Learning to Select Visual Tools from Experience

Zeyi Huang[1], Yuyang Ji[3], Anirudh Sundara Rajan[1], Zefan Cai[1],
Wen Xiao[2], Haohan Wang[3], Junjie Hu[1], Yong Jae Lee[1]
[1]University of Wisconsin-Madison [2]Microsoft [3]University of Illinois Urbaba-Champaign

Abstract

We introduce VisualToolAgent (VisTA), a new reinforcement learning framework that empowers visual agents to dynamically explore, select, and compose tools from a diverse library based on empirical performance. Existing methods for tool-augmented visual reasoning either rely on training-free prompting or large-scale supervised fine-tuning; both lack active tool exploration and typically assume limited tool diversity, and fine-tuning methods additionally demand extensive human supervision. In contrast, VisTA leverages end-to-end reinforcement learning to iteratively refine sophisticated, query-specific tool selection strategies, guided solely by task outcomes. Leveraging reinforcement learning with verifiable rewards (RLVR), our framework enables an agent to autonomously discover effective tool-selection pathways without requiring explicit reasoning supervision. Experiments on the ChartQA, Geometry3K, MathVerse, and BlindTest benchmarks demonstrate that VisTA achieves significant performance gains over training-free and fine-tuning baselines, especially on out-of-distribution examples. These results highlight VisTA's ability to enhance generalization, adaptively utilize diverse tools, and pave the way for flexible, experience-driven visual reasoning systems. Project website: https://oodbag.github.io/vista_web/.

1. Introduction

Recent advances in Large Language Models (LLMs) [2, 10, 60] and Vision Language Models (VLMs) [27, 33, 67] have unlocked impressive capabilities across tasks such as mathematical problem solving, code generation, and visual question-answering. However, these models are still inherently limited by the static nature of their architectures and the fixed information stored in their weights. To overcome these constraints, recent work explores augmenting LLMs and VLMs with external tools [11, 18, 20, 22, 25, 47, 55], dramatically expanding their functionality. Tool augmentation enables access to expert knowledge sources and dynamic computation, such as invoking a Python interpreter for self-verification, thereby enhancing reasoning performance on complex tasks.

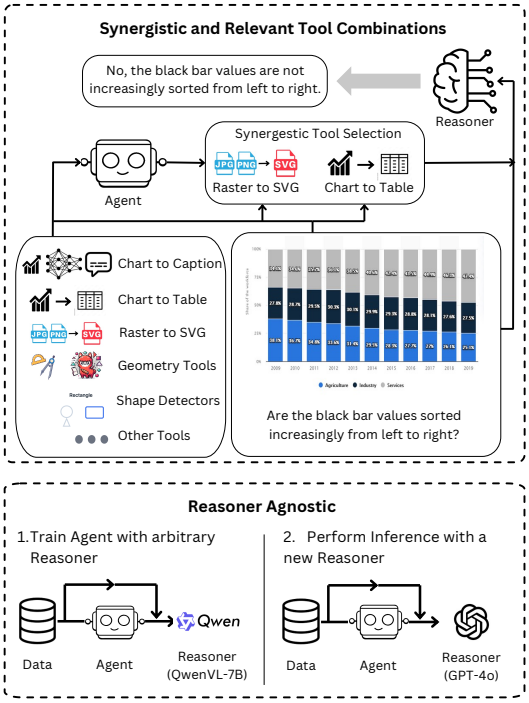

Figure 1. **Overview of VisTA**. (Top) Our method trains an agent to autonomously discover effective combinations of visual tools without human supervision. (Bottom) By decoupling the agent from the reasoner, the learned policy can be seamlessly integrated with a wide range of reasoning models.

However, the current paradigm for tool integration faces significant limitations in both LLMs and VLMs. Current approaches typically either rely on large-scale fine-tuning with human supervision to teach LLMs how to invoke tools [35, 51] or depend purely on the LLMs' internal world knowledge in a training-free manner [20, 25, 41]. These methods often rely on tool demonstrations [35, 51] or detailed tool descriptions to instruct LLMs on their usage [25, 41]. As a result, they lack the ability to automatically explore, select, or adapt tool choices based on the specific characteristics of each query, particularly when multiple tools of the same type with varying capabilities are available which is common in real world settings. The chal-

lenge is particularly pronounced when integrating tools with unknown, partially documented capabilities or inconsistently performing capabilities, where actual performance may differ from descriptions. When retrieving tools from diverse sources, the LLMs lack comprehensive knowledge of their strengths and weaknesses. Without a mechanism for experiential learning, the system cannot determine optimal tool selection or discover synergistic tool combinations that might emerge through collaborative deployment.

In realistic applications, tools vary substantially in functionality and applicability across different problem domains. Within each tool category, individual implementations exhibit varying capabilities that make them differentially effective across contexts. This presents a sophisticated decision-making challenge ideally suited for reinforcement learning (RL) [56]. RL's intrinsic exploration-exploitation mechanism enables agents to systematically assess and adaptively identify the most effective tools based on empirical performance rather than pre-specified rules. Through iterative interactions with its environment, an RL agent can learn adaptive strategies that dynamically adjust tool combination based on specific queries, or even potentially discover non-obvious tool utility patterns that may not be apparent from tool descriptions alone.

Therefore, we introduce a new RL framework, Visual-ToolAgent (VisTA), that trains autonomous agents to intelligently select optimal tools from multiple available options. Unlike training-free [20, 25, 41] and fine-tuning approaches [35, 51], our RL-based method inherently supports exploration-exploitation mechanisms, allowing agents to systematically experiment with various tool combinations through iterative interactions. In this work, we focus on the visual reasoning task. Our framework consists of an autonomous agent that learns through end-to-end RL training to dynamically select optimal tools for guiding a fixed VLM in solving complex visual reasoning problems. VisTA supports both efficient *single-shot* tool selection, where all tools are chosen in one step, and *multi-turn tool refinement*, where the agent incrementally updates its tool choices across multiple interactions based on intermediate observations and task feedback. The reasoner gradually constructs the final answer conditioned on accumulated tool outputs, and generation halts either through early stopping or upon reaching a fixed interaction budget.

Critically, our framework allows the VLM itself to remain frozen during RL training, which means that the agent's learned selection strategies can be transferred to different reasoning models *without retraining*, a critical advantage for deployment flexibility. Our framework employs reinforcement learning with verifiable rewards (RLVR) to enable our agent to autonomously discover effective tool-selection pathways entirely from scratch, without explicit reasoning examples. For a detailed look at how the agent performs inference and selects tools in practice, see the examples in Fig 6; more cases appear in the Supp.

We evaluate our method on five visual reasoning benchmarks: ChartQA [43], its out-of-distribution variant ChartQA-OoD, Geometry3K [39], MathVerse [70], and BlindTest [49]. These datasets span a diverse range of reasoning settings, including chart-based analysis, diagrammatic reasoning, mathematical question answering, and spatially grounded visual understanding. Our experimental results show that the proposed RL-based approach significantly outperforms both training-free and fine-tuning methods. The performance gap further widens on the more challenging ChartQA-OoD benchmarks, demonstrating VisTA's superior ability to generalize to novel visual scenarios and maintain robustness under distribution shifts.

2. Related Work

Tool-Augmented Reasoning. LLMs have shown significantly improved reasoning capabilities when augmented with external tools such as search engines [29], calculators [12], and Python interpreters [11, 18]. Programming-based approaches [11, 18], for example, integrate Python interpreters to simplify intermediate steps and validate final outputs, enhancing accuracy on mathematical tasks. Similar strategies have been adopted in the visual domain. Recent VLM methods [22, 25, 55] generate Python code to invoke specialized vision modules, decomposing complex visual tasks into simpler sub-tasks, each addressable by dedicated vision tools. However, existing approaches often rely on human demonstrations or annotations [35, 51], or operate in a training-free manner using only the model's internal knowledge [25, 41]. These methods typically offer limited tool diversity and depend heavily on explicit tool descriptions [25, 35, 41, 51], lacking the capacity to autonomously explore or adapt tool use to specific queries. In contrast, our proposed VisTA framework enables VLMs to autonomously explore and select tools based on empirical performance, without human-designed priors. By training an agent with RL, VisTA discovers context-dependent tool selection policies that adaptively tailor tool usage to the nuanced requirements of each visual reasoning task.

Reinforcement Learning for Enhanced Reasoning. RL has shown strong potential in enhancing complex reasoning abilities and enabling inference scaling. Models like OpenAI's o1 [46] and DeepSeek-R1 [21] have achieved notable success in tasks such as mathematical problem solving by leveraging long chain-of-thought (CoT) [61] reasoning. These models excel at strategies like mistake correction, step decomposition, and iterative refinement, resulting in more structured and extended reasoning. Recently, several studies [26, 37] have adapted the DeepSeek-R1 framework to visual reasoning, training models to generate CoT-based outputs directly from visual inputs. While these approaches

focus on finetuning the reasoning model itself to perform end-to-end visual inference, our work takes an orthogonal perspective. Instead of modifying or retraining the reasoning model, we propose to train an autonomous agent that reasons about which tools to select to best assist a *frozen* reasoning model in solving a given query. By learning to select supportive tools based on each query, our agent enhances performance without altering the model's internal parameters (thereby preserving generalization to other tasks). This design also ensures broad compatibility across different visual reasoning models and offers a flexible, modular strategy for improving multimodal reasoning systems. ReTool [15] is a concurrent work that uses RL to teach LLMs to invoke code execution tools for text-based reasoning. Similarly, models like OpenAI's o3 [28] demonstrate an ability to "think with images" by dynamically applying a limited set of visual tools, such as zooming or flipping, to improve visual understanding In contrast, our work tackles the challenging setting of visual reasoning with diverse tool choices, requiring agents to adaptively select the most effective tools.

Visual Reasoning Tasks. Some visual reasoning tasks, such as depth estimation and spatial reasoning [16], can be effectively solved using straightforward, specialized tools like depth estimators or object detectors. In contrast, we target more complex and cognitively demanding tasks, where optimal tool selection is query-dependent and may not be obvious. Chart understanding [40, 43] is a challenging task and a strong indicator of visual reasoning capabilities. It requires models to process numerical data, textual labels, and complex visual structures, often demanding precise quantitative reasoning (e.g., measuring bar heights). Geometry questions [39, 40, 70] pose a similarly demanding challenge, requiring fine-grained diagram understanding followed by text-based reasoning grounded in visual details. BlindTest [49] evaluates an orthogonal aspect of visual reasoning: fine-grained spatial perception on extremely simple images, such as counting intersections, where many state-of-the-art VLMs surprisingly fail despite the tasks being trivial for humans. The strong performance of our method across these benchmarks highlights the benefit of learning a tool-selection policy capable of adjusting to the visual demands of each query.

3. Method

In this section, we present VisTA, a reinforcement learning (RL) framework for tool-augmented visual reasoning. In contrast to previous methods that rely on training-free or fine-tuned strategies, VisTA empowers an agent to learn how to select tools through trial-and-error interaction, without requiring manual supervision. By harnessing the exploration-exploitation dynamics of RL, the agent adaptively chooses from a wide array of tools based on performance feedback. Notably, the core reasoning model is kept fixed, allowing the learned tool-selection policy to transfer across different reasoning backbones without the need for additional training.

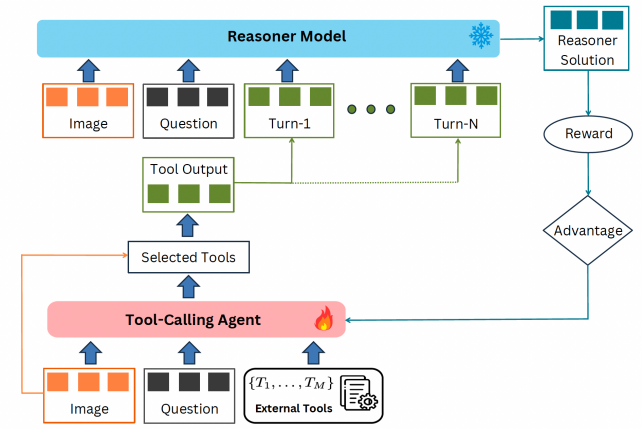

Figure 2. **Multi-turn Policy Optimization.** For each query, the agent selects tools from a unified pool and feeds their outputs to a frozen reasoner. The reasoner provides a scalar confidence, which the agent uses to decide whether another refinement turn is needed. This iterative interaction continues for up to three rounds during training. Rewards derived from the final prediction are used to train the multi-turn tool-selection policy.

3.1. Problem Formulation

Let (q, I) denote a vision-language query, consisting of an image I and an associated text query q, sampled from a task distribution $\mathcal{D}$. We consider a setting with a *frozen* vision-language model (the *reasoner*) f_θ and a library of external tools $\mathcal{T} = \{T_1, \ldots, T_M\}$, where M is the total number of available tools. These tools span a heterogeneous set of vision modules drawn from all tasks, including chart-analysis tools, diagram parsers, mathematical tools, object detectors, depth estimators and other specialized perception tools. Many tool types include multiple variants with different capability levels, reflecting realistic settings where tools differ in accuracy and robustness rather than function alone. The details of all the tools can be found in Supp.

We define the agent's observation or *state* for the first round as $s_1 = (q, I)$, encompassing both the image and its corresponding query. Our objective is to learn a *selection policy* $\pi_\phi(t \mid s_r)$, implemented as a vision-language model (the *agent*), that determines which tools to deploy at round r. In the single-turn case ($r = 1$), the policy outputs a sequence of selected tools $t_1 = \langle T^{(1)}, \ldots, T^{(K)} \rangle$ with $T^{(i)} \in \mathcal{T}$. Here $K \leq M$ adaptively varies with task complexity.

For the multi-turn setting introduced later, the agent's state at round $r > 1$ additionally includes the history of the agent's past decisions and the reasoner's feedback, denoted as $\{(t_1, c_1), \ldots, (t_{r-1}, c_{r-1})\}$, where $c_i \in [0, 1]$ is the confidence score output by the reasoner after observing the tool outputs selected at round i.

3.2. VisTA Framework

Our VisTA framework implements an end-to-end pipeline to learn a policy for dynamic tool selection in visual reasoning tasks (Figure 2), leveraging reinforcement learning to enable systematic exploration of tool combinations. In the single-turn version, the pipeline operates as follows:

1. The vision-language agent observes the state $s_1 = (q, I)$ and selects a sequence of tools $t_1 = \langle T^{(1)}, \ldots, T^{(K)} \rangle$ via a policy $\pi_\phi(t_1 \mid s_1)$.
2. Each selected tool is executed on the image to produce outputs $o_1 = \langle O^{(1)}, \ldots, O^{(K)} \rangle$.
3. These outputs are combined with the original inputs to form an augmented prompt.
4. The frozen reasoner f_θ processes the augmented prompt to produce the final answer $y_{\text{img+tools}} = f_\theta(q, I, o_1)$.

During training, we also compute a baseline prediction $y_{\text{img}} = f_\theta(q, I)$ using only the original query and image, which enables us to measure the impact of selected tools on reasoning performance, as detailed in our reward formulation.

3.3. Multi-Turn Tool Refinement

To further enhance VisTA's flexibility, we introduce a multi-turn tool refinement mechanism in which the agent iteratively interacts with the frozen reasoner to refine its tool choices over multiple rounds. This extension preserves the core single-turn structure while enabling deeper reasoning for complex queries where a single tool invocation may be insufficient.

At round r, the agent observes the query, image, and the full history of prior interactions $s_r = (q, I, \{(t_1, c_1), \ldots, (t_{r-1}, c_{r-1})\})$, where each $c_i \in [0, 1]$ is a scalar confidence score produced by the reasoner using the prompt: "Given the image, question, and tool outputs, output only a confidence score in $[0, 1]$ indicating how likely you can answer the question correctly with the current information." The confidence reflects how sufficient the reasoner believes the accumulated tool outputs are for answering the query. Conditioned on this state, the agent selects a new tool set t_r, which is executed to produce outputs $o_r = \{O_r^{(1)}, \ldots, O_r^{(K_r)}\}$. We denote the structured history of all tool sets and outputs up to round r as $o_{\leq r} = \{(t_1, o_1), \ldots, (t_r, o_r)\}$.

The reasoner then consumes this structured history and outputs an updated confidence score $c_r = f_\theta^{\text{conf}}(q, I, o_{\leq r})$. If c_r exceeds a threshold τ[1], the interaction stops early; otherwise, the agent proceeds to the next round. The final answer is generated only after termination, using the entire accumulated history $o_{\leq r^*}$ where r^* is the stopping round.

For multi-turn trajectories, we apply a token-wise loss mask to ignore losses on observation tokens not generated by the agent (e.g., the reasoner's confidence values). These tokens serve only as contextual information, ensuring that gradients are assigned solely to the agent's tool-selection decisions.

[1]Empirically, we find $\tau = 0.9$ to be a good choice.

3.4. Policy Optimization

To train the agent to discover effective tool combinations, we build upon GRPO [53] and adapt it to the visual tool-selection setting with a task-specific reward. At each training step, the agent samples G candidate tool sets $\{t^j\}_{j=1}^G$ from the policy $\pi_\phi(t \mid s)$, where each t^j denotes one sampled tool-selection action for the same query.

For each candidate t^j, we compare the reasoner's baseline prediction $y^j_{\text{img}} = f_\theta(q, I)$ with the prediction obtained using the selected tools, $y^j_{\text{img+tools}} = f_\theta(q, I, o^j)$, where o^j denotes the tool outputs associated with t^j. The reward is defined as: $r^j = +1$ if $y^j_{\text{img}} \neq y^*$ and $y^j_{\text{img+tools}} = y^*$ (tools help), $r^j = -0.5$ if $y^j_{\text{img}} = y^*$ and $y^j_{\text{img+tools}} \neq y^*$ (tools hurt), $r^j = 0$ if both fail, and $r^j = +1$ if both succeed. This encourages selecting helpful tools and penalizing harmful ones.

We compute the group-relative advantage $A^j = \frac{r^j - \text{mean}(r^1, \ldots, r^G)}{\text{std}(r^1, \ldots, r^G)}$, which contextualizes each reward relative to other candidates for the same query. The policy is updated by maximizing:

$$\mathcal{J}(\phi) = \mathbb{E}_{s\sim\mathcal{D}}\mathbb{E}_{\{t^j\}\sim\pi_{\phi_{\text{old}}}}\Bigg[\frac{1}{G}\sum_{j=1}^{G}\min\Big(\frac{\pi_\phi(t^j \mid s)}{\pi_{\phi_{\text{old}}}(t^j \mid s)}A^j,\ \text{clip}\big(\frac{\pi_\phi(t^j \mid s)}{\pi_{\phi_{\text{old}}}(t^j \mid s)},\ 1\pm\epsilon\big)\,A^j\Big) - \beta\,\mathbb{D}_{\text{KL}}(\pi_\phi \,\|\, \pi_{\text{ref}})\Bigg] \tag{1}$$

This objective stabilizes exploration through ratio clipping and KL regularization, enabling the agent to assign higher probability to tool combinations that consistently improve reasoning performance.

3.5. Tool Selection Prompting

We use a minimal prompt template that lists all tools by index (e.g., 0 to $M-1$) and their coarse functional categories (e.g., chart analysis, symbolic parsing, object detection and so on). No detailed descriptions or usage examples are provided.

"You are an expert agent selecting tools to help a reasoning model solve a visual-language query. You have access to multiple tools indexed from 0 to $M-1$, grouped into several functional types: type1, type2, type3... Function: 0: type1 (A), 1: type1 (B), 2: type1 (C), 3: type2 (D), 4: type2 (E) .... Given the image and query {Question}, output the index numbers of the tools you believe are most helpful, as a comma-separated list inside <answer> tags."

Method	Agent Model	Reasoning Model	ChartQA	ChartQA(OoD)	Geometry3K	MathVerse
Training-Free	-	QwenVL 7B	76.4	62.3	54.0	46.7
Training-Free	QwenVL 7B	QwenVL 7B	76.1	66.8	51.3	48.5
Training-Free	GPT4o	QwenVL 7B	73.0	66.4	51.5	47.9
RL	-	QwenVL 7B	77.5	64.3	41.0	49.2
Ours (Single-turn)	QwenVL 7B	QwenVL 7B	79.1	72.7	55.3	50.8
Ours (Multi-turn)	QwenVL 7B	QwenVL 7B	79.9	75.8	57.0	52.1

Table 1. **Main Results.** These results highlight VisTA's ability to support complex, multi-modal reasoning where tools provide complementary visual understanding. VisTA substantially improves accuracy across all tasks, with multi-turn refinement providing additional gains.

This lightweight prompting exposes only high-level structure, encouraging the agent to learn tool effectiveness entirely through reinforcement learning rather than relying on predefined tool semantics.

4. Experiments

We evaluate our VisTA framework on visual reasoning benchmarks, comparing to training-free baselines, alternative RL-based methods. We also analyze VisTA's tool selection strategies and distribution, and agent behavior dynamics over training.

4.1. Experimental Setup

We conduct experiments on five datasets that span a wide range of visual reasoning scenarios: ChartQA [43], ChartQA-OoD, Geometry3K [39], MathVerse [70], and BlindTest [49]. ChartQA and its OoD variant evaluate chart understanding under both standard and perturbed conditions (e.g., removing textual labels). Geometry3K and MathVerse focus on diagrammatic math reasoning, while BlindTest assesses low-level perceptual capabilities, an area where even strong VLMs (e.g., GPT-4o) continue to struggle.

To train a single general-purpose VisTA agent, we combine the training sets of ChartQA, Geometry3K, and BlindTest and randomly sample 800 examples from each, forming a diverse multi-domain pool. Since the agent learns to operate over a large and heterogeneous tool set, broad coverage of tool types matters more than full dataset size. RL-based selection policies are highly data-efficient in such settings, and we find that a few hundred examples per domain are sufficient for learning stable and effective tool preferences.

We train VisTA on 8 NVIDIA A100 GPUs using a batch size of 1 query per GPU, with 4 rollouts per query to encourage exploration of diverse tool subsets. We adopt the AdamW [38] optimizer with a learning rate of 5×10^{-5}, and train for 100 iterations. During inference, we allow up to three rounds of refinement in our multi-turn setting, which reflects the maximum number of interactions feasible within GPU memory limits. Early stopping is triggered automatically based on the reasoner's confidence estimates.

We construct a large, unified, and general-purpose tool pool designed to support a broad range of visual reasoning needs. Rather than tailoring tools to specific benchmarks, we assemble a diverse ecosystem of vision and reasoning utilities that can be reused across domains, enabling the agent to learn broadly applicable tool-selection policies. The pool spans chart analysis, diagram parsing, mathematical reasoning, and low-level perception, reflecting the diversity of real-world multimodal tasks. The chart-understanding tools include chart-to-table converters [32, 44, 64], chart-to-SVG extractors [9, 42, 66], and chart-captioning modules [6, 45, 62] (three variants per type). The geometry tools consist of symbolic parsers [52, 71] and visual–symbolic solvers [17, 48] (two variants each), which, while useful for geometry tasks, are general enough to apply to other visually grounded mathematical problems. To further broaden the pool, we add generic perception tools such as object detectors [23, 36], depth estimators [50, 68], and low-level visual element detectors [7, 14, 19, 24, 30, 69]. Across categories, many tools offer multiple capability variants, differing in accuracy, resulting in a 23-tool unified pool. Learning in this setting requires the agent to identify which tool families are relevant for a given query and to develop fine-grained preferences among tools with overlapping but distinct behaviors. This general-purpose design makes the learned policy broadly transferable and provides a realistic foundation for scalable multimodal tool selection.

4.2. Main Results

We first evaluate VisTA on four visual reasoning benchmarks: ChartQA, ChartQA-OoD, Geometry3K, and MathVerse, and compare against training-free baselines, RL-based alternatives, and strong modern VLMs [1, 3–5, 8, 13, 27, 57–59]. For all experiments, the tool-selection agent is QwenVL2.5-7B, and the reasoning model is kept frozen throughout training.

Table 1 presents the results. Both the single-turn and multi-turn versions of VisTA outperform all baselines across every benchmark, showing that RL-driven tool selection provides substantial gains over training-free prompting and direct RL finetuning of the reasoner. Even without refinement, our single-turn agent achieves strong results—79.1% on ChartQA, 72.7% on ChartQA-OoD, 55.3% on Geometry3K, and 50.8% on MathVerse, already surpassing all training-free baselines and a GRPO-trained reasoner that

Method	Agent Model	Reasoning Model	ChartQA	ChartQA(OoD)	Geometry3K	MathVerse
Training-Free	-	GPT4o	84.3	50.1	50.1	50.2
Training-Free	QwenVL 7B	GPT4o	82.3	67.1	48.7	51.4
Training-Free	GPT4o	GPT4o	84.6	73.3	49.5	53.6
Ours	QwenVL 7B	GPT4o	88.1	75.6	52.0	55.8

Table 2. **Transfer Results.** These results highlight VisTA's flexibility and compatibility with stronger frozen reasoning models like GPT-4o at deployment time. VisTA's tool-selection policy, trained with QwenVL-7B, transfers seamlessly to a stronger frozen reasoner (GPT-4o), yielding substantial gains across all benchmarks.

does not use tools.

Allowing the agent to revise its tool choices using confidence feedback leads to further improvements. The 3-turn agent achieves 79.9/75.8/57.0/52.1, consistently outperforming the single-turn version. These gains demonstrate that multi-turn refinement helps the agent correct suboptimal initial choices, request additional visual evidence, and adapt its tool usage to query difficulty.

The advantage of multi-turn refinement is especially evident on ChartQA-OoD, where VisTA improves over the GRPO-trained reasoner by 11.5 points (75.8 vs. 64.3). This suggests that iterative tool usage yields stronger visual grounding than directly optimizing the frozen model itself.

4.3. Transfer to Different Frozen Reasoning Model

A key strength of our method is its transferability. Without any retraining, the tool-selection policy learned with QwenVL 7B can be paired with GPT-4o as the reasoner. Table 2 shows the results. In this setting, we achieve 88.1% on ChartQA and 75.6% on ChartQA-OoD, surpassing the best training-free GPT-4o baseline by 3.5 and 2.3 points respectively, and also achieve 52.0% on Geometry3K, outperforming the best baseline (50.1%) by 1.9 points. This demonstrates VisTA's flexibility and compatibility with stronger reasoning models at deployment time.

4.4. Comparison to State-of-the-Art

Table 3 shows the comparison to state-of-the-art VLMs. VisTA achieves the best performance on Geometry3K and MathVerse, reaching 57.0% and 73.1% respectively, significantly outperforming all prior methods. On ChartQA, VisTA ranks third overall, only slightly behind Claude-3.5 Sonnet and InterVL2-76B (90.8 vs. 88.4 vs. 88.1), and surpasses other strong baselines such as Molmo-72B, Gemini 1.5 Pro, and Molmo-72B. This demonstrates that our approach is both highly effective on complex chart reasoning tasks and substantially more capable on geometric benchmarks.

4.5. Effect of Multi-Turn Refinement

Since our training setup supports up to three rounds of refinement, we evaluate the agent under 1-turn, 2-turn, and 3-turn settings to quantify the benefit of iterative interaction. While the single-turn agent selects tools once, multi-turn agents revise their choices using the reasoner's confidence feedback.

Method	ChartQA	Geometry3K	MathVerse
Ours	88.1	57.0	73.1
GPT-o1 [28]	-	–	67.7
Gemini 2.0 Flash [57]	-	–	65.6
DeepEyes [72]	-	–	47.3
Intern2VL-8B [58]	83.3	26.5	–
Intern2VL-8B-ShortCoT [58]	–	29.7	–
Geo-Intern2VL-8B [58]	–	30.7	–
G-LLAVA-7B [17]	–	22.4	–
Math-LLAVA-13B [54]	–	33.1	–
QvQ-72B-Preview	–	29.4	–
RedStar-Geo-8B [63]	–	33.6	–
GPT4v [2]	78.1	–	–
GPT4o-0513 [27]	85.7	–	–
Gemini 1.5 Flash [57]	85.4	–	–
Gemini 1.5 Pro [57]	87.2	–	–
Claude-3 Haiku [5]	81.7	–	–
Claude-3 Opus [5]	80.8	–	–
Claude-3.5 Sonnet [5]	90.8	–	58.2
PaliGemma-mix-3B [8]	33.7	–	–
Phi3.5-Vision-4B [1]	81.8	–	–
InternVL2-Llama-3-76B [58]	88.4	–	–
Pixtral-12B [3]	81.8	–	–
Llama-3.2V-11B-Instruct [4]	83.4	–	–
Llama-3.2V-90B-Instruct [4]	85.5	–	–
LLaVA-1.5-7B [34]	17.8	–	–
LLaVA-1.5-13B [34]	18.2	–	–
xGen-MM-interleave-4B [65]	60.0	–	–
Cambrian-1-8B [59]	73.3	–	–
Cambrian-1-34B [59]	75.6	–	–
LLaVA OneVision-7B [31]	80.0	–	–
LLaVA OneVision-72B [31]	83.7	–	–
MolmoE-1B [13]	78.0	–	–
Molmo-7B-O [13]	80.4	–	–
Molmo-7B-D [13]	84.1	–	–
Molmo-72B [13]	87.3	–	–

Table 3. **Comparison to state-of-the-art VLMs.** For our approach, the agent model is QwenVL 7B, and the reasoners are GPT4o for ChartQA, QwenVL 7B for Geometry3K and GPTo1 for MathVerse.

The trend is consistent across datasets: introducing a second turn already yields clear improvements, and the 3-turn agent achieves the best results.

Using confidence-based early stopping, the agent rarely exhausts all three turns. On average, it uses only 1.1 turns on ChartQA and 1.4–1.8 turns on the harder datasets (ChartQA-OoD, Geometry3K, MathVerse). This shows that refinement is invoked primarily for challenging cases, improving accuracy while keeping computation efficient. Examples where

	ChartQA	ChartQA(OoD)	Geometry3K	MathVerse
1-turn	79.1	72.7	55.3	50.8
Up to 2 turns	79.6	74.4	56.3	51.7
Up to 3 turns	79.9	75.8	57.0	52.1
Avg Turn #	1.1	1.8	1.4	1.5

Table 4. Performance across different numbers of refinement turns and the average turns used with confidence-based early stopping.

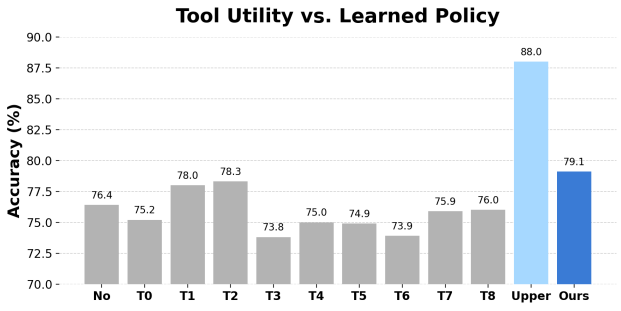

Figure 3. Comparison of ChartQA accuracy across each chart-specific tools (T0–T8), the no-tool baseline (No), our RL-based selection policy (Ours), and a pseudo-upper bound (Upper).

multi-turn refinement corrects single-turn errors are included in the supplementary material.

4.6. Tool Selection Analysis

To evaluate whether our tool selection policy effectively learns to combine and select the most appropriate tools for each query, we compare its performance against using individual tools in isolation. Specifically, we feed each tool's output along with the original input to the frozen reasoner and record its accuracy on the ChartQA benchmark. We also compute a pseudo-upper bound of 88.0% by treating a query as correct if any single tool enables the reasoner to produce the correct answer. This serves as a loose upper limit on what could be achieved with perfect single-tool selection, though it does not account for the benefits of combining multiple tools.

Fig. 3 shows the performance of each chart-specific tool (T0–T8), as well as the no-tool baseline (76.4%). While certain tools, such as T2 (78.3%) and T1 (78.0%), improve upon the no-tool baseline, the large gap to the pseudo-upper bound (88.0%) suggests that no single tool consistently performs best across all queries. Different tools appear to be optimal for different subsets of the data. Ideally, a well-trained policy should learn to select the most effective tool(s) for each specific query, achieving performance that surpasses any static tool choice. Our method achieves 79.1% accuracy, outperforming all individual tools. This suggests that the policy learns to go beyond fixed tool usage and adapt its selection based on query-specific context. Our policy also closes some gap between the best individual tool and the pseudo-upper bound, indicating progress toward optimal tool selection without explicit supervision. **Note.** For clarity, we present

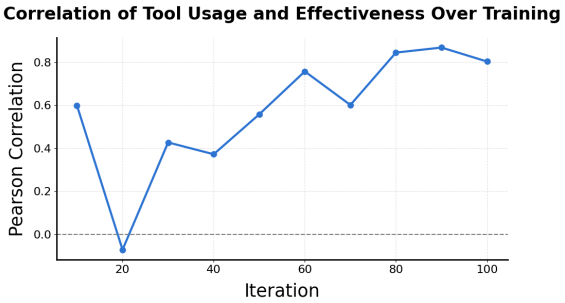

Figure 4. Pearson correlation between tool usage frequency and individual tool performance.

detailed analysis only on chart-specific tools (T0–T8), as they offer a stable and interpretable evaluation.

4.7. Agent Behavior Evolution During RL Training

To analyze whether the agent is learning to prefer more effective tools, we track the correlation between tool usage frequency and individual tool performance over training. Specifically, every 10 iterations, we compute the Pearson correlation coefficient between the usage counts of each tool and their corresponding standalone accuracy (as reported in Fig. 3).

Figure 4 shows the evolution of this correlation across training iterations. Despite some initial fluctuations, we observe a clear upward trend, with the correlation increasing from near zero to over 0.8 as training progresses. This indicates that the agent is gradually aligning its tool selection strategy with the relative utility of each tool, favoring those that contribute more to the reasoner's accuracy. These results suggest that our RL-based policy does not rely on fixed heuristics but instead learns to discriminate among tools based on their empirical contribution to task success. The emergence of this alignment over time provides further evidence that the agent is effectively adapting its behavior through reinforcement feedback.

4.8. Tools Selection Distribution

Figure 5 illustrates the tool selection distribution on the test set for our RL-trained agent, as well as for the training-free baselines using QwenVL-7B and GPT-4o. Again, we analyze the nine chart-understanding tools (T0–T8) to study selection behavior in a controlled ChartQA setting. Our method clearly shows a strong preference for Tool 1 and Tool 2, both are chart-to-table tools, which are among the most effective tools based on individual performance (see Fig. 3). In contrast, low-performing tools such as Tool 3 (chart-to-SVG) and Tool 6 (caption module), are selected far less frequently, suggesting that the learned policy has effectively adapted to favor high-utility tools based on empirical feedback.

The QwenVL-7B baseline, which operates in a training-free manner without reinforcement feedback, exhibits a more balanced selection pattern that resembles a near-normal dis-

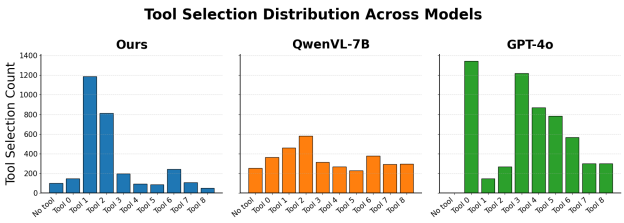

Figure 5. Tool selection frequency across our RL-trained agent, QwenVL-7B, and GPT-4o. Our method strongly favors effective tools (Tools 1 and 2) and avoids less useful ones, while QwenVL-7B shows a uniform distribution and GPT-4o selects broadly without clear alignment to tool performance.

Method	Agent Model	Reasoning Model	BlindTest
Training-Free	-	GPT4o	48.5
Training-Free	GPT4o	GPT4o	51.8
Ours	QwenVL 7B	GPT4o	**53.4**

Table 5. Performance of different tool selection strategies on BlindTest.

tribution. This indicates that it lacks strong preferences and does not consistently prioritize the most effective tools. Meanwhile, GPT-4o tends to select more tools per query, rarely opts for no tool, and distributes its selections across a broader set of tools. However, this broader usage still lacks clear alignment with tool effectiveness, showing no strong correlation between selection frequency and tool performance.

These differences highlight the benefit of learning a tool selection strategy through RL. Unlike training-free approaches that rely solely on static prompt understanding, our agent adapts its behavior based on downstream task outcomes, leading to more effective tool use.

4.9. Qualitative Results

In this section, we provide qualitative examples showing how our agent selects appropriate tools according to the query. From the Geometry3k dataset, we present an example in Figure 6. Based on the context, our agent selects a formal diagram parser tool, which represents the objects in the diagram and their relationships using formal language. This structured information is then used by the Reasoner model to correctly answer the question. More qualitative examples are provided in the supplementary material.

4.10. Results on BlindTest Benchmark

Finally, to further evaluate the generalizability of VisTA, we test on BlindTest [49], a benchmark designed to probe *low-level visual perception*, such as detecting line intersections, circle overlaps, and simple spatial relations, where even top VLMs (e.g., GPT-4o and Gemini 1.5 Pro) struggle despite their strong high-level reasoning ability. Using the same setting as in our main experiments, the VisTA agent learns to

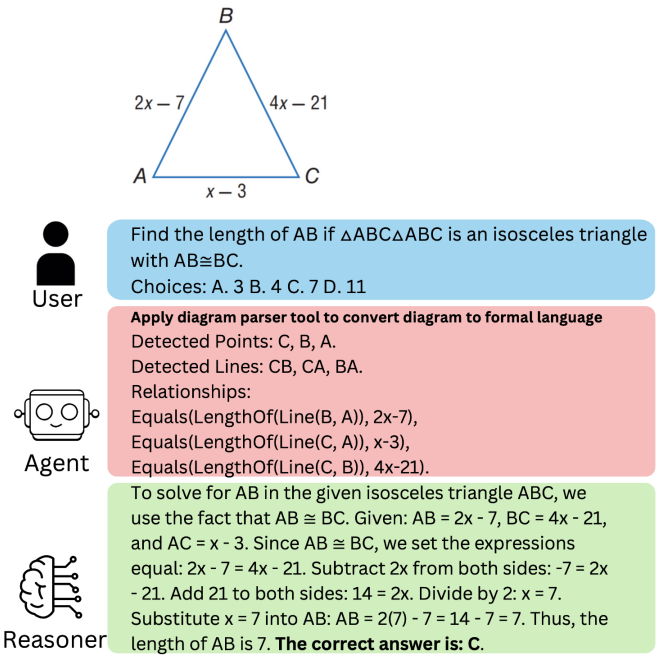

Figure 6. **Tool Selection for a Geometry Question**. Our agent selects the formal diagram parser which accurately extracts essential details from the diagram and represents them in a formal structure. Leveraging this representation, the reasoner is able to determine the correct answer.

select perception-oriented tools, such as low-level geometric analyzers, that complement the frozen reasoner's visual capabilities. As shown in Table 5, our method reliably improves accuracy over training-free baselines, demonstrating that reinforcement-learned tool policies can enhance fine-grained spatial reasoning even on tasks where VLMs alone often fail.

5. Discussion and Conclusion

We introduced VisTA, an RL framework enabling visual agents to autonomously select effective external tools for multimodal reasoning. Unlike prior methods, VisTA learns adaptive tool-selection strategies without explicit supervision. Our experiments showed significant accuracy gains over strong baselines, highlighting VisTA's potential for robust, flexible visual reasoning.

Limitations. VisTA currently abstracts each tool as a black-box module chosen through high-level selection, without modeling the full parameter structure of real-world tool interfaces. While this abstraction is sufficient for many visual tools, it does not yet capture scenarios that require explicit argument construction, such as specifying a bounding box when invoking a zoom-in tool. Extending VisTA to support parameterized tool invocation and richer argument generation is an exciting direction for future work, and we believe our RL-based framework provides a natural foundation for learning such more expressive tool-use behaviors.

Acknowledgments This work was supported in part by NSF IIS2404180, NetApp Inc., and Institute of Information & communications Technology Planning& Evaluation (IITP) grants funded by the Korea government (MSIT) (No. 2022-0-00871, Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration, (No. RS-2022-00187238, Development of Large Korean Language Model Technology for Efficient Pre-training), and (No. RS-2025-2543949. Environment-Aware and Domain-Adaptive Multimodal Embodied AI for Real-World Interaction).

References

[1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024. 5, 6

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1, 6

[3] Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Devendra Chaplot, Jessica Chudnovsky, Saurabh Garg, Theophile Gervet, Soham Ghosh, Amélie Héliou, Paul Jacob, et al. Pixtral 12b. *arXiv preprint arXiv:2410.07073*, 2024. 5, 6

[4] Meta AI. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 6

[5] Anthropic. The claude 3 model family: Opus, sonnet, haiku, 2024. 5, 6

[6] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 5, 1

[7] Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981. 5, 1

[8] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. PaliGemma: A versatile 3B VLM for transfer. *arXiv preprint arXiv:2407.07726*, 2024. 5, 6

[9] Gary Bradski. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000. 5, 1

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1

[11] Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*, 2022. 1, 2

[12] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. 2

[13] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*, 2024. 5, 6

[14] Konstantinos G Derpanis. The harris corner detector. *York University*, 2(1):2, 2004. 5, 1

[15] Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*, 2025. 3

[16] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pages 148–166. Springer, 2024. 3

[17] Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, et al. G-llava: Solving geometric problem with multi-modal large language model. *arXiv preprint arXiv:2312.11370*, 2023. 5, 6, 1

[18] Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR, 2023. 1, 2

[19] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 5, 1

[20] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. Tora: A tool-integrated reasoning agent for mathematical problem solving. *arXiv preprint arXiv:2309.17452*, 2023. 1, 2

[21] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 2

[22] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14953–14962, 2023. 1, 2

[23] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 5, 1

[24] John Edward Hershberger and Jack Snoeyink. Speeding up the douglas-peucker line-simplification algorithm. 1992. 5, 1

[25] Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *Advances in Neural Information Processing Systems*, 37:139348–139379, 2024. 1, 2

[26] Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*, 2025. 2

[27] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 1, 5, 6

[28] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024. 3, 6

[29] Mojtaba Komeili, Kurt Shuster, and Jason Weston. Internet-augmented dialogue generation. *arXiv preprint arXiv:2107.07566*, 2021. 2

[30] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018. 5, 1

[31] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 6

[32] Fangyu Liu, Julian Martin Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. Deplot: One-shot visual language reasoning by plot-to-table translation. *arXiv preprint arXiv:2212.10505*, 2022. 5, 1

[33] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 1

[34] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024. 6

[35] Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, et al. Llava-plus: Learning to use tools for creating multimodal agents. In *European Conference on Computer Vision*, pages 126–142. Springer, 2024. 1, 2

[36] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024. 5, 1

[37] Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785*, 2025. 2

[38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[39] Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv preprint arXiv:2105.04165*, 2021. 2, 3, 5

[40] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023. 3

[41] Pan Lu, Bowen Chen, Sheng Liu, Rahul Thapa, Joseph Boen, and James Zou. Octotools: An agentic framework with extensible tools for complex reasoning. *arXiv preprint arXiv:2502.11271*, 2025. 1, 2

[42] Junyu Luo, Zekun Li, Jinpeng Wang, and Chin-Yew Lin. Chartocr: Data extraction from charts images via a deep hybrid framework. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1917–1925, 2021. 5, 1

[43] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022. 2, 3, 5

[44] Ahmed Masry, Parsa Kavehzadeh, Xuan Long Do, Enamul Hoque, and Shafiq Joty. Unichart: A universal vision-language pretrained model for chart comprehension and reasoning. *arXiv preprint arXiv:2305.14761*, 2023. 5, 1

[45] Fanqing Meng, Wenqi Shao, Quanfeng Lu, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. Chartassisstant: A universal chart multimodal language model via chart-to-table pre-training and multitask instruction tuning. *arXiv preprint arXiv:2401.02384*, 2024. 5, 1

[46] OpenAI. Learning to reason with llms. https://openai.com/index/learning-to-reason-with-llms/, 2024. Accessed: 2025-05-13. 2

[47] Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. Gorilla: Large language model connected with massive apis. *Advances in Neural Information Processing Systems*, 37:126544–126565, 2024. 1

[48] Shuai Peng, Di Fu, Liangcai Gao, Xiuqin Zhong, Hongguang Fu, and Zhi Tang. Multimath: Bridging visual and mathematical reasoning for large language models. *arXiv preprint arXiv:2409.00147*, 2024. 5, 1

[49] Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza, Taesiri, and Anh Totti Nguyen. Vision language models are blind. In *ACCV*, 2024. 2, 3, 5, 8

[50] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020. 5, 1

[51] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023. 1, 2

[52] Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. Solving geometry problems: Combining text and diagram interpretation. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1466–1476, 2015. 5, 1

[53] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 4

[54] Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See-Kiong Ng, Lidong Bing, and Roy Ka-Wei Lee. Math-llava: Bootstrapping mathematical reasoning for multimodal large language models. *arXiv preprint arXiv:2406.17294*, 2024. 6

[55] Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11888–11898, 2023. 1, 2

[56] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998. 2

[57] Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. 5, 6

[58] OpenGVLab Team. Internvl2: Better than the best—expanding performance boundaries of open-source multimodal models with the progressive scaling strategy, 2024. 6

[59] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *NeurIPS*, 2024. 5, 6

[60] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1

[61] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 2

[62] Renqiu Xia, Bo Zhang, Hancheng Ye, Xiangchao Yan, Qi Liu, Hongbin Zhou, Zijun Chen, Peng Ye, Min Dou, Botian Shi, et al. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. *arXiv preprint arXiv:2402.12185*, 2024. 5, 1

[63] Haotian Xu, Xing Wu, Weinong Wang, Zhongzhi Li, Da Zheng, Boyuan Chen, Yi Hu, Shijia Kang, Jiaming Ji, Yingying Zhang, et al. Redstar: Does scaling long-cot data unlock better slow-reasoning systems? *arXiv preprint arXiv:2501.11284*, 2025. 6

[64] Zhengzhuo Xu, Bowen Qu, Yiyan Qi, Sinan Du, Chengjin Xu, Chun Yuan, and Jian Guo. Chartmoe: Mixture of expert connector for advanced chart understanding. *arXiv preprint arXiv:2409.03277*, 2024. 5, 1

[65] Le Xue, Manli Shu, Anas Awadalla, Jun Wang, An Yan, Senthil Purushwalkam, Honglu Zhou, Viraj Prabhu, Yutong Dai, Michael S Ryoo, et al. xGen-MM (BLIP-3): A family of open large multimodal models. *arXiv preprint arXiv:2408.08872*, 2024. 6

[66] Pengyu Yan, Saleem Ahmed, and David Doermann. Context-aware chart element detection. In *International conference on document analysis and recognition*, pages 218–233. Springer, 2023. 5, 1

[67] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. 1

[68] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10371–10381, 2024. 5, 1

[69] HK Yuen, John Princen, John Illingworth, and Josef Kittler. Comparative study of hough transform methods for circle finding. *Image and vision computing*, 8(1):71–77, 1990. 5, 1

[70] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer, 2024. 2, 3, 5

[71] Zeren Zhang, Jo-Ku Cheng, Jingyang Deng, Lu Tian, Jinwen Ma, Ziran Qin, Xiaokai Zhang, Na Zhu, and Tuo Leng. Diagram formalization enhanced multi-modal geometry problem solver. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025. 5, 1

[72] Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025. 6