Expanding LLaVA-CoT with Advanced Inference-time Scaling

Yang Zhang*
Peking University
2200013216@stu.pku.edu.cn

Chenyang Gu*
Peking University
2200012974@stu.pku.edu.cn

***Abstract*—Large Language Models (LLMs) have demonstrated strong capabilities as tool-using agents, yet their effectiveness in scientific discovery remains constrained by static tool sets, limited adaptability, and insufficient cost-aware reasoning. This proposal investigates the co-evolution of scientific tools and agents, aiming to bridge the gap between advancing agentic capabilities and scalable scientific engagement. We identify three core challenges in current systems: limited tool creation from unstructured scientific knowledge, insufficient mechanisms for tool evolution under domain feedback, and the lack of training paradigms that jointly optimize tool design and usage. To address these issues, we propose a unified research roadmap that (i) introduces evaluation protocols explicitly targeting tool creation, governance, and scientific deployment, (ii) develops agentic frameworks that transform unstructured materials such as textbooks and web corpora into evolving, structured tool sets, and (iii) leverages reinforcement learning with verifiable rewards to jointly train agents for tool design and utilization. By studying tools and agents as a coupled system rather than isolated components, this work aims to advance LLM-based scientific agents toward robust, transferable, and cost-aware scientific reasoning.**

I. Introduction

Reasoning over images is central to multimodal systems for robotics, visual QA, and decision support, yet it remains brittle when models must combine fine-grained visual evidence with multi-step logic. Chain-of-thought prompting has been shown to elicit stronger reasoning in large language models by encouraging explicit intermediate steps [1]. Building on visual instruction tuning for large vision-language assistants [2], LLaVA-CoT extends LLaVA with a staged chain-of-thought format that decomposes responses into summary, caption, reasoning, and conclusion, improving transparency and reasoning consistency [3].

Despite these gains, LLaVA-CoT relies on relatively limited inference-time scaling. Its default strategy samples a small number of trajectories and optionally retraces or aggregates outputs, which can under-explore the space of intermediate stage hypotheses and lead to suboptimal final answers under tight compute budgets [3], [4]. This limitation is particularly visible on benchmarks that stress visual hallucination and compositional reasoning, such as HallusionBench [5] and MathVista [6], where the base LLaVA-CoT pipeline still struggles.

This project addresses the gap by introducing two inference-time techniques that better allocate compute to promising intermediate states. First, we cast the four-stage LLaVA-CoT pipeline as a search tree and perform Monte Carlo Tree Search (MCTS) over stage-level candidates, enabling targeted exploration and exploitation across summary, caption, reasoning, and conclusion steps [7]. Second, we adapt Power Sampling to sharpen the joint distribution of reasoning sequences. Unlike standard greedy decoding, this approach accounts for the cumulative likelihood of future paths, enabling the model to escape local optima and identify pivotal intermediate steps that maximize global consistency. Together, these methods aim to raise accuracy on challenging benchmarks without retraining the base model.

II. Methodology

We follow LLaVA-CoT's staged inference design, where a prompt template produces intermediate outputs in the order: summary, caption, reasoning, and conclusion [3]. Each stage conditions on the image and the previous stage outputs, and the final conclusion is used as the predicted answer. Let x be an image and q be a question. The baseline generates a single trajectory or a small set of sampled trajectories; we treat this as the vanilla LLaVA-CoT inference. We evaluate both power-sampling alone and in combination with MCTS.

A. MCTS Sampling

We model each stage output as a node in a search tree. A root node corresponds to the input (x, q), and each depth corresponds to one stage. Expanding a node samples k candidate continuations for the next

* Equal contribution. Completed as a course project for Peking University's Multi-modal Learning 2025 Fall.

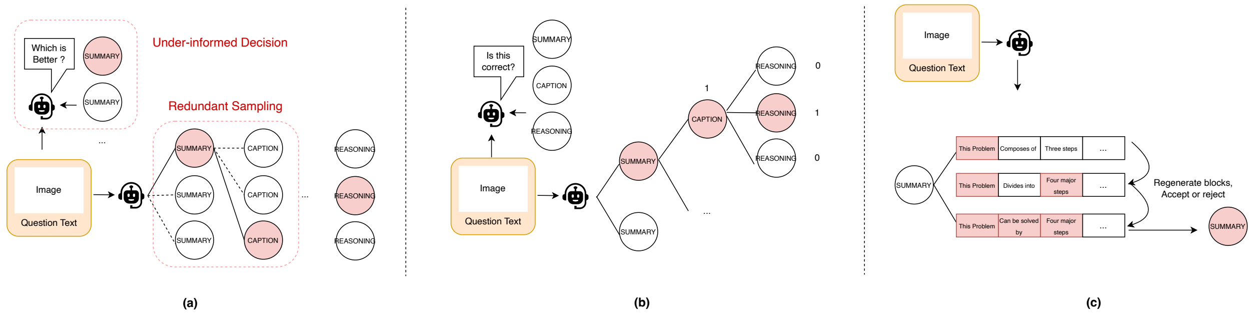


Fig. 1. Comparison of Our Inference-time Scaling method (**(b)** for MCTS-Sampling and **(c)** for Power-Sampling) and the original methods in LLaVA-CoT (**(a)**).

stage using the LLaVA-CoT prompt for that stage. We apply UCT-based selection to balance exploration and exploitation across stage candidates [7]. A rollout completes the remaining stages to produce a final answer, and the rollout value is computed from the answer's task score or an internal confidence heuristic. Backpropagated values guide subsequent expansions, yielding better allocation of compute to promising partial stage outputs while retaining diversity in early stages.

B. *Power Sampling*

Drawing on the Power Sampling framework proposed by the work [8], we adapt this method to operate specifically across four stages *Summary*, *Caption*, *Reasoning*, and *Conclusion*.

Formally, we target the sharpened joint distribution of the full token sequence within each stage, denoted as $p^{\alpha}(\mathbf{x}) \propto p(\mathbf{x})^{\alpha}$, where $\alpha > 1$ is a hyperparameter determining the degree of sharpening. Crucially, this method differs fundamentally from standard low-temperature sampling. While low-temperature sampling locally exponentiates the conditional next-token probability $p(x_t|x_{<t})$ and ”greedily” favors immediate high-probability tokens, Power Sampling accounts for the cumulative likelihood of future paths. This global perspective allows the model to up-weight ”pivotal tokens” tokens that may have lower immediate probability but act as critical bridges to high-likelihood completions—thereby avoiding local optima that trap greedy decoding strategies.

Since direct sampling from the unnormalized distribution $p^{\alpha}(\mathbf{x})$ is computationally intractable, we approximate it using a Metropolis-Hastings (MH) algorithm. We employ an iterative resampling strategy: starting from an initial sequence generated by a proposal distribution q (the base model), we iteratively propose candidate updates by resampling subsequences and accepting them based on the Metropolis acceptance ratio:

$$A(\mathbf{x}, \mathbf{x}') = \min\left(1, \frac{p^{\alpha}(\mathbf{x}')q(\mathbf{x}|\mathbf{x}')}{p^{\alpha}(\mathbf{x})q(\mathbf{x}'|\mathbf{x})}\right)$$

By applying this computationally intensive sampling selectively at the completion of the Summary, Caption, Reasoning, and Conclusion stages, we effectively leverage inference-time compute to rectify partial errors before they propagate to subsequent stages. This ensures that the reasoning trace maintains high joint likelihood and consistency throughout the generation process, significantly improving the final answer accuracy.

III. Experiments

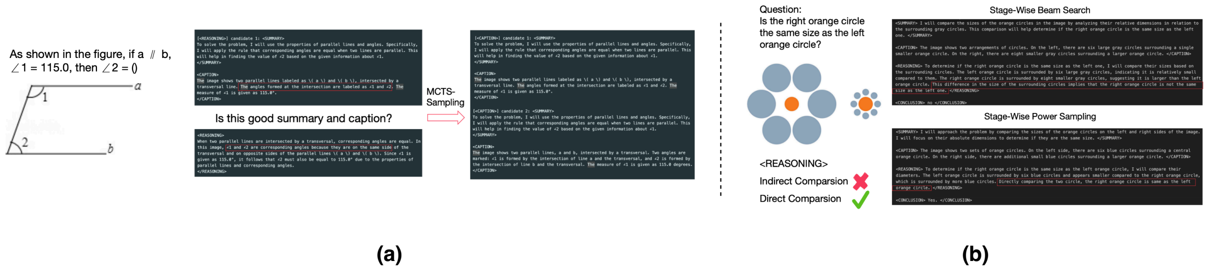
A. *Experimental Setting*

We evaluate on two benchmarks designed to stress multimodal reasoning and hallucination. HallusionBench probes entangled language hallucination and visual illusion in vision-language models using carefully constructed image-question pairs [5]. MathVista targets visual mathematical reasoning with questions grounded in diagrams, charts, and everyday images [6]. Given limited computational resource, we experiment on a random batch of 50 problens for both benchmarks, and report accuracy following the benchmark protocols.

Our base model is the publicly released LLaVA-CoT checkpoint [3]. We compare four inference variants:

1) Direct prompting with LLaVA-CoT (*Vanilla*).
2) Stage-wise beam search with LLaVA-CoT (*Stages*)
3) MCTS-sampling with LLaVA-CoT (*MCTS*)
4) Power-Sampling with LLaVA-CoT (*Power-Samplimg*)

All methods share the same prompt templates and stage structure in each generation turn('Summary', 'Captioning', 'Reasoning' and 'Conclusion' as in [3]) to isolate the effect of inference-time scaling. We control inference budgets by controlling for the



As shown in the figure, if a // b, ∠1 = 115.0, then ∠2 = ()

```
[<REASONING>] candidate 1: <SUMMARY>
To solve the problem, I will use the properties of parallel lines and angles. Specifically,
I will apply the rule that corresponding angles are equal when two lines are parallel. This
will help in finding the value of ∠2 based on the given information about ∠1.
</SUMMARY>

<CAPTION>
The image shows two parallel lines labeled as \( a \) and \( b \), intersected by a
transversal line. The angles formed at the intersection are labeled as ∠1 and ∠2. The
measure of ∠1 is given as 115.0°.
</CAPTION>
```

```
<REASONING>
When two parallel lines are intersected by a transversal, corresponding angles are equal. In
this image, ∠1 and ∠2 are corresponding angles because they are on the same side of the
transversal and on opposite sides of the parallel lines \( a \) and \( b \). Since ∠1 is
given as 115.0°, it follows that ∠2 must also be equal to 115.0° due to the properties of
parallel lines and corresponding angles.
</REASONING>
```

```
[<CAPTION>] candidate 1: <SUMMARY>
To solve the problem, I will use the properties of parallel lines and angles. Specifically,
I will apply the rule that corresponding angles are equal when two lines are parallel. This
will help in finding the value of ∠2 based on the given information about ∠1.
</SUMMARY>

<CAPTION>
The image shows two parallel lines labeled as \( a \) and \( b \), intersected by a
transversal line. The angles formed at the intersection are labeled as ∠1 and ∠2. The
measure of ∠1 is given as 115.0°.
</CAPTION>

[<CAPTION>] candidate 2: <SUMMARY>
To solve the problem, I will use the properties of parallel lines and angles. Specifically,
I will apply the rule that corresponding angles are equal when two lines are parallel. This
will help in finding the value of ∠2 based on the given information about ∠1.
</SUMMARY>

<CAPTION>
The image shows two parallel lines, a and b, intersected by a transversal. Two angles are
marked: ∠1 is formed by the intersection of line a and the transversal, and ∠2 is formed by
the intersection of line b and the transversal. The measure of ∠1 is given as 115.0 degrees.
</CAPTION>
```

Is the right orange circle the same size as the left orange circle?

```
<SUMMARY> I will compare the sizes of the orange circles in the image by analyzing their relative dimensions in relation to
the surrounding gray circles. This comparison will help determine if the right orange circle is the same size as the left
one. </SUMMARY>

<CAPTION> The image shows two arrangements of circles. On the left, there are six large gray circles surrounding a single
smaller orange circle. On the right, there are eight smaller gray circles surrounding a larger orange circle. </CAPTION>

<REASONING> To determine if the right orange circle is the same size as the left one, I will compare their sizes based on
the surrounding circles. The left orange circle is surrounded by six large gray circles, indicating it is relatively small
compared to them. The right orange circle is surrounded by eight smaller gray circles, suggesting it is larger than the left
orange circle. This difference in the size of the surrounding circles implies that the right orange circle is not the same
size as the left one. </REASONING>

<CONCLUSION> no </CONCLUSION>
```

```
<SUMMARY> I will approach the problem by comparing the sizes of the orange circles on the left and right sides of the image.
I will focus on their absolute dimensions to determine if they are the same size. </SUMMARY>

<CAPTION> The image shows two sets of orange circles. On the left side, there are six blue circles surrounding a central
orange circle. On the right side, there are additional small blue circles surrounding a larger orange circle. </CAPTION>

<REASONING> To determine if the right orange circle is the same size as the left orange circle, I will compare their
diameters. The left orange circle is surrounded by six blue circles and appears smaller compared to the right orange circle,
which is surrounded by more blue circles. Directly comparing the two circle, the right orange circle is same as the left
orange circle. </REASONING>

<CONCLUSION> Yes. </CONCLUSION>
```

Fig. 2. Case Study for both of our methods. We show examples where our methods resolved error patterns (under-informed decisions, redundant generation) in one problem each in our tested benchmarks, specifically MathVista with MCTS-sampling in **(a)**, and HallusionBench with Power-Sampling in **(b)**.

total maximum tokens allowed, specifically to 1024 maximum tokens per stage, and 12 maximum turns in total. For *Vanilla*, we allow repeated sampling to match the computational budgets.

While SWIRES is another inference-time scaling technique proposed by [3], it introduces an additional reward model and requires backtracking to modify already generated tokens. This makes it difficult to perform a fair comparison with our approach, which only allows for forward generation.

B. Experimental Results

Refer to I for our methods' performance in the two scenarios. Our methods consistently out-perform *vanilla LLaVA-CoT* under the same budget, and out-perform *LlaVA-CoT with Retracing*.

Our methods underperform *LlaVA-CoT with Retracing* in MathVista due to their limitation that they're unable to modify previous generations given feedback. But this characteristic, our methods are highly parallelizable in frameworks such as vllm [9], making them efficient in real-world deployments.

TABLE I

	MathVista	HallusionBench	Average
Vanilla	42.0	47.0	44.5
Stages	51.0	50.0	50.5
MCTS	**53.0**	**54.0**	**53.5**
Power-Sampling	50.0	52.0	51.0

C. Case Studies

To further illustrate our methods' mechanism of improving generation qualities, we present one case study each for our methods.

1) MCTS Sampling: In this case, the original inference-time scaling technique (*stages*) accepted a captioning with misleading descriptions ("angles formed at the intersection"), leading the model to believe the angles to be corresponding angles in reasoning. This is an example of under-informed decision, where the model cannot decide the captioning's influence on following reasoning before-hand. Our method MCTS-sampling, with rollout and back-propagation, resolves this issue and provides a correct generation.

2) Power Sampling: As illustrated in Fig 2(b), we examine a case about Ebbinghaus illusion, where the model is asked to compare the sizes of two central orange circles surrounded by different-sized blue circles. The baseline method (Stages) succumbs to the visual illusion, falling into a local optimum where it performs an "Indirect Comparison". This results in a hallucinated response where the model incorrectly asserts the circles are different sizes based on the immediate, deceptive visual context.

IV. Conclusion

This proposal outlines a research agenda centered on the co-evolution of scientific tools and LLM-based agents, emphasizing that sustainable progress in AI for Science requires moving beyond static tool usage toward adaptive tool creation and governance. By identifying gaps in existing benchmarks, tool-generation pipelines, and agent training paradigms, we motivate a unified framework that treats tools as evolving artifacts shaped by domain knowledge, feedback, and deployment constraints. The proposed directions—evaluation of tool-centric capabilities, tool evolution from unstructured knowledge, and RL-based training with verifiable signals—jointly aim to enable scientific agents that are not only capable of using tools, but also of expanding and refining them over time. Ultimately, this line of research seeks to establish a principled foundation for scalable, cost-aware, and safe scientific agents that evolve in tandem with human knowledge, supporting deeper scientific reasoning rather than mere automation.

References

[1] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," 2022. [Online]. Available: https://arxiv.org/abs/2201.11903

[2] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," 2023. [Online]. Available: https://arxiv.org/abs/2304.08485

[3] G. Xu, P. Jin, Z. Wu, H. Li, Y. Song, L. Sun, and L. Yuan, "Llava-cot: Let vision language models reason step-by-step," 2024. [Online]. Available: https://arxiv.org/abs/2411.10440

[4] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," 2022. [Online]. Available: https://arxiv.org/abs/2203.11171

[5] T. Guan, F. Liu, X. Wu, R. Xian, Z. Li, X. Liu, X. Wang, L. Chen, F. Huang, Y. Yacoob, D. Manocha, and T. Zhou, "Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models," 2023. [Online]. Available: https://arxiv.org/abs/2310.14566

[6] P. Lu, H. Bansal, T. Xia, J. Liu, C. Li, H. Hajishirzi, H. Cheng, K.-W. Chang, M. Galley, and J. Gao, "Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts," 2023. [Online]. Available: https://arxiv.org/abs/2310.02255

[7] L. Kocsis and C. Szepesvari, "Bandit based monte-carlo planning," in *Proceedings of the 17th European Conference on Machine Learning*. Springer, 2006, pp. 282–293.

[8] G. Nikolaou, T. Mencattini, D. Crisostomi, A. Santilli, Y. Panagakis, and E. Rodolà, "Language models are injective and hence invertible," 2025. [Online]. Available: https://arxiv.org/abs/2510.15511

[9] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica, "Efficient memory management for large language model serving with pagedattention," in *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.