See, Think, Learn: A Self-Taught Multimodal Reasoner

Sourabh Sharma[1] Sonam Gupta[2] Sadbhawna[1]

sourabh125ss@gmail.com sonam.gupta7@ibm.com sadbhawna.cse@mnit.ac.in

[1] Malaviya National Institute of Technology Jaipur [2] IBM Research

Abstract

Vision-Language Models (VLMs) have achieved remarkable progress in integrating visual perception with language understanding. However, effective multimodal reasoning requires both accurate ***perception*** *and robust* ***reasoning****, and weakness in either limits the performance of VLMs. Prior efforts to enhance reasoning often depend on high-quality chain-of-thought (CoT) data, obtained via labor-intensive human annotations, costly proprietary models or self-training methods that overlook perception. To address these limitations, we propose a simple yet effective self-training framework called "See-Think-Learn" (STL). At its core, STL introduces a structured reasoning template that encourages the model to see before thinking: first extracting visual attributes in textual form, then using them to guide reasoning. The framework jointly improves perception and reasoning by having the model generate and learn from its own structured rationales in a self-training loop. Furthermore, we augment the training data with negative rationales, i.e. explanations that justify why certain answer choices are incorrect, to enhance the model's ability to distinguish between correct and misleading responses. This fosters more discriminative and robust learning. Experiments across diverse domains show that STL consistently outperforms baselines trained directly only on answers or self-generated reasoning, while qualitative analysis confirms the high quality of its rationales. STL thus provides a cost-effective solution to enhance multimodal reasoning ability of VLMs. Our code is available at https://github.com/srbhcs/see-think-learn.*

1. Introduction

Large Language Models (LLMs) have recently demonstrated remarkable advances in complex reasoning through the use of Chain-of-Thought (CoT) prompting [13, 18, 28, 33]. By explicitly encouraging models to generate intermediate reasoning steps before arriving at the final answer, CoT significantly boosts performance on a wide range of textual tasks [3, 6, 22, 38]. Motivated by this success, recent works

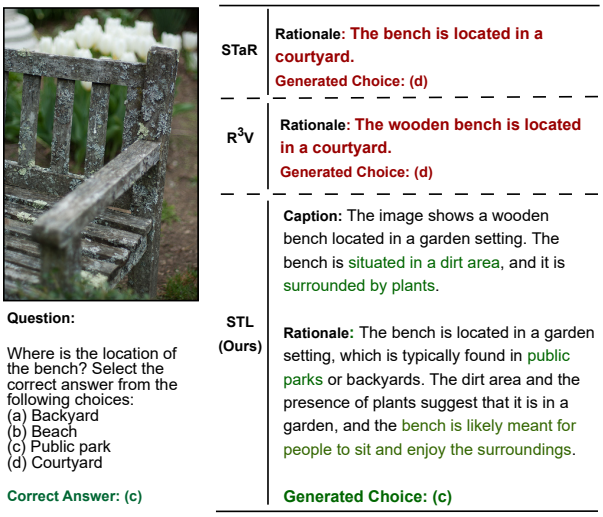

Figure 1. **Comparison of reasoning generated by our "See-Think-Learn" (STL) framework with STaR [37] and R^{3}V [9].** STL produces more detailed and perceptually grounded rationales, whereas STaR and R^{3}V tend to overlook contextual cues and provide shorter, less comprehensive explanations.

[19, 29, 30] have attempted to extend CoT prompting to multimodal extensions of LLMs, termed as Visual Language Models (VLMs) [1, 2, 10, 14, 15, 25]. However, reasoning in VLMs remains a fundamental challenge due to their limited capacity to jointly understand and reason over both visual and textual information.

A major bottleneck in training VLMs for CoT-style reasoning is the lack of high-quality supervision. Existing datasets [17, 23] are often limited to short answers with minimal or no explanatory rationales. For curating high-quality rationales, current research typically relies on labor-intensive human annotations (e.g., [5, 36]) which are difficult to scale or on proprietary black-box models like GPT-4V [20] or Gemini [36] which are expensive.

To overcome these limitations, we propose a simple yet effective self-training framework called See-Think-Learn (STL). STL leverages the model's existing perception and

reasoning capabilities to iteratively improve itself by generating and learning from its own rationales. A key question in this process is: *What should the structure of these rationales be?* Effectively answering a visual question requires strong perception and reasoning. To strengthen both components, STL introduces a structured rationale format grounded in the principle of "see before thinking". Specifically, the model is prompted to (1) first describe visual elements from the image, (2) then reason about them in context, and (3) finally produce an answer. This structure mirrors the natural human cognitive process and guides the model in better organizing its internal reasoning.

Relying only on rationales from correct answers provides limited supervision, as the model observes only successful reasoning paths. This weakens performance on complex tasks where distinguishing correct from incorrect reasoning is critical. To address this, we introduce negative rationales (explanations of why certain answers are wrong) alongside positive ones. This mirrors reflective human learning, where understanding improves by analyzing both successes and mistakes. Negative rationales expose flawed reasoning patterns and highlight contrasts between correct and incorrect answers, helping the model avoid pitfalls such as hallucinations and unsupported inferences.

We operationalize this idea in our Self-Taught Multimodal Reasoner (STL), a scalable self-training framework where a VLM learns to generate both positive and negative rationales and iteratively improves through retraining on these structured annotations. As illustrated in Figure 1, STL produces more detailed and accurate reasoning. Evaluations across commonsense, scientific, and language-based domains show that STL outperforms models trained on final answers alone and remains broadly comparable to models trained with human-annotated rationales (Table 3), despite some variation across datasets. These results highlight STL's potential as a practical alternative to annotation-heavy approaches.

In summary, the contributions of this work are as follows:

1. We introduce the "see-before-thinking" rationale structure, which explicitly separates perception and reasoning components within the generated rationale.
2. We enhance training data with negative rationales, allowing the model to learn to distinguish between correct and incorrect reasoning paths.
3. We present a self-training framework, STL that leverages structured rationales to jointly improve perception and reasoning, in contrast to prior approaches that emphasize reasoning alone.
4. We assess our method across domains such as commonsense, language, and science, and present detailed ablations showing that STL boosts performance while avoiding the limitations of human or proprietary supervision.

2. Background and Related Work

Vision-Language Reasoning. Reasoning [4, 31, 39] has been shown to play a critical role in enhancing the performance of Vision-Language Models (VLMs). While recent VLMs achieve strong results on general benchmarks [8, 16], effectively incorporating visual information into the reasoning process remains a persistent challenge, particularly for open-source models [2, 14, 15]. One direction has explored prompt-based strategies that assign functional roles to VLMs via system prompts, enabling modular step-by-step reasoning. For example, Cantor [11] structures reasoning into context analysis followed by high-level feature generation, while CCoT [19] leverages scene graphs to capture object-attribute relations and guide a two-stage process of graph construction and reasoning.

Another direction relies on learning-based approaches that fine-tune VLMs on datasets containing multimodal reasoning chains [24, 27, 30, 32]. Such datasets are typically curated using powerful teacher models (e.g. GPT-4o, Deepseek-R1) or through costly human annotations [8, 17], raising concerns about scalability. In contrast, our work avoids this reliance by enabling VLMs to enhance their reasoning abilities through self-learning, without requiring curated rationales.

Self-Training Methods. Self-training is a semi-supervised paradigm where a model improves by generating supervision from its own outputs. In Large Language Models (LLMs), it has been widely used to strengthen reasoning: models generate intermediate rationales (e.g. chain-of-thought) for unlabeled data, which are then reused as training signals in subsequent iterations. This iterative process enhances reasoning while reducing dependence on human annotations [7, 34]. Seminal works have advanced LLM reasoning by sampling rationales, filtering them for correctness, and fine-tuning on positive samples [12, 35, 37]. By contrast, self-training for VLMs remains relatively underexplored.

In the video domain, Video-STaR [40] extends STaR [37] by generating question-answer pairs from labeled video datasets for instruction tuning. More recently, R^3V [9] explored self-training for reasoning in VLMs. However, unlike STL, R^3V relies on knowledge distillation from GPT-4o [21] as a warm-up stage and requires additional bookkeeping to track responses that evolve from incorrect to correct across iterations.

Furthermore, both Video-STaR and R^3V omit explicit visual descriptions in their reasoning templates. STL differs by introducing a cost-effective reasoning template that integrates image descriptions, thereby improving perception and reasoning over successive iterations. Additionally, STL leverages discriminative learning to unlearn spurious correlations that otherwise lead to systematic errors.

3. Method

The Self-Taught Reasoner (STaR) algorithm [37] is a seminal self-training approach to improve reasoning in LLMs. For relevance to our setting, we first describe its extension to Vision-Language Models (VLMs). We then introduce our proposed framework, **See-Think-Learn (STL)**, which addresses the limitations of STaR by jointly refining perception and reasoning. We assume access to a multiple-choice VQA dataset

$$D = \{(I_i, x_i, C_i, a_i)\}_{i=1}^{N},$$

where I_i is an image, x_i a question, C_i the candidate answers, and a_i the correct answer.

3.1. STaR for Enhancing VLM Reasoning

A simple adaptation of STaR to the VLM setting involves providing its self-training loop with multimodal input, allowing the model to perceive images and reason over text. A VLM M maps the input (I, x, C) to a rationale-answer pair:

$$(r, \hat{a}) = M(I, x, C),$$

where r is a rationale in natural-language and $\hat{a}$ the predicted answer. At iteration n, the model produces

$$(r_i, \hat{a}_i) = M_{n-1}(I_i, x_i, C_i), \quad i = 1, \ldots, N.$$

We then partition these outputs into correct and incorrect predictions:

$$D_n^+ = \{(I_i, x_i, C_i, r_i, a_i) \mid \hat{a}_i = a_i\},$$
$$D_n^- = \{(I_i, x_i, C_i, r_i, \hat{a}_i) \mid \hat{a}_i \neq a_i\}.$$

In the STaR framework, both sets are used for retraining. Correct predictions (D_n^+) are directly fine-tuned, while incorrect predictions (D_n^-) undergo *positive rationalization*: the model is given the gold answer a_i and asked to produce a new rationale $\tilde{r}_i$ that supports it. These newly generated rationales are then filtered to retain only those that lead to the correct answer. This yields a corrected set:

$$\tilde{D}_n^+ = \{(I_i, x_i, C_i, \tilde{r}_i, a_i) \mid (I_i, x_i, C_i, r_i, \hat{a}_i) \in D_n^-\}.$$

The combined dataset

$$D_n = D_n^+ \cup \tilde{D}_n^+$$

is used to fine-tune the model M to yield the model M_n.

By iteratively generating, correcting and retraining on rationales, STaR bootstraps reasoning ability without additional human supervision. However, when applied to VLMs, it faces two key limitations:

1. **Perceptual grounding gap:** STaR's generated rationales mix perception with reasoning, preventing the reasoning from being properly conditioned on perceptual inputs.
2. **Noisy rationales from positive rationalization:** Because the gold answer is revealed, the model may generate superficially correct rationales while retaining flawed reasoning (Figure 5).

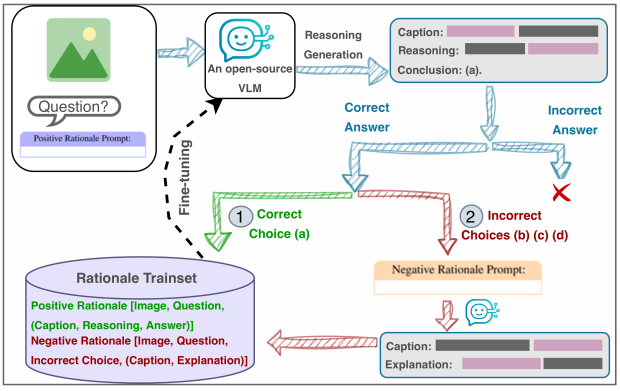

Figure 2. **An overview of our detailed "See-Think-Learn" (STL) framework.** In this framework, each image-question pair with multiple choices, together with a positive rationale prompt, is fed into the VLM to generate a caption, reasoning, and conclusion. If the model predicts the correct answer, the tuple [Image, Question, (Caption, Reasoning, Answer)] is stored as a **Positive Rationale** in the Rationale Trainset. The remaining incorrect choices are used to generate negative rationalizations, producing a caption and an explanation of why the choice is incorrect, which are stored as **Negative Rationales** [Image, Question, Incorrect Choice, (Caption, Explanation)]. The VLM is then iteratively fine-tuned on this dynamically constructed Rationale Trainset.

3.2. Self-Training Framework: See-Think-Learn (STL)

To address these limitations, we propose **STL**, a self-training framework that integrates perception into the reasoning loop. STL differs from STaR in three key aspects:

Structured rationale prompts: We introduce a structured rationale prompt that follows a "see before thinking" approach. As shown in Figure 3, our prompt has three parts: **(1) Caption**, which gives a detailed description of the image based on the question; **(2) Reasoning**, which involves a detailed thought process grounded in visual details; and **(3) Conclusion**, which gives the final answer based on reasoning. Specifically, the model is prompted to produce a tuple $(d_i, r_i, \hat{a}_i)$, comprising an image description d_i, a rationale r_i, and a predicted answer $\hat{a}_i$.

Selective positive rationales: Instead of correcting rationales for incorrect predictions, STL retains only high-quality rationales from correctly answered samples (obtained without revealing the gold answer), thereby reducing noise. Concretely, for each instance the model outputs

$$(d_i, r_i, \hat{a}_i) = M(I_i, x_i, C_i),$$

where d_i is an image description, r_i a rationale, and $\hat{a}_i$ the predicted answer. We retain only the correct predictions:

$$D_n^{\text{pos}} = \{(I_i, x_i, C_i, d_i, \hat{r}_i^+, a_i) \mid \hat{a}_i = a_i\}.$$

Positive Rationale Prompt:

```
You are an image based question-answering expert.
Given an image along with a multiple choice
question, your task is to select the correct
choice based on the image.
Your response should strictly follow the format
with three specific sections: CAPTION, REASONING
and CONCLUSION. Response:
###CAPTION: [Provide a detailed description of
the image, particularly emphasizing the aspects
related to the question.]
###REASONING: [Provide a detailed thought process
to answer the question.]
###CONCLUSION: [Provide the correct choice based
on the reasoning.]
Question: {question_and_choices}
Response:
```

Negative Rationale Prompt:

```
You are an image based question-answering expert.
Given an image along with a multiple choice
question and an answer, your task is to explain
why the answer is wrong.
Your response should strictly follow the
format with two specific sections: CAPTION
and EXPLANATION. Response:
###CAPTION: [Provide a detailed description of
the image, particularly emphasizing the aspects
related to the question.]
###EXPLANATION: [Provide a detailed explanation
for why the answer is wrong.]
Question: {question}
The correct choice is {correct_choice}.
Explain why this answer is wrong:
{incorrect_choice}
Response:
```

Figure 3. **Prompt templates** used for positive and negative rationalization in the STL framework.

This dataset is used for iterative fine-tuning, encouraging the model to improve both perception and reasoning.

Discriminative negative rationales: Rationales from correct samples capture only one side of the reasoning spectrum. To emulate the human strategy of reflective learning, where incorrect options are critically analyzed, we further augment the training data with *negative rationales*: explanations for why incorrect choices are wrong. This enhances the model's ability to distinguish the correct answer from the alternatives. We now describe this process in detail in the following subsection.

3.3. Negative Rationalization: Discriminative Learning

To further strengthen reasoning, STL introduces **negative rationalization**. For this, we focus only on samples that the model previously answered correctly. We assume that a correct prediction means that the model has a good understanding of the image and the question. Using this confidence, we prompt the model (see Figure 3) to first describe the image (d_i), then explain why each incorrect option $c \in C_i \setminus a_i$ is not valid. These explanations are generated with the correct answer a_i included in the prompt to guide accurate reasoning. Each explanation is stored as a tuple $(I_i, x_i, C_i, c, d_i, \hat{r}^-_{i,c})$, where $\hat{r}^-_{i,c}$ is the negative rationale for option c.

Later, when used for training, the gold answer is withheld from the prompt, forcing the model to reason independently about distractors. This yields the negative rationale dataset:

$$D_n^{\text{neg}} = \{(I_i, x_i, C_i, c, d_i, \hat{r}^-_{i,c}) \mid (I_i, x_i, C_i, d_i, \hat{r}^+_i, a_i) \in D_n^{\text{pos}}, c \in C_i \setminus \{a_i\}\}. \quad (1)$$

Fine-tuning on the combined dataset $D_n^{\text{pos}} \cup D_n^{\text{neg}}$ enhances three complementary abilities: (1) perceptual grounding via structured descriptions, (2) reasoning ability via positive rationales and (3) discriminative capability via negative rationales, allowing the model not only to generate coherent rationales for correct answers, but also to critically assess and reject misleading or implausible alternatives. Algorithm 1 summarizes the entire STL procedure and Figure 2 illustrates its workflow.

Algorithm 1 STL: SEE–THINK–LEARN

Require: Pretrained VLM M; MCQ dataset $\mathcal{D} = \{(I_i, x_i, \mathcal{C}_i, a_i)\}_{i=1}^{D}$

$M_0 \leftarrow M$; $n \leftarrow 0$
repeat
  $n \leftarrow n + 1$
  (A) Inference: Generate positive rationales (description, reasoning and prediction)
  for all $(I_i, x_i, \mathcal{C}_i, a_i) \in \mathcal{D}$ **do**
    $(d_i, \hat{r}_i^+, \hat{a}_i) \leftarrow M_{n-1}[\texttt{posprompt}, I_i, x_i, \mathcal{C}_i]$
  end for
  (B) Construct positive rationale dataset from correct predictions
  $\mathcal{D}_n^{\text{pos}} \leftarrow \{(I_i, x_i, C_i, d_i, \hat{r}_i^+, a_i) \mid \hat{a}_i = a_i\}$
  (C) Inference: Generate negative rationales (description and explanation)
  for all $(I_i, x_i, C_i, d_i, \hat{r}_i^+, a_i) \in \mathcal{D}_n^{\text{pos}}$ **do**
    for all $c \in \mathcal{C}_i \setminus \{a_i\}$ **do**
      $(d_i, \hat{r}^-_{i,c}) \leftarrow M_{n-1}[\texttt{negprompt}, I_i, x_i, C_i, a_i, c]$
    end for
  end for
  (D) Construct negative rationale dataset
  $\mathcal{D}_n^{\text{neg}} \leftarrow \{(I_i, x_i, C_i, c, d_i, \hat{r}^-_{i,c}) \mid c \in \mathcal{C}_i \setminus \{a_i\}\}$
  (E) Combine and fine-tune
  $\mathcal{D}_n \leftarrow \mathcal{D}_n^{\text{pos}} \cup \mathcal{D}_n^{\text{neg}}$
  $M_n \leftarrow \texttt{train}(M, \mathcal{D}_n)$
until converged

4. Experiments

To assess the effectiveness of our method, we conducted comprehensive evaluations across four knowledge domains using the LLaVA-v1.5-7B [15] model. To examine the generalizability of our method across different VLMs, we further conducted experiments with the Qwen2.5-VL-7B-Instruct

Table 1. **Performance Comparison on M3CoT Evaluation Splits on LLaVA [15].** Accuracy of various baselines along with the proposed STL across the four domains.

Method	Common sense	Natural- Science	Language- -Science	Social- -Science	Average
Zero-Shot Methods					
Direct VQA	57.58	36.40	45.02	29.62	42.16
CoT	54.94	35.5	35.54	24.68	37.66
Positive Prompt (Fig. 3)	53.40	33.20	40.28	27.55	38.61
Direct SFT					
Direct SFT	60.22	46.10	46.92	34.24	46.87
Self-Training Methods					
STaR[37]	64.98	**53.90**	48.82	41.88	51.21
R^3V[9]	62.64	-	45.97	-	-
Ours	**67.19**	50.45	**55.92**	**43.79**	**54.34**

Table 2. **Performance Comparison on M3CoT Evaluation Splits for Qwen [26].** Accuracy of various baselines along with the proposed STL across two domains.

Method	Commonsense	Language Science	Average
Zero-Shot Methods			
Direct VQA	82.20	72.51	77.36
CoT	80.00	57.34	68.67
Positive Prompt (Fig. 3)	80.48	73.46	76.97
Direct SFT			
Direct SFT	82.42	79.15	80.79
Self-Training Methods			
STaR[37]	81.54	80.57	81.06
R^3V[9]	80.44	74.88	77.66
Ours	**84.32**	**86.41**	**85.36**

model [26]. We start by detailing the datasets and baseline methods used for comparison, implementation details followed by the quantitative results and qualitative analysis.

4.1. Datasets

We evaluated our method across four domains, namely commonsense, natural science, language science, and social science, using samples drawn from the M3CoT dataset [5], which provides multi-domain, multiple-choice visual question-answering tuples paired with human-annotated rationales. The availability of these rationales allows a direct comparison between our approach and human-annotated reasoning. The commonsense domain assesses reasoning about physical, social, and temporal aspects depicted in images; natural science focuses on visually grounded questions in physics, chemistry, and biology; social science addresses topics related to geography, economics, and cognitive science; and language science encompasses questions involving figurative language, grammar, and reading comprehension.

4.2. Baselines

We compare our approach against several strong baselines. For zero-shot evaluation, we assess the base model's performance under different prompting strategies, including direct answer prompting, Chain-of-Thought (CoT) prompting, and our structured rationale prompt. We also include a direct SFT baseline, where the model is fine-tuned on (image, question, answer) tuples using direct prompting, instructing it to predict the answer without generating an intermediate rationale. Finally, we compare our method with two state-of-the-art self-training approaches: STaR [37] and R^3V [9].

4.3. Implementation Details

We fine-tuned both the LLaVA-v1.5-7B [15] and Qwen2.5-VL-7B-Instruct [26] models using LoRA with rank $r = 128$ and scaling factor $\alpha = 256$. To support memory-efficient training, we used DeepSpeed ZeRO-3 and enabled gradient checkpointing. Training was performed for one epoch with a batch size of 8 and 3 gradient accumulation steps under a cosine learning-rate schedule. All experiments were run on a single NVIDIA A6000 GPU (48 GB VRAM). The self-training methods were executed for 6–7 iterations, and we report test-set accuracy for all results.

5. Quantitative Evaluation

We evaluated STL against several baselines using accuracy, with results on LLaVA presented in Table 1. Zero-shot prompting struggles with multimodal reasoning, and Chain-of-Thought (CoT) prompting performs worse than direct prompting (Direct VQA). The higher accuracy observed in the Direct VQA setting compared to CoT prompting indicates the inherent shortcut-seeking behavior of MLLMs. When prompted to reason before answering, models such as LLaVA, which possess limited reasoning capabilities, often generate incorrect rationales, ultimately leading to incorrect answers. Direct SFT, which trains the model to answer questions without generating rationales, achieves a modest 4% gain over zero-shot methods.

Self-training methods leverage the base model's weak reasoning to bootstrap reasoning datasets, achieving substantial gains. STL performs best, surpassing STaR by 3% and R^3V by 7%, producing higher-quality rationales than the baselines. Notably, after STL training, the model can generate coherent rationales for new problems, contributing to its improved performance.

Question: What is this scene most likely a part of? Select the correct answer from the following choices:
(a) A Chinese restaurant
(b) A buffet breakfast
(c) A cooking class
(d) A hotel room

Correct Answer: (b)

a. CoT Prompt

Generated Choice: (c)
Rationale: (c) A cooking class

b. Training with positive rationales only

Generated Choice: (c)
Caption: The image shows **a bowl of noodles** with a **fork and a knife** on a dining table. The bowl is filled with noodles, and there are also **some vegetables** in the mix.
Rationale: The presence of a bowl of noodles, a fork, and a knife suggests that this scene is most likely a part of a **cooking class.** The person is likely learning how to prepare noodles with vegetables, and the utensils are provided for their convenience.

c. STL (Ours)

Generated Choice: (b)
Caption: The image shows a **bowl of food with a spoon** in it, placed on a dining table. The food appears to be a **pasta dish**.
Rationale: The presence of a bowl of food with a spoon in it, along with the dining table, suggests that this scene is most likely a part of a **buffet breakfast**. Buffet breakfasts typically include a variety of food items, such as pasta dishes, and are **served in bowls with spoons for guests to serve** themselves.

Figure 4. **Comparison of our "See-Think-Learn" (STL) framework with CoT Prompting.** The example is taken from the Commonsense Split of M3CoT Dataset [5]. Unlike CoT prompting a., our STL framework (b. and c.) effectively generates a detailed description and accurate reasoning for the image by leveraging the proposed Positive and Negative Rationale Prompts. In a., the answer is incorrect, and the image description is missing. In b., although a detailed description is provided, it is inaccurate. For example, it mentions a "fork" and "knife" that are not present in the image. In contrast, c. produces both the correct answer and an accurate description, capturing key elements such as "serve" and "buffet". Q: Question; O: Options;

Table 3. **Comparison with human-annotated rationale training.** Evaluation of STL vs. M3CoT [5] rationales on Language-Science and Commonsense splits.

Model	Language Science	Commonsense	Average
LLaVA (Human)	**66.82**	60.44	**63.63**
LLaVA (Ours)	55.92	**67.19**	61.56
Qwen (Human)	72.98	78.68	75.83
Qwen (Ours)	**86.41**	**84.32**	**85.36**

Table 4. **Ablation Study.** W/O: Without, W/: With, Neg: Negative Rationalization, Cap: Structured Rationale Prompt

Method	Language Science	Commonsense
W/O (Cap+Neg)	46.44	59.56
W/O Neg	48.34	64.62
Ours (W/ (Cap+Neg))	**55.92**	**67.19**

Interestingly, STaR outperforms STL on the Natural Science dataset. This is largely due to STaR's sample generation strategy: besides using high-quality positive samples, it employs positive rationalization by reprompting incorrect answers with hints emphasizing the correct choice. This enables the model to reach correct answers without true reasoning, a shortcut that can produce inconsistent or misaligned explanations. In contrast, STL relies solely on correctly answered samples and applies negative rationalization for augmentation, encouraging the model to differentiate correct from incorrect reasoning. Although STaR benefits from a greater exposure to training data, STL prioritizes robust, genuine reasoning. Figure 5 illustrates an example of shortcut learning in STaR.

To further assess the generalizability of our approach to stronger, modern VLMs, we applied the STL framework to Qwen2.5-VL-7B [26] on the Commonsense and Language Science domains. Compared to LLaVA, the Qwen model exhibits superior reasoning abilities, as reflected in Direct VQA performance. However, shortcut learning is still evident: performance drops noticeably when the model is prompted to generate reasoning before answering. Fine-tuning directly on the answers improves overall performance by 3%. As expected, self-training methods provide much larger gains, demonstrating that STL can also be used to enhance the reasoning capabilities of stronger models.

Comparison with human-annotated rationales: To compare the reasoning quality, we use the human-annotated rationales from M3CoT. Table 3 compares models fine-tuned on these human-annotated rationales with the models fine-tuned using See-Think-Learn (STL) self-generated rationales, across the Language Science and Commonsense domains. Although human-annotated rationales remain higher in quality, STL-generated samples achieve competitive or superior performance, demonstrating that STL can approximate human-level reasoning while providing a scalable alternative to manual annotation.

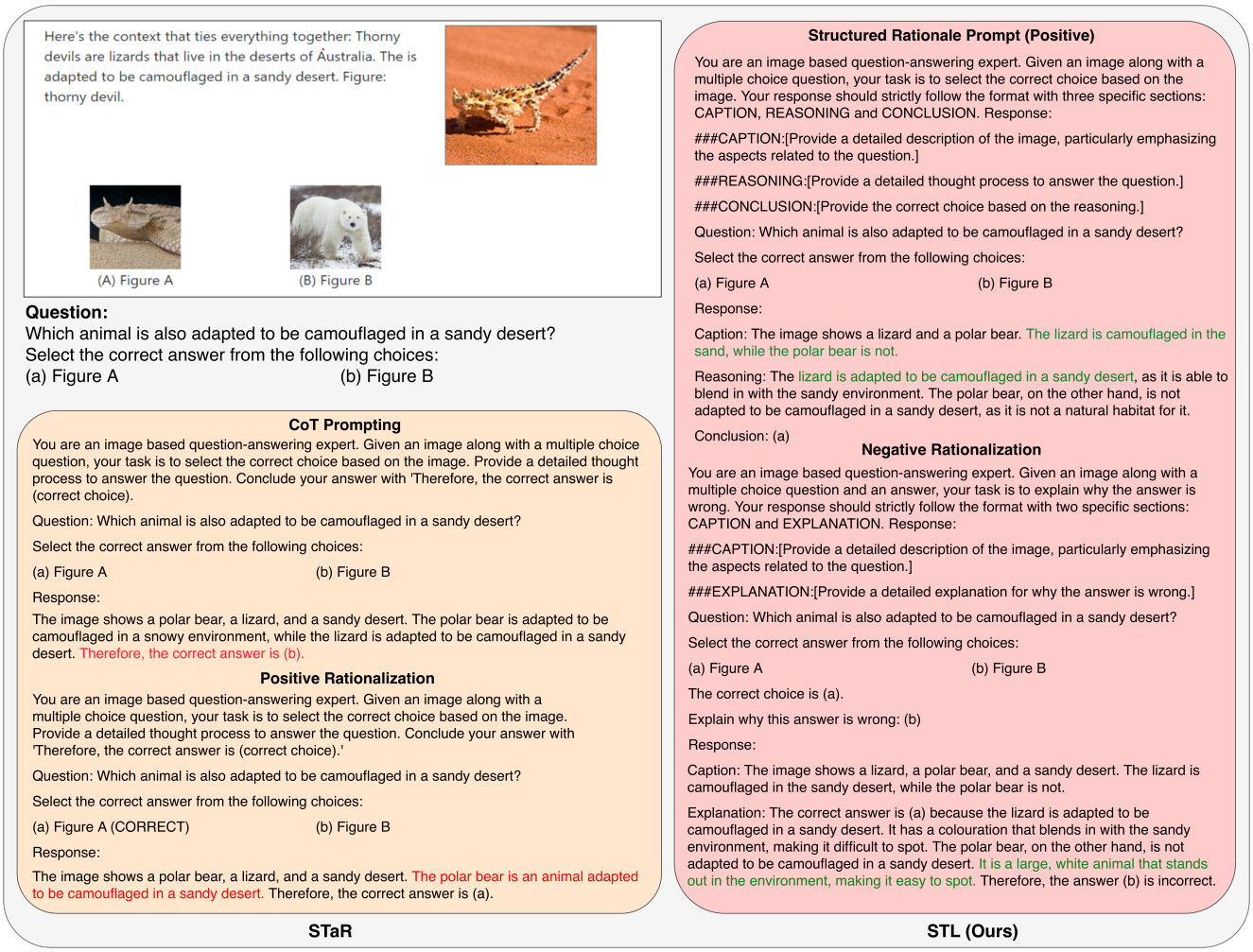

Figure 5. **Qualitative Comparison on Natural Science Domain.** Qualitative analysis shows that STL (ours) produces more coherent and logically consistent explanations than STaR, indicating deeper understanding and more faithful reasoning.

5.1. Ablation Studies

We perform two ablation studies to assess the impact of each component of the proposed STL framework on the language science and commonsense domains. The W/O (Cap+Neg) condition represents our STL framework implemented without the proposed structured rationale prompt and negative rationalization i.e. this setup utilizes a prompt without caption and fine-tunes on just the correctly answered rationales. The W/O Neg condition excludes only the negative rationalization component from our framework. As shown in Table 4, removing both the structured rationale prompt and negative rationalization results in noticeably lower performance, mainly due to the loss of enhanced perception and discriminative learning. Introducing the structured rationale prompt significantly improves accuracy by guiding the model to carefully analyze the image before reasoning, thereby enabling iterative refinement of its perception during training. Furthermore, incorporating negative rationalization augmentation yields an additional performance gain, as the model learns to better discriminate between correct and incorrect answers.

In Figure 4, we show a qualitative example, demonstrating the effect of the positive rationale template and discriminative learning through negative rationale template. The image depicts a bowl-based serving setup and the correct answer is *buffet breakfast*. Figure 4 a. shows the output of the model with CoT prompt (No description). The model fails to generate any rationale and answers the question incorrectly. When fine-tuning the model using self-training with Positive rationales (STL template) only, the model is able to generate a description which is partially correct leading to the same incorrect prediction. By contrast, STL fine-tuned using both positive and negative rationales corrects the answer and grounds reasoning in verifiable cues, such as a *bowl of food*

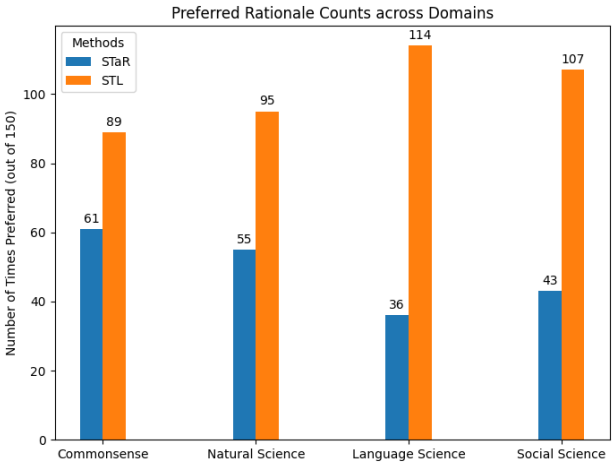

Figure 6. **Comparison of Preferred Rationale Counts across Domains.** Each subplot displays the number of times the rationale generated by each method (STaR and STL) was preferred (out of 150) within a domain. Across all domains, the reasoning generated by STL is preferred for more samples than that of STaR, highlighting its superior quality.

with a spoon and self-serve presentation, while avoiding nonexistent items. This suggests that STL improvements stem not from longer rationales but from the complementary effect of negative rationales, which suppress spurious tokens and guide the model toward subtle, task-relevant visual cues, enhancing both accuracy and rationale fidelity.

5.2. Qualitative Results

Qualitative Example on Commonsense: Figure 1 shows an example from the commonsense domain with reasoning generated by STaR, R^3V, and STL. Both STaR and R^3V tend to overlook perceptual details and produce short explanations that resemble descriptive answers rather than grounded reasoning. In contrast, STL attends to contextual cues such as the plants in the background and the dirt area, enabling it to generate a more comprehensive rationale that leads to the correct answer.

Qualitative Example of Natural Science Domain: Figure 5 provides a qualitative comparison of reasoning outputs on the Natural Science domain, highlighting differences between STaR and STL. In the example, the model is asked to identify which animal is adapted to be camouflaged in a sandy desert. The base model out of the box when prompted using CoT prompt is able to generate good reasoning but still end up predicting the incorrect choice demonstrating the inherent biases the model has when predicting the answers. STaR (with Positive Rationalization) encourages to learn incorrect reasoning. It mistakenly highlights the polar bear as camouflaged, and its rationales are inconsistent with the final answer, reflecting limited understanding of the visual context. In contrast, STL produces structured and logically consistent rationales. Using positive and negative rationalization, STL correctly identifies the lizard as adapted for camouflage and provides clear, step-by-step explanations. The caption accurately describes the scene, the reasoning links the observation to the correct choice, and the conclusion aligns perfectly with the rationale. The negative rationalization additionally reinforces the distinction by explaining why the polar bear is not the correct answer.

Overall, this example illustrates that STL generates more faithful and coherent explanations compared to STaR, indicating deeper understanding and stronger reasoning capabilities. Such qualitative improvements complement the quantitative gains observed in accuracy and suggest that STL promotes robust reasoning rather than relying on shortcuts. This also suggests that while STaR may achieve marginally better accuracy in some domains due to shortcut learning, STL fosters deeper understanding and more faithful reasoning. More qualitative samples are provided in the Supplementary Material.

6. Subjective Analysis

To assess rationale quality, we conducted a subjective evaluation comparing reasoning generated by STL and STaR for the LLaVA model. We randomly sampled 150 rationales per domain from questions both methods answered correctly and asked three annotators to compare them. Each annotator was shown the image, the corresponding question, and reasoning from both methods, without knowing which model produced them. Annotators ranked the two reasoning, and to ensure consistency, all assessed the same set of samples within each domain. Further details of the annotation procedure are provided in the Supplementary Material.

On average, STL reasoning were preferred 35% more often than those from STaR. Figure 6 summarizes these results, showing that across all domains, STL was consistently chosen more frequently. These findings demonstrate STL's effectiveness in producing rationales that better align with human preferences and provide higher-quality reasoning compared to existing methods.

7. Conclusion

We present the See–Think–Learn (STL) framework, a self-training strategy that enables vision–language models to enhance multimodal reasoning without costly human-annotated rationales. By leveraging structured prompts with both positive and negative rationales, STL improves visual understanding and discriminative learning. The framework achieves strong empirical performance across tasks, demonstrating that models can effectively learn from their own generated perceptions and explanations, laying the groundwork for more advanced multimodal reasoning.

References

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

[2] J Bai, S Bai, S Yang, S Wang, S Tan, P Wang, J Lin, C Zhou, and J Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. arxiv 2023. *arXiv preprint arXiv:2308.12966*, 2023. 1, 2

[3] Dibyanayan Bandyopadhyay, Soham Bhattacharjee, and Asif Ekbal. Thinking machines: A survey of llm based reasoning strategies. *arXiv preprint arXiv:2503.10814*, 2025. 1

[4] Franz Louis Cesista. Multimodal structured generation: Cvpr's 2nd mmfm challenge technical report. *arXiv preprint arXiv:2406.11403*, 2024. 2

[5] Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. M^3CoT: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8199–8221, Bangkok, Thailand, 2024. Association for Computational Linguistics. 1, 5, 6

[6] Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*, 2025. 1

[7] Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024. 2

[8] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 67(12):220101, 2024. 2

[9] Kanzhi Cheng, Yantao Li, Fangzhi Xu, Jianbing Zhang, Hao Zhou, and Yang Liu. Vision-language models can self-improve reasoning via reflection. *The North American Chapter of the Association for Computational Linguistics*, 2025. 1, 2, 5

[10] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Xilin Wei, Songyang Zhang, Haodong Duan, Maosong Cao, et al. Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *arXiv preprint arXiv:2401.16420*, 2024. 1

[11] Timin Gao, Peixian Chen, Mengdan Zhang, Chaoyou Fu, Yunhang Shen, Yan Zhang, Shengchuan Zhang, Xiawu Zheng, Xing Sun, Liujuan Cao, et al. Cantor: Inspiring multimodal chain-of-thought of mllm. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 9096–9105, 2024. 2

[12] Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. V-star: Training verifiers for self-taught reasoners. *arXiv preprint arXiv:2402.06457*, 2024. 2

[13] Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, pages 22199–22213. Curran Associates, Inc., 2022. 1

[14] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 1, 2

[15] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 2, 4, 5

[16] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. 2

[17] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022. 1, 2

[18] Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful chain-of-thought reasoning. In *The 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP-AACL 2023)*, 2023. 1

[19] Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. Compositional chain-of-thought prompting for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14420–14431, 2024. 1, 2

[20] OpenAI. GPT-4 technical report, 2023. Includes GPT-4V (Vision) multimodal capabilities. 1

[21] OpenAI. Gpt-4o system card, 2024. Large multimodal foundation model. 2

[22] Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Back. Reasoning with large language models, a survey. *arXiv preprint arXiv:2407.11511*, 2024. 1

[23] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European conference on computer vision*, pages 146–162. Springer, 2022. 1

[24] Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642, 2024. 2

[25] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1

[26] Qwen Team. Qwen2.5-vl, 2025. 5, 6

[27] Omkar Thawakar, Dinura Dissanayake, Ketan More, Ritesh Thawkar, Ahmed Heakl, Noor Ahsan, Yuhao Li, Mohammed

Zumri, Jean Lahoud, Rao Muhammad Anwer, et al. Llama-v-o1: Rethinking step-by-step visual reasoning in llms. *arXiv preprint arXiv:2501.06186*, 2025. 2

[28] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 1

[29] Jinyang Wu, Mingkuan Feng, Shuai Zhang, Ruihan Jin, Feihu Che, Zengqi Wen, and Jianhua Tao. Boosting multimodal reasoning with mcts-automated structured thinking. *arXiv preprint arXiv:2502.02339*, 2025. 1

[30] Guowei Xu, Peng Jin, Li Hao, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step. *URL https://arxiv. org/abs/2411.10440*, 2024. 1, 2

[31] Guowei Xu, Peng Jin, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step, 2024. 2

[32] Huanjin Yao, Jiaxing Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, Yuxin Song, Haocheng Feng, Li Shen, et al. Mulberry: Empowering mllm with o1-like reasoning and reflection via collective monte carlo tree search. *arXiv preprint arXiv:2412.18319*, 2024. 2

[33] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023. 1

[34] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models, 2025. 2

[35] Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023. 2

[36] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024. 1

[37] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STar: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, 2022. 1, 2, 3, 5

[38] Yadong Zhang, Shaoguang Mao, Tao Ge, Xun Wang, Adrian de Wynter, Yan Xia, Wenshan Wu, Ting Song, Man Lan, and Furu Wei. Llm as a mastermind: A survey of strategic reasoning with large language models. *arXiv preprint arXiv:2404.01230*, 2024. 1

[39] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023. 2

[40] Orr Zohar, Xiaohan Wang, Yonatan Bitton, Idan Szpektor, and Serena Yeung-Levy. Video-star: Self-training enables video instruction tuning with any supervision. *arXiv preprint arXiv:2407.06189*, 2024. 2