Chain of thought Prompting + LLAVA COT

Chain of Thought (CoT) is a reasoning mechanism in which a large language model (LLM) breaks down a task into a series of intermediate steps before arriving at a final solution.

Why is CoT Important?

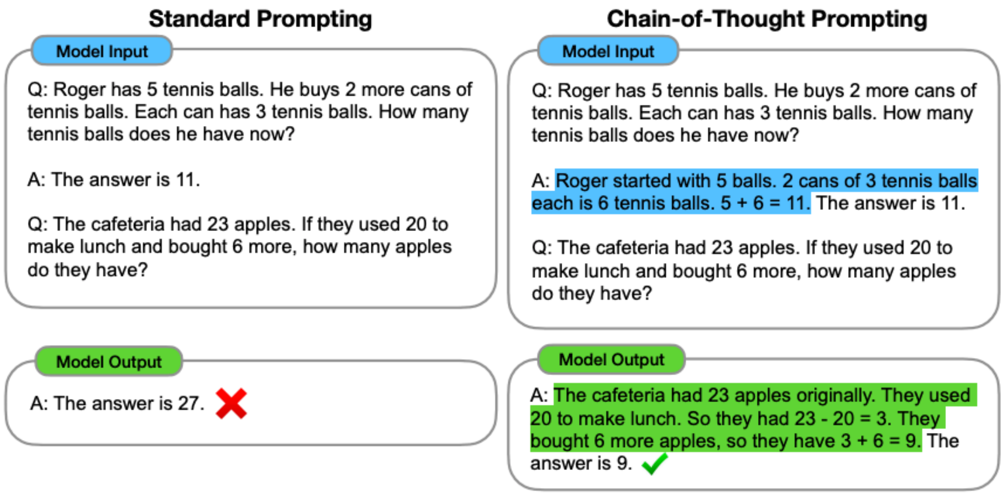
- **Improves performance on complex tasks**: Literature shows that LLMs equipped with CoT capabilities perform significantly better on problems that require multi-step reasoning.
- **Natural language rationales**: CoT enables LLMs to reason through problems in a human-like way, producing step-by-step explanations that are interpretable and insightful.

Limitation of Default LLMs

- **Instinctive responses**: They typically respond with direct answers rather than engaging in a reasoning process.

Since LLMs do not naturally engage in CoT reasoning, researchers have explored strategies to incorporate it effectively:

- ▼ **Few-Shot Prompting (In-Context Learning)**
  - Embeds example reasoning chains directly within the prompt to guide the model's response style.
  - Default few shot prompting would not work. The prompting should be done in a COT manner. Example below



▼ More COT prompt examples

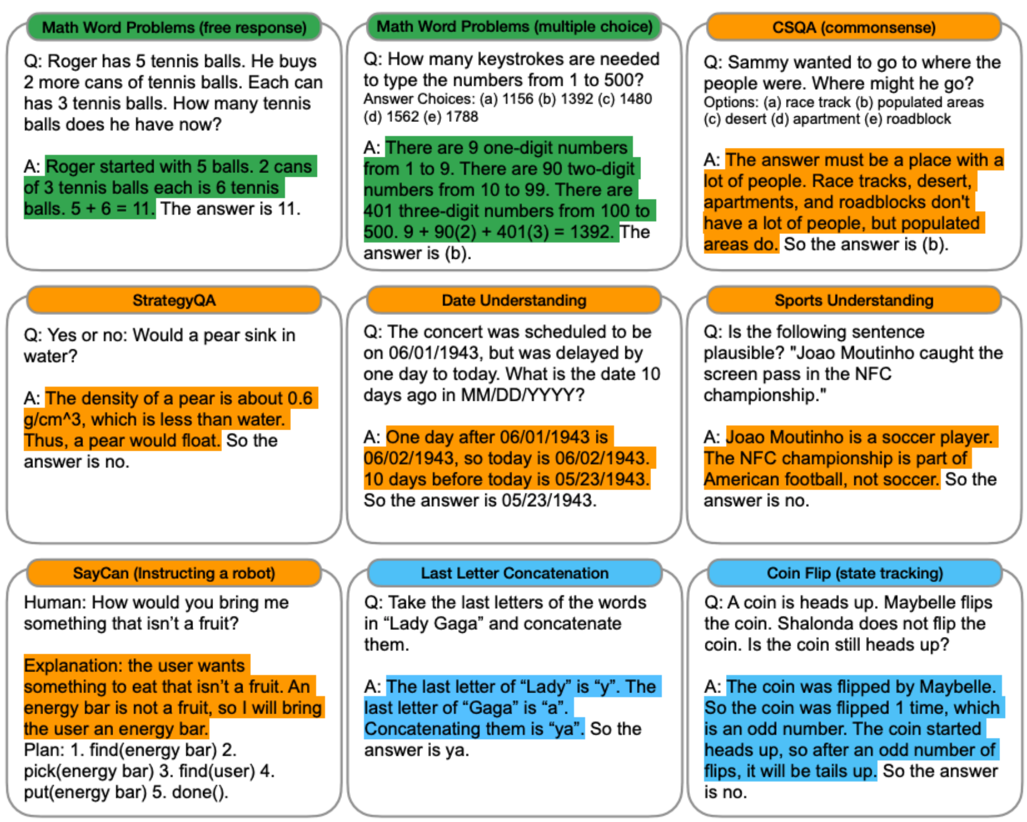
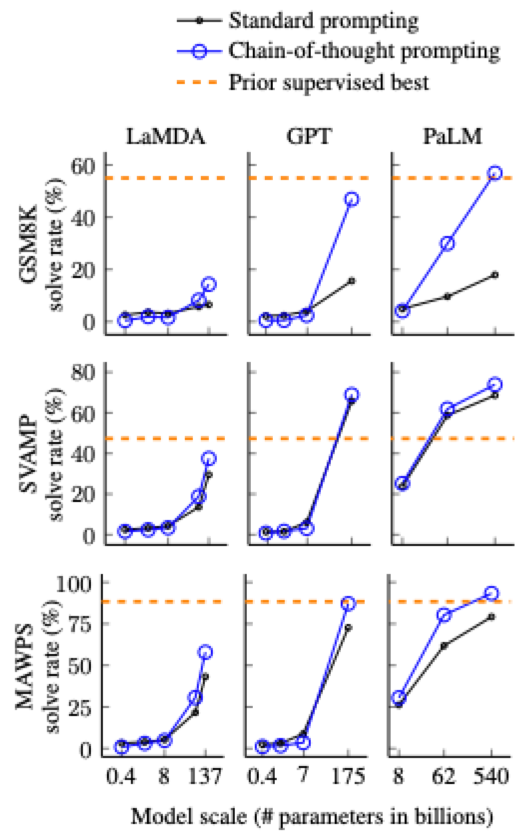
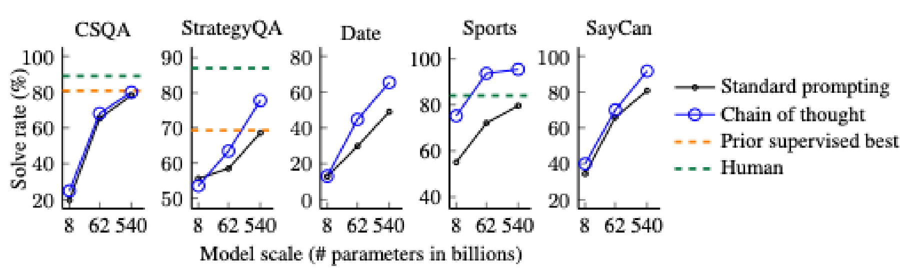


Figure 3: Examples of ⟨input, chain of thought, output⟩ triples for arithmetic, commonsense, and

▼ Arithmetic reasoning



▼ Common Sense Reasoning



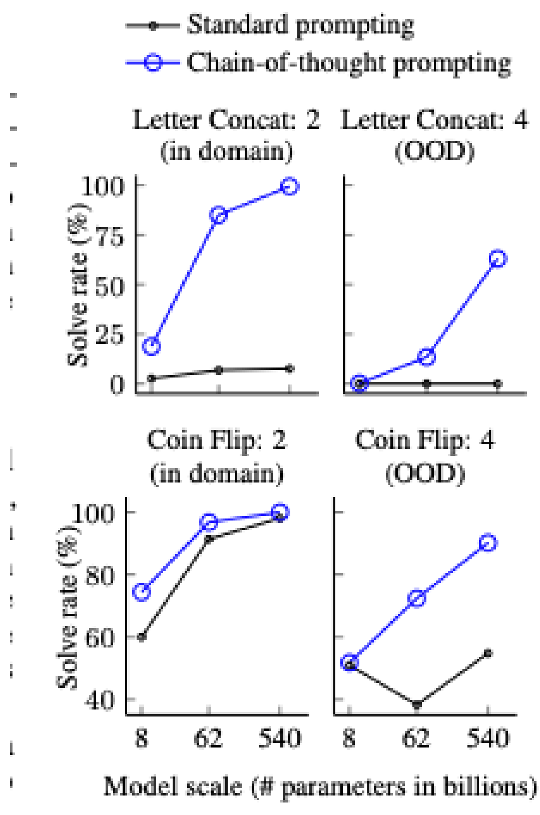
▼ Symbolic reasoning

▼ CoinFlip Example

"A coin is heads up. Phoebe flips the coin.
Osvaldo does not flip the coin. Is the coin still heads up?"
→ "no"

▼ Letter Concatenation:

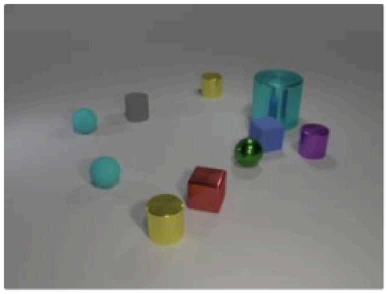
"Amy Brown" → "yn"



Vision-Language Models often exhibit limited Chain-of-Thought (CoT) reasoning capabilities. This limitation is stems from their training methodologies, which primarily involve learning from image-caption pairs, or image-one word answer pairs. Such training focuses on direct associations between images and textual descriptions, potentially neglecting the development of intermediate reasoning steps. Consequently, VLMs may struggle with tasks requiring complex, step-by-step reasoning.

Example of VLM struggling:

Example 1: Reasoning Problems



Question: Subtract all tiny shiny balls. Subtract all purple objects. How many objects are left?

Options: A. 4, B. 8, C. 2, D. 6

Answer from Llama-3.2-11B-Vision-Instruct:

To determine the correct answer, we need to follow the instructions provided.

The first instruction is to subtract all tiny shiny balls.
Since there are **three tiny shiny balls** in the image, we would be left with no shiny balls after performing this action.
The second instruction is to subtract all purple objects.
However, upon closer inspection, it becomes apparent that there are **no shiny purple objects present in the image**.
Therefore, we cannot subtract any purple objects.

Given that we have already removed all the tiny shiny balls and there were no purple objects to begin with, the correct answer is C. 2, as initially, **there were 2 purple objects, but they are not to be subtracted** according to the instruction.

The authors of LLAVA-CoT observed the following issues with VLMs:

- VLMs often initiate responses without adequately organizing the problem and the available information.
- They ofter deviate from logical reasoning towards conclusions.

Another paper that talks about this issue : Improve Vision Language Model Chain of thought reasoning : Mention that the training of VLMs mostly doesn't support CoT implicitly through prompting (few shot).

To address the limitations in reasoning, the authors propose **LLaVA-CoT**, specifically designed to perform inherent Chain-of-Thought (CoT) reasoning for vision-language tasks.

LLaVA-CoT is trained to generate reasoning outputs in four distinct stages, each serving a specific role in the thought process:

1. **Summary**: Provides a brief outline of the task at hand.
2. **Caption**: Describes the relevant parts of the image (if present), focusing on aspects connected to the question.
3. **Reasoning**: Offers a detailed, systematic analysis of the problem.
4. **Conclusion**: Presents a final answer derived from the reasoning.

Each of these stages are invoked at model's discretion.

To achieve this, they basically train the VLM with reasoning rationales, instead of just image caption pairs

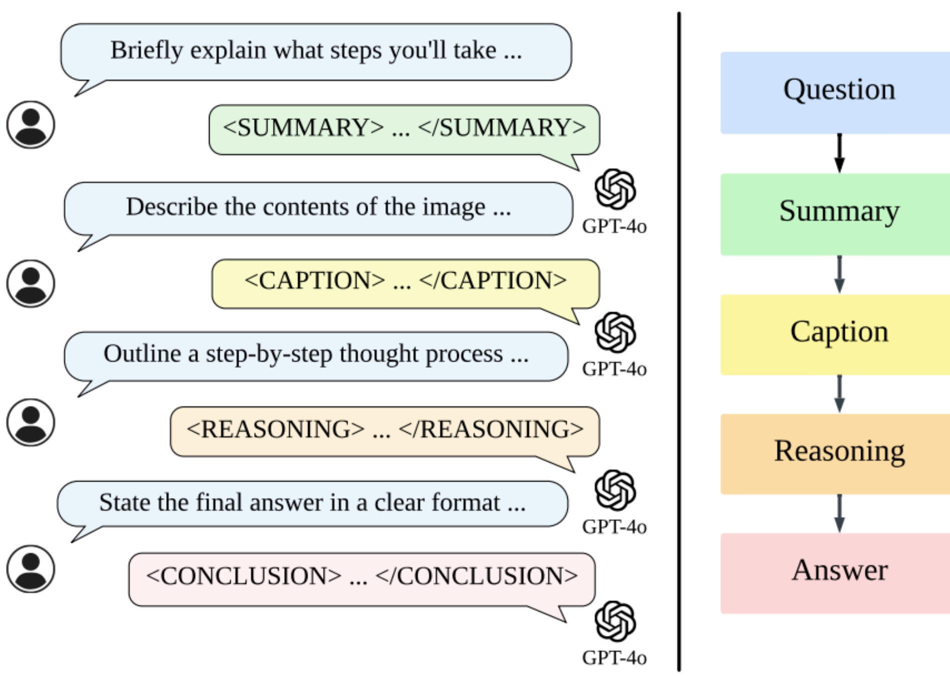
They construct a dataset named **LLaVA-CoT-100k**, containing samples generated stage-by-stage using GPT-4o.

Each stage is explicitly marked using a dedicated tag. Eg <summary></summary>

But to do so, they would need to train the model with a dataset that has this structure to it. They generate it using GPT-4o. They provide an image to GPT and ask it to create different stages explicitly through prompting.

> Do not like this - have seen a lot. Could end up conditioning the model to GPT 4o

Data Generation example



LLAVA-COT-100k

Dataset	Type	Size
ShareGPT4V [9]	General VQA	31.3k
ChartQA [40]	General VQA	17.2k
A-OKVQA [48]	General VQA	16.1k
AI2D [25]	Science-Targeted VQA	11.4k
GeoQA+ [7]	Science-Targeted VQA	11.4k
ScienceQA [36]	Science-Targeted VQA	5.6k
DocVQA [41]	General VQA	4.0k
PISC [30]	General VQA	1.0k
CLEVR [24]	General VQA	0.5k
CLEVR-Math [14]	Science-Targeted VQA	0.5k

▼ Dataset descriptions

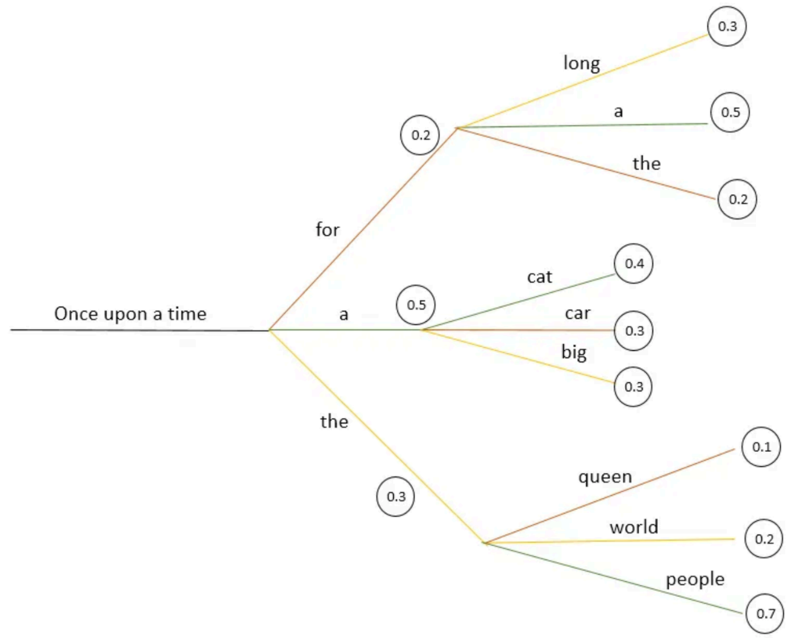
- **ShareGPT4V**: Multi-turn question-answering data sourced from GPT-4V interactions, representing realistic and complex user prompts.
- **ChartQA**: Focused on interpreting charts and graphs, requiring the model to extract and reason with visual data.
- **A-OKVQA**: Emphasis external knowledge reasoning, questions often require going beyond what is directly visible in the image.
- **DocVQA**: Involves document-based question answering, requiring textual comprehension of structured documents.
- **PISC**: Designed to assess understanding of social relationships in visual scenes.
- **CLEVR**: Targets object properties, spatial reasoning, and counting.
- **GeoQA+**: Focuses on **geometric reasoning** in visual tasks.
- **AI2D** and **ScienceQA**: Designed for answering science-based questions.

Introduce Stage level beam search:

How LLMs typically generate a output?

- beam search is one of many ways. Others include greedy etc.,

Example illustration of how beam search works: (beam set to 3)



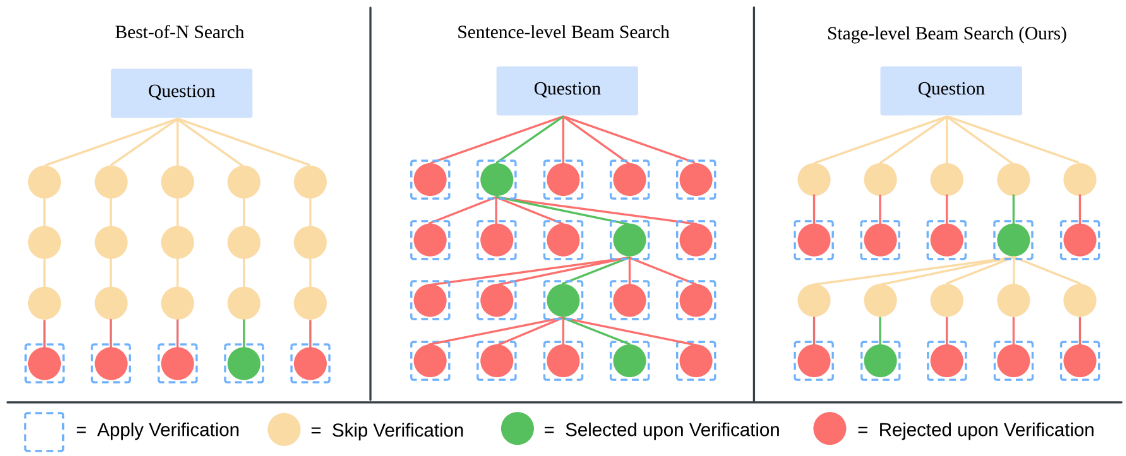
Beam search algorithm [2]

https://medium.com/@javaid.nabi/all-you-need-to-know-about-llm-text-generation-03b138e0ed19

Sentence level beam search : You apply beam search or any other technique to generate N sentences. Post generating the N sentences, pick that best sentence.

Best of N search : Here you allow the LLM to generate N candidate outputs and then use a reward model to decide which output is desirable. The reward model is typically trained using human feedback. I assume they used the base rewards model that came with the LLM for the purpose of the paper. They do not explicitly talk about it.

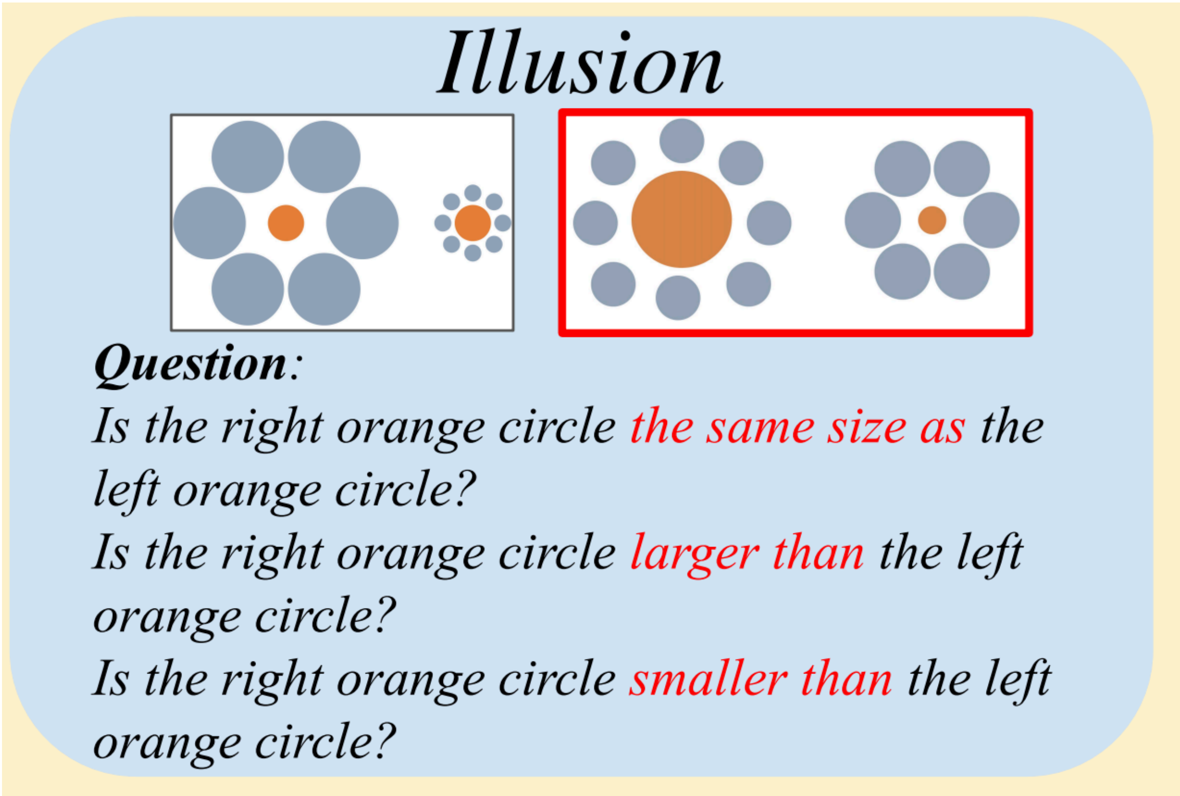
In the paper, they introduce a stage level beam search. In this, they generate N candidates for each stage and then select the best candidate using VLM itself. Do it for all the stages to end up with final output.



Impressions

They evaluate their model on different benchmarks. The MMStar, MMBench and MMVet are the ones that do regular VQA, where MMVet is the trickier one. MathVista and AI2D do math and science benchmarking respectively. Hallusion is to test for hallucinations

- ▼ Hallusion Example (from here)



They show that their method has a better average performance on these benchmark. And best performance on most of the benchmarks.

Model	MMStar	MMBench	MMVet	MathVista	AI2D	Hallusion	Average
Base Model							
Llama-3.2-11B-Vision-Instruct	49.8	65.8	57.6	48.6	77.3	40.3	56.6
Our Models							
LLaVA-CoT (with Direct Training)	54.3	76.2	49.9	49.5	81.2	42.9	59.0
LLaVA-CoT (w/o Structured Tags)	55.7	74.2	57.0	54.1	79.1	45.0	60.9
LLaVA-CoT	57.6	75.0	60.3	54.8	78.7	47.8	62.4

Furthermore they take the results from MMStar benchmark and divide them based on subtasks and show that their approach help improve significantly on the tasks requiring strong reasoning capabilities.

Model	CP	FP	IR	LR	Math	Science & Technology	Average
Base Model							
Llama-3.2-11B-Vision-Instruct	66.0	46.4	57.6	50.8	45.2	32.8	49.8
Our Models							
LLaVA-CoT (with Direct Training)	68.4	48.0	65.6	52.0	51.6	40.0	54.3
LLaVA-CoT (w/o Structured Tags)	68.4	48.0	60.0	55.2	64.4	38.0	55.7
LLaVA-CoT	68.8	46.8	63.2	58.0	64.0	44.8	57.6

Table 3. **Performance of different models on the MMStar benchmark across various skill areas.** Here, CP represents coarse perception, FP represents fine-grained perception, IR represents instance reasoning, and LR represents logical reasoning. As shown in the table, our model demonstrates substantial improvement over the base model in instance reasoning, logical reasoning, math, and science & technology, indicating that structured reasoning can significantly enhance the model's reasoning capabilities.

Stage level beam search :

Model	MMStar	MMBench	MMVet	MathVista	AI2D	Hallusion	Average
Base Model							
Llama-3.2-11B-Vision-Instruct	49.8	65.8	57.6	48.6	77.3	40.3	56.6
Our Models							
LLaVA-CoT	57.6	75.0	60.3	54.8	78.7	47.8	62.4
LLaVA-CoT(BS = 2)	58.1	75.6	61.7	56.1	78.8	48.2	63.1

Comparison between different methods of inference scaling:

Here the criteria of comparison is the underlying compute resources.

Method	Number of Beam	MMVet Score
No Inference Scaling	1	60.3
Best-of-N Search	10	60.9
Sentence-level Beam Search	2	58.4
Stage-level Beam Search	4	62.9

Method	Number of Beam	MMVet Score
No Inference Scaling	1	60.3
Stage-level Beam Search	2	61.7
Stage-level Beam Search	3	62.3
Stage-level Beam Search	4	62.9