DART: Leveraging Multi-Agent Disagreement for Tool Recruitment in Multimodal Reasoning

Nithin Sivakumaran[1]   **Justin Chih-Yao Chen[1]**   **David Wan[1]**   **Yue Zhang[1]**
Jaehong Yoon[2]   **Elias Stengel-Eskin[3]**   **Mohit Bansal[1]**
[1]UNC Chapel Hill   [2]Nanyang Technological University
[3]The University of Texas at Austin

Abstract

Specialized visual tools can augment large language models or vision language models with expert knowledge (e.g., grounding, spatial reasoning, medical knowledge, etc.), but knowing which tools to call (and when to call them) can be challenging. We introduce DART, a multi-agent framework that uses disagreements between multiple debating visual agents to identify useful visual tools (e.g., object detection, OCR, spatial reasoning, etc.) that can resolve inter-agent disagreement. These tools allow for fruitful multi-agent discussion by introducing new information, and by providing tool-aligned agreement scores that highlight agents in agreement with expert tools, thereby facilitating discussion. We utilize an aggregator agent to select the best answer by providing the agent outputs and tool information. We test DART on four diverse benchmarks and show that our approach improves over multi-agent debate as well as over single agent tool-calling frameworks, beating the next-strongest baseline (multi-agent debate with a judge model) by 3.4% and 2.4% on A-OKVQA and MMMU respectively. We also find that DART adapts well to new tools in applied domains, with a 1.3% improvement on the M3D medical dataset over other strong tool-calling, single agent, and multi-agent baselines. Additionally, we measure text overlap across rounds to highlight the rich discussion in DART compared to existing multi-agent methods. Finally, we study the tool call distribution, finding that diverse tools are reliably used to help resolve disagreement.[1]

1 Introduction

A key strength of human intelligence is the ability to debate and discuss our reasoning with others. However, everyone has had the experience of engaging in a debate that goes nowhere because of a lack of new information: without the incorporation of new knowledge, each party becomes entrenched in their point of view. While past work (Wang et al., 2024a; Chen et al., 2024b; Estornell et al., 2025a; Estornell and Liu, 2024) has found that large-language model agents can improve their performance by engaging in multi-agent debate, this has largely been explored in unimodal, text-only settings. We find that when extending debate to multimodal settings with vision-language models (VLMs), the discussion between agents frequently becomes unproductive. The lack of progress in conversation is in part due to a lack of new information, with the VLM-based agents participating in the discussion often being unable to resolve perceptual differences (see Fig. 1.A). Moreover, each VLM suffers from similar perceptual limitations: past work has found that VLMs often lag behind specially-trained expert models in domains like grounding, OCR, and spatial reasoning (Chiu et al., 2025; Chen et al., 2024a). Indeed, because of these limitations, a separate line of work has explored using LLMs to call expert models as tools rather than relying on VLMs (see Fig. 1.B) (Yang et al., 2023; Lu et al., 2023; Wu et al., 2023; Surís et al., 2023; Mialon et al., 2023; Yu et al., 2025b), benefiting from the reasoning abilities of LLMs and the visual perceptual abilities of expert models.

We unify these three approaches (VLMs, multi-agent debate, and tool calling) with **D**isagreement among **A**gents for **R**ecruitment of **T**ools (DART), a multi-agent approach which approaches visual question-answering (VQA) tasks by engaging VLMs in a multi-agent debate, resolving the disagreement between agents with information from expert tools (see Fig. 1.C). This has several key advantages: first, the novel information from expert tools helps move the discussion between agents forward productively and augments their perceptual abilities. Second, it simplifies the tool-calling process; unlike top-down LLM-based solutions which plan what tools to execute in a single step based

---

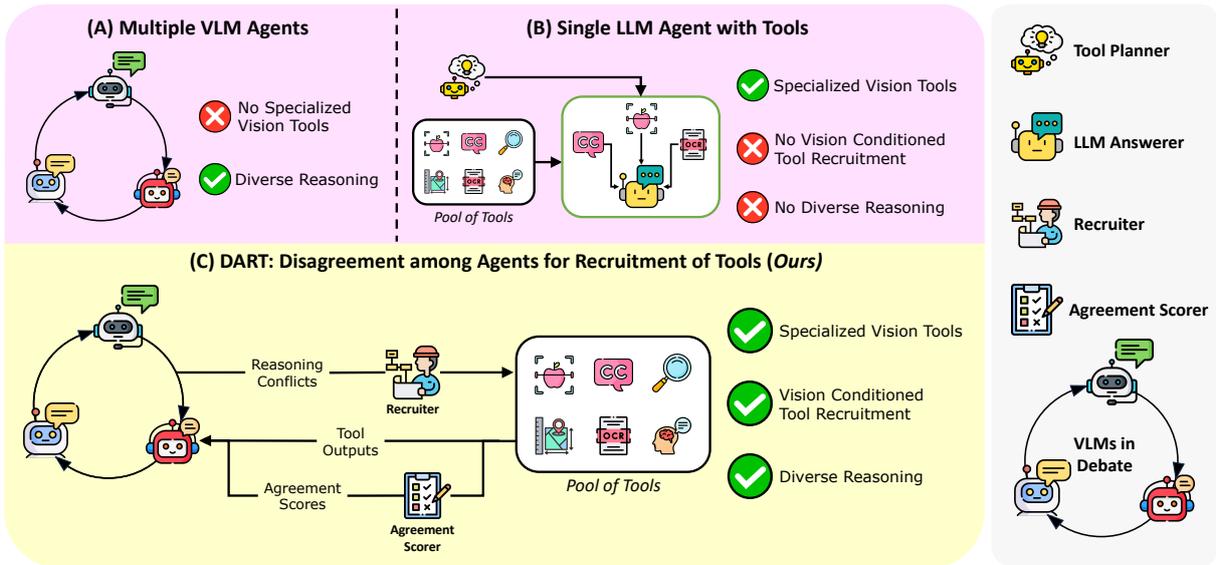[1]Code: https://github.com/nsivaku/dart

Figure 1: Previous work has explored using (A) multiple agents in debate to refine their reasoning, but this approach is limited to the abilities of the agents. Alternatively, some methods employ a (B) top-down LLM agent that invokes vision tools, yet they plan tool usage based solely on the question and overlook the visual information itself. In our method (C), we facilitate a discussion among multiple agents with targeted intervention from a pool of vision tools. These tools address disagreements detected in a debate of VLM agents, with their specialized vision outputs and agreement scores being used for future discussion.

only on the question – e.g., ViperGPT (Surís et al., 2023), Chameleon (Lu et al., 2023) – DART adaptively calls tools based only on the points which the VLMs disagree on. This has the added benefit of combining the strong reasoning abilities of VLMs (backed by LLMs) with the specialized perception of vision tools. Finally, the added information given by tools serves as an intuitive measure of confidence in each agent: the more a given agent agrees with the expert tools (which have stronger perceptual abilities), the more it can be relied on.

DART consists of five steps, outlined in Fig. 2. DART starts with (1) *Initial Answer Generation*, where each agent creates an answer and reasoning for an image-question pair. Next, our method uses the answers and reasoning chains from the answering agents in (2) *Tool-Based Disagreement Resolution*, which aims to resolve disagreements that occurred during the initial answering step by invoking tools based on the identified conflicts. Then, in (3) *Tool-Based Agreement Scoring*, DART compares the expert tool outputs and agent answers to calculate agreement scores. These new tool outputs and per agent agreement scores are used during (4) *Discussion* where agents refine their own answer and reasoning. Finally, DART composes the tool outputs, recalculated agreement scores, and post-discussion answer outputs into a final answer in the (5) *Aggregation* step.

We test DART on four different benchmarks that require strong and diverse perception capabilities and reasoning. We find that DART improves over the next best baseline (multi-agent debate with a judge model) by 3.4% on A-OKVQA (Schwenk et al., 2022), 2.4% on MMMU (Yue et al., 2024), and 1.2% on NaturalBench (Li et al., 2024). Similarly, DART outperforms tool-calling baselines like ViperGPT (Surís et al., 2023) and Chameleon (Lu et al., 2023) on A-OKVQA, MMMU, and NaturalBench by 19.0%, 11.5%, and 5.3% respectively. We also test how DART adapts to a subject-specific domain, adding in a relevant domain expert tool. Specifically, we test DART's adaptability to medical data by testing on the M3D dataset (Bai et al., 2024) while adding in MedGemma 4B (Google, 2025) into the expert tool set. We find that our system improves performance on M3D by 1.3% over Qwen2.5-VL (Bai et al., 2025), the strongest single agent baseline, and achieve positive gains in a multi-agent setting compared to naive multi-agent debate (which decreases by 2.7% compared to Qwen2.5-VL). Our results indicate that DART leads to stronger discussion, introducing more novel ideas into the discussion round compared to multi-agent debate. Moreover, DART benefits from calling a variety of tools based on reasoning disagreements from multiple models, without overly relying on any singular tool.

2 Method

The problem setup is a Visual Question Answering (VQA) task where we use a set of VLM agents and tools in DART to arrive at a final answer. The framework takes in an image and question as input and generates a single natural language answer. Formally, given a question Q and image I, we aim to generate a single answer a based on a discussion among generalist answering agents $\mathbf{A} = \{A_i\}_{i=1}^n$ and vision expert tools $\mathbf{V} = \{V_i\}_{i=1}^m$.

2.1 Initial Answer Generation

In initial answer generation (see Fig. 2 (Step 1), we prompt each answering agent A_i to use CoT to generate a candidate answer a_i, a corresponding reasoning chain r_i, and self-reported confidence score c_i. Before being passed to future stages of the pipeline, we group answers $\{a_i\}_{i=1}^n$, reasoning $\{r_i\}_{i=1}^n$, and confidence $\{c_i\}_{i=1}^n$ following Chen et al. (2024b) to form G_I. The grouping process lists each unique agent answer with the number of agents supporting it and their reasoning.

2.2 Tool-Based Disagreement Resolution

We expect answering agents with distinct outputs to have reasoning chains that disagree at specific points. These disagreements mean that agents diverge in their perception or analysis of the scene and question. As such, we aim to address these divergences with tool-based disagreement resolution to directly provide visual information from relevant vision expert tools that will ground future conversations (see Fig. 2 (Step 2). We detect disagreements among the agents' answers $\{a_i\}_{i=1}^n$ and answer reasoning chains $\{r_i\}_{i=1}^n$ using a recruitment agent. For each of our vision tools, the recruitment agent is parameterized by an LLM provided with the acceptable input parameters (either none, text query, list of text queries) as well as potential use cases for each tool in the prompt (e.g., *the grounder can be used if an agent mentioned an object that another agent missed*). Then, given a grouped answer G_I, the recruitment agent detects disagreements and proceeds to construct a vision tool set and corresponding input parameters. The recruitment agent is instructed to state the disagreement and justify how a given tool would address that disagreement (see Appendix E for the prompt). For example, if the recruitment agent finds that "*Agents disagree on the color and specific markings of the planes, with one agent mentioning a green color while others*

describe the distinctive blue," the recruitment agent would call on the attribute detection tool to "*clarify the colors and markings of the planes.*" DART uses grounding, object detection, OCR, spatial reasoning, captioning, and attribute detection tools.

The grounding, object detection, OCR, and spatial reasoning tools are implemented as expert, fine-tuned models. In contrast, reasoning, captioning, and attribute detection are handled by a general-purpose VLM instructed to focus on those specific tasks without leveraging task-specific models. We use a separate VLM from the debate agents to avoid bias, as using the same VLM might bias the output towards that particular VLM's idiosyncrasies.

The vision tools process image I with their respective generated input parameters to get expert tool outputs $\mathcal{E} = \{E_i\}_{i=1}^l$. Note that certain vision tools, such as grounding, object detection, and OCR, are not naturally suited for natural language outputs. For tools that do not natively output natural language (Grounder, Object Detector) we perform minimal post-processing to convert outputs into statements suitable for language-model consumption (see Appendix E for the prompt).

2.3 Tool-Based Agreement Scoring

Domain-specific models have been shown to be stronger than VLMs within their specific skill set (Chiu et al., 2025; Chen et al., 2024a). As such, we hypothesize that VLMs whose outputs agree more closely with those of specialized tool models are more likely to be accurate. The expert tools have better perception, meaning VLMs that match the experts are likely to have correctly perceived the image. Thus, these agreement scores serve as an indicator for whether or not to trust a VLM output.

For every pair $((a_i, r_i), E_j)$ of output from answering agent i and output from tool j, an agreement scorer LLM produces a binary agreement score between answering agent i and output from invoked tool j as $s_{ij} \in \{0, 1\}$ after comparing (a_i, r_i) with the $E_j \in \mathcal{E}$ (see Fig. 2 (Step 3)). The agreement scorer agent produces scores that represent the degree to which an agent agrees with one specific expert tool output E_j. Agreement of answering agent i with the full set of tools is then represented by the mean:

$$S_i = \frac{1}{|\mathcal{E}|} \sum_{E_j \in \mathcal{E}} s_{ij}, \qquad s_i \in [0, 1].$$

Each (a_i, r_i) is now annotated with its agreement score S_i and carried into the subsequent discussion
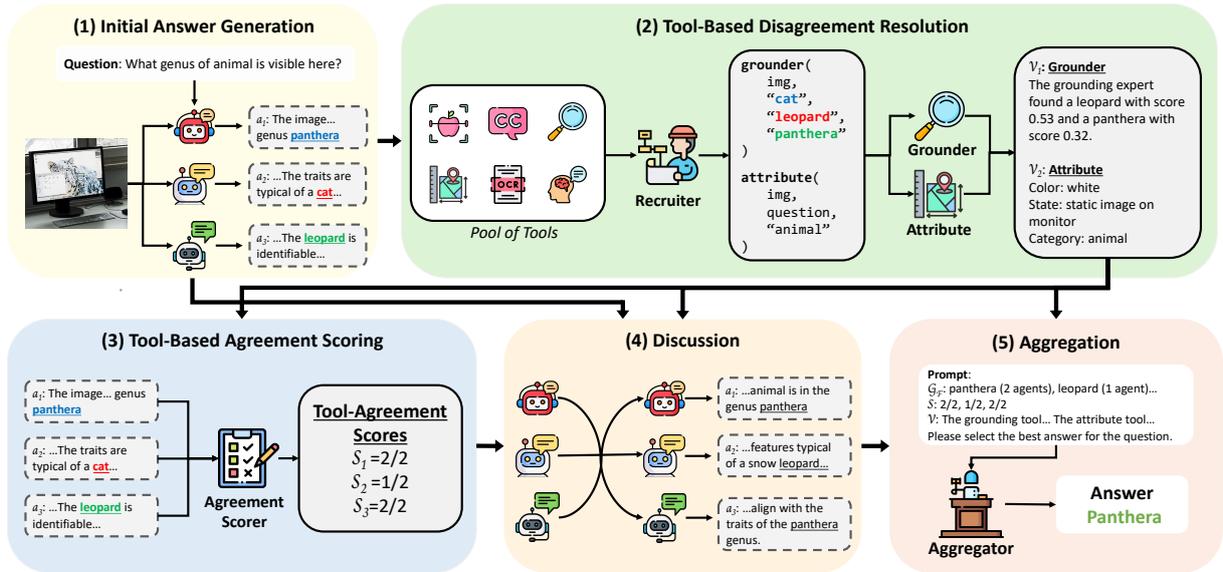
Figure 2: Overview of DART. We start with (1) *Initial Answer Generation* from a set of answering/reasoning agents. This is followed by (2) *Tool-Based Disagreement Resolution* and (3) *Agreement Scoring*. The newly generated tool outputs and agreement scores are incorporated into the (4) *Discussion* and (5) *Aggregation* phases.

and aggregation steps. These steps provide agreement scores such that more trust is placed on agents that align with expert information. The agents that do have high scores are more likely to have accurate visual understanding of the scene. We provide the agreement scorer prompt in Appendix E.

2.4 Discussion

In the discussion, each agent takes in the grouped input G_I, tool outputs $\mathcal{E}$, and the expert-based agreement scores S_I (see Fig. 2 (Step 4)). Recall that the grouped agent solution form includes self-reported confidence. This means that agents have both self-reported confidence and agreement scores during the discussion step. Agents are instructed to place high priority on the expert outputs, even over the agent solutions. Answering agents are also given a 3-shot prompt with simple synthetic examples to guide them to follow the provided output format (see Appendix E for the full prompt). If the agent solutions contain information that contradicts the tool outputs, models are told to assume that the vision tool is correct. Finally, agents are provided with the tool-based agreement scores as well as the self-reported confidence score. This outputs the final group of agent answers defined as G_F (see Fig. 2 (Step 3)), and we recalculate final confidence scores following Section 2.3 to obtain S_F.

2.5 Aggregation

Similar to the discussion step, we provide an aggregator agent with grouped solutions G_F, tool outputs $\mathcal{E}$, and tool-based agreement scores S_F (see Fig. 2 (Step 5)). The aggregator VLM is instructed to choose, with CoT, the best answer out of all the provided solutions (see Appendix E for prompt).

3 Results and Analysis

3.1 Experimental Setup

Datasets and Metrics. We evaluate our framework on four Visual Question Answering (VQA) datasets: **A-OKVQA** (Schwenk et al., 2022), **MMMU** (Yue et al., 2024), **NaturalBench** (Li et al., 2024), and **M3D-VQA** (Bai et al., 2024). We aim to test whether DART performs well on a dataset that focuses on reasoning and world knowledge, such as A-OKVQA. We evaluate on the direct answer subset using the metric from Antol et al. (2015) following the A-OKVQA direct answer evaluation scheme. We also test on MMMU to assess DART's effect on a large, diverse set of VQA data, and on NaturalBench, which is designed to rigorously test VLMs on their visual comprehension using adversarial VQA pairs that are easy for humans but challenging for VLMs. Additionally, we evaluate how well DART adapts to a new tool by testing it on M3D-VQA, which is a medical VQA dataset on 3D medical scans of real patients. MMMU, NaturalBench, and M3D are all multiple choice datasets, so we measure accuracy.

Implementation Details. We use a diverse set of answering agents and visual tools. For answering

Method	Agent	A-OKVQA	MMMU	NaturalBench
Tool Systems				
ViperGPT (Surís et al., 2023)	Qwen	49.9	54.0	75.3
Chameleon (Lu et al., 2023)	Qwen	47.5	51.6	77.2
Single Agent				
CoT (Wei et al., 2022)	QwenVL	61.3	58.8	79.2
	MiniCPM-o	55.7	50.4	77.9
	Ovis2	61.0	50.7	78.7
	LLaVA 1.6	62.8	39.4	70.9
Self-Refinement (Madaan et al., 2023)	QwenVL	61.8	56.5	79.0
	MiniCPM-o	56.7	46.7	78.2
	Ovis2	61.7	47.2	78.4
	LLaVA 1.6	62.9	39.7	70.6
Self-Consistency (5-way) (Wang et al., 2023)	QwenVL	63.7	60.8	80.1
	MiniCPM-o	58.2	52.5	78.1
	Ovis2	64.6	52.8	79.3
	LLaVA 1.6	64.2	41.3	72.4
Multiple Agents				
Debate with Consensus	3×QwenVL	61.9	60.5	80.0
	3×MiniCPM-o	57.4	53.2	78.3
	3×Ovis2	62.8	57.0	80.1
	Q, M, O	64.2	<u>63.1</u>	80.6
Debate with Judge	3×QwenVL	62.1	60.3	79.9
	3×MiniCPM-o	59.7	54.8	78.3
	3×Ovis2	63.0	58.3	80.4
	Q, M, O	<u>65.5</u>	<u>63.1</u>	<u>81.3</u>
DART (*Ours*)	Best Model	65.3	61.6	80.8
	Q, M, O	**68.9**	**65.5**	**82.5**

Table 1: Comparison of DART and different baselines on A-OKVQA, MMMU, and NaturalBench. Q, M, and O stand for QwenVL, MiniCPM-o, and Ovis2 respectively. For the DART (Best Model) setting, we take the single best performing agent from multi-agent debate with judge and run DART with three instances of that model. The best models are Ovis2, QwenVL, and Ovis2 for A-OKVQA, MMMU, and NaturalBench respectively.

agents, we test with Qwen2.5-VL 7B (Bai et al., 2025), MiniCPM-o 2.6 8B (Yao et al., 2024), and Ovis2 8B (Lu et al., 2024) as our VLMs. As an additional independent single agent baseline, we include LLaVA-1.6 (Mistral) (Liu et al., 2024). Then, we construct a pool of visual expert tools with GroundingDINO (Liu et al., 2023) (grounding), YOLOv11 (Khanam and Hussain, 2024) (object detection), SpaceLLaVA (Chen et al., 2024a) (spatial), OCR-Qwen (OCR), and InternVL-2.5 MPO (Wang et al., 2024b) (captioning, attribute, and reasoning) as vision tools. We use InternVL-2.5 to serve as an expert tool in multiple domains, as prior work has shown that models can be designed to be more effective in specific domains when prompted to do so (Xu et al., 2025; Wang et al., 2024c). We introduce a separate VLM for this task to reduce overlap between answering agent information and tool calling information. We also test the effectiveness of DART by ablating the VLM-based tools (see Section 3.3). We use Qwen2.5 (7B) as the recruitment agent and agreement scorer. We also find that

Ovis2 serves as the best aggregator among a pool of similar VLMs. When testing on M3D-VQA, we additionally insert MedGemma 4B (Google, 2025) into the pool of tools to serve as a medical expert.

Baselines. We compare DART to multiple single-agent, multi-agent, and tool-calling baselines. We include comparisons to training-free single-agent tool-calling baselines: ViperGPT (Surís et al., 2023) and Chameleon (Lu et al., 2023). ViperGPT utilizes a code generation model to predict executable code that leverages required tools for a given question. In contrast to the flexibility offered by free-form code generation in ViperGPT, Chameleon adopts a more robust compositional approach, utilizing an LLM-based planner to directly invoke various tools in sequence via word matching. We update these methods to use our pool of experts and also incorporate more recent models as planners to ensure a fair comparison.

Our single agent baselines are zero-shot CoT (Wei et al., 2022), self-refinement (Madaan et al.,

2023), and self-consistency (Wang et al., 2023). For multi-agent baselines, we use Debate with Consensus and Debate with Judge. Specifically, we follow the ReConcile LLM debate framework outlined by Chen et al. (2024b) for both consensus and InternVL2.5 as a judge. Debate with consensus counts the most frequent answer from the agents while debate with judge uses a separate VLM model to choose an answer. See Appendix B for more details about the baselines.

3.2 Results

DART is stronger than all tool-calling, single-agent, and multi-agent baselines. DART outperforms training-free tool-calling systems by a wide margin as seen in Table 1. On A-OKVQA, we see 19.0% and 21.4% improvements over ViperGPT and Chameleon respectively (see Table 1 for full results). The gap on both MMMU and NaturalBench compared to gap between DART and single/multi-agent baselines. These tool-calling systems plan their tool execution in a single step based solely on the question, and they lack the ability to reason directly on the image as an input. This is in contrast to our own system that effectively incorporates tools on top of existing VLM systems, allowing for better vision understanding with tools while building on the existing capabilities of VLMs. The performance gain of DART over LLM-based tool calling methods illustrates the strengths of our targeted tool-calling approach for a debate among multiple VLM agents.

We also see consistent gains across all three benchmarks over all baselines. On A-OKVQA, DART reaches 68.9%, exceeding the best single-agent baseline (Self-Consistency with Ovis2) by 4.3% and the best multi-agent baseline (multi-model debate with judge) by 3.4%. This trend continues across both MMMU and Natural-Bench, where DART consistently outperforms both the strongest single and multi-agent baselines. The gains are highest on A-OKVQA, which requires advanced reasoning and world knowledge, indicating that our system improves the overall reasoning process of the answering agents. These improvements indicate that resolving disagreements with expert tools and using tool-aligned agreement scores yields better reasoning and perception than prior single- or multi-agent approaches.

DART can adapt to diverse domains by adding appropriate expert tools. To show how DART can adapt to domains where its current pool of experts is unsuited for a task, we test on M3D with an additional medical expert. If DART is able to adapt to new experts, we expect the accuracy to increase, as the medical tool provides valuable info to guide a better discussion. In Table 2, our method outperforms the baselines by 1.54% even given the disparities in individual model strengths on this task (i.e., Ovis2 is 10.0% weaker than QwenVL on this dataset). While multi-agent debate underperforms compared to the strongest single-agent baseline by 2.72%, DART can effectively use the medical expert and multiple agent reasoning paths to bridge this gap and improve even further over the next best method (Qwen2.5-VL) by 1.5%. This indicates that DART can effectively incorporate new experts, allowing it to adapt to new tasks with minimal implementation effort.

DART leads to more diverse and fruitful discussion compared to existing multi-agent methods. To evaluate the diversity and novelty of ideas during discussion, we measure the textual overlap between agents' initial responses and their responses after one round of discussion, comparing DART with the standard multi-agent debate. For DART, tool outputs are appended to the initial responses, enabling the metrics to account for all available information during the discussion round. As shown in Table 3, DART exhibits substantially lower overlap across all metrics (ROUGE-1/2/L (Lin, 2004) and Jaccard Index), with ROUGE-L and Jaccard Index decreasing by 0.173 and 0.289 respectively compared to multi-agent debate (MAD) (Du et al., 2024). This indicates that discussions in DART are more diverse, with agents introducing novel reasoning paths rather than repeating prior content. The integration of tool-based resolution enables each agent to refine or challenge other perspectives with visual or textual evidence, resulting in richer discussion. In contrast, multi-agent debate seems to converge in terms of word usage, with agents repeating similar or nearly identical reasoning. These findings show that DART fosters more fruitful discussion among agents.

Additional Results. We conduct additional experiments to study calibration, multi-round performance, efficiency, tool benefits, and prompt sensitivity in DART. This additional analysis is included in Appendix D.

Method	Plane	Phase	Organ	Abnormality	Location	Avg.
CREMA (Yu et al., 2025a)	14.9	26.7	15.9	17.3	13.0	17.2
GPT-4o (OpenAI et al., 2024)	83.3	<u>42.7</u>	50.0	41.3	41.3	51.7
MEXA (Yu et al., 2025b)	65.0	**48.1**	**60.9**	44.8	48.0	53.3
Ovis2	59.6	34.0	39.8	55.4	43.0	46.4
MiniCPM-o	79.8	28.4	<u>53.9</u>	49.2	37.8	49.8
QwenVL	<u>91.2</u>	33.5	49.7	60.0	47.7	<u>56.4</u>
Debate with Consensus	81.2	32.0	43.1	<u>60.9</u>	**51.9**	53.7
Debate with Judge	79.5	32.0	45.6	**61.2**	51.7	53.4
DART (*Ours*)	**92.8**	40.7	47.6	55.9	<u>51.8</u>	**57.8**

Table 2: Accuracy on the M3D medical dataset by question type and overall. DART performs the best out of the tested methods.

Method	R-1	R-2	R-L	Jac. Index
Debate with Consensus	0.679	0.529	0.550	0.559
DART	0.452	0.310	0.377	0.270

Table 3: Comparison of DART and multi-agent debate across text similarity metrics. *R* and *Jac. Index* stand for ROUGE and Jaccard Index respectively.

Method	Accuracy
DART	**68.9**
w/o Diverse Answerers (only Ovis)	65.3
w/o Tool Agreement Scores	67.8
w/o Expert Tool Outputs[†]	65.7

Table 4: Ablations of DART on A-OKVQA. [†]Agreement scores are impossible to calculate without tool outputs, so they are omitted.

Method	Accuracy
DART	68.9
DART (VLM as all tools)	66.0
DART (no VLM-based tools)	66.7
DART (no specialized model tools)	65.6

Table 5: Comparison between the full DART system and its variant using VLMs as tools. VLM-based tools are the captioning, attribute detection, and reasoning tools. Specialized model tools are the grounding, object detection, spatial reasoning, and OCR tools.

Figure 3: Breakdown of total tool calling of DART on A-OKVQA.

3.3 Analysis and Ablation Study

DART calls on a variety of tools. In Figure 3, we present the distribution of tool calls made by DART when evaluated on A-OKVQA. The most common tools called were Grounder, Captioning, and Attribute Detection, with each called for over 20% of questions with tool calls. These tools are mostly focused on physical scene understanding and composition, indicating that the VLM agents often differ in their vision understanding capabilities. The less called tools were OCR, Reasoning, Spatial Reasoning, and Object Detection respectively, with each called for $\leq 10\%$ of questions. These are less holistic, more fine-grained vision skills that may not be needed for most questions. Overall, every tool is being invoked and contributing towards more detailed vision understanding.

Multiple different answerers lead to more disagreements. We compare using multiple models (i.e., QwenVL, MiniCPM-o, and Ovis2) as answers to multiple instances of the same model (i.e., Ovis2 with higher temperature). From Table 1, we see that using a single model in DART results in a decrease of 4.6% accuracy on A-OKVQA. Then, we find that using a single model results in less than half the disagreements than DART with multiple models, where we count 0.49 and 1.32 disagreements on average respectively. This indicates that the diverse perspectives inherent in a multi-model system yields a richer disagreement signal that allows for better expert tool calling.

Effect of using a VLM as an expert tool. While VLMs exhibit strong general capabilities across

perception and reasoning tasks, they often underperform compared to specialized models on domain-specific tasks (i.e., grounding, spatial reasoning, etc.). In our original setting, we use a general VLM for three tasks: captioning, attribute detection, and reasoning. We extend this to check whether a single VLM could indeed serve as a substitute for *all* tools. To quantify the trade-off between specialized models and general VLMs, we replace all specialized tools in DART with a single general-purpose VLM (InternVL2.5) prompted to handle OCR, spatial reasoning, grounding, and object detection on top of its original tasks. As shown in Table 5, this setting results in a 2.9% decrease in overall accuracy on A-OKVQA. Although VLMs are a convenient solution, their generality may come at the cost of reliability particularly for fine-grained visual understanding tasks.

We also measure the effect of ablating each tool type in this heterogeneous tool set: VLM-based tools and specialized model tools. Removing the VLM-based tools (captioning, attribute detection, reasoning) lowers accuracy from 68.9% to 66.7%, suggesting that the VLM tools are still able to provide a complementary signal for a domain-specific task even without designated expert capabilities. Removing only specialized model tools (grounder, object detection, spatial reasoning, OCR) while keeping VLM-based tools lowers accuracy to 65.3%, showing the importance of using specialized models in the tool pool. Overall, these ablations indicate that the strongest performance occurs when combining specialized model tools with VLM-based tools, with specialized expert tools contributing the larger share of the performance gains.

DART component ablations. We assess the impact of key components in our system, including diverse answering agents, expert tools, and tool-aligned agreement scores (results in Table 4). We conduct the ablation studies on A-OKVQA as it is specifically designed to test commonsense reasoning and needs precise visual understanding for questions reliant on world knowledge. Removing the multiple agent answerers or vision-expert components degrades performance compared to the full DART system. We evaluate DART without diverse answerers and with only the strongest performing model (i.e., Ovis2) on A-OKVQA. Using only the strongest single model (Ovis2) on A-OKVQA (i.e., without diverse answerers) leads to a decline in

accuracy from 68.9% to 65.3%, highlighting the value of model diversity. This indicates that DART benefits from having the diverse reasoning and expanded knowledge base of multiple VLMs. Furthermore, retaining a diverse set of VLM agents without expert confidence yields a 1.1% performance drop, showing that the calibrated confidence from the experts improves the discussion beyond regular self-reported confidence. Additionally, we test the discussion framework without vision expert outputs. This setting is almost equivalent to Debate with Judge (from Table 1) but with only one debate round. Accuracy drops by 3.2% in this experiment. This result suggests that visual information from experts leads to a higher-quality discussion compared to that of only answerers. In fact, the performance dip is comparable to that of having no diverse answerers, indicating that they are relatively equally important to the overall system. The ablations highlight the importance of having multiple VLMs and vision experts working together in discussion.

4 Related Work

Tool-Calling Systems. Past efforts have augmented large language models with external tools to overcome knowledge and reasoning limitations, especially in vision-language tasks (Surís et al., 2023; Lu et al., 2023; Qin et al., 2024; Gupta and Kembhavi, 2022; Chung-En et al., 2025; Yang et al., 2023; Shen et al., 2023; Schick et al., 2023; Parisi et al., 2022). This tool augmentation has enabled LLMs to tackle tasks such as finding real-time information through web search (Lu et al., 2023; Nakano et al., 2022), accessing vision information (Lu et al., 2023; Surís et al., 2023; Wu et al., 2023; Yang et al., 2023; Gupta and Kembhavi, 2022), and acquiring expert domain-specific knowledge (Das et al., 2024; Yu et al., 2025c). One direction has used a programmatic approach by leveraging LLMs inherent code generation capabilities to handle tool calls (Shen et al., 2024; Surís et al., 2023) while other work has experimented with directly prompting an LLM to construct a series of tool calls (Lu et al., 2023; Chung-En et al., 2025; Yang et al., 2023). While past work has mostly focused on using a single LLM agent for tool calling, we investigate how using disagreements in multimodal multi-agent debate can lead to more effective tool calling and discussion.

Multi-Agent Systems. Another line of research improves reasoning by enlisting multiple LLM

agents that collaborate or compete among one another to arrive at a stronger conclusion (Chen et al., 2024b; Estornell et al., 2025b; Qian et al., 2024; Zhao et al., 2024; Tang et al., 2024; Yin et al., 2024; Zhang et al., 2024; Du et al., 2023). A large body of work has explored designing a debate framework for agents to interact and refine their answers through multiple rounds of discussion (Chen et al., 2024b; Estornell and Liu, 2024; Du et al., 2023). Others have tried assigning agents specific functions or roles to accomplish different goals or subtasks for a given query (Estornell et al., 2025b; Hong et al., 2024; Wang et al., 2024c). Past work has also tested how to control the level of disagreement in a debate (Khan et al., 2024; Liang et al., 2024; Chang, 2025). Our method contributes several novelties along a number of axes. First, prior multi-agent debate methods have largely omitted tool use while we inject novel information via disagreement-triggered tool use. Moreover, we expand this into a less explored frontier by examining multimodal agents instead of text.

5 Conclusion

In this work, we introduce DART, a multi-agent framework that recruits domain-specific vision tools based on disagreements among VLM agents. By identifying points of disagreement in agent discussions and selectively invoking expert tools to resolve these disagreements, DART combines the reasoning flexibility of VLMs with the specialized perception capabilities of expert tools. Through this, DART sees consistent improvements across four diverse benchmarks over strong baselines on A-OKVQA, MMMU, NaturalBench, and M3D. DART represents a promising direction for multimodal reasoning that harnesses the strengths of generalist models and specialized tools through disagreement resolution.

Limitations

Our method, like other multi-agent systems, has a higher computational cost than single-agent to end-to-end setups. We explore the costs of using DART over other baselines in Appendix D. Moreover, our evaluation is currently focused on challenging visual question answering datasets requiring strong reasoning and/or vision understanding. Future work might expand this to other debate settings, including non-visual reasoning settings. Finally, we note that our system does not introduce

new risks, but it may still be susceptible to the risks faced by the underlying VLMs incorporated into the framework.

Acknowledgments

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*.

Fan Bai, Yuxin Du, Tiejun Huang, Max Q. H. Meng, and Bo Zhao. 2024. M3d: Advancing 3d medical image analysis with multi-modal large language models. *Preprint*, arXiv:2404.00578.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Edward Y. Chang. 2025. Evince: Optimizing multi-llm dialogues using conditional statistics and information theory. *Preprint*, arXiv:2408.14575.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. 2024a. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14455–14465.

Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. 2024b. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. *Preprint*, arXiv:2309.13007.

Ming-Chang Chiu, Fuxiao Liu, Karan Sapra, Andrew Tao, Yaser Jacoob, Xuezhe Ma, Zhiding Yu, and Guilin Liu. 2025. Aide: Agentically improve visual language model with domain experts. *Preprint*, arXiv:2502.09051.

Chung-En, Yu, Hsuan-Chih, Chen, Brian Jalaian, and Nathaniel D. Bastian. 2025. Hydra: An agentic reasoning approach for enhancing adversarial robustness and mitigating hallucinations in vision-language models. *Preprint*, arXiv:2504.14395.

Debrup Das, Debopriyo Banerjee, Somak Aditya, and Ashish Kulkarni. 2024. Mathsensei: A tool-augmented large language model for mathematical reasoning. *Preprint*, arXiv:2402.17231.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *Preprint*, arXiv:2305.14325.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *International Conference on Machine Learning*, pages 11733–11763. PMLR.

Andrew Estornell and Yang Liu. 2024. Multi-LLM debate: Framework, principals, and interventions. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Andrew Estornell, Jean-Francois Ton, Yuanshun Yao, and Yang Liu. 2025a. ACC-collab: An actor-critic approach to multi-agent LLM collaboration. In *The Thirteenth International Conference on Learning Representations*.

Andrew Estornell, Jean-Francois Ton, Yuanshun Yao, and Yang Liu. 2025b. Acc-collab: An actor-critic approach to multi-agent llm collaboration. *Preprint*, arXiv:2411.00053.

Google. 2025. Medgemma hugging face. https://huggingface.co/collections/google/medgemma-release-680aade845f90bec6a3f60c4.

Tanmay Gupta and Aniruddha Kembhavi. 2022. Visual programming: Compositional visual reasoning without training. *Preprint*, arXiv:2211.11559.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rocktäschel, and Ethan Perez. 2024. Debating with more persuasive llms leads to more truthful answers. *Preprint*, arXiv:2402.06782.

Rahima Khanam and Muhammad Hussain. 2024. Yolov11: An overview of the key architectural enhancements. *Preprint*, arXiv:2410.17725.

Baiqi Li, Zhiqiu Lin, Wenxuan Peng, Jean de Dieu Nyandwi, Daniel Jiang, Zixian Ma, Simran Khanuja, Ranjay Krishna, Graham Neubig, and Deva Ramanan. 2024. Naturalbench: Evaluating vision-language models on natural adversarial samples. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: Bootstrapping language-image pretraining with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. *Preprint*, arXiv:2305.19118.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. Llavanext: Improved reasoning, ocr, and world knowledge.

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*.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023. Chameleon: Plug-and-play compositional reasoning with large language models. In *The 37th Conference on Neural Information Processing Systems (NeurIPS)*.

Shiyin Lu, Yang Li, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Han-Jia Ye. 2024. Ovis: Structural embedding alignment for multimodal large language model. *arXiv:2405.20797*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *37th Conference on Neural Information Processing Systems*.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: a survey. *Preprint*, arXiv:2302.07842.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback. *Preprint*, arXiv:2112.09332.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI. 2025. GPT-5 system card. Technical report, OpenAI. OpenAI system card, August 13, 2025.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1).

Aaron Parisi, Yao Zhao, and Noah Fiedel. 2022. Talm: Tool augmented language models. *Preprint*, arXiv:2205.12255.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. Chatdev: Communicative agents for software development. *Preprint*, arXiv:2307.07924.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, dahai li, Zhiyuan Liu, and Maosong Sun. 2024. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *The Twelfth International Conference on Learning Representations*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Preprint*, arXiv:2302.04761.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. A-okvqa: A benchmark for visual question answering using world knowledge. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VIII*, pages 146–162. Springer.

Ruoyue Shen, Nakamasa Inoue, and Koichi Shinoda. 2024. Pyramid coder: Hierarchical code generator for compositional visual question answering. *Preprint*, arXiv:2407.20563.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface. In *Advances in Neural Information Processing Systems*.

Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. Vipergpt: Visual inference via python execution for reasoning. *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.

Xunzhu Tang, Kisub Kim, Yewei Song, Cedric Lothritz, Bei Li, Saad Ezzini, Haoye Tian, Jacques Klein, and Tegawende F. Bissyande. 2024. Codeagent: Autonomous communicative agents for code review. *Preprint*, arXiv:2402.02172.

Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. 2024a. Rethinking the bounds of llm reasoning: Are multi-agent discussions the key? In *Long Papers*, Proceedings of the Annual Meeting of the Association for Computational Linguistics, pages 6106–6131. Association for Computational Linguistics (ACL). The authors of this paper were supported by the NSFC Fund (U20B2053) from the NSFC of China, the RIF (R6020-19 and R6021-20), and the GRF (16211520 and 16205322) from RGC of Hong Kong. We also thank the support from the UGC Research Matching Grants (RMGS20EG01-D, RMGS20CR11, RMGS20CR12, RMGS20EG19, RMGS20EG21, RMGS23CR05, RMGS23EG08).; 62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024 ; Conference date: 11-08-2024 Through 16-08-2024.

Weiyun Wang, Zhe Chen, Wenhai Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Jinguo Zhu, Xizhou Zhu, Lewei Lu, Yu Qiao, and Jifeng Dai. 2024b. Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. *arXiv preprint arXiv:2411.10442*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2024c. Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. *Preprint*, arXiv:2307.05300.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

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

Benfeng Xu, An Yang, Junyang Lin, Quan Wang, Chang Zhou, Yongdong Zhang, and Zhendong Mao. 2025. Expertprompting: Instructing large language models to be distinguished experts. *Preprint*, arXiv:2305.14688.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. 2023. Mmreact: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint*.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, and 1 others. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Xiangyu Yin, Chuqiao Shi, Yimo Han, and Yi Jiang. 2024. Pear: A robust and flexible automation framework for ptychography enabled by multiple large language model agents. *Preprint*, arXiv:2410.09034.

Shoubin Yu, Jaehong Yoon, and Mohit Bansal. 2025a. CREMA: Generalizable and efficient video-language reasoning via multimodal modular fusion. In *The Thirteenth International Conference on Learning Representations*.

Shoubin Yu, Yue Zhang, Ziyang Wang, Jaehong Yoon, and Mohit Bansal. 2025b. Mexa: Towards general multimodal reasoning with dynamic multi-expert aggregation. *arXiv preprint arXiv:2506.17113*.

Shoubin Yu, Yue Zhang, Ziyang Wang, Jaehong Yoon, and Mohit Bansal. 2025c. Mexa: Towards general multimodal reasoning with dynamic multi-expert aggregation. *Preprint*, arXiv:2506.17113.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and 3 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*.

Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, Xiaojun Chang, Junge Zhang, Feng Yin, Yitao Liang, and Yaodong Yang. 2024. Proagent: Building proactive cooperative agents with large language models. *Preprint*, arXiv:2308.11339.

Xiutian Zhao, Ke Wang, and Wei Peng. 2024. An electoral approach to diversify llm-based multi-agent collective decision-making. *Preprint*, arXiv:2410.15168.

A Model Configurations

We use the default generation configurations from HuggingFace for Qwen2.5-VL 7B (Apache-2.0 license), MiniCPM-o 2.6 8B (Apache-2.0 license), and Ovis2 8B (Apache-2.0 license). Qwen2.5-VL uses a temperature of 10^{-6} and repetition penalty of 1.05. MiniCPM-o and Ovis2 both use a temperature of 0.7 and repetition penalty of 1.05. For instances of multi-agent debate and DART where we would like more answer diversity with a single model, we increase temperature to 0.9 for all agents.

We use the `grounding-dino-base` implementation in HuggingFace of GroundingDINO (Apache-2.0 license) with box and text thresholds of 0.3. We use OCR-Qwen (Apache 2.0 License), a Qwen2.5 7B model finetuned for OCR, with default generation configuration of temperature 0.1 and repetition penalty 1.05. We use SpaceLLaVA (Apache-2.0 License) for spatial reasoning with a temperature of 0.2 and repetition penalty of 1.1. InternVL2.5-MPO 8B (MIT License) uses temperature 0.8 for all three expert tool functionalities.

B Baseline Details

We compare DART to multiple single-agent and multi-agent baselines. The single agent baselines include zero-shot CoT (Wei et al., 2022), self-refinement (Madaan et al., 2023), and self-consistency (Wang et al., 2023). In zero-shot COT, we prompt models to first output their reasoning before providing their final answer. With self-refinement, models are asked to provide feedback on their own output and update their answer accordingly. This assesses whether models can improve themselves based on their own feedback. With self-consistency, we sample multiple reasoning paths from a model and take the most consistent answer. This takes advantage of the existence of multiple reasoning paths to arrive at an answer by increasing the number of generated samples per question.

We include comparisons to training-free single-agent tool-calling baselines: ViperGPT (Surís et al., 2023) and Chameleon (Lu et al., 2023). ViperGPT utilizes a code generation model to predict executable code that leverages required tools for a given question. In contrast to the flexibility offered by free-form code generation in ViperGPT, Chameleon adopts a more robust compositional approach, utilizing an LLM-based planner to directly invoke various tools in sequence via word matching. We update both implementations, using Qwen2.5 7B Instruct as the code generation model and incorporating our expert tools into both systems. For ViperGPT specifically, we also update its VLM

module (called on by the code generation model) from BLIP2 (Li et al., 2023) to InternVL-2.5 MPO for a fairer comparison with our system.

Finally, we also evaluate against strong multi-agent baselines designed for text-based tasks, including both consensus-based debate and debate with a judge. We follow the ReConcile LLM debate framework outlined by Chen et al. (2024b) for both consensus and InternVL2.5 as a judge. In the consensus setting, the majority – or consensus – answer is automatically taken (a random answer is taken if all agents disagree). In the judge setting, a judge model is given the image along with each answer, justification, and confidence before picking a single final answer.

Tool Conf.?	Metric	ECE↓
✗	Self-Reported	0.4581
✓	Self-Reported	**0.3859**
✓	Tool-Confidence	0.4901

Table 6: Expected calibration error (ECE) of DART with and without tool confidence on self-reported confidence metrics and our tool-based agreement score metric.

C Dataset Details

- **A-OKVQA** (Apache-2.0 License): We evaluate on the validation split of 1135 samples.

- **MMMU** (Apache-2.0 License): We evaluate on the validation split of 900 samples.

- **NaturalBench** (Apache-2.0 License): We evaluate on the full dataset of 7600 question-image pairs.

D Additional Results

DART provides a stronger confidence metric based on vision tools. We test the effectiveness of calculating confidence based on agreement to vision tools. For this, we run DART under two settings: 1) using only self-reported confidence, and 2) combining self-reported confidence with tool-based agreement scores. Both settings still involve disagreement resolution with tools. We then measure the effectiveness using expected calibration error (ECE) (Pakdaman Naeini et al., 2015) of self-reported confidence only, tool-based confidence, and self-reported confidence given tool-based confidence during discussion. Recall that both self-reported confidence and tool-based confidence are used in DART during discussion.
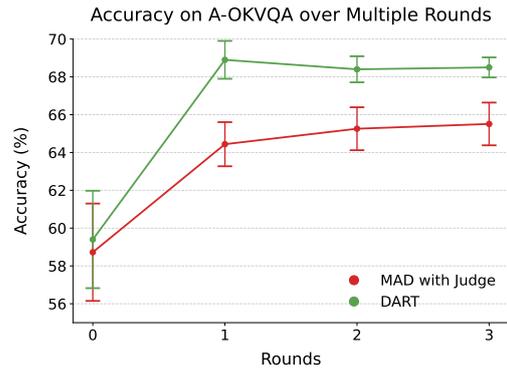
Figure 4: Performance of DART and multi-agent debate over three rounds. The error bars indicate the standard deviation among the individual answering agents.

We observe higher ECE when relying solely on self-reported confidence (i.e., without tool-based agreement scores) than when including those scores: 0.4581 vs. 0.3859 respectively (see Table 6). Note however that the raw tool-based confidence calculated in DART has ECE of 0.4901, which indicates weaker calibration than self-reported confidence on its own.

Round	QwenVL	MiniCPM	Ovis	w/ Aggregation
Initial	61.3	55.7	61.0	59.4
1	68.6	66.8	**69.2**	**68.9**
2	**68.7**	67.2	68.7	68.4
3	68.4	**67.4**	68.7	68.5

Table 7: DART performance over three rounds of disagreement resolution and expert confidence on A-OKVQA.

Multiple rounds of debate in multimodal multi-agent systems is ineffective. Encouraged by the success of multiple debate rounds in text-based multi-agent environments, we apply our framework across successive debate rounds to see if the same pattern emerges. We compare the improvement of DART to multi-agent debate with judge over three rounds in Fig. 4. DART after one round of expert tool intervention outperforms the round three performance of multi-agent debate with judge. Both methods have the largest improvement in round after the initial answer generation. The improvement is likely due to initial exposure to other agent outputs and, in the case of DART, tool outputs. However, after the first round of discussion both stagnate in subsequent rounds (with <2% accuracy gains between rounds 1 and 3). For multi-agent debate, the agents are not able to effectively refine

their reasoning and answer after round one as there is no new information being introduced. For DART, we observe that the most major disagreements are resolved in round 1. Future rounds are prone to resolving more minor disagreements that can distract from the question. We find that it is insufficient to give any piece of new information in a multimodal multi-agent system. In fact, the new expert tool information must be sufficiently good and relevant to advance discussion.

Model	All Choices	Accuracy
InternVL2.5	✗	68.3
Qwen2.5-VL	✗	68.7
MiniCPM-o 2.6	✗	67.8
Ovis2	✓	68.5
Ovis2	✗	**68.9**

Table 8: Aggregator Ablation on A-OKVQA. All choices indicates whether answers from initial generation are used.

Robustness of DART aggregator to model choice. We test four different VLMs as aggregator models: Qwen2.5-VL, MiniCPM-o, Ovis2, and InternVL-2.5 MPO. The results are shown in Table 8. Between the strongest and weakest aggregator (Ovis2 and MiniCPM-o, respectively), there is a difference in accuracy of 1.1%. In fact, among the top 3 models, accuracies are within 0.6% of each other. This indicates DART has relatively low sensitivity to aggregator choice and will generalize well to new models as aggregators. Additionally, we test how the aggregator model responds when answer choices from the initial generation stage are provided in addition to the discussion stage answers. We find that including these additional answers slightly lowers performance by 0.4%. This suggests that having a more concise answer set after using tools (i.e., post discussion answers) makes aggregation easier compared to lower quality answers from initial generation.

We also find that the tool-based agreement scores are beneficial in improving calibration during aggregation (see Table 6 in the Appendix.). When aggregating with agreement scores, the final self-reported confidence values in DART are better calibrated and obtain a relative decrease of 15.8% in expected calibration error (ECE) compared to self-reported confidence values without tool-based agreement scores. More discussion on the calibration calculations is in Appendix D.

Method	Accuracy
DART	**68.9**
w/o Aggregation (i.e., Majority Vote)	67.9
w/o Discussion	62.5

Table 9: Ablations of discussion and aggregation steps of DART on A-OKVQA.

Additional Method Ablations. We observe that removing aggregation and instead using a simple majority vote leads to reduced accuracy by 1.0% in Table 9, confirming that the aggregation step is more effective than naive majority vote. In fact, majority vote underperforms aggregation in all but one setting tested in Table 8, demonstrating that proper processing of agent reasoning and tool outputs creates better final answers. Finally, removing the discussion phase causes the largest performance drop by 6.4%, emphasizing its role in refining the preliminary answers. In fact, the quality of answers after the initial reasoning phase is substantially higher (up to 11.1% higher for the weakest agent as reported in Table 7). This indicates the discussion step allows for stronger reasoning and responses, yielding to a more accurate downstream system output. The ablations emphasize the importance of incorporating multiple VLMs, expert tool information, and discussion steps to the system.

Model	Avg. Tokens Generated
Ovis2	74.3
MiniCPM-o	36.7
QwenVL	80.5
MAD with Consensus	1031.8
MAD with Judge	1196.8
DART	928.5

Table 10: We calculate the average tokens generated of each method on A-OKVQA. This represents the relative cost of each method.

Method	Total Runtime (hrs)	TFLOPs	Latency (s)
Ovis2	0.13	21.17	0.38
MiniCPM-o	0.09	18.92	0.26
QwenVL (7B)	0.05	19.04	0.16
QwenVL (32B)	0.20	88.01	0.64
MAD w/ Consensus	7.97	215.66	21.84
MAD w/ Judge	8.88	236.83	27.92
DART	10.52	158.15	32.75

Table 11: Comparison of system efficiency across different methods on A-OKVQA.

DART provides a balance between performance and computation compared to both single and

multi-agent baselines. We investigate the cost of using DART compared to single and multi-agent baselines. We first count the average number of tokens generated per method on A-OKVQA and present the results in Table 10. As expected, single agent methods with CoT are the cheapest, with QwenVL (the most token-heavy single agent baseline) generating 80.5 tokens compared to DART's 928.5. However, compared to multi-agent systems, DART needs 103.3 fewer tokens on average than a multi-agent debate system.

We further assess computational efficiency in Table 11, reporting total runtime, floating-point operations (TFLOPs), and latency per query. While DART incurs higher runtime and latency than single-model systems such as QwenVL, its cost remains competitive with other multi-agent methods. In fact, in terms of FLOPs, DART requires fewer FLOPs than the multi-agent baselines (although still behind in total runtime and latency). The observed difference in runtime and latency between multi-agent debate and DART is likely due to inefficient tool-calling implementation relative to discussion/answering implementations.

Additionally, we test a larger VLM (Qwen2.5-VL 32B) to compare efficiency and performance when scaling versus using multiple agents. While Qwen2.5-VL 32B is more cost-effective, its performance gains are modest (scoring 62.7% on A-OKVQA). Although pricier, multi-agent approaches like DART may offer a viable path to improved results when scaling yields diminishing returns. Overall, DART achieves stronger performance from the incorporation of tools at the cost of reduced efficiency compared to multi-agent methods and single-agent methods.

Each tool complements DART's performance. We seek to test whether DART is able to effectively use the tools that it has been equipped with. From Table 12 we can infer the relative importance of each tool to DART's overall performance on A-OKVQA. In fact, we observe that the performance drops roughly correlate with the tool usage frequency recorded in Fig. 3. More specifically, the *grounder* and *captioning* tools – each contributing to over 20% of tool invocations on A-OKVQA – are the most critical, as their removal leads to a degradation of more than 2% in accuracy. The remaining tools (*OCR*, *spatial reasoning*, *attribute detection*, and *reasoning*) yield smaller decreases, typically within 1%, suggesting that DART remains

Tool Witheld	Accuracy
DART	68.9
w/o Grounder	66.5
w/o Captioning	67.5
w/o Attribute Detection	68.6
w/o OCR	68.1
w/o Reasoning	68.4
w/o Spatial Reasoning	68.0
w/o Object Detection	68.9

Table 12: Ablation for tools of DART on A-OKVQA. We measure performance impact on the entire system when we remove a single tool from the tool set. Tools are ordered in terms of most to least frequent as reported in Fig. 3.

relatively robust to their absence. Interestingly, the omission of object detection does not noticeably affect accuracy, implying that visual grounding and caption-based context already provide sufficient perceptual cues for most questions. Overall, these results highlight that while certain tools (particularly grounding and captioning) play a dominant role, the complementary interaction among all tools contributes to the system's strong performance, demonstrating that DART effectively leverages its tool reasoning capabilities.

DART is robust to prompt choice. We test DART's performance under an altered set of prompts to test the system's sensitivity to prompt choice. Each stage's prompt is passed into GPT-5 (OpenAI, 2025) with the prompt "I am creating a prompt for a system that uses tools and agents. Rephrase the following prompt:" while we manually adjust the newly generated prompt to ensure the output format and handling of string variables is consistent. Testing this new set of prompts on A-OKVQA, we achieve accuracy of 68.7%, a 0.2% decrease from our previous baseline of 68.9%. While there is a dip in performance, the change is relatively small, and our method still outperforms the other baselines.

E Prompts

We provide the prompts used in our methods for reproducibility. The details on how these prompts are used are provided in Section 2.

F Qualitative Examples

We provide two qualitative examples showcasing the full DART pipeline in Fig. 6 and Fig. 5.

Disagreement Resolution Prompt

Here was the initial prompt: <full initial answer generation prompt>

Carefully review the following solutions from other agents for the provided question. Now, analyze what disagreements are occurring between the different agents.

<grouped solution>
You have the ability to call on different experts, each with their own specialized capabilities. Based on the disagreements you observed, pick out the set of experts (could be just one) that would be best equipped to solve all the disagreements.

Here are all the experts, their inputs, and their capabilities/usage:

"spatial" (input: list. objects that have confused spatial relations) - Has perfect understanding of spatial relations between objects. Use this when agents are unsure about the placement of items in a scene. "ocr" (input: none)- Can correctly read all text in an image. Use this when agents have differing views on what the text is in an image. "grounder" (input: list. objects you are trying to find) - Will find any object if it is an image, otherwise it will return nothing. Use this when agents are not agreeing on what's present in an image. "detector" (input: none) - Will provide a list of objects in the image, their counts, and their bounding boxes. Only use this when agents are differing in their counts of objects in an image. "captioning" (input: list. objects you want captions for) - Can give a detailed description of what's going in the image relevant to the question. Use this when agents might need a better idea of the general scene or descriptions of specific objects. "attribute" (input: list. objects you want attributes for) - Will give information on different features of objects in the image, including color, properties, catgories, and more. Use this when agents are confused about the features of relevant objects and need many surface level features. "reasoning" (input: list. objects you want reasoning for) - Has better world knowledge and advanced reasoning capabilities about what might be going on in an image. Use this when agents are confused or conflicting in their inferences about the scene. This is essentially a meta-reasoning agent that intervenes when models have different conclusions based on the same assumptions.

Output the expert(s) you need to resolve the disagreements to answer the original question: . This should be a JSON with this format like this. You can call more or less experts than this as needed:

```
{
    "experts": ["grounder", "attribute", "ocr"],
    "inputs": {
        "grounder": {
            "disagreement": "Agent 1 mentioned that there is a cat, but Agent 2 said there is no cat and instead said it is a dog.",
            "justification": "The grounder will help resolve the disagreement about the presence of a cat or dog in the image.",
            "arguments": ["cat", "dog"]
        },
        "attribute": {
            "disagreement": "Agent 1 said the flower is red, Agent 2 said it is orange, and Agent 3 did not specifically mention anything about the flower.
There also was confusion about the details of the car.",
            "justification": "The attribute expert will help resolve the disagreement about the color of the flower and provide details about the car.",
            "arguments": ["flower", "car"]
        },
        "ocr": {
            "disagreement": "Agents conflict on the text they see in the image.",
            "justification": "The OCR expert will see the text in the image and resolve the disagreement.",
            "arguments": []
        }
    }
}
```

Now give the expert output in the given format for the previous agent solutions and the question provided above. Do not be redundant on disagreements unless the expert is adding new information that better resolves the disagreement. Do not add an expert unless absolutely necessary.

Reminder, the question is: <question>

Tool Agreement Prompt

You are an aligning agent. That means that you determine whether or not two different outputs are misaligned (0) or aligned (1). An expert output has provided a response to a disagreement between multiple different agents. Your task is to determine whether or not the expert output is aligned with a single agent's output. Reason about the alignment of the agent output with the expert output. Then, output a 0 if the agent output is not aligned with the expert's output, and a 1 if it is aligned.

Output Format:
```
{
    "reasoning": "Your reasoning for the alignment of the agent output with the expert output.",
    "alignment": "0 if the agent output is not aligned with the expert's output, and 1 if it is aligned."
}
```

Disagreement: <disagreement>
Expert Output: <expert_output>
Agent Output: <agent_output>

Discussion Prompt

Carefully review the following solutions from other agents as additional information, and provide your own answer and step-by-step reasoning to the question. Then, give your confidence in your new answer.

Clearly state which point of view you agree or disagree with and why.

<grouped_solutions>

Here is information about the image from expert(s). You should take any information presented in this as factual unless you truly notice something wrong with the expert info (and make sure to state this).

<tool_outputs>

Output your response in the following format:

Reasoning: [reasoning for your answer to the original question based on the other agents responses]
Answer: [new answer]
Confidence: [confidence]
Some example outputs:

Reasoning: I agree with Agent 1's reasoning that the answer is A because it aligns with the medical expert's analysis of the CT scan. The medical expert confirmed that the condition described in the question is consistent with the findings in the image.
Answer: Option A
Confidence: 0.65

Reasoning: I disagree with Agent 2's reasoning that the answer is B. The medical expert's analysis suggests that the condition is more likely to be A, as it matches the symptoms described in the question and the findings in the image.
Answer: Option A
Confidence: 0.95

Reasoning: I agree with Agent 1 and 3's reasoning that the answer is C. The medical expert's analysis supports this conclusion, as it aligns with the findings in the image and the symptoms described in the question.
Answer: Option C
Confidence: 0.30

You must follow the provided format no matter what. This rule is unbreakable.

Aggregator Prompt

You are an aggregator model that will be given an image, question, and set of answer choices. Your tasks is to select the best final answer for the question. You will also be given various sources of information to help inform your decision. Each answer was generated by a different agent, and these agent will provide their reasoning for why they gave their answer. You will also be given tool outputs from expert models that directly relate to the question and were used to resolve disagreement among the answering agents. Feel free to defer to these expert tool outputs if the answers contradict the info from the tools.

The question is: <question>

Here are the different agent answers:
<grouped_output>

Here are the tool outputs:
<tool_outputs>

Now based on the provided information, provide your step-by-step reasoning for selecting the best, most correct answer to the question provided. Then, give your confidence in your selected answer.

Output in the following format:

Reasoning: [reasoning]
Answer: [answer]
Confidence: [confidence]

Question: When does meter enforcement have their days off?

Gold Answer

Weekends

---

1. Initial Round

QwenVL

SUNDAY, the parking meter indicates that meter enforcement is active from 9am to 5pm Monday through Friday. Therefore, the days when meter enforcement has their days off would be Saturday and Sunday. (CL: 70%).

MiniCPM-o

Weekends and holidays, the meter enforcement hours are stated as "9am–6pm M–F", which means Monday to Friday. Therefore, the days off would be Saturday, Sunday, and all public holidays (which typically fall on weekends). (CL: 85%)

Ovis

Saturday and Sunday, the parking meter in the image indicates that meter enforcement occurs from 8am to 6pm, Monday through Friday. This implies that meter enforcement does not occur on weekends, specifically Saturday and Sunday. (CL: 90%)

2. Disagreement Resolution

Qwen (Recruiter)

```
{
    "experts": ["ocr"],
    "inputs": {
        "ocr": {
            "disagreement": "Agents conflict on the text they see in the image
regarding the meter enforcement hours.",
            "justification": "The OCR expert will read the text in the image and
resolve the disagreement about the meter enforcement hours.",
            "arguments": []
        }
    }
}
```

Tool Outputs

On the parking meter:
Top section:
- "2", "pay meter"
Bottom section:
- "METER ENFORCEMENT", "9am–6pm M–F"
On the sign in the background:
The text is partially visible and appears to include: "NO", "2 HOURS", "PARKING", "EXCEPT", "DISABLED", "PERMITS"

3. Final Round

QwenVL

Saturday and Sunday, the parking meter in the image clearly indicates that meter enforcement is active from 9am to 6pm, Monday through Friday. This means that meter enforcement does not occur on weekends, specifically Saturday and Sunday. The OCR tool confirms this information, as it reads "METER ENFORCEMENT 9am–6pm M–F" on the parking meter. Therefore, the days when meter enforcement has their days off are Saturday and Sunday. (CL: 90%, TA: 1/1)

MiniCPM-o

Saturday and Sunday, he parking meter indicates that enforcement is active from 9am to 6pm, Monday through Friday. This means the days when there is no meter enforcement would be Saturday and Sunday. (CL: 95%, TA: 1/1)

Ovis

Saturday and Sunday, the OCR tool provides clear information about the parking meter\'s operation hours, which are "9am–6pm M–F". This directly contradicts the agent solutions that suggest meter enforcement operates on weekends and holidays. The OCR output is definitive and should be considered authoritative in this context. (CL: 100%, TA: 1/1)

4. Aggregation

Ovis 2 (Aggregator)

The OCR tool output clearly shows that the parking meter operates from 9am to 6pm, Monday through Friday. This information is definitive and directly addresses the question about the days when meter enforcement is not active. The OCR tool's output is authoritative and aligns with the agent solutions that correctly identify Saturday and Sunday as the days when meter enforcement is off. The other agent solutions also correctly interpret the meter's information, reinforcing the conclusion. The tool output confirms that meter enforcement does not occur on weekends, specifically Saturday and Sunday.

Answer: Saturday and Sunday
Confidence: 100%

Figure 5: Qualitative Example for DART. We have input question "When does meter enforcement have their days off?" with gold answer **Weekends**. Ovis is the only model to get it correct, with MiniCPM-o jumping to an improper conclusion and QwenVL being objectively incorrect. The recruiter identifies this disagreement on what text is said on the meter and calls on the OCR tool to resolve the disagreement. The OCR tools correctly identifies the text as M-F 9am-6pm. As a result, the models and aggregators are able to get to the correct answer in subsequent steps.

Question: What direction is the narrow end of the flag pointing?

Gold Answer

Left

---

1. Initial Round

QwenVL

Left, the flag is positioned on top of the tower, with its narrow end pointing towards the left side of the image. This indicates that the flag is blowing in the wind from right to left, which is consistent with the direction of the flag's narrow end. (CL: 90%).

MiniCPM–o

Right, the flag is flying on the left side of the building, which would be to the west if we assume this image was taken in London. Given that it's a standard way for flags to fly (with the blue part at the top), and assuming no wind direction changes during the day when photos are often taken, the narrow end pointing right makes sense as it means the wind is blowing from east to west or north–south depending on what time of year the photo was taken. (CL: 95%)

Ovis

Down, the flag appears to be at half–mast, which means it is positioned with its narrow end pointing downward. This is a common practice in many countries to show respect or mourning. (CL: 80%)

---

2. Disagreement Resolution

Qwen (Recruiter)

```
{
  "experts": ["spatial", "captioning"],
  "inputs": {
    "spatial": {
      "disagreement": "One agent says the flag points left, another says it
points right, and another says it points down.",
      "justification": "The spatial expert can determine the actual direction of
the flag.",
      "arguments": ["direction of flag"]
    },
    "captioning": {
      "disagreement": "Agents also disagree about whether the flag is at half–
mast or flying normally.",
      "justification": "The captioning expert can give a broader scene
description, clarifying whether the flag is raised fully or at half–mast.",
      "arguments": ["half–mast", "flag"]
    }
  }
}
```

Tool Outputs

Spatial Tool Output: The flag is waving in the wind, which is blowing from the right side of the image towards the left side.

Captioning Tool Output: This image shows Westminster Abbey, the Houses of Parliament, and the clock tower known as the Elizabeth Tower in London. The flag is flying at the top left of the building, indicating it is not a half–mast or a sign of mourning. The Westminster Bridge is visible in front of the building, spanning across the River Thames. The sky is clear and blue.

---

3. Final Round

QwenVL

Left, the Spatial Tool indicates that the flag is waving in the wind from the right side of the image towards the left side. The Captioning Tool confirms that the flag is flying at the top left of the building, which is consistent with the direction of the wind. The agent solutions suggest that the narrow end of the flag is pointing left, right, or down. However, the Spatial Tool's analysis is more reliable in this context, as it directly addresses the direction of the flag's movement. Therefore, the narrow end of the flag is pointing to the left. (CL: 90%, TA: 2/2)

MiniCPM–o

Right, to determine the direction of the narrow end of the flag, we need to analyze the image provided. The experts have given us valuable information about the location and orientation of the flag.

1. **Spatial Tool Analysis**: According to the spatial tool analysis, the flag is waving in the wind from right to left.
2. **Captioning Tool Analysis**: The caption indicates that the flag is flying at the top left of a building (Westminster Abbey), which suggests it's blowing towards the bottom right if we consider the perspective shown in the image.

Given these two pieces of evidence:
– The flag appears to be moving from right to left based on its position relative to the tower.
– The flag's movement seems consistent with being blown by wind coming from the west or north depending on the time of year when the photo was taken.

Considering both analyses together, the most logical conclusion is that the narrow end of the flag is pointing to the right side of the image because this aligns well with the observed wind direction indicated by the flag's motion (CL: 80%, TA: 1/2)

Ovis

Left, I agree with Agent 1 who states that the flag is positioned on top of the tower, with its narrow end pointing towards the left side of the image. This interpretation matches the wind direction suggested by the flag's positioning as seen in the expert description. (CL: 95%, TA: 2/2)

---

4. Aggregation

Ovis 2 (Aggregator)

The Spatial Tool Output confirms that the flag is waving in the wind from the right side of the image towards the left side. The Captioning Tool Output supports this by indicating that the flag is flying at the top left of the building, which aligns with the direction of the wind. The Spatial Tool and Captioning Tool both agree that the flag's narrow end is pointing to the left. The agent solutions that agree with this analysis provide strong evidence that the narrow end of the flag is pointing to the left. The Spatial Tool's analysis is particularly reliable in this context, as it directly addresses the direction of the flag's movement. Thus, the narrow end of the flag is pointing to the left.

Answer: Left
Confidence: 95%

Figure 6: Qualitative Example for DART. We have input question "What direction is the narrow end of the flag pointing?" with gold answer **Left**. The models all initially disagree, each pointing out different directions (down, left, right). The recruiter identifies this disagreement and calls on the spatial and captioning tools to resolve the disagreement. With the intervention of these tools specifying the direction of the flag, the models refine their answers to get closer to the gold answer, and the aggregator successfully picks the correct answer.