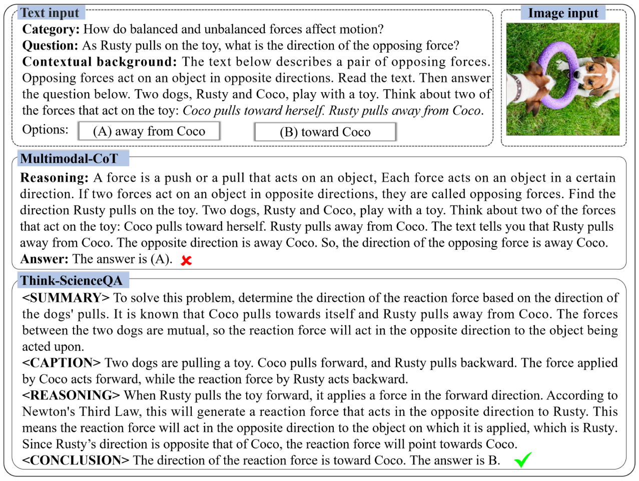
Let’s Think During Reasoning: Adapting Stage-wise Self-consistent Plan-and-Solve for ScienceQA

Anonymous submission

Abstract

Chain-of-Thought (CoT) prompting enhances reasoning performance but remains inherently limited by missing steps and semantic misunderstanding errors. Moreover, existing approaches struggle to effectively integrate multi-modal information and maintain coherent reasoning, limiting their performance on complex multi-modal tasks in ScienceQA. To address the above challenges, this paper proposes Think-ScienceQA, a novel stage-wise plan-and-solve self-consistent reasoning framework guided by a semantic adapter, which explicitly thinks through the entire generation process in ScienceQA tasks. Unlike previous multi-path CoT methods that focus on generating diverse reasoning trajectories, Think-ScienceQA introduces a word-level consistency voting mechanism that operates across multiple candidate paths during the reasoning generation process, aiming to construct a more coherent and reliable rationale. In addition, a lightweight semantic adapter is designed to bridge the semantic gap between frozen visual encoders and language models, enabling more precise and efficient multi-modal feature alignment. Despite having only 228M parameters, approximately 6% the size of the strongest fine-tuned baseline, Think-ScienceQA achieves state-of-the-art performance on both ScienceQA and A-OKVQA benchmarks. Extensive ablation studies demonstrate that the stage-wise reasoning design and self-consistency mechanism contribute performance gains of 6.7 and 4.1 percentage points, respectively.

code — https://think-scienceqa.github.io/


Text input

Category: How do balanced and unbalanced forces affect motion?
Question: As Rusty pulls on the toy, what is the direction of the opposing force?
Contextual background: The text below describes a pair of opposing forces. Opposing forces act on an object in opposite directions. Read the text. Then answer the question below. Two dogs, Rusty and Coco, play with a toy. Think about two of the forces that act on the toy: *Coco pulls toward herself. Rusty pulls away from Coco.*
Options: (A) away from Coco (B) toward Coco

Image input

Multimodal-CoT

Reasoning: A force is a push or a pull that acts on an object, Each force acts on an object in a certain direction. If two forces act on an object in opposite directions, they are called opposing forces. Find the direction Rusty pulls on the toy. Two dogs, Rusty and Coco, play with a toy. Think about two of the forces that act on the toy: Coco pulls toward herself. Rusty pulls away from Coco. The text tells you that Rusty pulls away from Coco. The opposite direction is away Coco. So, the direction of the opposing force is away Coco.
Answer: The answer is (A). ✗

Think-ScienceQA

<SUMMARY> To solve this problem, determine the direction of the reaction force based on the direction of the dogs' pulls. It is known that Coco pulls towards itself and Rusty pulls away from Coco. The forces between the two dogs are mutual, so the reaction force will act in the opposite direction to the object being acted upon.
<CAPTION> Two dogs are pulling a toy. Coco pulls forward, and Rusty pulls backward. The force applied by Coco acts forward, while the reaction force by Rusty acts backward.
<REASONING> When Rusty pulls the toy forward, it applies a force in the forward direction. According to Newton's Third Law, this will generate a reaction force that acts in the opposite direction to Rusty. This means the reaction force will act in the opposite direction to the object on which it is applied, which is Rusty. Since Rusty’s direction is opposite that of Coco, the reaction force will point towards Coco.
<CONCLUSION> The direction of the reaction force is toward Coco. The answer is B. ✓

Figure 1: An example compares the baseline method Multimodal-CoT and the proposed Think-ScienceQA in generating rationales and predicting answers. Despite the contextual cue that *“Coco pulls toward herself, while Rusty pulls away from Coco”*, Multimodal-CoT incorrectly predicts the answer as “(A) away from Coco” when asked about the direction of the opposing force on Rusty, indicating a hallucination error.. In contrast, Think-ScienceQA effectively mitigates such errors through planned stage-wise reasoning.

Introduction

In recent years, Large Language Models (LLMs) have made remarkable progress in multi-modal reasoning by effectively bridging the semantic gap between visual and textual modalities in Science Question Answering (ScienceQA) tasks (Horawalavithana et al. 2024; Chen et al. 2023; Lu et al. 2022). A key development in this area is Chain-of-Thought (CoT) prompting (Lu et al. 2022), which enables models to generate the reasoning step, thereby enhancing both interpretability and prediction accuracy. Despite its promise, most existing CoT-based methods (Wei et al. 2022) rely on a single reasoning trajectory, typically produced through greedy decoding or beam search. This single-path approach fails to capture the diversity of cognitive strategies that humans naturally employ when solving complex problems. Although recent efforts, such as Multi-modal CoT (Zhang et al. 2024) and LLaMA-Adapter (Horawalavithana et al. 2024), have attempted to address these limitations, they still struggle to maintain the integrity of multi-modal information integration and to ensure coherent reasoning. As a result, these models often converge on suboptimal solutions, particularly when confronted with ambiguous or uncertain queries.

A major cause of suboptimal or incorrect outputs in LLMs is the hallucination problem, which refers to the generation of plausible-sounding but factually inaccurate or fabricated content. This often results from an overreliance on prior knowledge at the expense of input-grounded evidence, or from oversimplified reasoning processes that fail to incorporate essential contextual information. For example, as shown in Figure 1, although the context clearly states that *“Coco pulls toward herself, while Rusty pulls away from*

Coco", the model incorrectly selects *"(A) away from Coco"* in response to the question *"What is the direction of the opposing force when Rusty pulls the toy?"*. This reflects a semantic misalignment between the visual and textual inputs and reveals a misunderstanding of basic physical principles. In contrast, the proposed framework addresses the above challenge through a planned stage-wise reasoning process, improving accuracy and obtaining the correct answer. This example underscores a key limitation of current CoT-based methods, the absence of intermediate step verification, often leading to incoherent and inaccurate final answers.

The proposed framework, Think-ScienceQA, is designed to *think* stage-wise to achieve reasoning consistency within the plan-and-solve process for ScienceQA. Inspired by the plan-and-solve prompting (Wang et al. 2023a), Think-ScienceQA decomposes the overall reasoning procedure into a sequence of planned stages: question summarization, image description, logical reasoning, and final conclusion. Unlike existing CoT approaches that follow a single reasoning path or an optimal answer selection from multiple paths, Think-ScienceQA leverages a stage-wise self-consistency voting mechanism to dynamically select the most reliable intermediate outcomes for generating optimal rationale. A key component of the proposed framework is a novel semantic adapter, which plays a pivotal role in bridging the gap between frozen visual encoders and language models. This adapter significantly enhances the model's visual understanding by enabling precise extraction and integration of critical information from visual inputs. Moreover, it facilitates flexible and fine-grained multi-modal feature fusion, thereby strengthening semantic alignment across modalities. This design not only improves the depth and accuracy of multi-modal reasoning but also supports the identification and correction of potential reasoning errors throughout the process.

Specifically, this paper makes three key contributions:

- This paper presents Think-ScienceQA, a novel framework that combines a lightweight semantic adapter with a self-consistency plan-and-solve to support stage-wise reasoning with optimal rationale generation. Unlike existing CoT approaches that either follow a single reasoning path or select the final answer from multiple complete paths, Think-ScienceQA introduces a novel stage-wise self-consistency voting mechanism that dynamically identifies the most reliable intermediate steps at each stage, enabling the generation of an optimal and coherent rationale.
- The proposed semantic adapter module enables fine-grained alignment and fusion of visual and textual features, thereby strengthening multi-modal representation learning. In the reasoning generation process, the semantic adapter guides the stage-wise self-consistent plan-and-solve scheme by selecting the most trustworthy reasoning from multiple intermediate candidates. In the answer inference stage, it further assists the model in synthesizing intermediate results to derive the final answer.
- In terms of efficiency and performance, Think-ScienceQA strikes an effective balance between model size and accuracy. On the ScienceQA dataset, it achieves 93.86% accuracy with only 228M parameters, just 1.75% of the baseline model's size, while outperforming several substantially larger models. On the A-OKVQA dataset, it attains 61.7% direct-answer accuracy, exceeding mainstream baselines, while using only 2.07% of the baseline's parameters.

Related Work

Reasoning Optimization for ScienceQA. Recent efforts have focused on improving reasoning efficiency in ScienceQA by leveraging prompt engineering and lightweight model architectures, aiming to reduce computational overhead while maintaining strong performance (Chen et al. 2023). CoT prompting (Lu et al. 2022) introduces step-by-step reasoning but often suffers from inconsistency in intermediate steps. To address this, SC-CoT (Wang et al. 2023b) employs multi-path decoding with majority voting, achieving a reduction in error propagation. MC-CoT (Tan et al. 2024) further enhances robustness by jointly voting across multiple rationales and answers, enabling smaller models to rival larger counterparts. Additionally, BLIP-2 (Li et al. 2023) shows that using frozen vision-language models with lightweight adapters can outperform fully end-to-end training, underscoring the parameter efficiency of modular architectures.

Multi-modal Reasoning. In multi-modal question answering, a range of reasoning frameworks (Lu et al. 2022) have been proposed to facilitate effective cross-modal integration. Early approaches such as ViLBERT (Lu et al. 2019) adopt dual-stream architectures with cross-modal attention but are limited by the use of frozen encoders, which restrict deep interaction between modalities. Multimodal-CoT (Zhang et al. 2024) introduces explicit reasoning chains through a two-stage training paradigm, yet it incurs high annotation costs (e.g., 83.4 hours for ScienceQA). More recently, C-Abstractor from the Honeybee framework (Horawalavithana et al. 2024) integrates deformable convolutions and geometric priors to improve symbolic reasoning.

Structured Representation and Inference. To better support complex reasoning tasks, structured representations (Marino et al. 2021) are increasingly utilized to organize intermediate reasoning steps and enhance interpretability. For example, scene graph-based methods (Liang, Jiang, and Liu 2021) improve spatial reasoning capabilities, although their scalability remains limited in cluttered visual scenes. Instruction-tuned models like GPT-4o (OpenAI 2023) demonstrate the ability to generate structured rationales for 89.7% of multi-step questions via CoT distillation. Similarly, LLaVA-CoT (Xu et al. 2024) introduces stage-level beam search on its LLaVA-CoT-100k dataset to achieve scalable and interpretable inference across vision-language tasks.

Limitations of Existing Approaches. Despite notable advancements in ScienceQA, existing methods still face several critical limitations. First, most current approaches rely on single-path reasoning chains, which lack structured guidance and error correction mechanisms. This often results in error accumulation during multi-step reasoning. Second, the

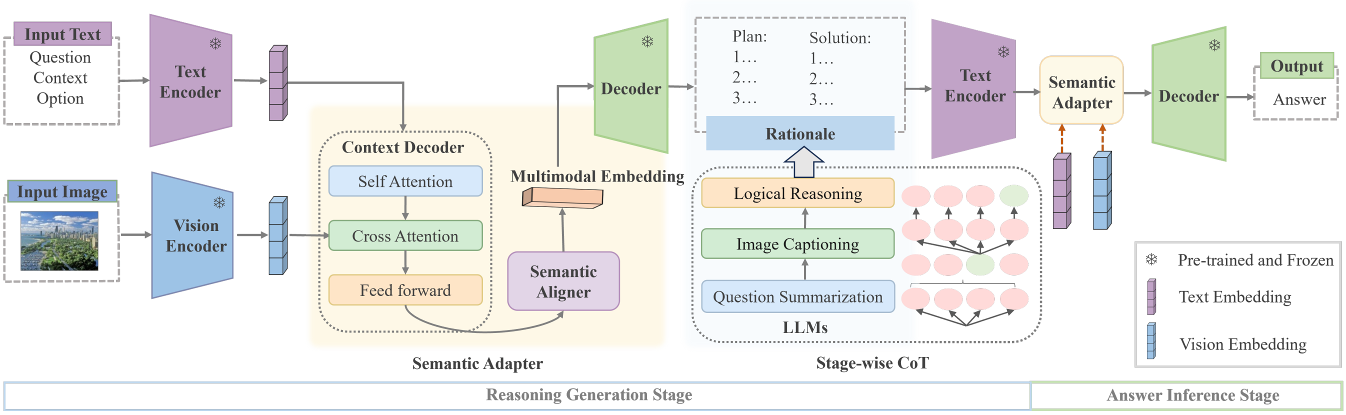


Figure 2: Overview of the Think-ScienceQA framework. The framework aligns textual and visual modalities using a lightweight semantic adapter. A Stage-wise CoT mechanism guides the generation of optimal rationales. A two-stage fine-tuning strategy further instructs the LLM to first produce structured reasoning, followed by final answer inference.

alignment between modalities remains suboptimal, with a risk of semantic pseudo-alignment between visual and textual inputs. To address these challenges, we propose Think-ScienceQA, a novel framework that incorporates a planned stage-wise reasoning strategy. It enhances both the stability of the reasoning process and the consistency of cross-modal semantics through stage-wise self-consistency voting. Furthermore, a semantic adapter module is introduced to facilitate fine-grained alignment and fusion of visual and textual features, thereby strengthening multi-modal representations.

Methodology

As shown in Figure 2, Think-ScienceQA adopts a planned stage-wise reasoning framework augmented with a semantic adapter. The model first leverages pre-trained encoders to extract modality-specific embeddings from both text and images. These embeddings are then fused within the semantic adapter using a context decoder equipped with self-attention and cross-attention mechanisms. A semantic alignment module is further employed to enhance cross-modal consistency. Guided by LLMs, the model generates structured plan-and-solve reasoning paths that span multiple subtasks, question summarization, image description, logical reasoning. Then, a stage-wise self-consistent voting mechanism is introduced to select the most reliable rationales. Finally, the semantic adapter and decoder collaboratively produce the conclusion and final answer.

Pre-training Semantic Adapter

The semantic adapter is a trainable module designed to bridge the gap between frozen image encoders and LLMs. It comprises two key components. The first is a context decoder, constructed using self-attention, cross-attention, and feedforward layers, which captures textual context and guides attention toward relevant visual information, facilitating effective multi-modal fusion. The second component is a semantic aligner, implemented as a linear transformation layer, which refines and aligns visual and textual semantic representations to enhance cross-modal consistency. During pre-training, the semantic adapter is combined with a frozen image encoder and a frozen language model and trained on paired image-text data. The context decoder receives both text and visual embeddings as input and integrates them through a Transformer decoder to model cross-modal interactions.

Given an input image I_v, a pre-trained visual encoder is employed to extract visual features $f_v = \mathcal{E}_V(I_v)$. For the input question Q, context C, and candidate options P, the textual input is defined as $I_t = \{Q, C, P\}$. A text encoder is then used to obtain textual features $f_t = \mathcal{E}_T(I_t)$. Then, the context decoder projects and fuses textual embeddings with frozen visual features into a shared representation space through self-attention and cross-attention mechanisms, producing the fused features, which is formulated as:

$$f_{\text{CD}} = \mathcal{D}_{\text{Context}}\big((f_t, f_v), \theta\big), \tag{1}$$

where θ denotes the trainable parameters of the context decoder $\mathcal{D}_{\text{Context}}$.

Subsequently, a semantic aligner is applied to refine and align the fused representation, enhancing cross-modal consistency. The final multi-modal features are given by:

$$f_{\text{SA}} = \mathcal{A}_{\text{Semantic}}(f_t, f_{\text{CD}}) + f_{\text{CD}}. \tag{2}$$

The refined features f_{SA} are then passed to the decoder of a language model to generate the output sequence $\mathbf{O}$.

The semantic adapter is pre-trained using an autoregressive objective. The probability of generating a target sequence $\mathbf{O} = (O_1, \dots, O_L)$ of length L is defined as:

$$\mathcal{P}_\phi(\mathbf{O} \mid I_v, I_t) = \prod_{i=1}^{L} \mathcal{P}_\phi(O_i \mid I_v, I_t, O_{<i}), \tag{3}$$

where ϕ denotes the parameters of the semantic adapter during pre-training, and $O_{<i}$ represents the previously generated tokens.

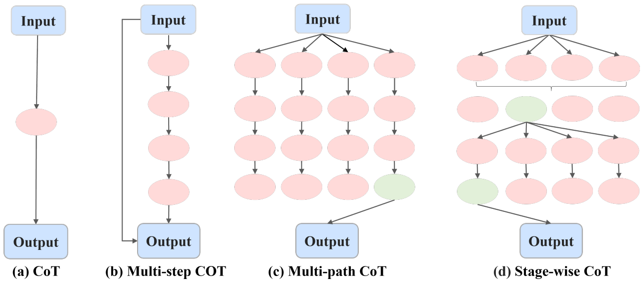


Figure 3: Comparison of CoT-based reasoning strategies. (a) Basic CoT performs single-step end-to-end reasoning. (b) Multi-step CoT extends this to sequential reasoning across multiple steps. (c) Multi-path CoT enhances reliability by applying self-consistency voting at the final step over multiple reasoning paths. (d) Stage-wise CoT (**Ours**) progressively selects the most consistent words through self-consistent voting during the generation of each reasoning step across multiple reasoning paths.

Stage-wise Plan-and-Solve CoT

Figure 3 compares the proposed Stage-wise CoT module with representative CoT-based reasoning strategies. While traditional methods either generate a single reasoning path or apply self-consistent voting only at the final answer stage across multiple paths, the proposed method introduces a planed reasoning process better reflects human problem-solving. Stage-wise CoT progressively selects the most consistent words through self-consistent voting during the generation of each reasoning step. In contrast to multi-path methods such as CoT-SC (Wang et al. 2023b), which focus on generating diverse reasoning trajectories, our approach aims to produce an optimal reasoning by introducing a voting mechanism during the reasoning generation process. This process reduces error propagation.

Stage-wise Reasoning and Conclusion. Stage-wise CoT decomposes the reasoning process into four planned stages, enhancing both reasoning quality and robustness. As illustrated in Figure 1, the process begins with the question summarization stage, which abstracts the core objective of the question to guide attention toward key information and reduce reasoning drift. Next, the image captioning stage generates a concise description of visual elements relevant to the question, improving cross-modal alignment and ensuring that essential visual cues are captured. This is followed by the logical reasoning stage, where the model integrates multi-modal inputs and performs step-by-step inference to derive intermediate reasoning steps, thereby enhancing transparency and minimizing errors from omission or misalignment. Finally, in the conclusion stage, the model synthesizes the reasoning chain to produce a logically consistent final answer.

During the reasoning generation process, multiple reasoning trajectories R^i are generated for the given text-image pair (I_v, I_t) through N_r rounds of sampling from the model, as formally defined:

$$R^i \sim \mathcal{P}(R \mid I_v, I_t), \quad i = 1, 2, \ldots, N_r. \tag{4}$$

The most consistent reasoning is obtained by progressively selecting the most consistent words across the reasoning trajectories. The best reasoning R^* is chosen as:

$$R^* = \text{Vote}(\{R_i^{(j)}\}), \tag{5}$$

where the word with the highest frequency at position j in the N_r generated rationales is selected to form the optimal reasoning token R_j^*.

In the conclusion phase, multiple answers are generated based on the best reasoning process R^* and the input text-image pair (I_v, I_t). Different answers are generated by sampling N_a times from the model A_i, as defined:

$$A_i \sim P(A \mid I_v, I_t, R^*), \quad i = 1, 2, \ldots, N_a. \tag{6}$$

Finally, the answer is selected by majority voting, as defined:

$$A^* = \text{Vote}(\{A_i\}). \tag{7}$$

Among the N_a generated candidate answers, the one with the highest occurrence frequency is chosen as the most reliable and consistent answer.

Voting Strategy. Unlike the Multimodal-CoT (Zhang et al. 2024) model, which directly employs the cross-entropy loss function to compute model loss, a combined loss voting strategy is adopted to further enhance the inference process. Specifically, the proposed method first calculates both the average prediction vector $\overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$ and the weighted average vector $\hat{Y} = \sum_{i=1}^{n} w_i Y_i$, where the weight $w_i = \frac{1}{1+\sigma_i}$, and σ_i denotes the standard deviation of the i-th prediction. The average vector $\overline{Y}$ captures the overall prediction trend and helps suppress the impact of random fluctuations. To estimate the statistical characteristics of the predictions, the model is sampled N_s times and compute the variance σ of the j-th dimension of the outputs as follows:

$$\sigma_i^2 = \frac{1}{N_a}\sum_{i=1}^{N_a}(Y_i - \mu)^2, \quad \mu = \frac{1}{N_s}\sum_{i=1}^{N_s} Y_i, \tag{8}$$

where μ denotes the mean of the j-th dimension of the outputs. A smaller σ_i indicates more stable predictions and is thus assigned a higher weight, while a larger σ_i reflects greater uncertainty and is given a lower weight. The weight vector $w_i = \frac{1}{1+\sigma_i}$ dynamically adjusts each prediction's contribution based on its confidence. As σ_i decreases, w_i increases, emphasizing more reliable outputs. This confidence-aware weighting enhances model robustness by prioritizing consistent predictions.

The final output is computed as a linear combination of the average prediction $\overline{Y}$ and the weighted average prediction $\hat{Y}$, given by:

$$Y^* = \alpha\hat{Y} + (1-\alpha)\overline{Y}, \tag{9}$$

where α is a balance parameter.

Finally, the cross-entropy loss is computed between the combined prediction Y^* and the ground-truth label A:

$$\mathcal{L} = \text{CrossEntropy}(Y^*, A) \tag{10}$$

Two-stage Fine-Tuning Process

The fine-tuning process consists of two stages: the reasoning generation stage and the answer inference stage. As shown in Figure 2, both stages share the same model architecture but differ in input-output formats. During training, the visual encoder is kept frozen to retain pre-trained visual representations, while the language model is fine-tuned with unfrozen parameters. In the reasoning generation stage, the model generates explicit reasoning chains based on multi-modal inputs (text and images). In the answer inference stage, the text input is concatenated with the generated reasoning chain from the first stage and used to predict the final answer.

In the reasoning generation stage, the model $\mathcal{M}_{\text{rational}}$ is trained to generate a reasoning process R conditioned on the multi-modal input X, defined as:

$$R = \mathcal{M}_{\text{rational}}(I). \tag{11}$$

Here, $I = \{I_v, I_t\}$ denotes the multi-modal input, where I_v is the visual input and $I_t = \{Q, C, O\}$ includes the question Q, context C, and candidate options O. The reasoning process R is generated in an auto-regressive manner, with its probability defined as:

$$\mathcal{P}(R \mid I_v, I_t) = \prod_{i=1}^{L} \mathcal{P}_\psi(R_i \mid I_v, I_t, R_{<i}), \tag{12}$$

where ψ denotes the trainable parameters of $\mathcal{M}$rational, L is the length of the target reasoning sequence, and $R< i$ refers to the previously generated tokens.

In the answer inference stage, a separate model $\mathcal{M}_{\text{answer}}$ is trained to predict the final answer A based on the original multi-modal input and the generated reasoning process. This can be formally expressed as:

$$A = \mathcal{M}_{\text{answer}}(I'), \tag{13}$$

where $I' = \{I'_t, I_v\}$ denotes the augmented multi-modal input, where $I'_t = I_t \oplus R$ is the concatenation of the original textual input with the generated reasoning sequence. The conditional probability of predicting the answer is defined as:

$$\mathcal{P}(A \mid I_v, I_t) = \prod_{i=1}^{L} \mathcal{P}\psi'(A_i \mid I_v, I_t, A< i), \tag{14}$$

where ψ' denotes the trainable parameters of the answer inference model, and $A_{<i}$ refers to the previously generated answer tokens.

During inference, the model first uses $\mathcal{M}_{\text{rational}}$ to generate an intermediate reasoning process R from the input. This reasoning is then passed to $\mathcal{M}_{\text{answer}}$, which is conditioned on both the original input and R, predicts the final answer A. This two-stage pipeline encourages explicit reasoning, thereby improving both interpretability and accuracy.

Experiments

This study conducts experiments on two benchmarks for complex reasoning in visual question answering, ScienceQA (Lu et al. 2022) and A-OKVQA (Schwenk et al. 2022), and evaluates model performance using accuracy. More details about datasets and metrics can be seen in Appendix D.

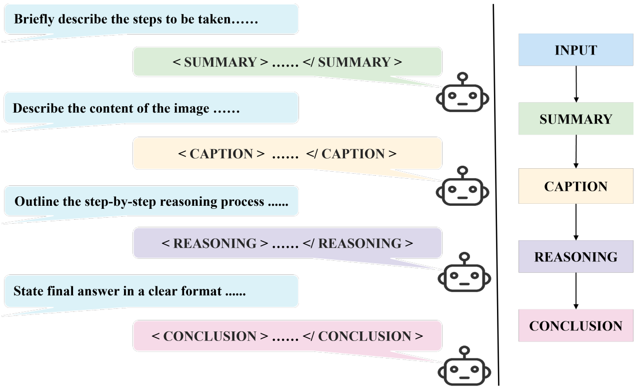
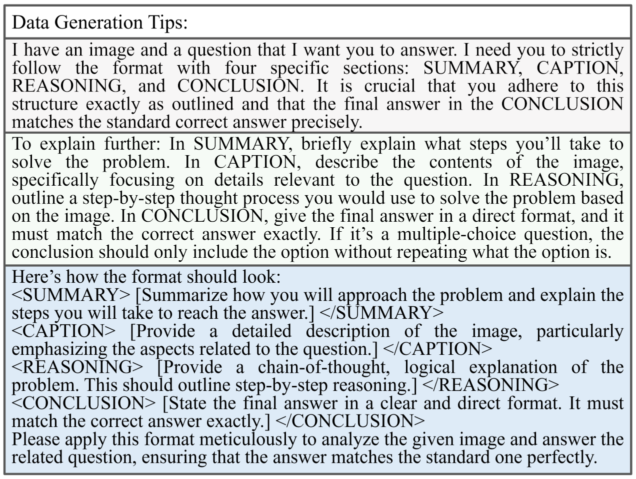

Briefly describe the steps to be taken……
< SUMMARY > …… </ SUMMARY >
Describe the content of the image ……
< CAPTION > …… </ CAPTION >
Outline the step-by-step reasoning process ……
< REASONING > …… </ REASONING >
State final answer in a clear format ……
< CONCLUSION > …… </ CONCLUSION >
INPUT
SUMMARY
CAPTION
REASONING
CONCLUSION

Figure 4: Overview of multi-stage in stage-wise reasoning.



Data Generation Tips:

I have an image and a question that I want you to answer. I need you to strictly follow the format with four specific sections: SUMMARY, CAPTION, REASONING, and CONCLUSION. It is crucial that you adhere to this structure exactly as outlined and that the final answer in the CONCLUSION matches the standard correct answer precisely.

To explain further: In SUMMARY, briefly explain what steps you'll take to solve the problem. In CAPTION, describe the contents of the image, specifically focusing on details relevant to the question. In REASONING, outline a step-by-step thought process you would use to solve the problem based on the image. In CONCLUSION, give the final answer in a direct format, and it must match the correct answer exactly. If it's a multiple-choice question, the conclusion should only include the option without repeating what the option is.

Here's how the format should look:
<SUMMARY> [Summarize how you will approach the problem and explain the steps you will take to reach the answer.] </SUMMARY>
<CAPTION> [Provide a detailed description of the image, particularly emphasizing the aspects related to the question.] </CAPTION>
<REASONING> [Provide a chain-of-thought, logical explanation of the problem. This should outline step-by-step reasoning.] </REASONING>
<CONCLUSION> [State the final answer in a clear and direct format. It must match the correct answer exactly.] </CONCLUSION>
Please apply this format meticulously to analyze the given image and answer the related question, ensuring that the answer matches the standard one perfectly.

Figure 5: Prompt templates for stage-wise CoT.

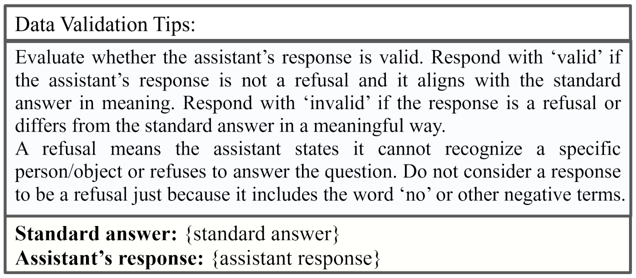
Experimental Setup

Datasets Re-annotation. To build a high-quality dataset for training the Think-ScienceQA model, we re-annotate the ScienceQA and A-OKVQA datasets by generating planned stage-wise reasoning chains using GPT-4o (OpenAI 2023). As shown in Figure 4, each sample is augmented with a four-stage reasoning process, including Summary, Caption, Reasoning, and Conclusion, based on the original question, image, and answer. Carefully designed prompts illustrated in Figure 5 guide GPT-4o to generate structured rationales, which are then validated for format consistency and filtered to remove invalid outputs. Finally, using the prompt in Figure 6, the content within the "</CONCLUSION>...</CONCLUSION>" tags is extracted, and samples where GPT-4o either rejects the answer or fails to match the ground-truth label are discarded. This process yields a clean, structured dataset optimized for multi-sta visual reasoning in Think-ScienceQA.

Implementation Details. Inspired by LLaVA (Liu et al. 2023), the semantic adapter is pre-trained with the frozen image encoder and frozen language model. The adapter itself is initialized from scratch. This strategy preserves the

Method	Size	NAT	SOC	LAN	TXT	IMG	NO	G1-6	G7-12	Avg.
Human (Lu et al. 2022)	-	90.23	84.97	87.48	89.60	87.50	88.10	91.59	82.42	88.40
MCAN (Yu et al. 2019)	95M	56.08	46.23	58.09	59.43	51.17	55.40	51.65	59.72	54.54
Top-Down (Anderson et al. 2018)	70M	59.50	54.33	61.82	62.90	54.88	59.79	57.27	62.16	59.02
BAN (Kim, Jun, and Zhang 2018)	112M	60.88	46.57	66.64	62.61	52.60	65.51	56.83	63.94	59.37
DFAF (Gao et al. 2019)	74M	64.03	48.82	63.55	65.88	54.49	64.11	57.12	67.17	60.72
ViLT (Kim, Son, and Kim 2021)	113M	60.48	63.89	60.27	63.20	61.38	57.00	60.72	61.90	61.14
Patch-TRM (Lu et al. 2021)	90M	65.19	46.79	65.55	66.96	55.28	64.95	58.04	67.50	61.42
VisualBERT (Li et al. 2020)	111M	59.33	69.18	61.18	62.71	62.17	58.54	62.96	59.92	61.87
UnifiedQA (Raffel et al. 2020)	223M	68.16	69.18	74.91	63.78	61.38	77.84	72.98	65.00	70.12
GPT-3 (Brown et al. 2020)	175B	74.64	69.74	76.00	74.44	67.28	77.42	76.80	68.89	73.97
GPT-3 (CoT) (Lu et al. 2022)	175B	75.44	70.87	78.09	74.68	67.43	79.93	78.23	69.68	75.17
GPT-4 (CoT) (Lu et al. 2023)	>1T	84.06	73.45	87.36	81.87	70.75	90.73	84.69	79.10	82.69
Chameleon (Lu et al. 2023)	>1T	89.83	74.13	89.82	88.27	77.64	92.13	88.03	83.72	86.54
Multimodal-CoT$_{base}$ (Zhang et al. 2024)	223M	87.52	77.17	85.82	87.88	82.90	86.83	84.65	85.37	84.91
Multimodal-CoT$_{large}$ (Zhang et al. 2024)	738M	95.91	82.00	90.82	95.26	88.80	92.89	92.44	90.31	91.68
LLaMA-Adapter (Horawalavithana et al. 2024)	7B	84.37	88.30	84.36	83.72	80.32	86.90	85.83	84.05	85.19
LLaVA (Liu et al. 2023)	13B	90.36	95.95	88.00	89.49	88.00	90.66	90.93	90.90	90.92
LLaVA+GPT-4 (Judge) (Liu et al. 2023)	13B	91.56	**96.74**	91.09	90.62	88.99	93.52	92.73	92.16	92.53
LaVIN-13B (Zhang et al. 2024)	13B	90.32	94.38	87.73	89.44	87.65	90.31	91.19	89.26	90.50
LLaMA-SciTune (Horawalavithana et al. 2024)	13B	89.30	95.61	87.00	93.08	86.67	91.75	84.37	91.30	90.03
PILL (Yin et al. 2024)	7B	90.36	95.84	89.27	89.39	88.65	91.71	92.11	89.65	91.23
Honeybee (Cha et al. 2024)	13B	95.20	96.29	91.18	94.48	93.75	93.17	95.04	93.21	94.39
Think-ScienceQA$_{Base}$ (**Ours**)	228M	94.84	94.21	91.59	94.91	94.92	92.10	93.73	94.10	93.86
Think-ScienceQA$_{Large}$ (**Ours**)	746M	**97.77**	93.55	**92.77**	**97.60**	**95.46**	**94.00**	**95.39**	**95.94**	**95.59**

Table 1: Accuracy comparison across eight question categories on the ScienceQA dataset. Bold and underlined values indicate **best** and second-best results, respectively.



Data Validation Tips:

Evaluate whether the assistant's response is valid. Respond with 'valid' if the assistant's response is not a refusal and it aligns with the standard answer in meaning. Respond with 'invalid' if the response is a refusal or differs from the standard answer in a meaningful way.
A refusal means the assistant states it cannot recognize a specific person/object or refuses to answer the question. Do not consider a response to be a refusal just because it includes the word 'no' or other negative terms.

Standard answer: {standard answer}
Assistant's response: {assistant response}

Figure 6: Prompt template for verifying generated data.

original knowledge of the backbone models while gradually incorporating visual information, leading to more stable learning during fine-tuning and improved visual understanding. For image encoding, this study adopts the CLIP visual encoder (Radford et al. 2021) pre-trained on large-scale data. Features are extracted from the penultimate layer of the Transformer encoder, as the final layer emphasizes global semantics while earlier layers capture more local details, providing a more balanced visual representation. The language backbone is built upon the T5 encoder-decoder architecture, with UnifiedQA (Raffel et al. 2020) adopted as the default. The batch size is set to 16 for the Base model and 8 for the Large model. The experiments utilize a learning rate of 5×10^{-5} and a maximum input sequence length of 512 tokens. α is empirically set to 0.5. All experiments are implemented in PyTorch 1.12 and run on four NVIDIA Tesla V100 GPUs with 32GB of memory each.

Results and Analysis

Results on the ScienceQA Dataset. The results on the ScienceQA dataset are given in Table 1. It shows the overall model size rather than just the tunable parameters to better reflect true capabilities of the models. Think-ScienceQA$_{\text{Base}}$, with only 228M parameters, achieves an average accuracy of 93.86%, surpassing the performance of LaVIN-13B (Zhang et al. 2024) and LLaVa-13B (Liu et al. 2023) both of which involve extensive fine-tuning of the language model. Furthermore, Think-ScienceQA$_{\text{Large}}$ achieves a 3.06% improvement in accuracy over the fine-tuned LLaVa+GPT-4 (Liu et al. 2023). It also outperforms Honeybee (Cha et al. 2024), a multi-modal LLM method, by 1.2%. Compared to multimodal-CoT$_{\text{Base}}$ (Zhang et al. 2024), Think-ScienceQA$_{\text{Base}}$ shows an 8.95% improvement, while Think-ScienceQA$_{\text{Large}}$ outperforms Multimodal-CoT$_{\text{Large}}$ by 3.91%.

Results on the A-OKVQA Dataset. Table 2 presents the experimental results on the A-OKVQA dataset. The Think-ScienceQA$_{\text{Base}}$ model significantly outperforms current state-of-the-art models in both direct answer accuracy and multiple-choice tasks. In the direct answer task, Think-ScienceQA$_{\text{Base}}$ achieves an accuracy of 61.7%, outperforming the second-best model BLIP-2 (with over 11 billion parameters) by 8.5%. In the multiple-choice task, Think-ScienceQA$_{\text{Base}}$ reaches an accuracy of 66.0%. These results demonstrate excellent parameter utilization efficiency of the model, surpassing larger models with more parameters. This is especially notable given the A-OKVQA dataset's complexity and its requirement for high-level reasoning across

Method	Model Size	Direct-answer	Multi-choice
Pythia (Biderman et al. 2023)	70M	25.2	49.0
ViLBERT (Lu et al. 2019)	300M	30.6	49.1
KRISP (Marino et al. 2021)	200M	33.7	51.9
GPV-2 (Kamath et al. 2022)	300M	48.6	60.3
BLIP-2 (Li et al. 2023)	11B	<u>53.2</u>	**70.2**
PaLM-CoT (Lu et al. 2022)	540B	41.5	48.1
PICa (Yang et al. 2022)	175B	42.4	46.1
IPVR (Chen et al. 2023)	66B	46.4	48.6
Multimodal-CoT$_{Base}$ (Zhang et al. 2024)	223M	-	50.6
Think-ScienceQA$_{Base}$ **(Ours)**	228M	**61.7**	<u>66.0</u>

Table 2: Comparisons on the A-OKVQA dataset. Bold and underlined values indicate **best** and <u>second-best</u> results.

Model	Rouge-L (%)	Average Accuracy(%)
Multimodal-CoT$_{Base}$	96.97	84.91
Think-ScienceQA$_{Base}$	98.23	93.86
Think-ScienceQA$_{Large}$	98.86	95.59

Table 3: Impact of reasoning path quality on ScienceQA.

Model	No. of Paths	Average Accuracy(%)
Multimodal-CoT$_{Large}$	0	91.68
Think-ScienceQA$_{Large}$	5	95.45
Think-ScienceQA$_{Large}$	10	95.59

Table 4: Results under varying numbers of reasoning paths on ScienceQA.

Method	Average Accuracy (%)	Δ (%)
w/ structured labels	93.86	-
w/o structured labels	87.05	↓ 6.81

Table 5: Impact of structured labels on ScienceQA.

both text and visual modalities. The performance improvement of Think-ScienceQA$_{\text{Base}}$ underscores the effectiveness of consistency training strategies.

Ablation studies

Impact of Reasoning Path Quality. Table 3 presents the evaluation results regarding the impact of reasoning path quality on the ScienceQA dataset. Specifically, we use the Rouge-L score (Lin 2004) to evaluate the quality of the generated reasoning paths. Rouge-L is one of the standard metrics for assessing the quality of generated text, as it measures the similarity between the generated output and the reference text based on the longest common subsequence. Compared to the Multimodal-CoT$_{\text{Base}}$ model, Think-Science$_{\text{Base}}$ achieves an improvement of 1.26% in Rouge-L score and an increase of 8.95% in answer accuracy. Although Think-ScienceQA$_{\text{Large}}$ surpasses Think-Science$_{\text{Base}}$ by only 0.63% in Rouge-L, its answer accuracy improves by approximately 2%. These results suggest that even small gains in reasoning path quality can lead to notable improvements in answer prediction performance for multi-modal reasoning models.

Impact of Multi-Path. To investigate how varying the number of intermediate reasoning paths affects accuracy and training efficiency on ScienceQA tasks, this study compare two settings: (1) generating 5 reasoning paths, which is the default configuration chosen to balance accuracy and computational cost, and (2) generating 10 reasoning paths to explore potential performance gains with increased reasoning diversity. As shown in Table 4, increasing the number of reasoning paths to 10 yields a marginal improvement of 0.14% in accuracy (from 95.35% to 95.59%). The results indicate that increasing the number of reasoning paths does not lead to a significant improvement in model performance. This suggests that only a few reasoning paths contribute meaningfully to solving the problem, while others may be redundant, overly similar, or even implausible. Moreover, generating 10 reasoning paths nearly doubles the computational cost compared to using 5, due to the additional overhead in both generation and evaluation. Therefore, in the experiments, a default of 5 reasoning paths is adopted as a balanced choice between performance and computational efficiency.

Impact of Structured Labels. To evaluate the role of structured labels, that is, <SUMMARY>, <CAPTION>, <REASONING>, <CONCLUSION>, we compare Think-ScienceQA with a variant trained without them. As shown in Table 5, removing these labels results in a notable performance drop of 6.81%, confirming that structural supervision improves reasoning and overall model effectiveness.

In addition, the theoretical justification for the multi-path reasoning generation and voting strategy, the visual comparison of model accuracy versus model size, and the analysis of challenging examples are provided in the Appendix.

Conclusions

This paper presented Think-ScienceQA, a novel framework aimed at improving robustness, interpretability, and efficiency in ScienceQA tasks. The approach decomposed complex reasoning into semantically planned stages, and incorporated stage-wise self-consistency voting to mitigate hallucinations and reasoning drift commonly observed in end-to-end generation. Additionally, a semantic adapter was introduced to enhance visual-text alignment, thereby strengthening multimodal representation learning. Extensive experiments demonstrate that Think-ScienceQA achieves state-of-the-art accuracy with far fewer parameters, with ablation studies confirming the effectiveness of structured labels and path quality in enhancing both performance and interpretability. While challenges related to scalability and generalization across broader domains remained, Think-ScienceQA represented a promising step toward building reliable, efficient, and explainable multimodal reasoning systems. Future work would explore cross-domain transfer, fine-grained control of reasoning strategies, and closer alignment with human reasoning patterns.

References

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6077–6086.

Biderman, S.; Schoelkopf, H.; Anthony, Q.; Bradley, H.; O'Brien, K.; Hallahan, E.; Khan, M. A.; Purohit, S.; Prashanth, U. S.; Raff, E.; Skowron, A.; Sutawika, L.; and van der Wal, O. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2397–2430.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language models are few-shot learners. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 1877–1901.

Cha, J.; Kang, W.; Mun, J.; and Roh, B. 2024. Honeybee: Locality-enhanced projector for multimodal LLM. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 13817–13827.

Chen, Z.; Zhou, Q.; Shen, Y.; Hong, Y.; Zhang, H.; and Gan, C. 2023. See, think, confirm: Interactive prompting between vision and language models for knowledge-based visual reasoning. arXiv:2301.05226.

Gao, P.; Jiang, Z.; You, H.; Lu, P.; Hoi, S. C. H.; Wang, X.; and Li, H. 2019. Dynamic fusion with intra- and inter-modality attention flow for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6639–6648.

Horawalavithana, S.; Munikoti, S.; Stewart, I.; Kvinge, H.; and Pazdernik, K. 2024. SCITUNE: Aligning large language models with human-curated scientific multimodal instructions. In *Proceedings of the 1st Workshop on NLP for Science (NLP4Science)*, 58–72.

Kamath, A.; Clark, C.; Gupta, T.; and Kembhavi, A. 2022. Webly supervised concept expansion for general purpose vision models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 662–681.

Kim, J.-H.; Jun, J.; and Zhang, B.-T. 2018. Bilinear attention networks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 1571–1581.

Kim, W.; Son, B.; and Kim, I. 2021. ViLT: Vision-and-language transformer without convolution or region supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 5583–5594.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 19730–19742.

Li, L. H.; Yatskar, M.; Yin, D.; Hsieh, C.-J.; and Chang, K.-W. 2020. What does BERT with vision look at? In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 5265–5275.

Liang, W.; Jiang, Y.; and Liu, Z. 2021. GraphVQA: Language-guided graph neural networks for scene graph question answering. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 79–86.

Lin, C.-Y. 2004. ROUGE: A Package for automatic evaluation of summaries. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 74–81.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual instruction tuning. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 1–25.

Lu, J.; Batra, D.; Parikh, D.; and Lee, S. 2019. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 13–23.

Lu, P.; Mishra, S.; Xia, T.; Qiu, L.; Chang, K.; Zhu, S.; Tafjord, O.; Clark, P.; and Kalyan, A. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 1–15.

Lu, P.; Peng, B.; Cheng, H.; Galley, M.; Chang, K.; Wu, Y. N.; Zhu, S.; and Gao, J. 2023. Chameleon: Plug-and-play compositional reasoning with large language models. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 1–32.

Lu, P.; Qiu, L.; Chen, J.; Xia, T.; Zhao, Y.; Zhang, W.; Yu, Z.; Liang, X.; and Zhu, S.-C. 2021. IconQA: A new benchmark for abstract diagram understanding and visual language reasoning. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks)*, 1–14.

Marino, K.; Chen, X.; Parikh, D.; Gupta, A.; and Rohrbach, M. 2021. KRISP: Integrating implicit and symbolic knowledge for open-domain knowledge-based VQA. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 14111–14121.

OpenAI. 2023. GPT-4 technical report. Technical report, OpenAI. Available at https://arxiv.org/abs/2303.08774.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 8748–8763.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140): 1–140.

Schwenk, D.; Khandelwal, A.; Clark, C.; Marino, K.; and Mottaghi, R. 2022. A-OKVQA: A benchmark for visual question answering using world knowledge. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 146–162.

Tan, C.; Wei, J.; Gao, Z.; Sun, L.; Li, S.; Guo, R.; Yu, B.; and Li, S. Z. 2024. Boosting the power of small multimodal reasoning models to match larger models with self-consistency training. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 305–322.

Wang, L.; Xu, W.; Lan, Y.; Hu, Z.; Lan, Y.; Lee, R. K.-W.; and Lim, E.-P. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2609–2634.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E. H.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023b. Self-consistency improves chain of thought reasoning in language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 1–24.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E. H.; Le, Q. V.; and Zhou, D. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 1–14.

Xu, G.; Jin, P.; Li, H.; Song, Y.; Sun, L.; and Yuan, L. 2024. LLaVA-CoT: Let vision language models reason step-by-step. arXiv:2411.10440.

Yang, Z.; Gan, Z.; Wang, J.; Hu, X.; Lu, Y.; Liu, Z.; and Wang, L. 2022. An empirical study of GPT-3 for few-shot knowledge-based VQA. In *Proceedings of the Annual AAAI Conference on Artificial Intelligence*, 3081–3089.

Yin, Y.; Zhang, F.; Wu, Z.; Qiu, Q.; Liang, T.; and Zhang, X. 2024. PILL: Plug into LLM with adapter expert and attention gate. *Applied Soft Computing*, 165(11): 112115.

Yu, Z.; Yu, J.; Cui, Y.; Tao, D.; and Tian, Q. 2019. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6281–6290.

Zhang, Z.; Zhang, A.; Li, M.; Zhao, H.; Karypis, G.; and Smola, A. 2024. Multimodal chain-of-thought reasoning in language models. *Transactions on Machine Learning Research*, 4(5): 1–25.

Reproducibility Checklist

1. General Paper Structure

1.1. Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes/partial/no/NA) yes

1.2. Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes/no) yes

1.3. Provides well-marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (yes/no) yes

2. Theoretical Contributions

2.1. Does this paper make theoretical contributions? (yes/no) yes

If yes, please address the following points:

2.2. All assumptions and restrictions are stated clearly and formally (yes/partial/no) yes

2.3. All novel claims are stated formally (e.g., in theorem statements) (yes/partial/no) yes

2.4. Proofs of all novel claims are included (yes/partial/no) yes

2.5. Proof sketches or intuitions are given for complex and/or novel results (yes/partial/no) yes

2.6. Appropriate citations to theoretical tools used are given (yes/partial/no) yes

2.7. All theoretical claims are demonstrated empirically to hold (yes/partial/no/NA) yes

2.8. All experimental code used to eliminate or disprove claims is included (yes/no/NA) yes

3. Dataset Usage

3.1. Does this paper rely on one or more datasets? (yes/no) yes

If yes, please address the following points:

3.2. A motivation is given for why the experiments are conducted on the selected datasets (yes/partial/no/NA) yes

3.3. All novel datasets introduced in this paper are included in a data appendix (yes/partial/no/NA) NA

3.4. All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no/NA) NA

3.5. All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations (yes/no/NA) yes

3.6. All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available (yes/partial/no/NA) yes

3.7. All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing (yes/partial/no/NA) NA

4. Computational Experiments

4.1. Does this paper include computational experiments? (yes/no) yes

If yes, please address the following points:

4.2. This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting (yes/partial/no/NA) yes

4.3. Any code required for pre-processing data is included in the appendix (yes/partial/no) yes

4.4. All source code required for conducting and analyzing the experiments is included in a code appendix (yes/partial/no) yes

4.5. All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/-partial/no) yes

4.6. All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes/-partial/no) yes

4.7. If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results (yes/partial/no/NA) NA

4.8. This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks (yes/partial/no) yes

4.9. This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics (yes/partial/no) yes

4.10. This paper states the number of algorithm runs used to compute each reported result (yes/no) yes

4.11. Analysis of experiments goes beyond singledimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information (yes/no) yes

4.12. The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank) (yes/partial/no) yes

4.13. This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments (yes/partial/no/NA) yes