression is one of a typical day in an urban setting, with a bus waiting for its passengers amidst the hustle and bustle of city life.

Figure 9: CogCoM demonstrates the flexible capabilities for adapting to different multimodal scenarios, including evidential visual reasoning, Visual Grounding, Grounded Captioning, Image Captioning, Multi Choice, and Detailed Captioning.

# NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please refer to Line 6 to Line 20, and Line 48 to Line 72.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to Appendix B.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Please refer to Line 74 to Line 92 for pilot experiments.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The data collection with Section 3, model training process with Section 4, and the experimental settings with Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will open-source the code, model weights, and all collected data. The generation process and statistics of data are available at Section 3 and Appendix C.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The experimental settings are listed in each subsection of benchmark evaluation, which is Section 5.1, Section 5.2 and Section 5.3. The training settings with hyperparameters and optimizations are listed at Appendix D.2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We include the error analysis with textual descriptions in Appendix C.5, as well as the reasoning accuracy with limitations in Section 5.1.2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Detailed compute resources are listed in Section D.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: All code anonymity.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discussed the societal impacts in details in Appendix B.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The training data collected from public datasets with research purpose does not face the safety risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The public available datasets used in this paper are cited properly.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The newly created benchmark CoM-test is documented in detailed in Section 5 and Appendix C.2, and the manually annotated math data is described in Section 3.2.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This study does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This study does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.