WITH TOOLS OR BECAUSE OF TOOLS? TRACING CAPABILITY GAINS IN MULTIMODAL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal agents increasingly rely on tools for image inspection and code execution, and their benchmark gains are often attributed to these interactions. Yet aggregate accuracy does not reveal whether tools expand the set of solvable problems or merely provide alternative paths to answers already within reach. We study DeepEyesV2, Thyme, and Qwen3-VL-8B-Instruct across 15 benchmarks, comparing each agent with its tool-free counterpart. We additionally train a Pure-Text Reasoner from DeepEyesV2 data and edit tool-use trajectories to probe the roles of generated calls and returned results. Tool access yields no consistent performance advantage, while the Pure-Text Reasoner matches or exceeds the two specialized agents on several benchmarks. Across DeepEyesV2 and Thyme, most successes under tool-enabled inference are reproduced by at least one non-tool reference. Some remaining successes persist after returned results are removed, whereas tool outputs judged novel, correct, and properly integrated often occur on problems already solvable without tools. The full tool-use loop yields clear gains in specific settings, but much of its apparent advantage remains attainable through non-tool reasoning, revealing a gap between tool-enabled success and tool-contributed capability.

1 INTRODUCTION

A multimodal agent is asked what material a glove in a street-food image is made of. It writes a Python snippet, calls the code interpreter, and answers. Yet the snippet does little: it simply reloads and displays the original image, with no cropping, zooming, or visual processing. Our Pure-Text Reasoner (Section 3.3), given no tools, answers correctly (Figure 1). The trajectory contains a tool call, but the call supplies no new visual evidence.

Multimodal agents increasingly use tools to inspect image regions, manipulate visual inputs, and carry out calculations during reasoning (Zheng et al., 2026; Su et al., 2025; Zhang et al., 2026b; Hong et al., 2026). Their benchmark gains are often attributed to these interactions. Yet a correct answer after a tool call leaves an important question open: did the tool make the problem solvable, or did the agent take a different path to an answer already within reach? Aggregate accuracy cannot tell these cases apart. Two settings can achieve similar scores while sharing nearly all their successes or solving quite different problems.

We trace these gains at the level of individual questions. We study DeepEyesV2, Thyme, and Qwen3-VL-8B-Instruct (Qwen below) across 15 benchmarks spanning real-world understanding, OCR, charts, and mathematical reasoning. For each agent, we compare the full loop with the same checkpoint run without tools. We also train a tool-free multimodal reasoner on the DeepEyesV2 source pool, providing a second route learned without tool-calling supervision. We then edit call messages and returned results and use process annotations to examine what the interactions contribute.

Most full-loop successes have an observed non-tool solution. The two references cover most of DeepEyesV2's and Thyme's Full-run successes, while Qwen also shares most of its successes with its own Tool-Free run. Tools also lose answers available without them: on called questions, DeepEyesV2 gains and loses nearly equal numbers of successes relative to Tool-Free. Some successes missed by the references remain correct under No-Result. Outputs judged informative also frequently occur on questions the same agent solves without tools. An output can contribute information to a trajectory even when the agent can solve the question without it.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

The full loop nevertheless produces clear gains in specific settings. Qwen gains 7.1 points on HRBench-4K, and its score falls below Tool-Free when the returned crop is withheld. We examine these positive cases alongside the losses, rather than treating all tool calls as equivalent.

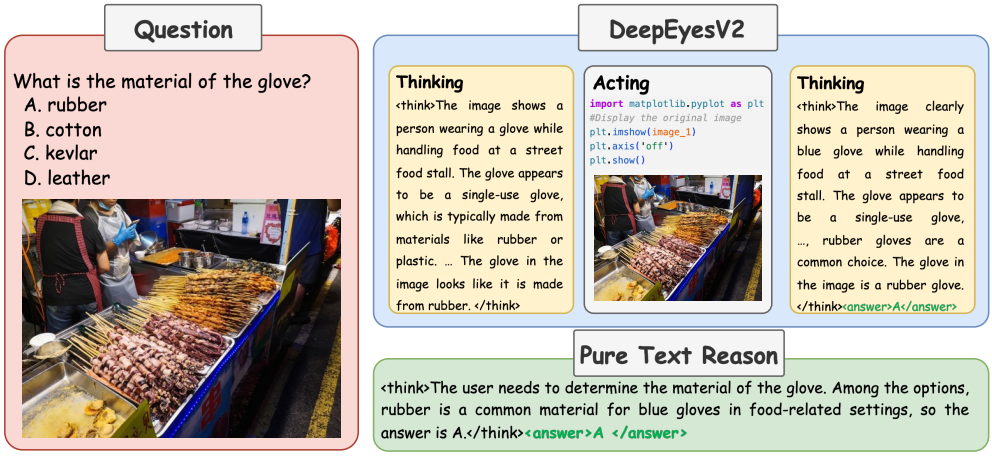


Figure 1: A motivating example illustrating that tool interaction can become procedural rather than genuinely helpful: the agent performs extra tool-mediated steps, yet this added interaction does not necessarily improve the final reasoning outcome.

Our contributions are threefold. First, we develop an evaluation framework that traces tool gains to individual solved questions, combining each agent's own Tool-Free run with a shared Pure-Text Reasoner, trajectory edits, and process annotations. Second, across three agents, we show that most Full-run successes have a non-tool solution. DeepEyesV2's gains and losses nearly cancel, and Qwen loses more successes than it adds. Third, we distinguish gains that disappear when returned results are withheld from those that persist, and show why even an informative tool output need not expand the solved set.

2 Related Work

Tool-augmented multimodal reasoning. A growing line of work moves multimodal models from passive image understanding toward active visual reasoning, often described as "thinking with images" (OpenAI, 2025). Training-free systems orchestrate vision modules through executable programs (Gupta & Kembhavi, 2023; Surís et al., 2023) or produce visual artifacts as intermediate reasoning steps (Hu et al., 2024). Recent systems are trained, mostly with reinforcement learning (RL), to crop and zoom (Zheng et al., 2026; Su et al., 2025; Zhang et al., 2025), to search over many turns (Lai et al., 2026), or to edit images while reasoning (Wu et al., 2026). DeepEyesV2 (Hong et al., 2026) and Thyme (Zhang et al., 2026b) use code execution for image manipulation and numerical computation.

Evaluating tool gains. Several recent studies examine what these interactions contribute. MED (Ma et al., 2026) measures intrinsic changes under a tool-free protocol across crop-and-zoom RL checkpoints, separates them from tool-induced gains and harms, and validates its reference with a no-tool RL control. Wang et al. (2026b) audit visual tool use at the policy, trajectory, and step levels, including the replacement of individual observations under a fixed prefix. They find that aggregate gains are concentrated in a minority of well-calibrated rollouts. TextCall (Shao et al., 2026) retains tool-call text while withholding returned images, and tests this design with low-rank adaptation (LoRA), full fine-tuning, and RL. See2Think (Yan et al., 2026) compares rendered, unrendered, and corrupted visual states, relating their effects on answers to process judgments. Tool-Use Tax (Zhang et al., 2026a) separates protocol costs from execution gains in language agents under semantic distractors. Its Capability Overlap Principle states that many apparent tool gains fall on questions already solvable by native chain-of-thought. We study this overlap in multimodal agents by matching each agent's successes to its own tool-free answers and to the Pure-Text Reasoner built from the DeepEyesV2

source pool. The study covers code-execution and crop-and-zoom agents across perception, OCR, charts, and mathematics.

Faithfulness of intermediate reasoning. Plausible reasoning traces need not explain the model's answer (Turpin et al., 2023). Lanham et al. (2023) test this by perturbing chain-of-thought text and measuring changes in predictions. MME-CoT (Jiang et al., 2025) evaluates multimodal reasoning beyond final correctness, while MAPO (Yang et al., 2026) addresses the gap between textual reasoning and visual actions. Our process analysis checks whether outputs judged informative occur on questions the same agent also solves without tools.

Agentic benchmarks. TIR-Bench (Li et al., 2025) covers 13 image-processing and manipulation tasks, while O3-Bench (Li et al., 2026) requires reasoning over fine details in dense charts and maps. AgentVista (Su et al., 2026) tests extended tool use in challenging visual scenarios. We include TIR-Bench and O3-Bench as additional tests of the two specialized agents, alongside the main benchmark suite.

3 Study Design

3.1 Agents, Benchmarks, and Evaluation

DeepEyesV2 (Hong et al., 2026) and Thyme (Zhang et al., 2026b) use code-execution interfaces for image manipulation and computation. We evaluate their released checkpoints with the project inference wrappers. Qwen3-VL-8B-Instruct (Bai et al., 2025a) uses a crop-and-zoom interface: each call specifies a region, and the wrapper returns its crop. It does not expose arbitrary Python execution. Appendix B describes the interfaces, prompts, and evaluation settings.

The main suite spans real-world understanding, optical character recognition (OCR), charts, and mathematical reasoning. DeepEyesV2, Thyme, and Qwen3-VL-8B-Instruct are evaluated under all five settings on all 15 benchmarks, with 28,150 records per model in each setting. We report native benchmark scores and use record-level correctness for success-set comparisons; the two measures differ for composite benchmarks. Scoring uses VLMEvalKit (Duan et al., 2024), with GPT-4o as the judge where a benchmark requires model-based evaluation. Additional TIR-Bench and O3-Bench results cover the two specialized agents.

3.2 Tool-Free Runs and Trajectory Edits

Full and Tool-Free. Full uses the agent's tool-enabled inference loop. Tool-Free uses the same checkpoint with calls suppressed by the prompt and execution disabled. Both receive the original question and image. The Pure-Text Reasoner below provides a second non-tool reference for all three agents.

Changes to the interaction. No-Result retains the model-generated code or call parameters as part of the reasoning, while withholding the returned result. No-Call-Message (No-Call below) retains the execution output and removes the call text. Empty-Call replaces the specialized agents' task code with `pass`; for Qwen it masks the crop parameters and withholds the crop. For DeepEyesV2 and Thyme, these edits operate on already-generated tool trajectories before the model continues to its final answer.

Appendix E gives the ablation scores and correctness agreement with their corresponding Full runs.

3.3 Training the Pure-Text Reasoner

We train the Pure-Text Reasoner from the DeepEyesV2 source pool, starting from Qwen2.5-VL-7B-Instruct (Bai et al., 2025b). It receives images as well as text and reasons in text without tool calls. From the 65K supervised fine-tuning (SFT) pool, we retain 53K visual reasoning examples and ask Qwen2.5-VL-72B to generate replacement rationales from each question and image. We then train with group relative policy optimization (GRPO) (Shao et al., 2024), using final-answer correctness as the reward. Appendix A gives the construction and configuration. This reasoner tests another learning route from the source data; it is not a matched ablation of the agents' full training procedures.

3.4 Outcome and Process Measures

Success coverage and outcome changes. Let F, T, and P denote records solved by Full, the same agent's Tool-Free run, and the Pure-Text Reasoner. Own-reference coverage is $|F \cap T|/|F|$; union coverage is $|F \cap (T \cup P)|/|F|$. All three agents use the same Pure-Text Reasoner, matched by question. On records with a Full call, we count both corrections and introduced errors; Appendix D defines the four outcome rates and their conditional counterparts.

Ablation measures. Correctness agreement is the fraction of questions on which an edited setting and Full have the same binary correctness, including shared successes and shared failures. We also report the aggregate coverage of Full successes after adding No-Result to the non-tool references. Retention measures the fraction of called Full successes also solved under an edit.

Statistics. Paired comparisons use exact McNemar tests for benchmarks other than DynaMath and WeMath. For their variants and components, we exchange the two settings within whole source-question groups and recompute the native score difference. The comparison family comprises 45 model–benchmark pairs, with Holm correction for multiple testing. Scores and success sets use benchmark correctness labels. Coverage intervals use 10,000 paired bootstrap resamples stratified by benchmark, retaining these question groups. Wilson intervals accompany count-based descriptive proportions. Appendix D gives the statistical procedures.

Process labels. Qwen3-VL-30B-A3B-Instruct labels the novelty, correctness, and integration of tool outputs in DeepEyesV2 and Thyme trajectories. Outputs that are novel, correct, and properly integrated form the informative subset. We measure Tool-Free correctness in this subset and compare it with the rate on all called questions. Two independent annotators audited a random 30% subset of the judged tool-using trajectories, with 94% agreement. Appendix F gives the annotation prompts, label mapping, sample sizes, intervals, and validation procedure.

Table 1: Perception results. Full and Tool-Free share a checkpoint within each model. Bold and underline mark the best and second-best scores within the Qwen2.5-VL-7B group above the separator; displayed ties share a mark. Qwen3 is shown separately. OCRBench uses the 0 to 1000 scale.

Model	Setting	Real-World Understanding				OCR		Chart		
		V* Bench	HRBench 4K	HRBench 8K	Tree Bench	OCR Bench	SEED 2-Plus	CharXiv Descr.	CharXiv Reason.	Chart QA
DeepEyesV2	Full	81.7	75.8	70.5	<u>41.5</u>	814	69.8	75.7	43.7	**88.1**
	Tool-Free	80.1	74.8	71.1	33.1	866	**71.2**	<u>77.0</u>	**46.2**	83.1
Thyme	Full	82.7	<u>77.3</u>	**72.5**	38.8	865	70.2	73.3	<u>43.8</u>	<u>87.3</u>
	Tool-Free	**83.8**	**78.5**	<u>72.3</u>	37.5	<u>875</u>	69.7	73.3	43.0	87.2
Pure-Text Reasoner		<u>83.2</u>	<u>77.3</u>	**72.5**	**42.2**	**888**	<u>70.7</u>	**77.5**	**46.2**	83.4
Qwen3-VL-8B	Full	86.4	84.0	74.6	44.9	855	70.7	87.0	40.1	52.8
	Tool-Free	83.2	76.9	75.8	45.4	838	71.6	89.4	50.4	67.0

4 Results

4.1 Performance with and without Tools

Tool access moves scores in both directions. Tables 1 and 2 compare each agent with its own Tool-Free run. The 45 model–benchmark comparisons show no consistent advantage for Full over Tool-Free across the suite. Across the 45 paired comparisons, Full is significantly better in 3 and significantly worse in 8 after Holm correction. Thyme's Full and Tool-Free scores differ by at most 1.3 points on a 0–100 scale. Appendix D gives every paired test, and Figure 2 shows the score differences.

The Pure-Text Reasoner exceeds both specialized agents on a majority of the benchmarks. On WeMath, it scores 46.3, compared with 37.1 for DeepEyesV2 and 37.5 for Thyme. It also scores higher than both on OCRBench, TreeBench, both CharXiv splits, and MathVista. It achieves these results without tool-calling supervision.

On TIR-Bench and O3-Bench, which emphasize image manipulation and visual search, the two specialized agents' full-loop scores remain within 0.8 points of their own Tool-Free scores (Table 3). No-Result is slightly above Full and Empty-Call slightly below in all four comparisons, with each gap from Full at most 1.5 points.

4.2 Which Successes Are New?

Similar accuracy can hide different sets of solved questions. We therefore check each full-loop success against the agent's Tool-Free answer and the Pure-Text Reasoner. Table 4 reports the counts and intervals; Figure 3 separates the specialized agents' overlap by task family.

The agent's own Tool-Free run reproduces most Full successes for each model. Adding the Pure-Text Reasoner extends this coverage by finding solutions missed by the same-checkpoint reference. The two references therefore test complementary routes to an answer: suppressing tools at inference and

Table 2: Mathematical reasoning results. Bold and underline mark the best and second-best scores within the Qwen2.5-VL-7B group above the separator; displayed ties share a mark. Qwen3 is shown separately. WeMath uses its strict composite score; DynaMath reports the Average score.

Model	Setting	Math Reasoning					
		MathVista	MathVerse	MathVision	WeMath	DynaMath	LogicVista
DeepEyesV2	Full	72.5	**51.8**	26.4	37.1	<u>57.6</u>	48.1
	Tool-Free	<u>73.8</u>	48.6	<u>28.5</u>	<u>43.0</u>	**57.9**	**49.2**
Thyme	Full	72.5	47.3	24.9	37.5	54.1	<u>48.3</u>
	Tool-Free	71.6	47.9	24.9	37.5	54.3	47.4
Pure-Text Reasoner		**74.6**	<u>51.6</u>	**29.3**	**46.3**	56.8	**49.2**
Qwen3-VL-8B	Full	68.4	54.0	54.1	44.6	62.5	49.9
	Tool-Free	76.9	58.8	55.7	52.6	66.7	51.7

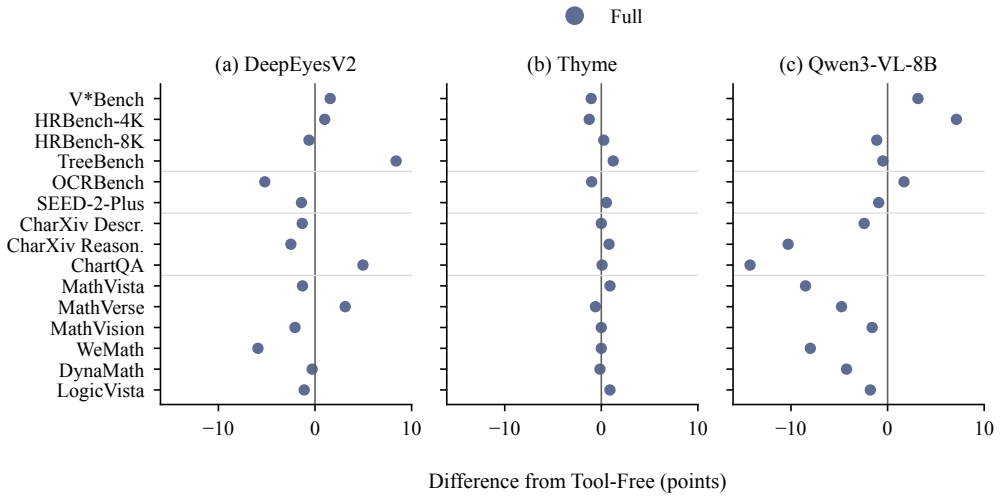


Figure 2: Full scores relative to the same model's Tool-Free score across 15 benchmarks. OCRBench is rescaled to 0 to 100.

learning to reason without tool-calling supervision. Figure 3 shows how their overlap varies across task families.

Coverage remains high on called successes (call rates: Appendix B). Restricting the comparison to these questions separates success after an actual interaction from success in a run that merely allows tools. Table 4 reports the called-subset outcomes alongside coverage of all Full successes.

Some successes missed by a reference contain no tool call at all. These differences cannot be credited to a tool return; changing the prompt or the generated reasoning can also change which questions are answered correctly. This distinction matters when interpreting the additional coverage from an independently generated No-Result run.

Tools also lose successes. On called questions, DeepEyesV2 solves 2,605 questions that Tool-Free misses, while Tool-Free solves 2,638 that Full misses. Table 4 reports both directions of change and their conditional rates for all three agents. Correction and induced error condition on reference failures and successes, respectively. Their percentages can differ even when the numbers of gained and lost successes are nearly equal.

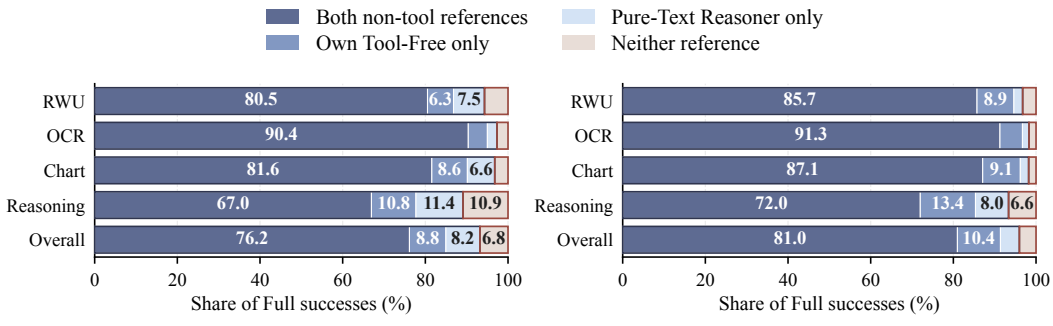
4.3 The Role of Calls and Returned Results

Adding No-Result extends coverage beyond the two non-tool references (Figure 4). For Thyme, much of the additional coverage comes from questions with no Full call. On those questions, the comparison captures another successful run rather than the removal of an observed tool result. The called subset directly tests whether success persists when generated calls are retained but their results are withheld.

Agreement with Full differs across the edited settings. For DeepEyesV2, No-Result has the highest agreement with Full in every task family, exceeding Tool-Free agreement by 3.2 to 7.5 points. Replacing the code with `pass` as well (Empty-Call) lowers agreement by only 0.9 to 1.7 points, so most of this advantage remains. For Thyme, No-Call has the highest agreement throughout (Table 5).

Table 3: Reported scores on TIR-Bench (1,215 questions) and O3-Bench (345 questions).

	Thyme		DeepEyesV2	
Setting	**TIR-Bench**	**O3-Bench**	**TIR-Bench**	**O3-Bench**
Full	16.6	26.7	18.4	27.2
Tool-Free	16.4	27.1	18.9	28.0
No-Result	17.2	27.1	19.3	28.4
No-Call	16.3	25.8	17.6	26.1
Empty-Call	15.8	25.4	17.1	25.7



(a) DeepEyesV2 (b) Thyme

Figure 3: Which full-loop successes are also solved without tools? Each bar partitions successes reproduced by both non-tool references, the agent's own Tool-Free run alone, the Pure-Text Reasoner alone, or neither. RWU denotes real-world understanding. Overall percentages use pooled counts.

Qwen's trajectory edits provide a further comparison. Table 17 compares the retention of called Full successes under No-Result, No-Call, and Tool-Free inference. No-Result can reproduce answers missed by the independent Tool-Free run even when its overall retention is lower. Thus, the performance of an interrupted tool loop and the existence of an alternative successful route answer different questions.

4.4 Informative Tool Use and Additional Successes

Process labels describe how tool outputs enter the reasoning trace. Figure 5 shows task-family distributions, sample sizes, intervals, and a Tool-Free baseline. For DeepEyesV2, failed or non-contributory calls dominate real-world understanding, while confirmatory calls are common on OCR and charts. Thyme calls much less often, and a larger share of its real-world and OCR calls are labeled informative. Its chart analysis has only 16 calls and is marked accordingly.

An informative output often appears on a question the same agent already solves without tools. For DeepEyesV2 the fraction is 59.7%, compared with 61.7% on all called questions. For Thyme it is 79.4%, above the 75.2% baseline. The two models therefore do not show a common pattern of informative calls concentrating on Tool-Free failures. The disjoint comparison in Appendix F finds a modest enrichment of such failures for DeepEyesV2 and the opposite direction for Thyme. In both, most informative traces concern a question with an observed non-tool solution. Thyme's crop of a road sign reading "ONE WAY" provides a concrete example: the crop is judged informative, yet Tool-Free also answers correctly (Appendix G).

Table 4: Success coverage and called-subset outcomes (%). Panel A conditions on Full being correct; panel B conditions on a recorded call.

A. Coverage of Full successes

Model	Full correct	Tool-Free	Either reference	Uncovered
DeepEyesV2	17,335	85.0 [84.3, 85.6]	93.2 [92.8, 93.6]	6.8
Thyme	16,934	91.4 [90.9, 91.8]	96.0 [95.7, 96.3]	4.0
Qwen3-VL-8B	18,278	90.5 [90.0, 91.0]	95.4 [95.0, 95.7]	4.6

B. Outcomes on called records

Model	Called	TRR	TFR	TER	TIR	Correction	Induced error
DeepEyesV2	28,144	52.3	29.0	9.3	9.4	24.2 [23.4, 25.0]	15.2 [14.7, 15.7]
Thyme	804	71.3	20.3	4.5	4.0	18.1 [13.4, 24.0]	5.3 [3.8, 7.4]
Qwen3-VL-8B	15,322	59.6	22.2	6.4	11.8	22.3 [21.1, 23.6]	16.6 [15.9, 17.3]

Either reference combines own Tool-Free and the Pure-Text Reasoner. Panel B compares each agent with its own Tool-Free run: TRR/TFR, both correct/incorrect; TER/TIR, corrected/lost by Full (Appendix D). Brackets give 95% intervals (grouped bootstrap in A, Wilson in B).

Table 5: Reported correctness agreement with Full across all questions in each task family (%). Agreement counts shared successes and shared failures.

Model	Family	Tool-Free	No-Result	Empty-Call	No-Call
DeepEyesV2	RWU	82.9	90.0	88.7	82.8
	OCR	90.7	93.9	93.0	92.8
	Chart	83.3	90.8	89.1	85.8
	Reasoning	76.8	82.9	81.3	77.7
Thyme	RWU	92.4	99.2	95.1	99.4
	OCR	95.9	96.5	96.1	99.0
	Chart	93.0	95.9	95.7	96.3
	Reasoning	84.8	88.8	88.0	89.6

Empty-Call replaces task-relevant code with a `pass` statement. RWU denotes real-world understanding.

Restricting the comparison to Full-correct trajectories keeps final correctness fixed across groups. Tool-Free solves 80.0% of DeepEyesV2's informative successes and 94.7% of Thyme's; among the other called successes, the rates are 86.9% and 80.0%, respectively (Appendix F).

4.5 Where the Full Loop Helps

The clearest positive case is Qwen on HRBench-4K. Its full loop scores 84.0, compared with 76.9 for Tool-Free. The 7.1-point gain reverses to a 1.0-point loss under No-Result and a 3.0-point loss under Empty-Call. No-Call retains a 3.5-point gain. On this high-resolution benchmark, keeping the returned crop supports performance more than keeping the crop request alone (Table 6).

DeepEyesV2 scores 41.5 on TreeBench with Full, 39.0 with No-Result, and 34.1 with No-Call. MathVerse is more sensitive to both edits: the reported scores are 51.8 with Full, 48.0 with No-Result, and 40.2 with No-Call.

Qwen's HRBench-4K gain disappears without the returned crop, whereas much of DeepEyesV2's TreeBench gain persists with the call and code alone. Tool use is most valuable when the return supplies information or computation the agent otherwise lacks. A successful call, by itself, does not establish that contribution.

5 Implications

Report a paired Tool-Free evaluation. An agent's score with tools enabled should be accompanied by the same checkpoint's score without execution. This comparison exposes gains and losses that comparisons between separately trained models cannot isolate. Report the two directions of correctness change alongside the net score, with paired tests and the units used for resampling. For an agent with a low call rate, also report coverage on called successes: otherwise, questions that never enter the tool loop can dominate the comparison. Include the call rate beside these results.

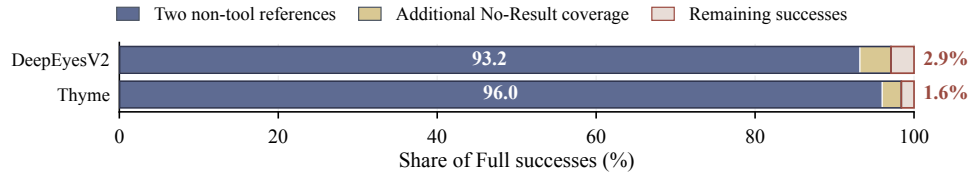
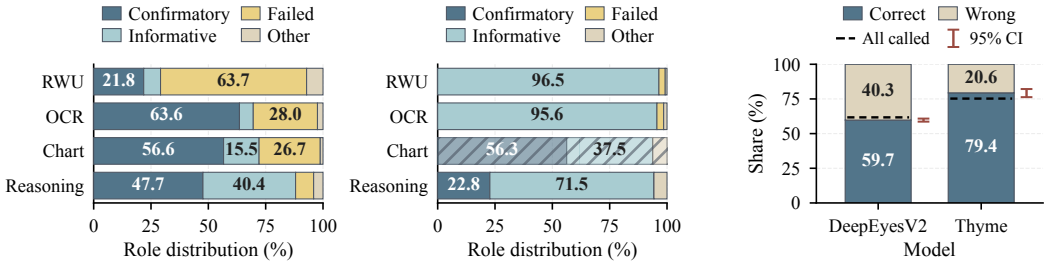


Figure 4: Reported coverage of Full successes by the two non-tool references and by No-Result. Thyme's added coverage mostly comes from questions with no Full call. Appendix D gives the counts and sources.



(a) DeepEyesV2 (b) Thyme (c) Informative subset

Figure 5: Process labels on called trajectories. (a,b) Role distributions. Counts for RWU, OCR, chart, and reasoning are 2,195, 3,275, 7,499, and 15,175 for DeepEyesV2 and 255, 410, 16, and 123 for Thyme; hatching marks $n < 30$. (c) Tool-Free correctness among informative trajectories ($n = 7,649$ and 732), with 95% Wilson intervals. Dashed lines show Tool-Free correctness on all called questions.

Check the return before crediting the call. Compare Full with No-Result and No-Call-Message to distinguish the effects of the generated call and its returned content. Correctness agreement shows which part of the loop preserves behavior; success coverage shows whether an alternative reaches the same answer. Comparing No-Result with Empty-Call tests whether code content matters beyond the call format. Neither provides a tool result; Empty-Call also replaces the specialized agents' code with a single `pass` statement.

Table 6: Scores under trajectory edits on selected benchmarks.

Model	Benchmark	Full	No-Result	No-Call
DeepEyesV2	TreeBench	41.5	39.0	34.1
DeepEyesV2	MathVerse	51.8	48.0	40.2
Qwen3-VL-8B	HRBench-4K	84.0	75.9	80.4

Scores are rounded to one decimal place. No-Call denotes No-Call-Message.

Use more selective training signals. Final-answer rewards can credit a redundant image redisplay along with a useful crop. The present results motivate rewards that account for what the returned information changes. Toolformer selects calls using improvement in prediction of subsequent text (Schick et al., 2023); MED studies intrinsic gains and tool-induced harms during visual-tool training (Ma et al., 2026). Testing a call-specific counterfactual reward would additionally require interventions at a fixed prefix. A training comparison could hold the source questions, initialization, and compute fixed, then vary whether calls receive credit for final correctness alone or for the improvement after a returned observation. Tracking both newly solved questions and previously solvable questions lost under tool use would show whether such a reward improves the balance between gains and harms. When pooled within each task family, recorded end-to-end runtimes are 1.4 to 3.1 times those of the Pure-Text Reasoner (Table 7), so unnecessary calls also cost time.

Table 7: End-to-end runtime relative to the Pure-Text Reasoner, by task family.

Model	Real-world understanding	OCR	Chart	Reasoning
DeepEyesV2	1.9×	3.1×	2.1×	3.0×
Thyme	1.4×	2.1×	1.7×	1.4×

Each ratio divides the agent's total wall-clock time by the Pure-Text Reasoner's total time over the same task family. Timings include model generation and tool interaction, using four H20 GPUs, fixed batching and concurrency, a local executor, and at most one retry after an execution failure.

Scope. The results concern image-based perception and reasoning. Extensions to web navigation, embodied interaction, and retrieval of facts absent from the input would test how the balance changes when tools provide access to information beyond the original question and image.

6 Conclusion

A successful tool-use trace shows that an agent succeeded with tools; it does not establish that the agent succeeded because of them. Across three multimodal agents, most Full successes are also reached without tools. DeepEyesV2 loses about as many Tool-Free successes as it adds, and Qwen loses more. The reported ablations further separate the roles of calls and returns: DeepEyesV2 retains most of its TreeBench gain without returned results (39.0 versus 41.5 with Full and 33.1 without tools), while Qwen's HRBench-4K gain disappears when the crop is withheld. Even outputs judged informative often occur on questions already solvable without tools. These findings establish a gap between learning to produce tool interactions and learning when those interactions add capability. Evaluation should measure newly solved questions and lost successes alongside aggregate accuracy, and training should reward the information a tool contributes.

AI USE STATEMENT

We used AI assistants to help organize and edit the manuscript and prepare figures and supplementary analyses from existing experimental records. The authors reviewed and revised the AI-assisted material and take full responsibility for the final claims, analyses, and conclusions.

REPRODUCIBILITY STATEMENT

Appendix A describes the reasoner's training data and configuration. Appendix B gives the model interfaces, evaluation settings, and scoring procedures. Appendices D, E, and F provide the success-set counts, statistical procedures, trajectory-editing protocols, annotation prompts, and human-validation procedure. The paired analyses use an outcome ledger and analysis scripts; separate reported summaries provide the specialized-agent ablation scores and human-validation results.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report, 2025a. URL https://arxiv.org/abs/2511.21631.

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, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report, 2025b. URL https://arxiv.org/abs/2502.13923.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, Dahua Lin, and Kai Chen. VLMEvalKit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 11198–11201. Association for Computing Machinery, 2024. doi: 10.1145/3664647.3685520. URL https://doi.org/10.1145/3664647.3685520.

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*, pp. 14953–14962, 2023.

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, Guohai Xu, and Xing Yu. DeepEyesV2: Toward agentic multimodal model. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=yDKawwfJ5O.

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.

Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanwei Li, Yu Qi, Xinyan Chen, Liuhui Wang, Jianhan Jin, Claire Guo, Shen Yan, Bo Zhang, Chaoyou Fu, Peng Gao, and Hongsheng Li. MME-CoT: Benchmarking chain-of-thought in large multimodal models for reasoning quality, robustness, and efficiency. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL https://proceedings.mlr.press/v267/jiang25n.html.

Xin Lai, Junyi Li, Wei Li, Tao Liu, Tianjian Li, and Hengshuang Zhao. Mini-o3: Scaling up reasoning patterns and interaction turns for visual search. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Zp2y9O3wEj.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning, 2023. URL https://arxiv.org/abs/2307.13702.

Bohao Li, Yuying Ge, Yi Chen, Yixiao Ge, Ruimao Zhang, and Ying Shan. SEED-Bench-2-Plus: Benchmarking multimodal large language models with text-rich visual comprehension, 2024. URL https://arxiv.org/abs/2404.16790.

Kaican Li, Lewei Yao, Jiannan Wu, Tiezheng YU, Jierun Chen, Haoli Bai, Lu Hou, Lanqing HONG, Wei Zhang, and Nevin L. Zhang. InSight-o3: Empowering multimodal foundation models with generalized visual search. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=vlraTIgUD3.

Ming Li, Jike Zhong, Shitian Zhao, Haoquan Zhang, Shaoheng Lin, Yuxiang Lai, Chen Wei, Konstantinos Psounis, and Kaipeng Zhang. TIR-Bench: A comprehensive benchmark for agentic thinking-with-images reasoning. *arXiv preprint arXiv:2511.01833*, 2025. URL https://arxiv.org/abs/2511.01833.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. OCRBench: On the hidden mystery of OCR in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024.

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. In *International Conference on Learning Representations*, volume 2024, pp. 23439–23554, 2024.

Yan Ma, Weiyu Zhang, Tianle Li, Du Linge, Xuyang Shen, and Pengfei Liu. What does vision tool-use reinforcement learning really learn? Disentangling tool-induced and intrinsic effects for crop-and-zoom. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=iTWu7ziS5z.

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. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.177. URL https://aclanthology.org/2022.findings-acl.177/.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, Ping Luo, Yu Qiao, Qiaosheng Zhang, and Wenqi Shao. MM-Eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning, 2025. URL https://arxiv.org/abs/2503.07365.

OpenAI. GPT-4o system card. System card, 2024. URL https://openai.com/index/gpt-4o-system-card/.

OpenAI. Thinking with images. https://openai.com/index/thinking-with-images/, 2025. Accessed: 2026-09-24.

Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma GongQue, Shanglin Lei, YiFan Zhang, Zhe Wei, Miaoxuan Zhang, Runfeng Qiao, Xiao Zong, Yida Xu, Peiqing Yang, Zhimin Bao, Muxi Diao, Chen Li, and Honggang Zhang. We-Math: Does your large multimodal model achieve human-like mathematical reasoning? In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20023–20070, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.983. URL https://aclanthology.org/2025.acl-long.983/.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551. Curran Associates, Inc., 2023. doi: 10.52202/075280-2997. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/d842425e4bf79ba039352da0f658a906-Paper-Conference.pdf.

Jiahao Shao, Yuanbo Yang, Yiyi Liao, Yujun Shen, Ceyuan Yang, and Yinghao Xu. Thinking with tools, not with pixels: Tool calls as text scaffolds for visual reasoning, 2026. URL https://arxiv.org/abs/2608.09682.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel Reasoner: Incentivizing pixel space reasoning via curiosity-driven reinforcement learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 8222–8251. Curran Associates, Inc., 2025. doi: 10.52202/085713-0280. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/0c38f54740062529aa4117a04b583f3c-Paper-Conference.pdf.

Zhaochen Su, Jincheng Gao, Hangyu Guo, Zhenhua Liu, Lueyang Zhang, Xinyu Geng, Shijue Huang, Peng Xia, Guanyu Jiang, Cheng Wang, Yue Zhang, Yi R. Fung, and Junxian He. AgentVista: Evaluating multimodal agents in ultra-challenging realistic visual scenarios, 2026. URL https://arxiv.org/abs/2602.23166.

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*, pp. 11888–11898, 2023.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 74952–74965. Curran Associates, Inc., 2023. doi: 10.52202/075280-3275. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/ed3fea9033a80fea1376299fa7863f4a-Paper-Conference.pdf.

Haochen Wang, Xiangtai Li, Zilong Huang, Anran Wang, Jiacong Wang, Tao Zhang, Jiani Zheng, Sule Bai, Zijian Kang, Jiashi Feng, Zhuochen Wang, and Zhaoxiang Zhang. Traceable evidence enhanced visual grounded reasoning: Evaluation and method. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=zwMpimSZEq.

Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. VL-Rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 30865–30891. Curran Associates, Inc., 2025a. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/2c84844a559e4f962752570bff456ae4-Paper-Conference.pdf.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with MATH-Vision dataset. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 95095–95169. Curran Associates, Inc., 2024a. doi: 10.52202/079017-3014. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ad0edc7d5fa1a783f063646968b7315b-Paper-Datasets_and_Benchmarks_Track.pdf.

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception

in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7907–7915, 2025b.

Zhiheng Wang, Bo Peng, Lai Wei, and Chaochao Lu. The illusion of visual tool-use: A causal audit of thinking with images, 2026b. URL https://arxiv.org/abs/2608.06270.

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, Alexis Chevalier, Sanjeev Arora, and Danqi Chen. CharXiv: Charting gaps in realistic chart understanding in multimodal LLMs. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 113569–113697. Curran Associates, Inc., 2024b. doi: 10.52202/079017-3609. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/cdf6f8e9fd9aeaf79b6024caec24f15b-Paper-Datasets_and_Benchmarks_Track.pdf.

Mingyuan Wu, Jingcheng Yang, Jize Jiang, Meitang Li, Kaizhuo Yan, Hanchao Yu, Minjia Zhang, ChengXiang Zhai, and Klara Nahrstedt. VTool-R1: VLMs learn to think with images via reinforcement learning on multimodal tool use. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Idst6X6gmy.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13084–13094, 2024.

Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. LogicVista: Multimodal LLM logical reasoning benchmark in visual contexts, 2024. URL https://arxiv.org/abs/2407.04973.

Siyu Yan, Zhuoran Yan, Haiying Xu, Panhao Zhou, Jingyu Chen, Chenhao Ji, Shuo Cao, Yongheng Zhang, Haoze Liu, Siyu Zhang, Xiwen Gu, Yihao Liu, and Alex Jinpeng Wang. See2Think: Do multimodal models really use intermediate visual states?, 2026. URL https://arxiv.org/abs/2607.26769.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Wenhao Yang, Yu Xia, Jinlong Huang, Shiyin Lu, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, Yuchen Zhou, Xiaobo Xia, Yuanyu Wan, Lijun Zhang, and Tat-Seng Chua. Walk the talk: Bridging the reasoning-action gap for thinking with images via multimodal agentic policy optimization, 2026. URL https://arxiv.org/abs/2604.06777.

Kaituo Zhang, Zhen Xiong, Mingyu Zhong, Zhimeng Jiang, Zhouyuan Yuan, Zhecheng Li, and Ying Lin. Are tools all we need? Unveiling the Tool-Use Tax in LLM agents, 2026a. URL https://arxiv.org/abs/2605.00136.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, Peng Gao, and Hongsheng Li. MathVerse: Does your multi-modal LLM truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pp. 169–186. Springer, 2024.

Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, and Qing Li. Adaptive chain-of-focus reasoning via dynamic visual search and zooming for efficient VLMs, 2025. URL https://arxiv.org/abs/2505.15436.

Yi-Fan Zhang, Xingyu Lu, Shukang Yin, Chaoyou Fu, Wei Chen, Xiao Hu, Bin Wen, Kaiyu Jiang, Changyi Liu, Tianke Zhang, Haonan Fan, Kaibing Chen, Jiankang Chen, Haojie Ding, Kaiyu Tang, Zhang Zhang, Liang Wang, Fan Yang, Tingting Gao, and Guorui Zhou. Thyme: Think beyond images. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=gCWLkqK45O.

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. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=xUyMXkI958.

Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. DynaMath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=VOAMTA8jKu.

A Training the Pure-Text Reasoner

Source data and re-annotation. The Pure-Text Reasoner starts from Qwen2.5-VL-7B-Instruct and receives the original question and image. We construct its supervision from the DeepEyesV2 source pool. Of the original 65K SFT examples, we retain 53K visual reasoning examples after removing text-only tasks. Qwen2.5-VL-72B generates alternative rationales from the questions and images with tool access disabled. The generation prompt does not include the original tool-use rationale. The resulting targets contain reasoning text and a final answer, with no tool-calling supervision in this training pipeline.

The re-annotation uses temperature 0.6, top-p 1.0, presence penalty 1.2, and an 8,192-token generation limit. The system instruction is `You are a helpful assistant. Please solve the following question.` The following formatting instruction is appended to the question:

```
You MUST provide your thinking process and your final answer as the following format:
<think>
your thinking process
</think>
<answer>
your final answer to the question
</answer>
```

Filtering and reinforcement learning. The original RL pool contains 82K examples. To reduce training time, we apply a pass@8 filter: we sample eight responses per example and retain examples with between one and seven correct responses, yielding 30K training examples. This pass@8 filter removes examples that are always correct or always incorrect under the sampled responses. Filtering uses temperature 1.0, top-p 0.95, repetition penalty 1.2, and an 8,192-token response limit. We then train with GRPO and a binary final-answer reward. The synchronous reward function extracts the content within `<answer>` tags and asks Qwen3-235B-A22B-Instruct-2507 (Yang et al., 2025) to compare it with the reference answer. A missing final-answer field receives zero reward. The reward does not assign credit for producing tool calls or for a particular reasoning format beyond extracting the answer.

Table 8: SFT and GRPO configuration supplied with the training code. The GRPO columns list settings shared by the provided launch configurations.

Parameter	SFT	GRPO
Training examples	53K	30K
Learning rate	2×10^{-5}	10^{-6}
Optimization batch	4/GPU, accumulation 4	128 prompts
Samples per prompt	1 supervised target	16 rollouts
Sequence limits	4,096 tokens	1,024 prompt; 4,096 response
Training schedule	2 epochs, cosine	200-epoch limit
SFT warmup ratio	0.1	n/a
Vision tower / projector	Frozen	n/a
Language model	Full fine-tuning	Policy optimization
KL loss / reward	n/a	Disabled / disabled
Entropy coefficient	n/a	0

KL denotes Kullback–Leibler divergence; the KL loss and reward penalty are disabled in the listed GRPO configuration.

B Models and Evaluation Protocols

Model interfaces. DeepEyesV2 and Thyme use their code-execution wrappers for image processing and computation. Qwen3-VL-8B-Instruct uses a local crop-and-zoom wrapper with `image_zoom_in_tool`. The official repository provides an example of this interface.[1] Our wrapper's call specifies a bounding box, a region label, and an image index. The wrapper crops with PIL and returns the crop with textual metadata. It does not expose general Python execution to Qwen. The supplied Qwen wrapper permits six tool rounds, retains the source image resolution, and sends JPEG images with the API image-detail field set to `high`. Coordinates are expressed on a 0 to 1000 scale; the returned crop keeps the selected pixel region at its native resolution.

Decoding settings. The main benchmark evaluations use temperature 0.7. Across the tested temperatures, each model's accuracy or normalized score varied by at most 1.9 points. Training-data generation uses the separate settings described in Appendix A.

All models receive the original visual input under Tool-Free inference. The execution path is disabled even if a response contains residual code text. In the main suite, code-marker suppression is 100% for DeepEyesV2 and 98.88% for Thyme. Thyme's 316 code-bearing responses all occur in OCRBench; these responses do not execute code. This distinction preserves the meaning of Tool-Free while separating it from compliance with the requested output format.

Benchmarks and scoring units. The main suite contains 28,150 evaluation records. Real-world understanding includes V*Bench (Wu & Xie, 2024), HRBench-4K and HRBench-8K (Wang et al., 2025b), and TreeBench (Wang et al., 2026a). OCR includes OCRBench (Liu et al., 2024) and SEED-Bench-2-Plus (Li et al., 2024); chart understanding includes the Descriptive and Reasoning splits of CharXiv (Wang et al., 2024b) and ChartQA (Masry et al., 2022). Mathematical reasoning includes MathVista (Lu et al., 2024), MathVerse (Zhang et al., 2024), MathVision (Wang et al., 2024a), WeMath (Qiao et al., 2025), DynaMath (Zou et al., 2025), and LogicVista (Xiao et al., 2024). The two HRBench resolutions and two CharXiv splits are separate evaluation settings. These counts refer to benchmark records: DynaMath contains 5,010 evaluated variants, and WeMath records include the component questions used in its composite metric.

All benchmark evaluations use VLMEvalKit (Duan et al., 2024). For each benchmark, the same evaluation protocol is used across models and settings. GPT-4o is the judge for benchmarks requiring model-based answer evaluation. Native scores and sample-level correctness are taken from the corresponding evaluation outputs. OCRBench is reported on its native 0 to 1000 scale; WeMath uses its strict composite score; DynaMath uses its Average score over evaluated variants. For success-set comparisons, a record is correct when its binary correctness field equals one. Native aggregate scores and record-level success fractions are kept separate for composite metrics.

Table 9: Analyses, tool-call rates, and data sources. n/a denotes an analysis not included for that model.

Analysis	DeepEyesV2	Thyme	Qwen3-VL-8B	Source
Main-suite performance	Yes	Yes	Yes	Records / aggregate*
Own Tool-Free success sets	Yes	Yes	Yes	Records
Coverage with Pure-Text Reasoner	Yes	Yes	Yes	Records
Trajectory-edit benchmark scores	Yes	Yes	Yes	Reported / records*
Process annotations	Yes	Yes	n/a	Records
TIR-Bench / O3-Bench performance	Yes	Yes	n/a	Reported scores
Full tool-call rate (%)	99.98	2.9	54.4	Records

*Specialized-agent No-Result and No-Call scores and Qwen MathVision scores use the reported evaluations; the other performance entries are computed from their corresponding records.

[1] `https://github.com/QwenLM/Qwen3-VL/blob/main/cookbooks/think_with_images.ipynb`

TIR-Bench and O3-Bench contain 1,215 and 345 questions, respectively, and form an additional evaluation of DeepEyesV2 and Thyme.

Tool-call rates. Tool-call rates count Full records containing at least one call (Table 9).

Tool-Free instructions. The system instructions below are reproduced from the supplied wrappers. They operate with the benchmark question and original image. DeepEyesV2 also appends the user instruction shown below; benchmark-specific answer instructions, such as selecting an option or giving a short phrase, remain part of the user message.

DeepEyesV2 system instruction.

```
You are a helpful assistant. Please keep reasoning until the user's query is completely
↪  resolved before providing your final answer.
Solve the following problem step by step using only natural-language reasoning. You
↪  should analyze the visual and textual information directly without writing any code
↪  or using any external tools.
You MUST plan extensively and think carefully about the visual content before arriving at
↪  an answer. Reflect on your reasoning process at each step to ensure correctness.
Note:
1. Do NOT generate any Python code or use any code-execution tools.
2. Do NOT use <code></code>, <sandbox_output></sandbox_output>, or any similar tags.
3. Reason purely from what you can observe in the image and the question text.
4. If the image contains fine-grained details that are hard to perceive, reason carefully
↪  based on your best visual interpretation rather than requesting image processing.
```

DeepEyesV2 appended user instruction.

```
You must put your answer inside <answer> </answer> tags, i.e., <answer> answer here
↪  </answer>. Please reason step by step using only natural-language reasoning. Do not
↪  generate any code or use any tools. Format strictly as <think> </think> <answer>
↪  </answer>.
```

Thyme system instruction.

```
You are a helpful assistant.

Solve the following visual problem step by step using only natural-language reasoning.

Rules:
1. Do not generate or output any Python code.
2. Do not use <code>...</code>, <sandbox_output>...</sandbox_output>, or any
↪  code-execution instructions.
3. Reason from the visual and textual information directly, then provide a concise final
↪  answer.
4. If the image is unclear or ambiguous, reason based on your best visual interpretation
↪  rather than requesting image processing.
```

Qwen system instruction.

```
You are a helpful visual reasoning assistant.

You must solve the problem by reasoning over the provided image(s) directly.
You do NOT have access to any tool. Do NOT output any tool call, function call,
code block, or <tool_call> tags, and do NOT try to crop or zoom the image.
Reason step by step over the given image and put your final answer within
↪  <answer></answer>.
```

ChartQA and OCRBench scoring. We use benchmark correctness labels for scores and success-set comparisons. ChartQA uses relaxed matching with 5% numeric tolerance, and OCRBench uses normalized substring matching. Each benchmark's evaluation protocol is shared across models and settings.

C Complete Benchmark Results

The following tables report the Full and Tool-Free scores underlying the paired comparisons in Figure 2. Qwen's table also includes its trajectory edits. Appendix E gives the specialized agents' trajectory-edit scores. OCRBench retains its native 0 to 1000 score in tables; plots and percentage-point differences divide this score by ten.

Table 10: Complete benchmark scores for DeepEyesV2. OCRBench uses the 0 to 1000 scale; WeMath uses its Strict score and DynaMath its Average score.

Benchmark	Full	Tool-Free
V* Bench	81.68	80.10
HRBench-4K	75.75	74.75
HRBench-8K	70.50	71.13
TreeBench	41.48	33.09
OCRBench	814	866
SEED-2-Plus	69.78	71.19
CharXiv Descr.	75.68	77.00
CharXiv Reason.	43.70	46.20
ChartQA	88.08	83.12
MathVista	72.50	73.80
MathVerse	51.75	48.63
MathVision	26.45	28.52
WeMath	37.14	43.05
DynaMath	57.60	57.90
LogicVista	48.10	49.22

Appendix E gives the trajectory-edit scores; Appendix B describes scoring.

Table 11: Complete benchmark scores for Thyme. OCRBench uses the 0 to 1000 scale; WeMath uses its Strict score and DynaMath its Average score.

Benchmark	Full	Tool-Free
V* Bench	82.72	83.77
HRBench-4K	77.25	78.50
HRBench-8K	72.50	72.25
TreeBench	38.77	37.53
OCRBench	865	875
SEED-2-Plus	70.22	69.70
CharXiv Descr.	73.25	73.25
CharXiv Reason.	43.80	43.00
ChartQA	87.32	87.24
MathVista	72.50	71.60
MathVerse	47.26	47.87
MathVision	24.87	24.87
WeMath	37.52	37.52
DynaMath	54.11	54.25
LogicVista	48.32	47.43

Appendix E gives the trajectory-edit scores; Appendix B describes scoring.

Table 3 reports the aggregate results for TIR-Bench and O3-Bench. These additional results do not enter the paired tests or success-set analyses, which use the main-suite records.

Table 13 gives additional reference models. These background scores are separate from the paired Full and Tool-Free runs used for success-set and intervention analyses.

Table 12: Complete benchmark scores for Qwen3-VL-8B. OCRBench uses the 0 to 1000 scale; WeMath uses its Strict score and DynaMath its Average score.

Benchmark	**Full**	**Tool-Free**	**No-Result**	**No-Call**	**Empty-Call**
V* Bench	86.39	83.25	80.63	82.72	80.10
HRBench-4K	84.00	76.88	75.88	80.38	73.88
HRBench-8K	74.63	75.75	70.00	73.88	69.00
TreeBench	44.94	45.43	41.73	43.70	40.25
OCRBench	855	838	858	859	840
SEED-2-Plus	70.66	71.59	70.84	72.42	69.65
CharXiv Descr.	86.95	89.38	83.65	86.08	79.25
CharXiv Reason.	40.10	50.40	44.40	47.50	42.20
ChartQA	52.80	67.04	58.96	60.08	57.12
MathVista	68.40	76.90	70.40	71.50	68.10
MathVerse	53.98	58.76	51.19	51.73	50.23
MathVision	54.10	55.70	52.30	52.50	52.10
WeMath	44.57	52.57	48.10	45.62	45.62
DynaMath	62.50	66.75	65.71	65.85	64.91
LogicVista	49.89	51.68	44.97	48.32	40.49

No-Call denotes No-Call-Message.

Table 13: Additional reference models. These background scores are separate from the paired agent runs analyzed in this paper; n/a denotes an unreported entry. OCRBench uses its native 0 to 1000 score.

A. Real-world understanding and OCR

Model	Tools	V*	HR-4K	HR-8K	Tree	OCR	SEED2+
Qwen2.5-VL (Bai et al., 2025b)	None	78.5	71.6	67.9	37.0	864	70.4
Pixel-Reasoner (Su et al., 2025)	Crop	84.3	74.0	66.9	39.0	n/a	n/a
DeepEyes (Zheng et al., 2026)	Crop	85.6	75.1	72.6	37.5	n/a	n/a

B. Chart understanding

Model	Tools	CharXiv Descriptive	CharXiv Reasoning	ChartQA
Qwen2.5-VL	None	72.7	40.2	86.2

C. Mathematical reasoning

Model	Tools	MathVista	MathVerse	MathVision	WeMath	DynaMath	LogicVista
Qwen2.5-VL	None	68.3	45.6	25.6	34.6	53.3	45.9
MM-Eureka (Meng et al., 2025)	None	72.6	n/a	28.1	21.8	n/a	46.3
VL-Rethinker (Wang et al., 2025a)	None	73.7	n/a	28.4	36.3	n/a	42.7
DeepEyes	Crop	70.1	47.3	26.6	38.9	55.0	47.7

D SOLVED SETS, OUTCOMES, AND PAIRED TESTS

Success-set counts. Let F, T, and P denote correct records from Full, the same agent's Tool-Free setting, and the Pure-Text Reasoner. Comparisons are joined by benchmark and record identifier. Conditioning on F, the four disjoint groups for each agent are

$$F \setminus (T \cup P), \qquad F \cap T \setminus P,$$
$$F \cap P \setminus T, \qquad F \cap T \cap P.$$

Their counts sum to $|F|$. The first is the group not reproduced by the specified references; the other three together give coverage by at least one reference. For Qwen, the existing Pure-Text Reasoner labels are matched to the agent records by benchmark and question ID; question text, reference answers, and available answer choices agree throughout. The same Qwen2.5-VL-based reasoner serves as an auxiliary reference for each agent. Qwen's call-outcome and replay comparisons continue to condition on its own Tool-Free run. These comparisons count observed successful non-tool solutions, including cases where the two references solve different problems.

Table 14: Benchmark-level success coverage for DeepEyesV2. Coverage columns are percentages of Full successes; remaining columns are counts. P denotes the Pure-Text Reasoner. Uncovered, TF only, P only, and Both partition Full successes.

Benchmark	Full correct	Own TF	With P	Uncovered	TF only	P only	Both
V* Bench	156	91.7	96.2	6	3	7	140
HRBench-4K	606	89.8	95.7	26	31	36	513
HRBench-8K	564	92.2	96.1	22	39	22	481
TreeBench	168	53.6	81.5	31	21	47	69
OCRBench	814	98.4	99.3	6	24	7	777
SEED-2-Plus	1,589	93.3	96.3	58	87	49	1,395
CharXiv Descr.	3,027	92.6	97.7	71	185	154	2,617
CharXiv Reason.	437	74.1	86.7	58	72	55	252
ChartQA	2,202	90.1	97.6	53	230	166	1,753
MathVista	725	91.6	97.4	19	64	42	600
MathVerse	2,039	73.1	87.0	266	205	282	1,286
MathVision	804	57.2	74.3	207	137	137	323
WeMath	1,103	85.2	95.3	52	111	111	829
DynaMath	2,886	81.0	90.9	264	281	283	2,058
LogicVista	215	68.4	82.3	38	39	30	108

Coverage intervals. The intervals in Table 4 use 10,000 paired bootstrap samples, stratified by benchmark with seed 20260923. Each resampled record retains all of its setting labels. DynaMath variants are sampled together by their source question, and WeMath components by their composite question; this preserves dependence within the 501 DynaMath groups and 525 WeMath groups. We recompute the success-conditioned coverage ratio in every sample and report its 2.5th and 97.5th percentiles.

Coverage counts and sources. Table 17 reports coverage after adding No-Result. The two references are combined before adding successes from No-Result. For Thyme, the table separates Full trajectories with and without calls, since another successful run on a question with no Full call does not measure the effect of removing a returned result. The remaining shares are not a direct measure of causal necessity.

Corrections and introduced errors. These rates are computed on the subset C of records with a Full call. The baseline is the agent's own Tool-Free run in the main analysis; the appendix also reports comparisons against the Pure-Text Reasoner separately. We define **TRR** (Tool Redundant Rate) as the fraction of samples where both the baseline and the full agent are correct, **TFR** (Tool Failure Rate) as the fraction where both are wrong, **TER** (Tool Effective Rate) as the fraction where

Table 15: Benchmark-level success coverage for Thyme. Coverage columns are percentages of Full successes; remaining columns are counts. P denotes the Pure-Text Reasoner. Uncovered, TF only, P only, and Both partition Full successes.

Benchmark	Full correct	Own TF	With P	Uncovered	TF only	P only	Both
V* Bench	158	97.5	98.7	2	8	2	146
HRBench-4K	618	96.8	98.2	11	48	9	550
HRBench-8K	580	94.7	96.7	19	48	12	501
TreeBench	157	83.4	89.2	17	31	9	100
OCRBench	865	99.3	99.8	2	33	4	826
SEED-2-Plus	1,599	95.3	97.5	40	101	35	1,423
CharXiv Descr.	2,930	96.3	98.3	50	195	59	2,626
CharXiv Reason.	438	84.5	93.6	28	93	40	277
ChartQA	2,183	98.4	99.0	21	217	13	1,932
MathVista	725	93.5	97.8	16	65	31	613
MathVerse	1,862	85.9	94.2	108	245	154	1,355
MathVision	756	66.4	80.2	150	150	104	352
WeMath	1,136	86.4	95.0	57	124	98	857
DynaMath	2,711	88.4	95.2	130	364	185	2,032
LogicVista	216	75.9	85.6	31	41	21	123

Table 16: Benchmark-level success coverage for Qwen3-VL-8B. Coverage columns are percentages of Full successes; the last three columns are counts.

Benchmark	Full correct	Own TF	With P	Uncovered	TF only	P only	Both
V* Bench	165	90.9	96.4	3.6	13	9	137
HRBench-4K	672	87.9	94.0	6.0	62	41	529
HRBench-8K	597	90.6	94.5	5.5	74	23	467
TreeBench	182	80.8	87.4	12.6	42	12	105
OCRBench	855	94.2	98.7	1.3	29	39	776
SEED-2-Plus	1,609	92.6	96.5	3.5	126	62	1,364
CharXiv Descr.	3,478	95.3	97.9	2.1	482	91	2,832
CharXiv Reason.	401	77.3	86.0	14.0	69	35	241
ChartQA	1,320	95.9	98.3	1.7	95	31	1,171
MathVista	684	92.8	96.5	3.5	89	25	546
MathVerse	2,127	86.9	93.6	6.4	414	142	1,435
MathVision	1,645	85.0	94.1	5.9	908	150	490
WeMath	1,189	91.6	97.3	2.7	183	68	906
DynaMath	3,131	88.9	93.6	6.4	591	146	2,194
LogicVista	223	77.1	85.2	14.8	59	18	113

P denotes the Pure-Text Reasoner. Question IDs, question text, reference answers, and available answer choices match for the benchmarks in this table.

Table 17: Coverage of Full successes and Qwen successful-answer retention. The two references are own Tool-Free and the Pure-Text Reasoner; No-Result is added to their union.

A. Coverage (%)

Model	Benchmarks	Two references	With No-Result	Remaining
DeepEyesV2	15	93.2	97.1	2.9
Thyme	15	96.0	98.4	1.6

B. Qwen called-success retention

Setting / subset	Correct	Full successes	Retention (%)
Tool-Free	9,136	10,112	90.3
No-Result	8,731	10,112	86.3
No-Call	9,130	10,112	90.3
Empty-Call	8,302	10,112	82.1
No-Result, missed by Tool-Free	470	976	48.2

For Thyme, the references cover 16,252/16,934 successes and No-Result adds 411; 405 of these have no Full call and six have a call.

the baseline is wrong but the full agent becomes correct, and **TIR** (Tool Interference Rate) as the fraction where the baseline is correct but the full agent becomes wrong.

For a reference R, let n_{11} count shared successes, n_{00} shared failures, n_{10} Full-correct/reference-wrong records, and n_{01} Full-wrong/reference-correct records within C. Then

$$\mathrm{TRR} = \frac{n_{11}}{|C|}, \quad \mathrm{TFR} = \frac{n_{00}}{|C|}, \quad \mathrm{TER} = \frac{n_{10}}{|C|}, \quad \mathrm{TIR} = \frac{n_{01}}{|C|}.$$

The four rates sum to one and are displayed as percentages. Conditional correction and induced error instead use

$$\frac{n_{10}}{n_{00} + n_{10}} \quad \text{and} \quad \frac{n_{01}}{n_{11} + n_{01}},$$

conditioning on reference failures and successes, respectively. These are the Correction/Induced error columns in the main table and Corr./Harm in the benchmark tables. For an edited setting with correct-record set E, retention of called Full successes is $|F \cap C \cap E|/|F \cap C|$; this is distinct from correctness agreement, which also counts shared failures.

Conditional correction and error-introduction rates use 95% Wilson intervals. These and the coverage bootstrap intervals quantify uncertainty across benchmark questions for the evaluated runs.

Paired significance tests. For benchmarks with binary per-record outcomes, we use a two-sided exact McNemar test on all paired records, including records without a call. DynaMath and WeMath instead use 100,000 paired group swaps at the source question level. DynaMath uses its Average score and WeMath its Strict composite score. Holm correction controls the family of paired comparisons. The family comprises all 45 model–benchmark comparisons. Tables 23 and 24 report the gains, losses, score differences, and test results.

Table 18: Called-subset outcomes for DeepEyesV2 against its own Tool-Free setting (%). Corr. conditions on Tool-Free being wrong; Harm conditions on Tool-Free being correct. Zero denominators are n/a.

Benchmark	Called	TRR	TFR	TER	TIR	Corr.	Harm
V* Bench	191	74.9	13.1	6.8	5.2	34.2	6.5
HRBench-4K	800	68.0	17.5	7.8	6.8	30.7	9.0
HRBench-8K	800	65.0	23.4	5.5	6.1	19.0	8.6
TreeBench	404	22.3	47.5	19.3	10.9	28.9	32.8
OCRBench	999	80.2	12.0	1.3	6.5	9.8	7.5
SEED-2-Plus	2,276	65.1	24.1	4.7	6.1	16.3	8.6
CharXiv Descr.	3,999	70.1	17.4	5.6	6.9	24.5	9.0
CharXiv Reason.	1,000	32.4	42.5	11.3	13.8	21.0	29.9
ChartQA	2,500	79.3	8.1	8.8	3.8	51.9	4.6
MathVista	1,000	66.4	20.1	6.1	7.4	23.3	10.0
MathVerse	3,940	37.8	37.5	13.9	10.8	27.1	22.2
MathVision	3,039	15.1	60.2	11.3	13.4	15.8	46.9
WeMath	1,740	54.0	23.6	9.4	13.0	28.4	19.5
DynaMath	5,009	46.7	31.2	10.9	11.2	25.9	19.4
LogicVista	447	32.9	35.6	15.2	16.3	30.0	33.2

Table 19: Called-subset outcomes for Thyme against its own Tool-Free setting (%). Corr. conditions on Tool-Free being wrong; Harm conditions on Tool-Free being correct. Zero denominators are n/a.

Benchmark	Called	TRR	TFR	TER	TIR	Corr.	Harm
V* Bench	31	67.7	16.1	3.2	12.9	16.7	16.0
HRBench-4K	100	73.0	17.0	2.0	8.0	10.5	9.9
HRBench-8K	109	74.3	19.3	3.7	2.8	16.0	3.6
TreeBench	15	40.0	53.3	0.0	6.7	0.0	14.3
OCRBench	363	85.7	11.8	0.3	2.2	2.3	2.5
SEED-2-Plus	47	68.1	23.4	2.1	6.4	8.3	8.6
CharXiv Descr.	5	40.0	40.0	0.0	20.0	0.0	33.3
CharXiv Reason.	2	0.0	100.0	0.0	0.0	0.0	n/a
ChartQA	9	88.9	11.1	0.0	0.0	0.0	0.0
MathVista	1	0.0	100.0	0.0	0.0	0.0	n/a
MathVerse	48	37.5	39.6	22.9	0.0	36.7	0.0
MathVision	8	12.5	75.0	0.0	12.5	0.0	50.0
WeMath	2	50.0	50.0	0.0	0.0	0.0	0.0
DynaMath	64	29.7	40.6	25.0	4.7	38.1	13.6
LogicVista	0	n/a	n/a	n/a	n/a	n/a	n/a

Table 20: Called-subset outcomes for Qwen3-VL-8B against its own Tool-Free setting (%). Corr. conditions on Tool-Free being wrong; Harm conditions on Tool-Free being correct. Zero denominators are n/a.

Benchmark	Called	TRR	TFR	TER	TIR	Corr.	Harm
V* Bench	187	79.7	7.5	8.0	4.8	51.7	5.7
HRBench-4K	781	73.6	12.9	10.4	3.1	44.5	4.0
HRBench-8K	757	69.6	15.3	7.4	7.7	32.6	9.9
TreeBench	373	34.0	48.8	8.8	8.3	15.3	19.6
OCRBench	958	80.6	11.2	5.0	3.2	31.0	3.9
SEED-2-Plus	2,090	65.7	22.8	5.2	6.4	18.5	8.8
CharXiv Descr.	3,197	80.7	7.0	4.8	7.5	40.7	8.5
CharXiv Reason.	975	30.6	40.6	9.1	19.7	18.4	39.2
ChartQA	970	41.1	31.0	3.6	24.2	10.4	37.1
MathVista	599	55.8	21.0	5.7	17.5	21.3	23.9
MathVerse	1,155	34.1	39.0	9.9	17.0	20.2	33.2
MathVision	971	52.4	24.5	6.3	16.8	20.4	24.3
WeMath	227	46.3	33.5	7.9	12.3	19.1	21.1
DynaMath	1,763	51.6	25.5	5.1	17.9	16.7	25.7
LogicVista	319	26.6	44.5	12.9	16.0	22.4	37.5

Table 21: Called-subset outcomes for DeepEyesV2 against the Pure-Text Reasoner (%). Corr. and Harm condition on reference failures and reference successes, respectively.

Benchmark	Called	TRR	TFR	TER	TIR	Corr.	Harm
V* Bench	191	77.0	12.6	4.7	5.8	27.3	7.0
HRBench-4K	800	68.6	16.1	7.1	8.1	30.6	10.6
HRBench-8K	800	62.9	20.9	7.6	8.6	26.8	12.1
TreeBench	404	28.7	44.6	12.9	13.9	22.4	32.6
OCRBench	999	78.5	8.7	3.0	9.8	25.6	11.1
SEED-2-Plus	2,276	63.4	23.7	6.4	6.5	21.2	9.2
CharXiv Descr.	3,999	69.3	15.3	6.4	9.0	29.5	11.5
CharXiv Reason.	1,000	30.7	42.0	13.0	14.3	23.6	31.8
ChartQA	2,500	76.8	7.7	11.3	4.2	59.5	5.2
MathVista	1,000	64.2	19.4	8.3	8.1	30.0	11.2
MathVerse	3,940	39.8	35.4	12.0	12.9	25.3	24.4
MathVision	3,039	15.1	60.0	11.3	13.6	15.9	47.3
WeMath	1,740	54.0	20.1	9.4	16.5	31.8	23.4
DynaMath	5,009	46.7	31.9	10.9	10.5	25.4	18.3
LogicVista	447	30.9	38.9	17.2	13.0	30.7	29.6

Table 22: Called-subset outcomes for Thyme against the Pure-Text Reasoner (%). Corr. and Harm condition on reference failures and reference successes, respectively.

Benchmark	Called	TRR	TFR	TER	TIR	Corr.	Harm
V* Bench	31	64.5	19.4	6.5	9.7	25.0	13.0
HRBench-4K	100	67.0	16.0	8.0	9.0	33.3	11.8
HRBench-8K	109	71.6	19.3	6.4	2.8	25.0	3.7
TreeBench	15	33.3	33.3	6.7	26.7	16.7	44.4
OCRBench	363	82.9	8.0	3.0	6.1	27.5	6.8
SEED-2-Plus	47	66.0	25.5	4.3	4.3	14.3	6.1
CharXiv Descr.	5	40.0	20.0	0.0	40.0	0.0	50.0
CharXiv Reason.	2	0.0	50.0	0.0	50.0	0.0	100.0
ChartQA	9	44.4	0.0	44.4	11.1	100.0	20.0
MathVista	1	0.0	100.0	0.0	0.0	0.0	n/a
MathVerse	48	39.6	35.4	20.8	4.2	37.0	9.5
MathVision	8	12.5	62.5	0.0	25.0	0.0	66.7
WeMath	2	0.0	0.0	50.0	50.0	100.0	100.0
DynaMath	64	28.1	35.9	26.6	9.4	42.5	25.0
LogicVista	0	n/a	n/a	n/a	n/a	n/a	n/a

Table 23: Paired Full versus Tool-Free tests. n_{10} and n_{01} count record-level correctness gains and losses; Δ uses the benchmark-native score in percentage points. The family for Holm correction comprises 45 model–benchmark pairs.

Benchmark	n_{10}	n_{01}	Δ	**Test**	p	p_{Holm}
DeepEyesV2						
V* Bench	13	10	+1.57	Exact	0.678	1.000
HRBench-4K	62	54	+1.00	Exact	0.516	1.000
HRBench-8K	44	49	-0.63	Exact	0.679	1.000
TreeBench	78	44	+8.40	Exact	0.003	0.088
OCRBench	13	65	-5.20	Exact	1.8×10^{-9}	7.2×10^{-8} ↓
SEED-2-Plus	107	139	-1.41	Exact	0.048	1.000
CharXiv Descr.	225	278	-1.33	Exact	0.020	0.650
CharXiv Reason.	113	138	-2.50	Exact	0.130	1.000
ChartQA	219	95	+4.96	Exact	2.0×10^{-12}	8.9×10^{-11} ↑
MathVista	61	74	-1.30	Exact	0.302	1.000
MathVerse	548	425	+3.12	Exact	9.0×10^{-5}	0.003 ↑
MathVision	344	407	-2.07	Exact	0.024	0.732
WeMath	163	227	-5.90	Group	0.002	0.079
DynaMath	547	562	-0.30	Group	0.797	1.000
LogicVista	68	73	-1.12	Exact	0.736	1.000
Thyme						
V* Bench	4	6	-1.05	Exact	0.754	1.000
HRBench-4K	20	30	-1.25	Exact	0.203	1.000
HRBench-8K	31	29	+0.25	Exact	0.897	1.000
TreeBench	26	21	+1.23	Exact	0.560	1.000
OCRBench	6	16	-1.00	Exact	0.052	1.000
SEED-2-Plus	75	63	+0.53	Exact	0.349	1.000
CharXiv Descr.	109	109	0.00	Exact	1.000	1.000
CharXiv Reason.	68	60	+0.80	Exact	0.536	1.000
ChartQA	34	32	+0.08	Exact	0.902	1.000
MathVista	47	38	+0.90	Exact	0.386	1.000
MathVerse	262	286	-0.61	Exact	0.326	1.000
MathVision	254	254	0.00	Exact	1.000	1.000
WeMath	155	143	0.00	Group	1.000	1.000
DynaMath	315	322	-0.14	Group	0.849	1.000
LogicVista	52	48	+0.89	Exact	0.764	1.000

Exact: two-sided exact McNemar. Group: 100,000 paired group swaps, using 501 DynaMath source questions or 525 WeMath source questions; WeMath uses its Strict composite metric. The family for Holm correction comprises 45 model–benchmark pairs.

Table 24: Paired Full versus Tool-Free tests. n_{10} and n_{01} count record-level correctness gains and losses; Δ uses the benchmark-native score in percentage points. The family for Holm correction comprises 45 model–benchmark pairs.

Benchmark	n_{10}	n_{01}	Δ	**Test**	p	p_{Holm}
Qwen3-VL-8B						
V* Bench	15	9	+3.14	Exact	0.307	1.000
HRBench-4K	81	24	+7.13	Exact	2.1×10^{-8}	8.1×10^{-7} ↑
HRBench-8K	56	65	-1.13	Exact	0.467	1.000
TreeBench	35	37	-0.49	Exact	0.906	1.000
OCRBench	50	33	+1.70	Exact	0.078	1.000
SEED-2-Plus	119	140	-0.92	Exact	0.214	1.000
CharXiv Descr.	164	261	-2.43	Exact	2.9×10^{-6}	1.1×10^{-4} ↓
CharXiv Reason.	91	194	-10.30	Exact	1.0×10^{-9}	4.1×10^{-8} ↓
ChartQA	54	410	-14.24	Exact	8.3×10^{-69}	3.7×10^{-67} ↓
MathVista	49	134	-8.50	Exact	2.5×10^{-10}	1.1×10^{-8} ↓
MathVerse	278	466	-4.77	Exact	5.5×10^{-12}	2.4×10^{-10} ↓
MathVision	247	295	-1.58	Exact	0.0434063	1.000
WeMath	100	178	-8.00	Group	1.0×10^{-5}	3.7×10^{-4} ↓
DynaMath	346	559	-4.25	Group	2.0×10^{-5}	7.2×10^{-4} ↓
LogicVista	51	59	-1.79	Exact	0.505	1.000

Exact: two-sided exact McNemar. Group: 100,000 paired group swaps, using 501 DynaMath source questions or 525 WeMath source questions; WeMath uses its Strict composite metric. The family for Holm correction comprises 45 model–benchmark pairs.

E TRAJECTORY EDITING AND CONDITIONAL CORRECTNESS

Changes for DeepEyesV2 and Thyme. No-Result retains the generated call, including its code, and replaces the execution result with an empty placeholder before the model continues. No-Call-Message, abbreviated No-Call in tables, keeps the returned result and removes the call text. These correspond to Tool Format Only and Tool Result Only in the original experiments. Empty-Call corresponds to the additional Format-NoSemantics control: it replaces the task-relevant code with `pass`, retaining the wrapper without its computation. Original images remain available.

Tables 25 and 26 give the ablation scores.

Final-context replay for Qwen. Each Qwen intervention run first generates its own tool-enabled conversation. The wrapper then reconstructs the successful crop rounds with the chosen changes, adds a final-answer instruction, and requests one new completion. No-Result retains the generated call parameters but replaces the crop return with a withholding message. No-Call-Message removes the call syntax while retaining its surrounding reasoning text and the returned crop. Empty-Call additionally replaces the bounding box with `[0, 0, 0, 0]`, the region label with `region`, and the image index with zero. Thus, Qwen's Empty-Call masks crop parameters; the Python-based control uses `pass`.

Failed crop rounds without a returned image do not enter the replay. Runs with no successful crop still receive the final-answer instruction, but have zero edited crop rounds.

Agreement and successful-answer retention. Table 5 reports correctness agreement with Full over all questions in each task family. Agreement counts both shared successes and shared failures. For Qwen, Table 17 also reports successful-answer retention under each comparison setting. These conditional fractions use successful questions as the denominator and differ from the all-question agreement measure.

Table 25: Trajectory-edit benchmark scores for DeepEyesV2. OCRBench uses its 0 to 1000 scale.

Benchmark	Full	No-Result	No-Call
V* Bench	81.68	83.77	81.68
HRBench-4K	75.75	76.00	73.50
HRBench-8K	70.50	70.88	68.75
TreeBench	41.48	39.01	34.07
OCRBench	814	805	828
SEED-2-Plus	69.78	69.78	69.87
CharXiv Descr.	75.68	76.05	73.78
CharXiv Reason.	43.70	44.70	45.90
ChartQA	88.08	87.08	80.24
MathVista	72.50	71.20	69.70
MathVerse	51.75	47.99	40.20
MathVision	26.45	27.27	28.06
WeMath	37.14	36.19	29.90
DynaMath	57.60	57.53	48.42
LogicVista	48.10	47.18	43.37

No-Result retains the call and withholds its result; No-Call retains the result and removes the call.

Table 26: Trajectory-edit benchmark scores for Thyme. OCRBench uses its 0 to 1000 scale.

Benchmark	Full	No-Result	No-Call
V* Bench	82.72	83.25	83.25
HRBench-4K	77.25	77.25	77.50
HRBench-8K	72.50	72.13	72.38
TreeBench	38.77	39.26	39.26
OCRBench	865	865	879
SEED-2-Plus	70.22	70.05	70.36
CharXiv Descr.	73.25	73.15	73.00
CharXiv Reason.	43.80	42.70	42.20
ChartQA	87.32	87.08	87.56
MathVista	72.50	71.30	70.90
MathVerse	47.26	46.70	47.39
MathVision	24.87	26.18	25.26
WeMath	37.52	38.67	37.43
DynaMath	54.11	53.45	54.21
LogicVista	48.32	50.56	47.20

No-Result retains the call and withholds its result; No-Call retains the result and removes the call.

F Process Annotations and Validation

Annotation input. We use Qwen3-VL-30B-A3B-Instruct to assign process labels to DeepEyesV2 and Thyme trajectories. The annotation template includes the question, reasoning, calls, returned content, final answer, and final score. The three label dimensions are information gain, output quality, and integration status. The prompt templates are reproduced below.

Labels and aggregation. The judge assigns information gain (Novel, Confirmatory, Irrelevant), output quality (Useful/Correct, Partially useful, Wrong/Failed), and integration status (Used correctly, Ignored, Misused/Misinterpreted). For aggregation, Useful/Correct becomes Correct and Partially useful becomes Partially correct. We apply the following priority rule: misinterpreted output is Misuse or harmful integration; Novel, Correct, and Used correctly is Genuine tool contribution; Novel and at least Partially correct but Ignored is Missed opportunity; Confirmatory and at least Partially correct, used or ignored, is Redundant confirmation; Wrong/Failed output, used or ignored, is Failed or non-contributory call. Other combinations receive Other.

We display Redundant confirmation as *Confirmatory*, Genuine tool contribution as *Informative*, and Failed or non-contributory call as *Failed*. Misuse, Missed opportunity, and Other form the *Other* display group. This grouping keeps the recorded label membership. Tables 27 and 28 give counts and Wilson intervals for all benchmarks; intervals are wide where Thyme has few called records.

The informative group contains 7,649 DeepEyesV2 trajectories and 732 Thyme trajectories. Their own Tool-Free settings answer 4,569 and 581 of these questions correctly, respectively. The denominator includes all trajectories in the informative group, without an additional Full-correct filter. The resulting overlaps are 59.7% and 79.4%. Table 29 compares these overlaps with all called records and with the disjoint non-informative group. Its second row for each model applies a Full-correct filter to every comparison group. The Fisher exact tests describe associations between process labels and Tool-Free correctness.

Validation. Two human annotators independently audited a random 30% subset of the judged tool-using trajectories. They saw the question, original image, full trajectory including the call and its returned output, and final answer. They labeled information gain, output quality, and integration status without seeing the automatic labels or each other's decisions; the composite role followed from the fixed mapping above. Human agreement was 94%. Disagreements were concentrated at the partially useful boundary and were resolved through discussion.

Table 27: Process-label proportions for DeepEyesV2 (%), with 95% Wilson intervals. Each benchmark uses its annotated called records as the denominator.

Benchmark	n	Confirmatory	Informative	Failed	Other
V* Bench	191	23.6 [18.1, 30.1]	11.5 [7.7, 16.8]	52.9 [45.8, 59.8]	12.0 [8.2, 17.4]
HRBench-4K	800	24.8 [21.9, 27.9]	7.9 [6.2, 9.9]	59.9 [56.4, 63.2]	7.5 [5.9, 9.5]
HRBench-8K	800	22.5 [19.7, 25.5]	7.9 [6.2, 9.9]	62.3 [58.8, 65.5]	7.4 [5.8, 9.4]
TreeBench	404	13.9 [10.8, 17.6]	3.2 [1.9, 5.4]	79.2 [75.0, 82.9]	3.7 [2.3, 6.0]
OCRBench	999	54.3 [51.2, 57.3]	3.8 [2.8, 5.2]	40.8 [37.8, 43.9]	1.1 [0.6, 2.0]
SEED-2-Plus	2,276	67.7 [65.7, 69.6]	7.0 [6.0, 8.1]	22.4 [20.7, 24.1]	3.0 [2.4, 3.8]
CharXiv Descr.	3,999	44.8 [43.3, 46.4]	16.7 [15.5, 17.8]	37.6 [36.1, 39.1]	0.9 [0.7, 1.2]
CharXiv Reason.	1,000	48.9 [45.8, 52.0]	11.5 [9.7, 13.6]	36.7 [33.8, 39.7]	2.9 [2.0, 4.1]
ChartQA	2,500	78.7 [77.0, 80.2]	15.2 [13.8, 16.7]	5.1 [4.3, 6.1]	1.0 [0.7, 1.5]
MathVista	1,000	57.5 [54.4, 60.5]	26.9 [24.2, 29.7]	12.6 [10.7, 14.8]	3.0 [2.1, 4.3]
MathVerse	3,940	43.9 [42.3, 45.4]	50.3 [48.7, 51.8]	0.6 [0.4, 0.9]	5.2 [4.6, 6.0]
MathVision	3,039	54.3 [52.6, 56.1]	31.1 [29.4, 32.7]	8.2 [7.3, 9.3]	6.4 [5.6, 7.3]
WeMath	1,740	50.0 [47.7, 52.3]	44.4 [42.0, 46.7]	3.6 [2.8, 4.6]	2.0 [1.4, 2.8]
DynaMath	5,009	42.4 [41.0, 43.7]	41.3 [40.0, 42.7]	13.4 [12.5, 14.4]	2.9 [2.4, 3.4]
LogicVista	447	63.5 [59.0, 67.9]	21.0 [17.5, 25.0]	13.0 [10.2, 16.4]	2.5 [1.4, 4.4]
Overall	28,144	49.9 [49.3, 50.5]	27.2 [26.7, 27.7]	19.6 [19.1, 20.0]	3.4 [3.2, 3.6]

Original labels are grouped without new judging. Other combines the remaining original categories; a zero denominator is n/a. Process roles follow the annotation protocol in Appendix F.

Table 28: Process-label proportions for Thyme (%), with 95% Wilson intervals. Each benchmark uses its annotated called records as the denominator.

Benchmark	n	Confirmatory	Informative	Failed	Other
V* Bench	31	0.0 [0.0, 11.0]	100.0 [89.0, 100.0]	0.0 [0.0, 11.0]	0.0 [0.0, 11.0]
HRBench-4K	100	0.0 [0.0, 3.7]	100.0 [96.3, 100.0]	0.0 [0.0, 3.7]	0.0 [0.0, 3.7]
HRBench-8K	109	0.0 [0.0, 3.4]	93.6 [87.3, 96.9]	5.5 [2.5, 11.5]	0.9 [0.2, 5.0]
TreeBench	15	0.0 [0.0, 20.4]	86.7 [62.1, 96.3]	6.7 [1.2, 29.8]	6.7 [1.2, 29.8]
OCRBench	363	0.0 [0.0, 1.0]	95.0 [92.3, 96.8]	3.3 [1.9, 5.7]	1.7 [0.8, 3.6]
SEED-2-Plus	47	0.0 [0.0, 7.6]	100.0 [92.4, 100.0]	0.0 [0.0, 7.6]	0.0 [0.0, 7.6]
CharXiv Descr.	5	0.0 [0.0, 43.4]	100.0 [56.6, 100.0]	0.0 [0.0, 43.4]	0.0 [0.0, 43.4]
CharXiv Reason.	2	0.0 [0.0, 65.8]	50.0 [9.5, 90.5]	0.0 [0.0, 65.8]	50.0 [9.5, 90.5]
ChartQA	9	100.0 [70.1, 100.0]	0.0 [0.0, 29.9]	0.0 [0.0, 29.9]	0.0 [0.0, 29.9]
MathVista	1	100.0 [20.7, 100.0]	0.0 [0.0, 79.3]	0.0 [0.0, 79.3]	0.0 [0.0, 79.3]
MathVerse	48	25.0 [14.9, 38.8]	75.0 [61.2, 85.1]	0.0 [0.0, 7.4]	0.0 [0.0, 7.4]
MathVision	8	37.5 [13.7, 69.4]	62.5 [30.6, 86.3]	0.0 [0.0, 32.4]	0.0 [0.0, 32.4]
WeMath	2	100.0 [34.2, 100.0]	0.0 [0.0, 65.8]	0.0 [0.0, 65.8]	0.0 [0.0, 65.8]
DynaMath	64	15.6 [8.7, 26.4]	73.4 [61.5, 82.7]	0.0 [0.0, 5.7]	10.9 [5.4, 20.9]
LogicVista	0	n/a	n/a	n/a	n/a
Overall	804	4.6 [3.4, 6.3]	91.0 [88.9, 92.8]	2.4 [1.5, 3.7]	2.0 [1.2, 3.2]

Original labels are grouped without new judging. Other combines the remaining original categories; a zero denominator is n/a. Process roles follow the annotation protocol in Appendix F.

Table 29: Tool-Free correctness in informative trajectories and comparison groups (%). G denotes informative labels and C called records. The Full-correct rows apply that filter to every column.

Model	Subset	$\lvert G\rvert$	G	C	$C \setminus G$	$G - (C \setminus G)$	p
DeepEyesV2	All called	7,649	59.7	61.7	62.4	-2.72	3.1×10^{-5}
	Full correct	4,845	80.0	85.0	86.9	-6.96	2.5×10^{-29}
Thyme	All called	732	79.4	75.2	33.3	46.04	2.7×10^{-15}
	Full correct	584	94.7	94.1	80.0	14.69	0.012

The difference compares the disjoint informative and other-called groups. Two-sided Fisher exact p values are exploratory, without multiplicity correction. They describe association in the saved records, not causal tool utility or independent-run stability.

GPT-4o (OpenAI, 2024) independently re-labeled the trajectories initially annotated by Qwen3-VL-30B-A3B-Instruct. The two judges agreed on 96% of roles, and both agreed with the human audit on approximately 93%.

Executed annotation prompts. The following system message and user template specify the annotation procedure.

System message.

```
You are an expert judge for multimodal agent tool-use analysis.

Your task is to inspect an agent trajectory and label the role of tool use using three
↪  dimensions.

Judge only from observable evidence in the trajectory. Do NOT give credit to tool use
↪  just because:
- the final answer is correct,
- the model says things became "clearer",
- or a tool call happened.

You must distinguish between truly new evidence and mere repetition / self-confirmation.
You must also keep the three dimensions strictly separate. Never reuse a label from one
↪  dimension in another dimension.

Definitions:
1. Information Gain
- Novel: the tool provides new task-relevant information not already available from the
↪  model's unaided reasoning.
- Confirmatory: the tool mainly confirms or repeats an already available conclusion.
- Irrelevant: the tool output is not materially useful for solving the task.

2. Tool Output Quality
- Useful/Correct: the tool output is correct and meaningfully usable.
- Partially useful: the tool output is partially correct, incomplete, or only weakly
↪  useful.
- Wrong/Failed: the tool output is wrong, failed, or unusable.

3. Integration Status
- Used correctly: the final reasoning meaningfully uses the tool output in a correct way.
- Ignored: useful or potentially useful tool output is not actually used.
- Misused/Misinterpreted: the model uses the tool output incorrectly or is misled by it.

Important decision rules:
1. Novel should be rare. Only assign Novel if the tool output clearly reveals or computes
↪  information that was not already available from the pre-tool reasoning.
2. If the tool only redisplays the original image (for example `plt.imshow(image_1)` or
↪  equivalent) without crop, zoom, OCR, enhancement, measurement, parsing, or other
↪  transformation, do NOT label it as Novel.
3. If the tool output is empty, trivial, or only repeats the original image view, it
↪  should usually be Confirmatory or Irrelevant, not Novel.
4. Do not infer tool usefulness from final correctness. A correct final answer can still
↪  correspond to Confirmatory, Partially useful, or even Irrelevant tool use.
5. Useful/Correct requires that the tool output itself is materially informative and
↪  correct. If it adds little or no new evidence, prefer Partially useful or
↪  Wrong/Failed as appropriate.
6. If the post-tool answer mostly repeats the same conclusion already stated before the
↪  tool call, that is evidence against Novel information gain.
7. `Irrelevant` is ONLY valid for `information_gain`. It is NOT a valid value for
↪  `tool_output_quality` or `integration_status`.
8. If the tool output is empty, blank, trivial, or unusable, then `tool_output_quality`
↪  should usually be `Wrong/Failed`.
9. If the tool output is merely the same unprocessed image view, with no crop / zoom / OCR
↪  / enhancement / measurement / parsing, then `tool_output_quality` should usually be
↪  `Partially useful` at most, and often `Wrong/Failed` if the execution adds nothing.
```

```
10. If the tool output adds nothing and the final answer comes from the same pre-tool
↪  reasoning, prefer `integration_status = Ignored`, not `Used correctly`.

Field-specific allowed values:
- information_gain must be exactly one of: Novel, Confirmatory, Irrelevant
- tool_output_quality must be exactly one of: Useful/Correct, Partially useful,
↪  Wrong/Failed
- integration_status must be exactly one of: Used correctly, Ignored,
↪  Misused/Misinterpreted

Before answering, silently check:
- Did the tool output reveal new evidence, or only repeat existing evidence?
- Was the tool output itself actually informative and correct?
- Did the final reasoning rely on the tool output, or mostly ignore it?

Concrete examples:
- If the assistant already says "the cart is green", then runs `plt.imshow(image_1)` and
↪  afterwards again says "the cart is clearly green", this is not Novel. It is at most
↪  Confirmatory.
- If the assistant crops or zooms into a small region and the crop newly reveals the
↪  object color or text, that can be Novel.
- If the tool output is blank or non-informative and the final answer is still produced
↪  from the model's own visual reasoning, the tool output is not Useful/Correct just
↪  because the answer is right.
- If the tool output is blank or unchanged and the model still answers correctly from its
↪  own prior reasoning, a common label pattern is:
  information_gain = Confirmatory or Irrelevant,
  tool_output_quality = Wrong/Failed or Partially useful,
  integration_status = Ignored.

Return ONLY a JSON object with exactly these keys:
{
  "information_gain": "...",
  "tool_output_quality": "...",
  "integration_status": "...",
  "confidence": 0-100,
  "rationale": "short explanation"
}

The rationale must be one short sentence with at most 25 words.
```

User message template.

```
Analyze the following multimodal agent trajectory.

Metadata:
- Model setting: {model_key}
- Benchmark: {benchmark}
- Sample index: {sample_index}
- Tool use times: {tool_use_times}
- Final score: {score}

Trajectory:
<trajectory>
{trajectory_text}
</trajectory>

Return only the JSON object.
```

G EXAMPLE TRAJECTORIES

Image redisplay with an unchanged answer. V*Bench record 0 asks, "What is the material of the glove?" with options rubber, cotton, kevlar, and leather. Before invoking a tool, DeepEyesV2 already describes the glove as rubber. It then executes:

```
import matplotlib.pyplot as plt
plt.imshow(image_1)
plt.axis('off')
plt.show()
```

The final answer is A, rubber. Our Pure-Text Reasoner also answers A, without manipulating the image. Figure 1 compares these two models. The call redisplays the original image; no crop, measurement, or other image transformation appears in its code. The trajectory contains a tool call, but the call supplies no new visual evidence.

An informative crop on an already solvable question. HRBench-8K record 78 asks for the text on the road sign below the American flag. Thyme crops the region and answers "ONE WAY." Its process labels are Novel, Useful/Correct, and Used correctly, placing it in the informative group. Its own Tool-Free response also answers "ONE WAY." The example shows why information attributed to a step and additional success must be checked separately.

A crop and a changed answer. For Qwen, we select the lowest question ID in HRBench-4K for which Full is correct while both Tool-Free and No-Result are wrong. Record 48 asks, "Where is the person standing in relation to the recycle bin?" The reference is A, "In front of the recycle bin." Full answers A, Tool-Free answers D, "To the left of the recycle bin," and No-Result answers C, "To the right of the recycle bin." Full crops the box `[443, 213, 624, 666]` on the 0 to 1000 coordinate scale. The saved pixel box is `[1786, 859, 2516, 2685]`, producing a 730×1826 crop. The No-Result run's internal tool-enabled parent uses the same box and also answers A before replay. This example records an answer change when the crop is withheld.