Taking Notes Brings Focus? Towards Multi-Turn Multimodal Dialogue Learnings

Anonymous ACL submission

Abstract

Multimodal large language models (MLLMs), built on large-scale pre-trained vision towers and language models, have shown great capabilities in multimodal understanding. However, most existing MLLMs are trained on single-turn vision question-answering tasks, which do not accurately reflect real-world human conversations. In this paper, we introduce MMDiag, a multi-turn multimodal dialogue dataset. This dataset is collaboratively generated through deliberately designed rules and GPT assistance, featuring strong correlations between questions, between questions and images, and among different image regions; thus aligning more closely with real-world scenarios. MMDiag serves as a strong benchmark for multi-turn multimodal dialogue learning and brings more challenges to the grounding and reasoning capabilities of MLLMs. Further, inspired by human vision processing, we present DiagNote, an MLLM equipped with multimodal grounding and reasoning capabilities. DiagNote consists of two modules (Deliberate and Gaze) interacting with each other to perform Chain-of-Thought and annotations respectively, throughout multi-turn dialogues. We empirically demonstrate the advantages of DiagNote in both grounding and jointly processing and reasoning with vision and language information over existing MLLMs.

1 Introduction

In recent years, large language models (LLMs) have achieved remarkable advances in various natural language applications, including chatbots (Bai et al., 2023a; Achiam et al., 2023; Reid et al., 2024), programming assistants (Cursor, 2024), and rhetorical aides (DeepL, 2024). The success has further spurred the development of multimodal large language models (MLLM) (Liu et al., 2024b; Zheng et al., 2025). However, most existing MLLMs are trained as single black-box systems to handle multimodal instructions, often struggling with inaccuracies and hallucinations, especially in complex multi-turn dialogues (Tan et al., 2024; Zheng et al., 2024). We hypothesize such challenges arise from the MLLM's difficulty in maintaining focus on target regions throughout the conversation, especially for high-resolution images with overly long visual tokens. In this paper, we seek to address these issues by moving beyond a black-box approach to an explicit target-grounding solution. Here, we summarize two key goals for multi-turn multimodal dialogue learning: ❶ "saliency tracking", where models must keep tracking different relevant regions over the course of the dialogue, and ❷ "saliency recall", where models need to consistently retain focus on the same critical information across multiple question-answering (QA) rounds. For example, in the dialogue illustrated in Figure 1, completing the Minigrid (Chevalier-Boisvert et al., 2023) task requires the MLLM to accurately locate both the agent (*i.e.*"red triangle") and the target (*i.e.*"purple key") to answer the initial question. The following question then builds upon this information, requiring the MLLM to reason about the agent's starting position based on the previously identified location of the key. This example illustrates the need for sustained and explicit grounding to multiple specific visual details in multi-turn multimodal dialogue.

To achieve these two goals, we draw inspiration from how humans maintain focus while studying. For instance, when working through documents, people may lose concentration, but can quickly refocus by using simple techniques such as jotting down notes or highlighting key points. Even basic marks, such as circling or underlining, can significantly enhance focus without requiring elaborate explanations. These visual cues guide attention, making it easier to track, recall, and revisit important information. In contrast, existing MLLMs lack such tracking capabilities, prompting us to ask: "Can an MLLM be designed to equip similar

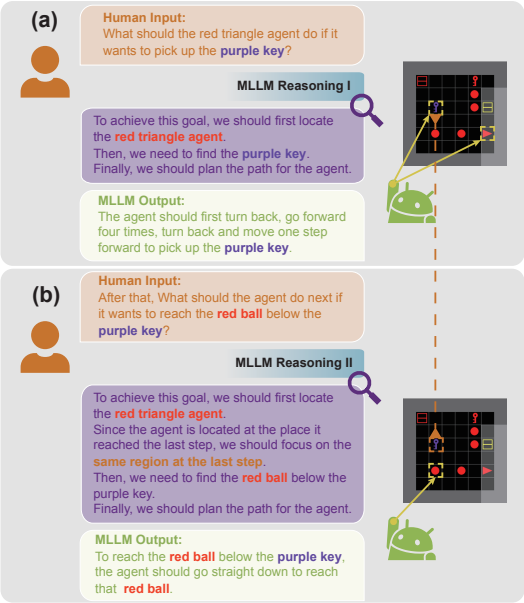


Figure 1: Multi-turn multimodal dialogue: **(a) Saliency tracking.** The MLLM needs to focus on both the red triangle agent and the purple key, which scatter on the image, to answer the question correctly. **(b) Saliency recall.** The MLLM needs to retain focus on the region where the agent will stop after the last question.

attention-guiding abilities? If so, what would that model design entail?"

To answer this question, we first review existing tuning methods for MLLMs and identify a critical gap: the lack of quality multi-turn multimodal QA datasets that adequately reason over both visual and text information. Existing datasets, such as MMDU (Liu et al., 2024c) and SciGraphQA (Li and Tajbakhsh, 2023), primarily consist of single-turn QA pairs, where most questions can be answered independently without relying on prior context. To bridge this gap, we introduce a novel dataset, MMDiag, designed as a foundational benchmark for challenging multi-turn multimodal dialogue. This dataset offers visually detailed multi-turn dialogues across a range of scenarios.

Furthermore, recent studies have introduced various modules to help keep focus in multi-turn multimodal dialogues. However, these methods either "zoom in" to progressively narrow focus areas with the aid of external grounding and OCR tools (Qi et al., 2024), or identify a single region of interest per question before generating an answer (Shao et al., 2024). These approaches lead to severe limitations: the zoom-in method restricts the focus to smaller regions, potentially missing broader context, while the single-region method isolates specific areas, overlooking multiple relevant details that could enrich responses. To address these limitations, we propose DiagNote, a model designed to enhance focus and reasoning in multi-turn multimodal dialogue. DiagNote comprises two main modules: Deliberate and Gaze. The Deliberate module guides the Gaze module in dynamically adjusting regions of visual focus, while the Gaze module highlights crucial areas for subsequent processing by the Deliberate module. These two modules interact across multiple dialogue turns, emulating human visual processing to produce an answer accompanied by optional reasoning and grounding steps. Through this interactive mechanism, DiagNote can achieve more effective reasoning with multimodal information, resulting in accurate and context-aware responses throughout dialogues.

Our main contributions are summarized as follows: ❶ To address the need for robust multimodal grounding and reasoning, we build a new large-scale multi-turn multimodal dialogue dataset – MMDiag – across several QA scenarios (*e.g.*daily life and tabular data), using rule-based searching and GPT-4o-mini (OpenAI) capabilities. ❷ Inspired by human visual processing, we propose DiagNote and its two key modules – Deliberate and Gaze – to enhance the model's capacity for multimodal information integration and reasoning. ❸ We evaluate DiagNote's reasoning and grounding abilities on MMDiag and other benchmarks and the results demonstrate that the introduction of MMDiag and DiagNote significantly improves performance in multimodal conversations, while the MMDiag itself can also serve as a more challenging benchmark for this area.

2 Related Work

2.1 Multimodal Large Language Models

The introduction of Transformers (Vaswani et al., 2017; Liu et al., 2021) and large-scale training has significantly advanced model capabilities, enabling powerful vision encoders (Radford et al., 2021a) and large language models (LLMs)(Chiang et al., 2023; Touvron et al., 2023). Building on these foundations, multimodal large language models (MLLMs)(Liu et al., 2024b; Zheng et al., 2024) have achieved strong performance across diverse tasks, with promising applications in VR/AR and game agents (Xu et al., 2024; Feng et al., 2024). MLLMs typically comprise three core components: modality encoders, modality interfaces, and LLMs (Yin et al., 2023). The encoders and LLMs handle visual and linguistic inputs separately, while

interfaces align non-language modalities with the language space. Some models further incorporate generators to produce other modalities, such as actions (Driess et al., 2023) or images (Zheng et al., 2024). Training MLLMs usually involves two stages. The first aligns vision and language via pretraining on large-scale image-caption datasets (Liu et al., 2024b; Schuhmann et al., 2022; Changpinyo et al., 2021). The second fine-tunes models on tasks like visual question answering (VQA)(Liu et al., 2024b; Singh et al., 2019) to enhance instruction-following abilities. This two-stage pipeline underpins many state-of-the-art models, including PALI-X(Chen et al., 2023), Qwen-VL (Bai et al., 2023b), and LLaVA (Liu et al., 2024b), serving as a foundation for recent MLLM advances.

2.2 Grounding and Reasoning Benefit MLLMs

MLLMs benefit from language models' in-context learning (Brown, 2020) and Chain-of-Thought (CoT) (Wei et al., 2022) for generalization and reasoning. However, MLLMs sometimes rely excessively on LLM components, leading to overlooking visual details and hallucinations. To address these limitations, Qi et al. (2024) introduce "Chain of Manipulations", allowing MLLMs to perform reasoning with external grounding and OCR models, which enable incremental task-solving. Although this approach improves performance, it is limited to zooming in on specific areas and may miss key scattered details. Similarly, Shao et al. (2024) enhance performance by focusing on a single region of interest per question. However, a single grounding and reasoning round is often insufficient for complex problems. To overcome these challenges, we propose two modules: **Deliberate** for reasoning and **Gaze** for grounding, enabling multiple rounds of reasoning. This iterative approach allows for better problem-solving by refining both grounding and reasoning across interactions, making it more effective in handling complex tasks, like multi-turn multimodal QAs.

2.3 Multi-Turn Multimodal Dialogue

Multi-turn dialogue involves sustained interaction between a human and an MLLM-based agent, spanning casual exchanges (Shuster et al., 2018), feedback-driven refinement (Chen et al., 2024c), cooperative tasks (Chen et al., 2024a), and structured QA scenarios (Lin et al., 2014; Singh et al., 2019), which is our focus. In language-only dialogues, a key challenge lies in handling question interdependence, where earlier answers serve as context for later queries. Introducing visual input adds complexity: the model must ❶ integrate language context, ❷ align it with visual input, and ❸ cope with diminishing visual focus in extended dialogues. Dialogues with independent questions reduce the task to single-turn QA. Existing multi-turn datasets (Das et al., 2017; Liu et al., 2024c; Li and Tajbakhsh, 2023) often feature weakly connected QA pairs. Seo et al. (2017) include spatial reasoning but with simple tasks, while Tian et al. (2024) address referential challenges by rule-based word substitution (*e.g.*, it), which harms coherence and introduces ambiguity. Our method overcomes these issues by first generating correlated QA drafts with rules, then refining them using GPT-4o-mini (OpenAI), resulting in a more realistic and complex multimodal, multi-turn dialogue dataset.

3 MMDiag: A New Benchmark for Multi-Turn Multimodal Dialogue

In the following section, we first motivate the choice of three scenarios: everyday, tabular, and Minigrid. Next, we illustrate how to construct the QA pairs for our MMDiag dataset. We then explain the evaluation process in Section 3.3. Finally, we compare MMDiag with existing multimodal dialogue datasets in Section 3.4. Examples of QA pairs are given in Appendix A.2. Both MMDiag and its generation code will be publicly released.

3.1 Chosen Scenarios

The three selected scenarios — *Everyday*, *Tabular*, and *Minigrid* — are chosen to evaluate distinct yet complementary challenges in multimodal reasoning. *Everyday* scenes test common-sense understanding and multi-turn interactions, reflecting real-world AI applications. *Tabular* scenarios require structured data comprehension and numerical reasoning, which many MLLMs struggle with. And *Minigrid* focuses on spatial reasoning and planning, essential for navigation and decision-making. This diverse selection ensures a comprehensive assessment of multimodal understanding. Empirically, all three settings pose significant challenges even for state-of-the-art models like GPT-4o (Figure 3), with notable failures, such as Visual CoT's inability to generate positive grounding predictions in Tabular tasks (Table 2).

Dataset	QA Scale	GND Scale	Generation Process	Average Turns	Multi-Turn	Multi-Region	Dialogue Correlation
CB-300k (Tian et al., 2024)	463k	254k	GPT-4/Rule-based	5.49	✓	✗	◯
Visual CoT (Shao et al., 2024)	438k	438k	GPT-4/OCR	1	✗	✗	✗
CoM (Qi et al., 2024)	76k	-	GPT-4/Tree-Search/Human	1	✗	◯	✗
MMDU (Liu et al., 2024c)	410k	-	LLM-filtered/GPT-4o	9	✓	✗	✗
MMDiag	639k	1139k	Graph-search/OCR/GPT-4o-mini	2.19	✓	✓	✓

Table 1: Comparison between MMDiag and other multimodal dialogue datasets. ◯: Features are considered, but implemented weakly.

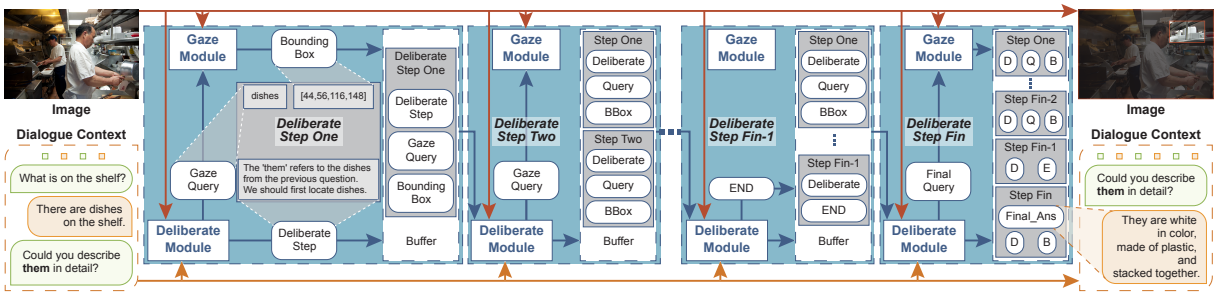


Figure 2: Model architecture of DiagNote. Regions with blue backgrounds represent a deliberation step and the interaction between the Deliberate and Gaze modules. At each turn, the Deliberate module processes the original image, dialogue context, and buffers from both modules. It produces two outputs: (1) a Deliberate step, stored in the Deliberate buffer, and (2) a Gaze query, which is processed by the Gaze module. The resulting bounding boxes are then stored in the Gaze buffer.

3.2 Dataset Curation

Everyday Scene Subset. The source dataset (Krishna et al., 2017) includes 108K images with detailed annotations, allowing us to construct a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ for each image, where $\mathcal{V}$ are objects and $\mathcal{E}$ are their relationships. Each QA pair is represented as a subgraph $\mathcal{G}qa = (\mathcal{V}qa, \mathcal{E}_{qa})$, containing nodes and edges involved in either question or answer. If a QA pair shares no nodes or edges with others, it is considered independent, as it doesn't add to dialogue complexity or rely on cross-QA information. We extend QA pairs into multi-turn QAs by building a subgraph pattern $\mathcal{M} = \bigcup_{i=1}^{n} \mathcal{G}qa^i$, ensuring each $\mathcal{G}qa^i$ overlaps with at least one other (i.e., $\exists j \neq i$ such that $\mathcal{V}qa^i \cap \mathcal{V}qa^j \neq \varnothing$), so answering any pair depends on others. Subgraph matching is then used to identify instances of $\mathcal{M}$ in $\mathcal{G}$, enabling the generation of diverse multi-turn QAs. We use GPT-4o-mini (OpenAI) to produce natural questions, answers, and reasoning steps, along with ground-truth object locations. The prompt is detailed in Appendix A.1.

Tabular Scene Subset. This subset is sourced from ChartQA (Masry et al., 2022), which contains 18K real-world charts and 23.1K human-authored QA pairs. As ChartQA consists only of single-turn QA, it does not meet our multi-turn dialogue requirements. To generate multi-turn question answering, we use GPT-4o-mini, primarily relying on chart images due to the questionable reliability of table-type metadata. To ensure interrelated dialogues, where certain regions are referenced as pronouns to increase complexity, we explicitly emphasize this requirement in the prompt. However, GPT-4o-mini struggles with maintaining this structure, requiring supplementary prompts to guide generation more effectively. Details on the prompt design are provided in Appendix A.1. Finally, we use EasyOCR (JaidedAI, 2024) to match keywords with corresponding chart regions, enabling generation of bounding boxes for relevant areas.

Minigrid Scene Subset. Minigrid (Chevalier-Boisvert et al., 2023) is a Gymnasium-based (Towers et al., 2024) collection of 2D grid-world environments with goal-oriented tasks. The agent, represented as a triangular figure with a discrete action space, navigates maze-like maps and interacts with objects such as doors, keys, and boxes. These tasks test the model's ability to focus on image details, spatial reasoning, and action planning, with some requiring numerous steps to complete, making them particularly challenging. To construct this subset, we use Minigrid and BabyAI (Chevalier-Boisvert et al., 2019) to generate grid worlds, tasks, and step-by-step action plans, which are formatted as prompts for GPT-4o-mini. Further details on environment generation and prompt design are in Appendix A.1.

Common Visual-Text Subset. To enable MLLMs with robust capabilities to answer the question, we

also add additional visual-text pairs with high quality from previous works (Liu et al., 2024b) to enhance their instruction-following ability.

3.3 Multi-Turn Multimodal Dialogue Evaluation

MMDiag outputs three components: reasoning process, grounded key regions, and final answers, which we evaluate separately. For the reasoning and answers—both in natural language and variable in phrasing—we follow standard practice by inputting images, questions, ground-truth and generated answers into a strong MLLM for scoring. To avoid evaluation bias, we use Gemini-1.5-Pro (Reid et al., 2024) instead of GPT-4o-mini (OpenAI), which was used in dataset generation. Following prior work (Lee et al., 2024; Stureborg et al., 2024; Chen et al., 2024b), we adopt "ad-hoc" reasoning-based scoring across five categories on a 0–10 scale for consistency and interpretability; the full prompt is in Appendix A.3. We also evaluate grounding using key region queries and bounding boxes, forming a GND subset. As these queries often describe objects or regions with detailed attributes and relations, the subset effectively assesses grounding for complex cases. Grounding accuracy is measured via Intersection over Union (IoU).

3.4 Multimodal Dialogue Datasets Comparison

We compare MMDiag with prior datasets designed for vision-language understanding and reasoning. As shown in Table 1, MMDiag is the first to feature multi-turn, multi-region dialogues with strong QA dependencies, reinforced by a thorough generation process. In contrast, datasets like CB-300k (Tian et al., 2024) and MMDU (Liu et al., 2024c) lack mechanisms to enforce such dependencies, reducing multi-turn dialogues to mere concatenations of independent QA pairs. Although MMDiag has relatively short dialogues, the inherent dependence between turns presents significant challenges for MLLMs, including GPT-4o, as demonstrated in Figure 3. The grounding and QA test splits include 1,000 unseen images and QA pairs, respectively.

4 DiagNote

In this section, we introduce our proposed DiagNote and its training process. Using two essential modules named Deliberate and Gaze, DiagNote is trained on the train split of MMDiag to meet the requirements for multi-turn multimodal dialogue, which provides capabilities of stepwise reasoning and grounding corresponding salient visual regions for each dialogue.

4.1 Model Architecture

The overall framework of our model is illustrated in Figure 2. We adopt the same architecture, LLaVA-1.5 (Liu et al., 2024b,a), for both the Deliberate and Gaze modules, with no shared parameters. To leverage the generalization capability of MLLMs, we avoid using dedicated grounding models such as Grounding DINO (Liu et al., 2023) for the Gaze. Each module consists of an LLM backbone, a pre-trained ViT (Radford et al., 2021b) as vision encoder, and an MLP projection for vision-language alignment, with distinct parameters for the two modules. Given an image $\mathbf{I}_\mathrm{v}$ and a dialogue of T turns $\left(\mathbf{I}_\mathrm{q}^1, \mathbf{I}_\mathrm{a}^1, \cdots, \mathbf{I}_\mathrm{q}^T, \mathbf{I}_\mathrm{a}^T\right)$, where $\mathbf{I}_\mathrm{q}^t$ and $\mathbf{I}_\mathrm{a}^t$ denote the t-th question and answer, the model performs multi-step interactions between Deliberate and Gaze at each turn to generate the answer $\mathbf{I}_\mathrm{a}^t$.

At turn t, given question $\mathbf{I}\mathrm{q}^t$, the Deliberate module $\mathbb{D}$ takes the image $\mathbf{I}_\mathrm{v}$ and dialogue context $\mathbf{C}^t = \left(\mathbf{I}_\mathrm{q}^1, \mathbf{I}_\mathrm{a}^1, \cdots, \mathbf{I}_\mathrm{q}^{t-1}, \mathbf{I}_\mathrm{a}^{t-1}, \mathbf{I}_\mathrm{q}^t\right)$ to produce a Deliberate step $\mathbf{S}_1^t$ and a Gaze query $\mathbf{Q}_1^t$, stored in buffers $\mathbf{B}_d^t$ and $\mathbf{B}_g^t$ respectively. The Gaze $\mathbb{G}$ then outputs bounding box $\mathbf{o}_1^t$ based on $\mathbf{Q}_1^t$, also stored in $\mathbf{B}_g^t$. In each subsequent round i, the Deliberate receives $\mathbf{I}_\mathrm{v}$, context $\mathbf{C}^t$, Gaze buffer $\mathbf{B}_g^t$, and Deliberate buffer $\mathbf{B}_d^t$ to generate new $\mathbf{S}_i^t$ and $\mathbf{Q}_i^t$, while Gaze returns $\mathbf{o}_i^t$. The process repeats until the Deliberate outputs 'END' as query $\mathbf{Q}\mathrm{Fin} - 1^t$, indicating that the Deliberate and Gaze back-and-forth process is complete.

Finally, the image, the dialogue context, and all the buffers are fed into the Deliberate module $\mathbb{D}$ to produce the final answer $\mathbf{S}_\mathrm{Fin}^t$ (i.e., $\mathbf{I}_\mathrm{a}^t$) and the Gaze query $\mathbf{Q}_\mathrm{Fin}^t$. The Gaze module $\mathbb{G}$ then provides the bounding box of the salient area $\mathbf{o}_\mathrm{Fin}^t$ for the t-th dialogue turn. The final output is $\mathbf{S}_\mathrm{Fin}^t$, along with the optional key region bounding box $\mathbf{o}_\mathrm{Fin}^t$, as well as the Deliberate process $\left(\mathbf{S}_1^t, \cdots, \mathbf{S}_{\mathrm{Fin}-1}^t\right)$, if required. The final answer $\mathbf{I}_\mathrm{a}^t$ is then appended to the dialogue context for the next dialogue turn.

4.2 Model Training

The training process of both Deliberate and Gaze modules follows that of LLaVA, and DiagNote provides two prompt templates p^d and p^g for Deliberate and Gaze respectively. At the i-th round of Deliberate and Gaze for Question $\mathbf{I}_\mathrm{q}^t$, the instruc-

Model	Train Data	MMDiag GND Testset			GND Dataset		Average
		Everyday	Tabular	Minigrid	MSCOCO	RefCOCO	
Grounding DINO (Liu et al., 2023)	-	0.384	0.001	0.209	**0.715**	0.469	0.356
LLaVA (Liu et al., 2024b)	LCS558K+Mixed665K	0.237	0.006	0.142	0.365	0.414	0.233
Visual CoT (Shao et al., 2024)	VisCoT	0.220	0.003	0.160	0.321	0.362	0.213
DiagNote	COCO	0.307	0.008	0.199	0.662	0.765	0.388
DiagNote	MMDiag	0.369	0.466	**1.0**	0.259	0.257	0.471
DiagNote	MMDiag + COCO	0.399	**0.487**	0.988	0.624	0.742	0.648
DiagNote	MMDiag + COCO + VisCoT	**0.433**	0.281	0.910	**0.662**	**0.837**	0.625

Table 2: Comparison results with existing MLLMs on Grounding benchmarks (GND) to demonstrate the challenging characteristics of our dataset MMDiag. We use Intersection over Union (IoU) as the evaluation metric.

tion $\mathbf{R}\text{in}_i^{\text{d}}$ for the Deliberate module is:

$$\mathbf{R}\text{in}_i^{\text{d}} = \begin{cases} p^{\text{d}}\big(\mathbf{I}_{\text{v}}, \mathbf{C}^t\big), & i = 1 \\ p^{\text{d}}\big(\mathbf{I}_{\text{v}}, \mathbf{C}^t, \mathbf{B}_g^t, \mathbf{B}_d^t\big), & 1 < i < \text{Fin} \\ p^{\text{d}}\big(\mathbf{I}_{\text{v}}, \mathbf{C}^t, \mathbf{B}_g^t, \mathbf{B}_d^t, \text{Fin}\big), & i = \text{Fin}, \end{cases} \tag{1}$$

where $\mathbf{B}_d^t = \big(\mathbf{S}_1^t, \cdots, \mathbf{S}_{i-1}^t\big)$ and $\mathbf{B}_g^t = \big(\mathbf{Q}_1^t, \cdots, \mathbf{Q}_{i-1}^t\big)$. The instruction $\mathbf{R}\text{in}_i^{\text{g}}$ for the Gaze module is:

$$\mathbf{R}\text{in}_i^{\text{g}} = p^{\text{g}}\big(\mathbf{I}_{\text{v}}, \mathbf{Q}_i^t\big), \quad i \leq \text{Fin}, i \neq \text{Fin} - 1. \tag{2}$$

We fine-tune the LLM on the prediction tokens, utilizing the auto-regressive training objective to optimize. We compute the probability of the target output $\mathbf{R}\text{out}_i^{\text{x}}$ with length L at i-th round by:

$$p\left(\mathbf{R}\text{out}_i^{\text{x}} \mid \mathbf{R}\text{in}_i^{\text{x}}\right) = \prod_{l=1}^{L} p_{\boldsymbol{\theta}^{\text{x}}}\left(r_l \mid \mathbf{R}\text{in}_i^{\text{x}}, \mathbf{R}\text{out}_{,<l}^{\text{x}}\right), \quad \text{where } \text{x} \in \{\text{d}, \text{g}\}. \tag{3}$$

$\boldsymbol{\theta}^{\text{x}}$ is the trainable parameters of Deliberate and Gaze modules respectively, with $\text{x} \in \{\text{d}, \text{g}\}$. $\mathbf{R}\text{in}_i^{\text{x}}$ are input tokens of i-th round of the Deliberate and Gaze interaction process. $\mathbf{R}\text{out}_{,<l}^{\text{x}}$ are answer tokens before the current prediction token r_l.

Our Deliberate and Gaze modules take LLaVA-1.5 as base models. For the Gaze module, since grounding such salient areas as words and objects with detailed descriptions is quite challenging, we first fine-tune it with an additional grounding dataset, and then fine-tune Deliberate and Gaze modules together. We combine the fine-tuning dataset from LLaVA (Liu et al., 2024b) with the grounding split of MMDiag to generate the grounding dataset; and we also combine the fine-tuning dataset from LLaVA with the training split of the MMDiag dataset to generate the entire training dataset. For data points in LLaVA, DiagNote does not add Deliberate prompts for the Deliberate module, thus instructing the Deliberate module to maintain the ability to output answers in general format.

5 Experiments

5.1 Implementation Details

We use LLaVA-1.5-7B (Liu et al., 2024a) as the foundation model for both Deliberate and Gaze modules, with CLIP-ViT-Large-Patch14-336 (Radford et al., 2021b) as vision tower. Training is conducted on 8 × A800 GPUs with a learning rate of 2e-5. Deliberate and Gaze are optimized separately via supervised learning with ground-truth outputs per round. During inference, the Gaze module signals reasoning completion by outputting "END" for turn T_x (Table 4), with the round number dynamically determined by DiagNote. Additional training details are provided in the Appendix B,C.

5.2 Results on MMDiag

5.2.1 Visual Grounding

This section focuses on how the MMDiag dataset enhances grounding performance in MLLMs. Grounding is essential for enabling MLLMs to attend to salient regions and reveal the reasoning process, rather than acting as black boxes. We evaluate DiagNote on standard grounding (GND) benchmarks (Lin et al., 2014; Kazemzadeh et al., 2014; Tian et al., 2024) and the MMDiag GND benchmark, using average IoU scores, as shown in Table 2. Compared to benchmarks like MSCOCO, DiagNote shows a notable performance drop on MMDiag, indicating its higher difficulty. Existing models like Visual CoT, despite incorporating region-based attention, perform poorly on GND tasks—*e.g.*, scoring -0.394 vs. Grounding DINO on MSCOCO and underperforming LLaVA—revealing their limited robustness in grounding relevant image areas. In contrast, Diag-

Model	Gaze	Train Data	MMDiag						Average
			Everyday		Tabular		Minigrid		
			reasoning	answer	reasoning	answer	reasoning	answer	
LLaVA (Liu et al., 2024b)	✗	LCS558K+Mixed665K	2.55	4.85	1.00	1.28	2.29	0.42	2.21
CogCoM (Qi et al., 2024)	✗	-	3.05	5.45	0.50	1.25	0.53	0.96	2.20
Visual CoT (Shao et al., 2024)	✗	VisCoT	4.15	4.90	1.23	1.95	1.09	2.50	2.81
DiagNote	✗	MMDiag	4.25	4.95	3.61	4.20	4.95	4.27	4.32
DiagNote	✓	MMDiag	5.82	**6.15**	3.95	4.05	5.10	4.15	4.92
DiagNote	✓	MMDiag+COCO	6.35	5.97	3.95	4.30	5.75	4.93	5.18
DiagNote	✓	GT	**6.85**	5.80	**6.32**	**7.76**	**7.37**	**9.15**	**7.00**

Table 3: Comparison of the evaluation score with baselines to validate the Gaze module, we use Gemini-1.5-Pro to evaluate the performance of the reasoning process and the final answer. The evaluation process is detailed in Section 3.3.

Model	Tabular							
	Reasoning				Answer			
	T1	T2	T3	T4	T1	T2	T3	T4
CogCoM	0.55	0.91	1.15	0.67	1.75	0.73	0.85	0.35
Visual CoT	1.50	1.05	1.33	1.02	1.86	1.24	1.03	0.88
LLaVA	2.34	0.35	1.00	0.58	1.42	0.50	0.97	0.50
w/o Gaze	**4.01**	3.05	2.15	1.66	**3.47**	2.03	1.65	1.63
with Gaze	3.86	**3.34**	**2.31**	**2.53**	3.25	**2.65**	**2.17**	**1.98**

Table 4: The Gemimi-1.5-Pro evaluation of the reasoning process and the final answer, scaling to 0-10, at turns 1 to 4 under the tabular scenario, where T$*$ denotes the $*$-th turn in the dialogue.

Note—trained on limited GND annotations from MMDiag and MSCOCO—achieves clear improvements on MSCOCO and RefCOCO, and outperforms others across all MMDiag subsets. Importantly, MSCOCO is used solely to enhance grounding, and we deliberately restrict GND data size to avoid scale bias. As shown in Row 4, training solely on MSCOCO leads to the weakest performance, underscoring the necessity and advantages of MMDiag.

5.2.2 Multi-Turn Reasoning

We evaluate our model's multi-turn reasoning capabilities using the MMDiag benchmark. Beyond final answer correctness, the evaluator also assesses the coherence and logic of the reasoning process within the Deliberate module, with detailed results in Table 3. "GT" denotes settings where the Deliberate receives ground-truth inputs during reasoning, serving as an upper bound. Other settings use Gaze queries generated by DiagNote, preventing information leakage. As expected, the GT setting significantly outperforms others, highlighting room for improvement. To validate the effectiveness of our proposed module, we observe that Gaze improves performance on specific reasoning tasks. For example, in everyday scenarios, models with Gaze achieve higher accuracy, showing enhanced focus and reasoning accuracy. When similar objects differ in location or attributes, the model may fail to identify the referenced one. Annotating the target in the image helps the model maintain focus and avoid such errors as reasoning progresses.

We further compare DiagNote with CogCoM (Qi et al., 2024) and Visual CoT (Shao et al., 2024), which also handle region-focused multimodal dialogue. DiagNote shows notable advantages, especially in tabular and Minigrid scenarios, reflecting the dataset's complexity and strengths of two modules. Table 4 shows a breakdown of tabular results across dialogue rounds: DiagNote consistently outperforms others in rounds two through four, underscoring its strength in long-context reasoning. Gaze brings more noticeable gains in longer dialogues (*e.g.*T3,4), further validating its benefit for extended multimodal understanding. Note that Table 3 includes QA pairs of lengths 2–4, while Table 4 focuses only on 4-turn dialogues.

5.3 Qualitative Results.

In this section, we provide additional grounding and reasoning examples of DiagNote. More visualization results can be found in Appendix D,F.

Visual Grounding. The Gaze module offers both grounding and OCR capabilities across diverse scenarios. As illustrated in Figure 4b, Grounding DINO (Liu et al., 2023) struggles in complex scenes where multiple objects of the same category exist with different attributes or relationships, therefore often failing to locate the target object precisely. In contrast, DiagNote's Gaze module effectively manages such situations, as shown in Figure 4a. Additionally, when faced with tasks requiring text recognition, the Gaze module exhibits

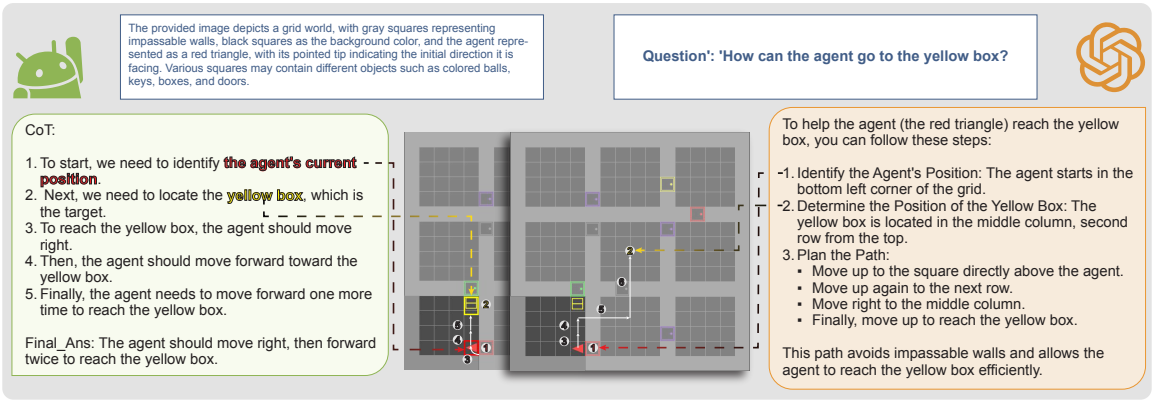


Figure 3: Comparison for an example of the Minigrid scenario, one of the subsets in MMDiag. We give DiagNote and GPT-4o the same environmental description and question. DiagNote focuses on the key regions and gives the correct reasoning process and the final answer. In contrast, GPT-4o fails to locate the object and thus gives the wrong answer. Examples for the MMDiag subsets of everyday scenarios and tabular scenes can be found in Appendix F.

(a) DiagNote (b) Grounding DINO

Figure 4: A grounding comparison between Grounding DINO and DiagNote's Gaze module , with the Gaze query "pink and white sign". In (a), the red bounding box represents the ground-truth answer, while the blue one indicates the output generated by the Gaze module in DiagNote. In (b), the red bounding boxes show the outputs produced by Grounding DINO.

more robust OCR capabilities, accurately identifying and localizing specific keywords.

Multi-Turn Reasoning. With the incorporation of the Gaze module, our model can also more effectively focus on fine-grained details distributed across the image, offering a clear advantage in tasks that demand cohesive reasoning across both visual and linguistic information. As shown in Figure 3, a comparison between our DiagNote and GPT-4o within a simple Minigrid environment highlights this benefit. Despite detailed descriptions provided in the prompt, GPT-4o struggles with completing a short-range, single-subgoal task, underscoring the strengths of our dataset and methodology.

5.4 Ablation Study

We observe a counterintuitive performance trend when comparing DiagNote with and without the Gaze module. To analyze its impact, we fine-tune DiagNote and Visual CoT on MMDiag and confirm Gaze's effectiveness. However, its gains are limited, likely due to low-resolution image inputs. Failure cases show that when dialogues reference tiny key regions (under 0.2% of the image), Gaze often produces inaccurate bounding boxes, confusing the Deliberate module. The CLIP-ViT-Large-Patch14-336 encoder further limits resolution, contributing to errors. On standard multimodal benchmarks, DiagNote performs comparably or slightly lower, as it targets complex multi-region dialogues without in-domain training data. Ablation details are in Appendix E.

6 Conclusion

In this paper, we focus on a key challenging task scenario for MLLMs—multi-turn multimodal dialogue. To address it, we first introduce a specially designed dataset, MMDiag, where accomplishing tasks requires properly integrating visual information across different regions of an image and connecting multimodal information across various QA pairs. This setting closely resembles natural conversations and poses significant challenges to current MLLMs. To solve this, we construct MMDiag across three distinct scenarios—everyday, tabular, and Minigrid—using a combination of rule-based methods and GPT-4o-mini to ensure robustness and diversity. Experimental results highlight challenges posed by MMDiag. Therefore, we propose DiagNote, an MLLM inspired by human visual processing, composed of two modules: Gaze and Deliberate. Deliberate performs reasoning step by step, with the assistance of Gaze, which provides annotations of salient regions to focus on. Experiments show that DiagNote enhances both grounding and reasoning capabilities, effectively addressing MMDiag challenges. We hope our work helps advance the development of more intelligent MLLMs.

Limitations

Although MMDiag contains diverse data, our methods can be expected to generate even more scenarios and complex questions, resulting in even more challenging datasets for multi-turn multimodal dialogue. Larger sub-graph patterns can be used to search for longer and more complex dialogues. While qualitative results and case studies demonstrate the effectiveness of our approach, there remains considerable room for improvement. The potential performance drops with the introduction of Gaze module may stem from failures in queries involving extremely tiny objects. Fine-tuning Gaze to abstain from answering when uncertain or replacing the vision encoder backbone may enhance its robustness. Further exploration of training paradigms and model architecture could also potentially lead to enhanced performance.

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*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023a. Qwen Technical Report. *arXiv preprint arXiv:2309.16609*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023b. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *arXiv preprint arXiv:2308.12966*.

Tom B. Brown. 2020. Language Models are Few-Shot Learners. *arXiv preprint arXiv:2005.14165*.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. 2021. Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts. In *CVPR*, pages 3558–3568.

Guangyao Chen, Siwei Dong, Yu Shu, Ge Zhang, Jaward Sesay, Börje F. Karlsson, Jie Fu, and Yemin Shi. 2024a. AutoAgents: A Framework for Automatic Agent Generation. In *IJCAI*.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024b. Humans or llms as the judge? a study on judgement biases. *arXiv preprint arXiv:2402.10669*.

Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, and 1 others. 2023. PaLI-X: On Scaling up a Multilingual Vision and Language Model. *arXiv preprint arXiv:2305.18565*.

Yangyi Chen, Karan Sikka, Michael Cogswell, Heng Ji, and Ajay Divakaran. 2024c. Dress: Instructing large vision-language models to align and interact with humans via natural language feedback. In *CVPR*.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. 2019. BabyAI: First Steps Towards Grounded Language Learning With a Human In the Loop. In *ICLR*.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. 2023. Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks. *CoRR*, abs/2306.13831.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Cursor. 2024. The AI Code Editor. https://www.cursor.com/.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, Jose M. F. Moura, Devi Parikh, and Dhruv Batra. 2017. Visual Dialog. In *CVPR*.

DeepL. 2024. Better writing with DeepL Write. https://www.deepl.com/en/write.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, and 1 others. 2023. PaLM-E: An Embodied Multimodal Language Model. *arXiv preprint arXiv:2303.03378*.

Yicheng Feng, Yuxuan Wang, Jiazheng Liu, Sipeng Zheng, and Zongqing Lu. 2024. LLaMA-Rider: Spurring Large Language Models to Explore the Open World. In *NAACL*, pages 4705–4724.

JaidedAI. 2024. EasyOCR. https://github.com/JaidedAI/EasyOCR.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. ReferItGame: Referring to Objects in Photographs of Natural Scenes. In *EMNLP*, pages 787–798.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, and 1 others. 2017. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *IJCV*, 123:32–73.

Noah Lee, Jiwoo Hong, and James Thorne. 2024. Evaluating the Consistency of LLM Evaluators. *arXiv preprint arXiv:2412.00543*.

Shengzhi Li and Nima Tajbakhsh. 2023. SciGraphQA: A Large-Scale Synthetic Multi-Turn Question-Answering Dataset for Scientific Graphs. *arXiv preprint arXiv:2308.03349*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*, pages 740–755. Springer.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved Baselines with Visual Instruction Tuning. In *CVPR*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024b. Visual Instruction Tuning. *NeurIPS*, 36.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and 1 others. 2023. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. *arXiv preprint arXiv:2303.05499*.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *ICCV*, pages 10012–10022.

Ziyu Liu, Tao Chu, Yuhang Zang, Xilin Wei, Xiaoyi Dong, Pan Zhang, Zijian Liang, Yuanjun Xiong, Yu Qiao, Dahua Lin, and 1 others. 2024c. MMDU: A Multi-Turn Multi-Image Dialog Understanding Benchmark and Instruction-Tuning Dataset for LVLMs. *arXiv preprint arXiv:2406.11833*.

Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In *ACL*.

OpenAI. GPT-4o mini: advancing cost-efficient intelligence.

Ji Qi, Ming Ding, Weihan Wang, Yushi Bai, Qingsong Lv, Wenyi Hong, Bin Xu, Lei Hou, Juanzi Li, Yuxiao Dong, and 1 others. 2024. CogCoM: Train Large Vision-Language Models Diving into Details through Chain of Manipulations. *arXiv preprint arXiv:2402.04236*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021a. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, pages 8748–8763. PMLR.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021b. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, pages 8748–8763. PMLR.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, and 1 others. 2022. LAION-5B: An open large-scale dataset for training next generation image-text models. *NeurIPS*, 35:25278–25294.

Paul Hongsuck Seo, Andreas Lehrmann, Bohyung Han, and Leonid Sigal. 2017. Visual Reference Resolution using Attention Memory for Visual Dialog. *NeurIPS*, 30.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. 2024. Visual CoT: Advancing Multi-Modal Language Models with a Comprehensive Dataset and Benchmark for Chain-of-Thought Reasoning. *arXiv preprint arXiv:2403.16999*.

Kurt Shuster, Samuel Humeau, Antoine Bordes, and Jason Weston. 2018. Image Chat: Engaging Grounded Conversations. *arXiv preprint arXiv:1811.00945*.

Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. 2019. Towards VQA Models That Can Read. In *CVPR*, pages 8317–8326.

Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. 2024. Large language models are inconsistent and biased evaluators. *arXiv preprint arXiv:2405.01724*.

Weihao Tan, Ziluo Ding, Wentao Zhang, Boyu Li, Bohan Zhou, Junpeng Yue, Haochong Xia, Jiechuan Jiang, Longtao Zheng, Xinrun Xu, and 1 others. 2024. Towards General Computer Control: A Multimodal Agent for Red Dead Redemption II as a Case Study. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Yunjie Tian, Tianren Ma, Lingxi Xie, Jihao Qiu, Xi Tang, Yuan Zhang, Jianbin Jiao, Qi Tian, and Qixiang Ye. 2024. ChatterBox: Multi-round Multimodal Referring and Grounding. *arXiv preprint arXiv:2401.13307*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal

Azhar, and 1 others. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, and 1 others. 2024. Gymnasium: A Standard Interface for Reinforcement Learning Environments. *arXiv preprint arXiv:2407.17032*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *NeurIPS*, 30.

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. *NeurIPS*, 35:24824–24837.

Xinrun Xu, Yuxin Wang, Chaoyi Xu, Ziluo Ding, Jiechuan Jiang, Zhiming Ding, and Börje F. Karlsson. 2024. A Survey on Game Playing Agents and Large Models: Methods, Applications, and Challenges. *arXiv preprint arXiv:2403.10249*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2023. A Survey on Multimodal Large Language Models. *arXiv preprint arXiv:2306.13549*.

Sipeng Zheng, Jiazheng Liu, Yicheng Feng, and Zongqing Lu. 2024. Steve-Eye: Equipping LLM-based Embodied Agents with Visual Perception in Open Worlds. In *ICLR*.

Sipeng Zheng, Bohan Zhou, Yicheng Feng, Ye Wang, and Zongqing Lu. 2025. UniCode: Learning a Unified Codebook for Multimodal Large Language Models. In *ECCV*, pages 426–443. Springer.

A Dataset

We use GPT-4o-mini (OpenAI) to generate our MMDiag dataset. Our dataset mainly consists of three parts: everyday scenes, tabular scenes, and Minigrid settings. We adopt different prompts for the generation of datasets under different scenes.

A.1 Dataset Collection

We design prompts for different scenarios, and the same devising ideas can be used in other scenarios for data collection.

Everyday Scenes. For everyday scenes, we generate our dataset from the Visual Genome dataset (Krishna et al., 2017). Since the original dataset has human-annotated attributes and relationship data, we extract the subsets that represent the QA pairs and feed them to GPT-4o-mini to generate corresponding dialogues. Figure 5,6,7 show several example prompts.

Please generate a new list based on a dictionary (`dict`) structured as follows:
[Image_Dict]

The resulting list should be structured as follows:
[Result_Dict]

Explanation:

There are two dictionaries in the generated list.

- The first dictionary's question is based on the relation to the first object in the `answer`. The first two items in the `CoT` (Chain of Thought) list correspond to the first list in `gnd`, breaking the question down into two steps of grounding reasoning. The final `CoT` item provides a complete and concise answer to the question.
- The second dictionary's question refers to the attributes of the object from the first question's answer and is presented using a pronoun. The first `CoT` item deduces the referent, the second extracts the attribute information, and the last item provides a complete and concise answer to the question. The `Question` and `CoT` answers should be diverse and natural. The `Query` contains a concise, detailed description of the object in that step, and `Bbox` includes the object's coordinates from `obj_info`.

Only output the dict in JSON format.

IMPORTANT: The order of objects in the CoT reasoning should follow the order of objects in the `gnd` list.

Human:**{Current_Image_Dict}**

Figure 5: The first example prompt for generating data samples in everyday scenes.

Tabular Scenes. For tabular scenes, we generate our dataset from the ChartQA dataset (Masry et al., 2022). In general, we use different types of graphs to capture various visualization intuitions, providing corresponding chart examples in the prompts. Figure 8 illustrates the main structure of the prompt, while Figure 9,10,11 show examples for line, pie, and bar charts, respectively.

Minigrid Settings. For Minigrid settings, we generate our dataset from the Minigrid database (Chevalier-Boisvert et al., 2023). Since we observe that GPT-4o-mini struggles to solve the mission without ground-truth planning, we first use BabyAI (Chevalier-Boisvert et al., 2019) to collect the plan needed to complete the mission for each environment generated by the Minigrid database.

Please generate a new `dict` based on the provided one. The provided `dict` is structured as follows:
[Image_Dict]

The generated `dict` should look like this:
[Result_Dict]

Explanation:

- The `Question` should be generated based on the `relation` predicates and the `attributes` of the last object in the `gnd`.
- The `CoT` (Chain of Thought) list's first three entries MUST correspond to the `gnd` objects list, which break the problem into three steps of grounding reasoning. The `Query` MUST correspond to the `gnd` objects list.
- The fourth item in the `CoT` list refers to the attributes of the target object.
- The last `CoT` entry provides a concise final answer to the question.
- The `Question` and `CoT.Ans` should be varied and natural. `Query` is a brief, specific description of the object, while `Bbox` corresponds to the object's `coordinates` in `obj_info`.

Only output the dict in JSON format.

IMPORTANT: The order of objects in the CoT reasoning should follow the order of objects in the `gnd` list.

Human:**{Current_Image_Dict}**

Figure 6: The second example prompt for generating data samples in everyday scenes.

Please generate a new `dict` based on the given one. The provided `dict` is structured as follows:
[Image_Dict]

The new `dict` should follow this structure:
[Result_Dict]

Explanation:

- The first `dict` asks a question based on the first object in the `relation[0]` and uses the first object from the `answer`. The `CoT` list contains step-by-step reasoning, aligning with the first item in `gnd`, breaking the problem into two steps of grounding reasoning. The final item in the `CoT` list provides a simple and concise answer to the question.
- The second `dict` asks about the attributes of the object answered in the first question, referring to it with a pronoun. The first `CoT` item infers the referred object, the second item extracts the attributes, and the final item provides a full, concise answer.
- The third `dict` asks a question about the related object from `relation[1]`, again referring to it with a pronoun. The `CoT` steps involve reasoning to identify the referred object and then the related object, ending with a complete, concise answer.

IMPORTANT: The order of objects in the CoT reasoning must match the order of objects in the `gnd` list.

Human:**{Current_Image_Dict}**

Figure 7: The third example prompt for generating data samples in everyday scenes.

We then combine the positions of all objects with the mission and plan, as shown in Figure 12, and feed them to GPT-4o-mini. For details, Minigrid creates environments based on specific constraints, saving grid world data as both rendered images and lists of special objects with bounding boxes. BabyAI then identifies feasible solutions by analyzing the agent's field of view and determining subgoal-aligned actions. To simplify QA generation, we make the entire grid world visible, allowing MLLMs to guide the agent from a top-down perspective. GPT-4o-mini then generates natural questions, reasoning steps, key region queries, and concise final answers. The prompt structure is illustrated in Figure 13.

A.2 Dataset Format

Examples of the final MMDiag dataset are shown in Figure 14,15,16. Figure 14a,15a,16a display

Please generate a new list based on the provided chart and table data. The main reference should be the chart content, as the table content might contain errors. The format of the new list should be similar to the following example:
[QA_and_CoT]

This list consists of two dictionaries corresponding to two rounds of Q&A. Each question is based on the chart, providing a reasoning process and an answer. The CoT (Chain of Thought) consists of multiple steps with "Ans" representing the answer broken down into steps, and "Query" indicating the key terms in the chart relevant to that step. The final step of CoT provides a complete and concise answer to the question, and the "Query" highlights the key terms in the chart that are relevant to the question.

The Question and CoT answers should be diverse and natural.

Important: The second question should refer back to the answer from the first question, meaning that you can't answer the second question unless you know the answer of the first question. The answer of the first question is presented using a pronoun in the second question, and shouldn't appear in the second question.
You only need to output the list in JSON format.

Human:**{Current_QA_and_CoT}**

Figure 8: The prompt structure to generate samples in tabular scenes.

```
[
  {
    "Question": "In which year did the highest percentage of voters care about the election outcome, and what was the percentage?",
    "CoT": [
      {
        "Ans": "To solve this, we should first find the highest point of the brown line, which is 83.",
        "Query": "83"
      },
      {
        "Ans": "Next, we can identify that this occurred in 2020.",
        "Query": "2020"
      },
      {
        "Ans": "In 2020, 83% of voters cared the most about the election result.",
        "Query": "83"
      }
    ]
  },
  {
    "Question": "What percentage of voters didn't care about the election result four years before that year?",
    "CoT": [
      {
        "Ans": "The referenced year is 2020 from the previous question, and four years earlier would be 2016.",
        "Query": "2016"
      },
      {
        "Ans": "The yellow line in 2016 indicates a value of 22.",
        "Query": "22"
      },
      {
        "Ans": "In 2016, 22% of voters did not care about the election outcome.",
        "Query": "22"
      }
    ]
  }
]
```

Figure 9: The question-answer (QA) and Chain-of-Thought (CoT) examples for line charts.

the original images from the source datasets and environments, while Figure 14b,15b,16b show the data format of MMDiag generated by GPT-4o-mini and standardized according to specific rules.

A.3 Evaluation

Since GPT-4o-mini contributes to generating our datasets, we use Gemini-1.5-Pro (Reid et al., 2024) for evaluation. There are multiple reasons for choosing it for this task: answer formatting and the Chain of Thought (CoT) processes may be diverse, making a simple similarity score insufficient for evaluation. Additionally, recent works (Liu et al., 2024b; Zheng et al., 2024) commonly apply LLMs for judgment. We provide the MLLM with images, ground-truth answers, and generated

```
[
  {
    "Question": "What did most Americans favor when it comes to spending on policing, and what was the percentage?",
    "CoT": [
      {
        "Ans": "To solve this, we should first locate the largest part of the pie chart, which is 42%.",
        "Query": "42"
      },
      {
        "Ans": "Next, we can see that this part represents people who favored maintaining the same level of spending on policing.",
        "Query": "Stay about the same"
      },
      {
        "Ans": "The largest group, with 42%, favored maintaining current spending levels on policing.",
        "Query": "42"
      }
    ]
  },
  {
    "Question": "How does this group compare to those who favored reduced spending?",
    "CoT": [
      {
        "Ans": "This group refers to the one mentioned in the previous answer, which represents 42%.",
        "Query": "42"
      },
      {
        "Ans": "Now, we need to compare it with those who favored reduced spending, indicated by the label 'Decreased'.",
        "Query": "Decreased"
      },
      {
        "Ans": "The portion of people who favored reduced spending is represented by the purple section of the pie chart, at 25%.",
        "Query": "25"
      },
      {
        "Ans": "The difference in percentage is 42 - 25 = 17.",
        "Query": ""
      },
      {
        "Ans": "This group is 17 percentage points larger than those who favored reduced spending.",
        "Query": "17"
      }
    ]
  }
]
```

Figure 10: The question-answer (QA) and Chain-of-Thought (CoT) examples for pie charts.

responses, and ask it to score the accuracy of the generated answers across five categories. We notice that the MLLM provides more reasonable rankings when asked to explain the 'ad-hoc' reason before their final score. As a result, we include this reasoning step in the prompt, as shown in Figure 17.

B DiagNote

Our DiagNote consists of two MLLMs, one for Deliberate, and one for Gaze. For each input question, DiagNote appends buffer information and queries to the respective prompts for Deliberate and Gaze. For images from Minigrid, a description of the Minigrid environment, as shown in Figure 20, is included in both training and testing. The remaining components of the Deliberate prompt and Gaze prompt are consistent across all three scenes.

Deliberate Prompt. For deliberating, DiagNote provides the dialogue context and Chain of Thought (CoT) history for the current question in the prompt, as shown in Figure 21. When the 'END' token appears in the latest 'Query' from the Deliberate module, signaling the end of the CoT process, DiagNote provides a new prompt, as

```
[
  {
    "Question": "Which region had the second smallest consumption of Ozone-Depleting Substances in tonnes in 1998?",
    "CoT": [
      {
        "Ans": "To solve this, we first need to find the second smallest consumption in tonnes, which is 143 tonnes.",
        "Query": "143 tonnes"
      },
      {
        "Ans": "Next, we can determine that this bar refers to Malta.",
        "Query": "Malta"
      },
      {
        "Ans": "In 1998, Malta had the second smallest consumption of Ozone-Depleting Substances, with 143 tonnes.",
        "Query": "143 tonnes"
      }
    ]
  },
  {
    "Question": "How many times greater was the highest consumption of Ozone-Depleting Substances compared to that region?",
    "CoT": [
      {
        "Ans": "The region in question is Malta, with 143 tonnes.",
        "Query": "143 tonnes"
      },
      {
        "Ans": "The highest consumption to compare it with is 2,262 tonnes.",
        "Query": "2,262 tonnes"
      },
      {
        "Ans": "The ratio is calculated as 2,262 / 143 = 15.8.",
        "Query": ""
      },
      {
        "Ans": "The region with the highest consumption used 15.8 times more Ozone-Depleting Substances than Malta.",
        "Query": ""
      }
    ]
  }
]
```


Figure 11: The question-answer (QA) and Chain-of-Thought (CoT) examples for bar charts.


```
{
  "mission": "open the grey door, then open the green door",
  "object": {
    "grey door": "[256, 320, 288, 352]",
    "red triangle agent": "[288, 288, 320, 320]",
    "green door": "[320, 288, 352, 320]"
  },
  "plan_list": [
    [
      "Actions.left",
      "(GoNextToSubgoal: grey door None, reason: Open)"
    ],
    [
      "Actions.forward",
      "(GoNextToSubgoal: grey door None, reason: Open)"
    ],
    [
      "Actions.left",
      "(GoNextToSubgoal: grey door None, reason: Open)"
    ],
    [
      "Actions.toggle",
      "(OpenSubgoal)"
    ],
    [
      "Actions.left",
      "(GoNextToSubgoal: green door None, reason: Open)"
    ],
    [
      "Actions.forward",
      "(GoNextToSubgoal: green door None, reason: Open)"
    ],
    [
      "Actions.toggle",
      "(OpenSubgoal)"
    ]
  ]
}
```


Figure 12: The mission and plan input example of Minigrid settings.

Based on the provided image and the given mission and object information, generate a new dict. The provided image is a grid world, where gray squares represent impassable walls, black squares are the background color, and the agent is a red triangle, with the pointed tip indicating the initial direction the agent is facing. Different squares may contain various objects such as colored balls, keys, boxes, doors, etc. The mission provides the task that the agent needs to accomplish, the plan list provides the action and subgoal for each step, and the object provides the coordinates of these objects. The format of mission and object is as follows:
[Mission_and_Plan]

The format of the new dict should be similar to the following example:
[QA_and_CoT]

Each dict should consist of a Question, a CoT (Chain of Thought) process, and a Final_Ans. The Question is generated based on the mission. The CoT consists of multiple steps, where each step has "Ans" for the explanation ,"Query" for identifying the key elements in the image relevant to that step and "Bbox" for the coordinates of the object in "Query". The Final_Ans provides a clear and concise solution to the question, with the "Query" highlighting the key terms in the image corresponding to the solution.

Ensure the Question, CoT answers, and Final_Ans are diverse and natural.
The Bbox should contains all the bounding boxes of the Query.
Output the dict in JSON format only.

Human:**{Current_QA_and_CoT}**

Figure 13: The prompt structure to generate data samples in Minigrid settings.

shown in Figure 22, to the Deliberate module for generating the final answer.

Gaze Prompt. For gazing, DiagNote extracts the 'Query' from the output of the Deliberate module and provides it to the Gaze module along with the prompt shown in Figure 23. The output from the Gaze module, which includes the bounding box of the query, is then saved in the Deliberate buffer to support the next turn of Deliberating.

C Implementation

The detailed parameters of implementation are shown in Table 5,6.

D Qualitative Comparison of Grounding

Figure 18,19 show a comparison of grounding ability between DiagNote and Grounding DINO (Liu et al., 2023). As illustrated in Figure 18b, Grounding DINO struggles with grounding tasks involving Optical Character Recognition (OCR). In contrast, DiagNote leverages the generalization capability of LLMs, enabling it to effectively locate the target words, as shown in Figure 18a. Figure 19b illustrates that Grounding DINO fails to handle objects with attributes. Although the grey key has a marginally higher confidence, accurately locating the 'grey' key in the image confuses Grounding DINO. In contrast, DiagNote accurately identifies the grey key in Figure 19a, which aids the subsequent actions of the Deliberate module.

E Ablation Study

We observe a counterintuitive performance trend in Table 3 in the main paper: Gaze provides only limited performance gains and, in some cases, even reduces performance, particularly in tabular and Minigrid scenarios. As shown in Figure 24, Gaze incorrectly identifies the bounding box for a critical but tiny piece of information—the year 2019—misleading Deliberate to focus on the wrong color bar. This issue accounts for most failure cases.

To further analyze this, we evaluate the propor-

hyper-parameters	value
deepspeed	zero3
base model	LLaVA-1.5-7B
conversation template	Vicuna v1
vision tower	CLIP-ViT-Large-Patch14-336
modality projector type	mlp2x_gelu
image aspect ratio	pad
training epochs	1
training batch size	16
learning rate	2e-5
weight decay	0
warm-up ratio	0.03
model max length	2048
data loader workers	4

Table 5: The implementation details of the Deliberate module.

hyper-parameters	value
deepspeed	zero3
base model	LLaVA-1.5-7B
conversation template	Vicuna v1
vision tower	CLIP-ViT-Large-Patch14-336
modality projector type	mlp2x_gelu
layer selected for fine-tuning vision tower	-2
image aspect ratio	pad
training epochs	1
training batch size	32
learning rate	2e-5
weight decay	0
warm-up ratio	0.03
model max length	2048
data loader workers	4
fine-tune vision tower	True/False

Table 6: The implementation details of the Gaze module.

Model	Fine-tuning Data	Gaze	T1	T2	T3	T4
Visual CoT-13B	MMDiag	-	2.00	1.43	0.40	0.95
DiagNote-14B	MMDiag	✗	3.15	2.35	1.78	1.23
DiagNote-14B	MMDiag	✓	4.20	3.10	2.55	1.95

Table 7: Tabular scenes results of MLLMs fine-tuned on MMDiag, using the same evaluation metrics as the previous evaluation.

Benchmark	MMBench	MM-Vet	RefCOCO+	RefCOCOg
DiagNote-14B	63.7	28.5	0.834	0.775

Table 8: DiagNote performance on general datasets.

Scenario	$\leq 0.2\%$	$\leq 1\%$	$\leq 3\%$	$\leq 5\%$	$\leq 10\%$
Everyday	7.57%	27.62%	47.99%	57.49%	69.91%
Tabular	87.17%	99.24%	99.80%	99.92%	100%
Minigrid	6.98%	66.61%	96.99%	99.41%	100%

Table 9: MMDiag tiny key regions percentage.

tion of tiny key regions across different scenarios in MMDiag (Table 9). In tabular and Minigrid scenes, nearly all key regions occupy less than 3% of the total image area, making them particularly challenging for Gaze to detect accurately. To mitigate this, we curate an alternative test dataset for tabular scenes, excluding questions that require attention to extremely small regions. We then fine-tune Visual CoT and DiagNote with MMDiag and evaluate them on this revised tabular split. As shown in Table 7, Gaze's impact becomes more pronounced. Table 8 demonstrates that DiagNote performs comparably or slightly lower on standard multimodal benchmarks, as it targets complex multi-region dialogues without in-domain training data.

F Qualitative Comparison of Multi-Turn Multimodal Dialogue

We present several cases comparing models in everyday scenarios and tabular scenes. Figure 25,26 show examples from unseen everyday scenarios. In Figure 25, CogCoM (Qi et al., 2024) completely fails to answer the two-turn questions correctly. Despite the assistance of the counting expert, CogCoM is unable to answer the first counting question. Although LLaVA-1.5-13B (Liu et al., 2024a) and Visual CoT (Shao et al., 2024) can answer the first questions accurately, both encounter hallucinations when responding to the second question, mistakenly identifying white plates as cups and bowls, respectively. In contrast, our DiagNote performs well on both questions, demonstrating the effectiveness of the Gaze module in ensuring DiagNote stays grounded in visual details. In Figure 26, CogCoM fails to provide a clear answer to the first

question, instead offering a confusing single word 'jean'. Again, LLaVA-1.5-13B and Visual CoT answer the first question correctly, but imagine the man was holding a frisbee. Both CogCoM and DiagNote understand the context, with DiagNote accurately describing the can based on the visual details. In contrast, CogCoM mistakenly assumes it is a can of beer, which may not be the case.

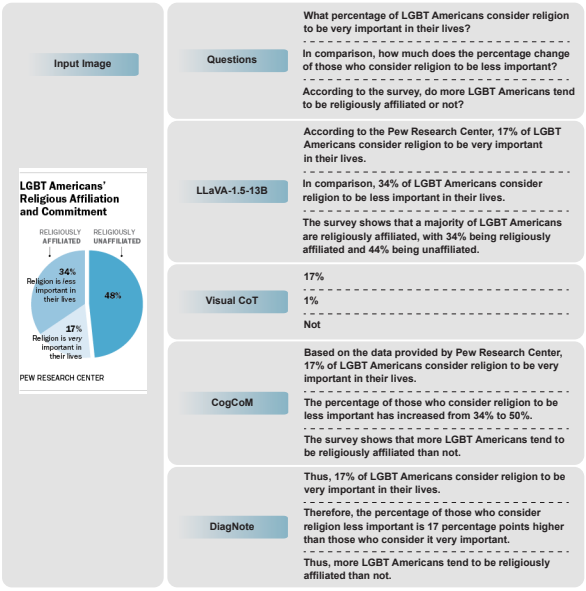


Figure 27: One example of comparison between different MLLMs under tabular scenes.

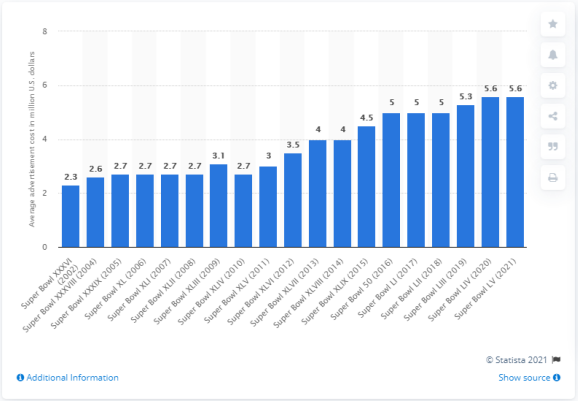
Figure 27 presents examples of unseen tabular scenes. All models answer the first question correctly. However, Visual CoT provides a completely incorrect answer to the second question, while CogCoM introduces an unfounded '50%'. LLaVA-1.5-13B correctly identifies the visual detail '34%', but overlooks the keyword 'change' in the question, which requires a calculation between two percentages. Only DiagNote answers the question precisely. The final question requires the models to understand the entire pie chart. The model should compare the sum of two parts on the right side of the pie chart with the left part to obtain the final answer 'yes'. Visual CoT fails to provide this correct answer, and LLaVA-1.5-13B misinterprets the unaffiliated percentage and derives an incorrect affiliated percentage. Both CogCoM and DiagNote reach the right conclusion. Overall, DiagNote performs well on all questions, demonstrating its ability to focus on both visual and language details and to comprehend the full picture the chart conveys. This strong ability can be attributed to the Gaze and Deliberate structure, which enables it to zoom in on specific details while integrating multimodal information for a holistic understanding.

(a) the original image


```
{
  "QA_pairs": [
    {
      "Question": "What's the woman holding?",
      "CoT": [
        {
          "Ans": "To address this question, we should first identify the woman.",
          "Query": "woman",
          "Bbox": [211, 46, 478, 255]
        },
        {
          "Ans": "Next, we can observe that she is holding a cup.",
          "Query": "END"
        },
        {
          "Final_Ans": "The woman is holding a cup.",
          "Query": "cup",
          "Bbox": [309, 118, 338, 154]
        }
      ]
    },
    {
      "Question": "Could you describe it in detail?",
      "CoT": [
        {
          "Ans": "The 'it' in the question refers to the cup from the previous question, so
we first need to locate the cup.",
          "Query": "cup",
          "Bbox": [309, 118, 338, 154]
        },
        {
          "Ans": "We can see that the cup is made of paper.",
          "Query": "END"
        },
        {
          "Final_Ans": "The cup is a paper cup.",
          "Query": "cup",
          "Bbox": [309, 118, 338, 154]
        }
      ]
    }
  ],
  "image": "2353699.jpg",
  "question_id": 16
}
```


(b) the sample format

Figure 14: One example of the original image and the generated sample from Visual Genome in JSON format.

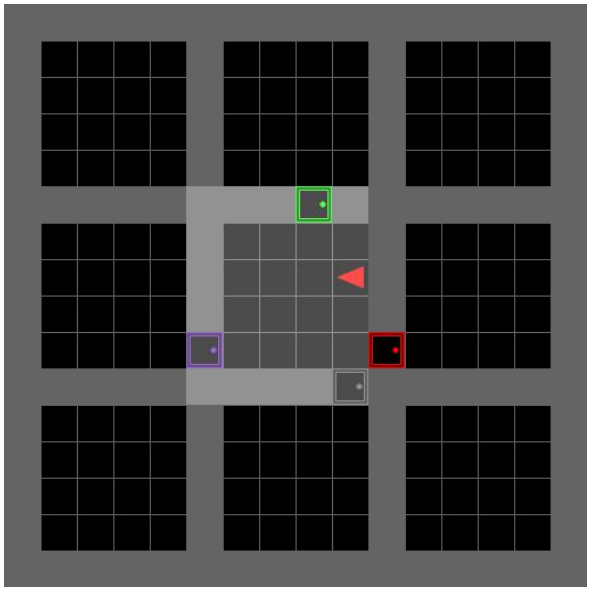


(a) the original image


```
{
  "QA_pairs": [
    {
      "Question": "What was the average advertisement cost during Super Bowl XXXVI
(2002)?",
      "CoT": [
        {
          "Ans": "To find the average advertisement cost for Super Bowl XXXVI, we look
at its specific entry.",
          "Query": "Super Bowl XXXVI (2002)"
        },
        {
          "Ans": "The average cost listed is 2.3 million U.S. dollars.",
          "Query": "END"
        },
        {
          "Final_Ans": "Thus, the average advertisement cost during Super Bowl XXXVI
was 2.3 million U.S. dollars.",
          "Query": "2.3 million U.S. dollars"
        }
      ]
    },
    {
      "Question": "How much more did the average advertisement cost for Super Bowl
LV (2021) compared to that event?",
      "CoT": [
        {
          "Ans": "The average advertisement cost for Super Bowl LV is 5.6 million U.S.
dollars.",
          "Query": "5.6 million U.S. dollars"
        },
        {
          "Ans": "The cost difference can be calculated as 5.6 - 2.3.",
          "Query": ""
        },
        {
          "Ans": "This results in a difference of 3.3 million U.S. dollars.",
          "Query": "END"
        },
        {
          "Final_Ans": "Therefore, the average advertisement cost for Super Bowl LV was
3.3 million U.S. dollars more than that event.",
          "Query": ""
        }
      ]
    }
  ],
  "image": "two_col_383.png"
}
```


(b) the sample format

Figure 15: One example of the original image and the generated data point from ChartQA in JSON format. The bounding boxes of the queries are generated using EasyOCR (JaidedAI, 2024) and thus are not shown in the example.



(a) the original image


```
{
  "id": "BabyAI-OpenDoorsOrderN4-v0_185",
  "QA_pairs": {
    "Question": "How can the agent open the green door first, and then open the grey
door?",
    "CoT": [
      {
        "Ans": "To solve this, we first need to locate the agent's position.",
        "Query": "red triangle agent",
        "Bbox": [288, 224, 320, 256]
      },
      {
        "Ans": "Next, we need to find the green door.",
        "Query": "green door",
        "Bbox": [256, 160, 288, 192]
      },
      {
        "Ans": "To open the green door, the agent should move forward, then turn right,
move forward again, and finally toggle to open the door.",
        "Query": "green door",
        "Bbox": [256, 160, 288, 192]
      },
      {
        "Ans": "Now, we need to locate the grey door.",
        "Query": "grey door",
        "Bbox": [288, 320, 320, 352]
      },
      {
        "Ans": "To go to the grey door, the agent should turn right, move forward, turn
right again, and move forward several times to reach the grey door, then toggle to
open it.",
        "Query": "END"
      },
      {
        "Final_Ans": "The agent first needs to move forward, turn right, move forward
again to open the green door. Then, it should turn right, move forward, turn right again,
move forward several times, and finally open the grey door.",
        "Query": "grey door",
        "Bbox": [288, 320, 320, 352]
      }
    ]
  },
  "image": "BabyAI_frame_0_with_action_full_obs_with_attr/BabyAI-OpenDoor-
sOrderN4-v0/185.jpg"
}
```


(b) the sample format

Figure 16: One example of the original image and the generated sample from Minigrid in JSON format.

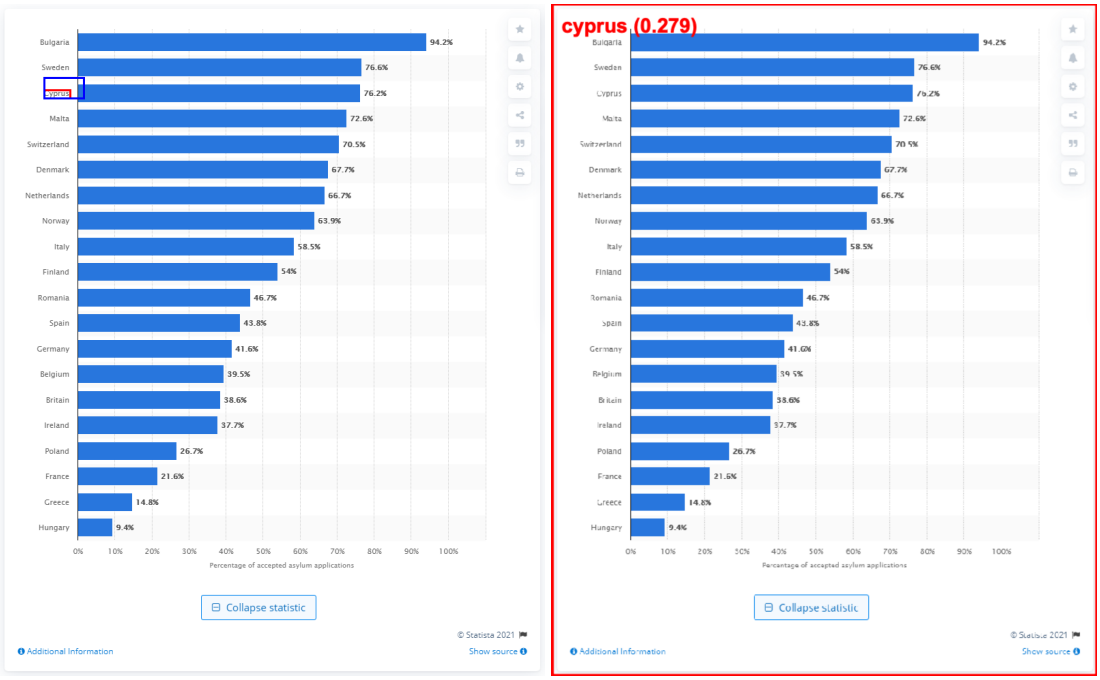
You are an evaluator. Your task is to assess the given answer based on its accuracy in response to the provided picture, related question, and the ground truth answer. Your evaluation should be based on ad-hoc reasoning. First, provide a detailed reasoning for your judgment, then explicitly state the final category in the format:
Reason: ... Judgment: ...
Use the following five categories for your judgment:
Incorrect: The answer is entirely wrong or unrelated.
Partially Correct: The answer contains some relevant elements but is mostly incorrect.
Medium: The answer captures partial correctness but lacks significant details or has notable inaccuracies.
Almost Correct: The answer is mostly accurate but has minor errors or omissions.
Correct: The answer is fully accurate and aligns well with the ground truth.
[Please give a detailed Chain-of-Thought process.]

Question: **{Question}**

Ground Truth Answer: **{GroundTruthAnswer}**

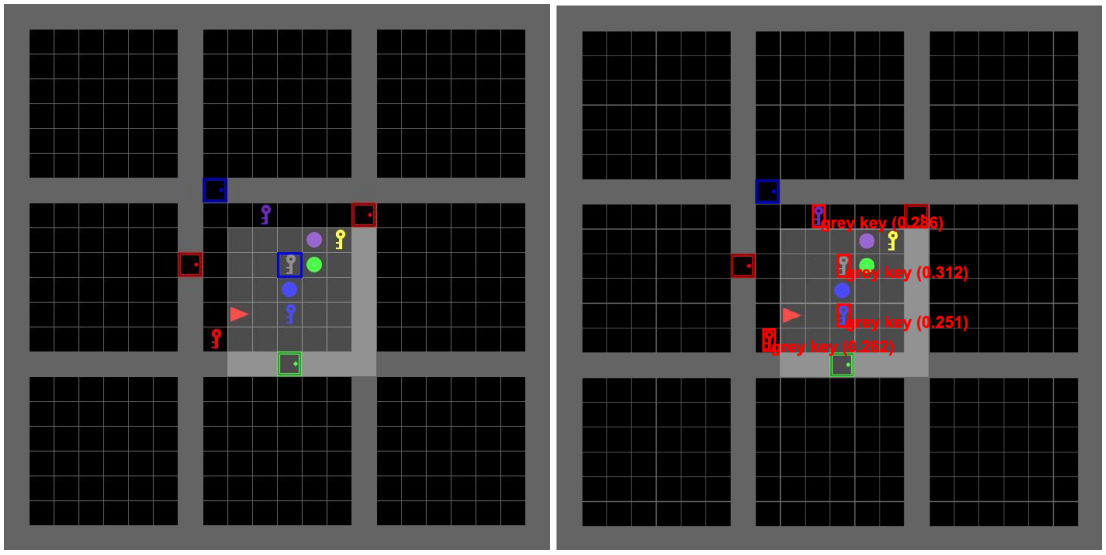
Given Answer: **{GivenAnswer}**

Figure 17: The evaluation prompt structure given to Gemini-1.5-Pro. The content in '[]' is added when the CoT process is evaluated.



(a) DiagNote (b) Grounding DINO

Figure 18: The grounding comparison between Grounding DINO and the Gaze module of DiagNote in Tabular Scene. The grounding query is "Cyprus". The red bounding box in (a) is the ground-truth answer, while the blue one is the bounding box generated by our Gaze module. The red bounding box in (b) is the output of Grounding DINO.



(a) DiagNote (b) Grounding DINO

Figure 19: The grounding comparison between Grounding DINO and the Gaze module of DiagNote in Minigrid Scene. The grounding query is "grey key". The blue bounding box in (a) is generated by the Gaze module of DiagNote, which overlaps the ground-truth red bounding box. Meanwhile, the red bounding box in (b) is the output of Grounding DINO.

The provided image depicts a grid world, with gray squares representing impassable walls, black squares as the background color, and the agent represented as a red triangle, with its pointed tip indicating the initial direction it is facing.

Various squares may contain different objects such as colored balls, keys, boxes, and doors.

Figure 20: The description of Minigrid Scene added to the prompts.

This is the context, which includes the previous questions and answers related to the image.
{Context}

Current problem and its reasoning process (CoT) includes reasoning, focused objects, and their bounding boxes (bbox):
{Question_and_CoT}

Please provide the detailed reasoning process and focused objects for the question in dictionary form, outputting one dictionary in the following format each time: {{'Ans': , 'Query': }}. Here, Ans represents the current reasoning step, and Query indicates the object of focus. If the reasoning is complete, set 'Query' to END.

Figure 21: The prompt structure of the Deliberate module when the last Query output of the Deliberate module is not 'END'.

This is the context, which includes the previous questions and answers related to the image.
{Context}

Current problem and its complete reasoning process (CoT) includes reasoning, focused objects, and their bounding boxes (bbox):
{Question_and_CoT}

Please provide a concise answer to the question in dictionary form, outputting a dictionary in the following format: {{'Final_Ans': , 'Query': }}, where Final_Ans is the concise answer to the question, and Query is the core object of focus in the image related to the question.

Figure 22: The prompt structure of the Deliberate module when the last Query output of the Deliberate module is 'END'.

Please refer to the image and provide the exact rectangular coordinates of the **{Query}** in the format of a four-dimensional integer array [x1, y1, x2, y2], where x1 and y1 represent the coordinates of the top-left corner of the rectangle, and x2 and y2 represent the coordinates of the bottom-right corner.

Figure 23: The prompt structure of the Gaze module.

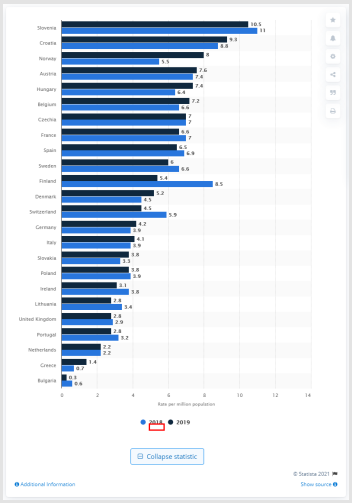
Questions	**Which country had the highest rate per million population in 2019?**
Deliberate One	**To find the highest rate in 2019, we look at the values for that year.** **Query: 2019 values** **Bbox: [337, 969, 373, 985]**
Bounding Box	

Figure 24: The second example of comparison between different MLLMs under everyday scenes.

Input Image	
Questions	**How many orange plates are there in front of the girl?** **What are placed above them?**
LLaVA-1.5-13B	**There are two orange plates in front of the girl.** **There are two cups placed above the orange plates.**
Visual CoT	**There are two orange plates in front of the girl.** **There are two bowls placed above the orange plates.**
CogCoM	**1** **The image captures a lively scene in a restaurant. A family of four is seated at a table, their attention directed towards the young boy who sits between them. He's engaged with his food, holding a fork and spoon in his hands as he enjoys his meal. \n\nThe table is adorned with plates filled with various dishes, each one adding to the vibrant color scheme of the room. There are glasses on the table, perhaps used for drinking or serving. \n\nIn the background, you can see other diners, suggesting that this is not just a family outing but also an event where others are present. The walls are painted in a warm yellow hue, providing a welcoming atmosphere to all those who enter.\n\nOverall, the image paints a picture of a happy moment shared by a family in a cozy setting.**
DiagNote	**There are two orange plates in front of the girl.** **The two white plates are placed above the orange plates.**

Figure 25: The first example of comparison between different MLLMs under everyday scenes.

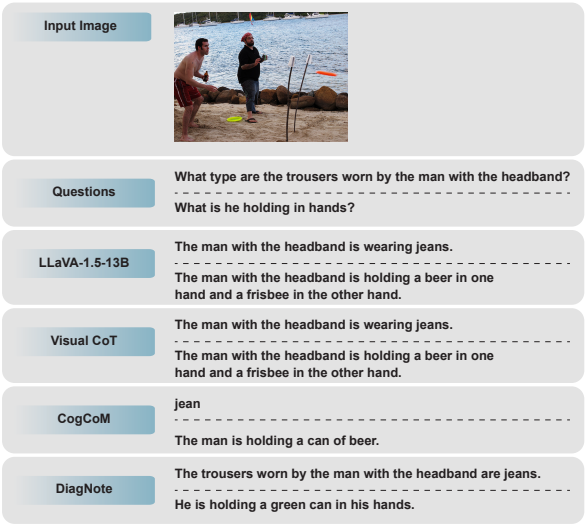


Figure 26: The second example of comparison between different MLLMs under everyday scenes.