Aha Moment Revisited: Are VLMs Truly Capable of Self Verification in Inference-time Scaling?

Anonymous ACL submission

Abstract

Inference-time techniques such as decoding-time scaling and self-refinement have been shown to substantially improve reasoning in large language models (LLMs), driven by emergent self-correction and self-verification behaviors often elicited through reinforcement learning (RL). In this work, we investigate whether these inference-time scaling methods similarly benefit vision-language models (VLMs), especially those fine-tuned with RL. Through extensive evaluation, we find that while strategies like majority vote and best-of-N with self-verification enhance VLM performance, majority vote significantly outperforms verification-centric ones. Furthermore, inference time scaling behaviors commonly associated with RL-tuned models, such as the "A-ha moment," do not yield consistent performance gains. Our analysis identifies a key limitation: current RL-trained VLMs exhibit weak self-verification across both visual and textual modalities, limiting the effectiveness of inference-time scaling.

1 Introduction

The reasoning capabilities of large language models (LLMs) have seen notable improvements in recent years (DeepSeek-AI, 2025; OpenAI, 2024). Although larger model scales and higher-quality pretraining datasets are major contributing factors to these improvements, emerging strategies that instead leverage **inference-time computation** (Snell et al., 2024) have also been proven effective: Providing models with zero-shot "think step by step" prompts or few-shot demonstrations augmented with intermediate reasoning steps (Wei et al., 2022) have enabled the models to generate extended reasoning chains even when not explicitly fine-tuned to do so. Likewise, methods such as decoding-time majority vote (Wang et al., 2023) and chain-of-thought decoding (Wang and Zhou, 2024) have enabled outputting of higher-quality answers without external feedback. More recently, inference-time self-correction (Kumar et al., 2025; DeepSeek-AI, 2025) has emerged as another form of scaling: models are trained with Reinforcement Learning (RL) to revise earlier mistakes and generate additional reasoning steps to arrive at improved reasoning answers. This is known as the "**A-ha moment**", where the model generates "Wait, I made a mistake in my prior response"—and initiates a second round of reasoning to refine its answer.

A central question we explore in this work is whether **self-verification generalizes well to VLMs.** Notably, several recent efforts (Zhou et al., 2025; Chen et al., 2025b; Zhang et al., 2025; Huang et al., 2025; Liu et al., 2025; Deng et al., 2025; Wang et al., 2025) adopt similar RL-based training strategies and report the emergence of "A-ha moments" in VLM reasoning to suggest that VLMs similarly contain the "hidden knowledge" of self-verification capacity present in LLMs and can be elicited via RL. However, a key question remains: **Are RL-trained VLMs genuinely effectively performing self-verification and self-correction during inference, or are these behaviors merely surface-level artifacts of training which contribute little to model performance?**

We study this problem by contrasting inference-time scaling strategies such as Majority Vote and Self-Verified Best-of-N. Then, for RL-trained models, we probe the self-verification mechanism in more detail to study possible causes of this gap by comparing between (1) giving the (self-)verifier access to the original image input and (2) withholding it during verification within the best-of-N setup. Our experiment shows that the verifier counterintuitively performs better **without the image input**. This behavior highlights a possible core limitation of VLMs' self-verification, in that they currently do not sufficiently leverage visual information for self-verification, which may explain the fundamental limitations that prevent current VLMs to effectively perform inference-time computation.

In this paper, we explicitly demonstrate that inference-time decoding strategies improve reasoning performance in RL-tuned VLMs. We also show that the emergence of "A-ha moments" in RL-tuned VLMs does not lead to gains in final reasoning accuracy—largely due to the model's limited self-verification capabilities. We design and perform extensive experimentation with various inference-time scaling frameworks to support our findings.

2 Related Works

2.1 LLM/VLM, Reinforcement Learning for Reasoning

Reinforcement learning (RL) was introduced to LLM fine-tuning via RL from human feedback (RLHF) (Ouyang et al., 2022), which learns a reward model from human preferences and optimizes the LLM policy, using Proximal Policy Optimization (PPO) (Schulman et al., 2017). Beyond alignment, RL has also been shown to enhance LLM reasoning and self-correction capabilities (Kumar et al., 2025; DeepSeek-AI, 2025; Zeng et al., 2025). Several studies (Gandhi et al., 2025; Zeng et al., 2025) further investigate what intrinsic properties enable effective self-improvement and how "A-ha moments" emerge as a result of RL-based training. In the vision-language domain, similar ideas have been extended to improve VLM reasoning. A number of recent works apply RL to incentivize multimodal reasoning behaviors, typically using PPO or GRPO to fine-tune VLMs. These studies report positive signs of RL to train VLM to generate "A-ha moments" in VLMs (Zhou et al., 2025; Chen et al., 2025b; Zhang et al., 2025; Huang et al., 2025; Liu et al., 2025; Deng et al., 2025; Wang et al., 2025).

2.2 Inference-Time Scaling

Inference-time scaling (Snell et al., 2024; Brown et al., 2024) has emerged as an effective strategy for improving LLM reasoning without additional fine-tuning. Several methods fall under this umbrella. Simple parallel decoding approaches—such as chain-of-thought decoding (Wang and Zhou, 2024) and self-consistency sampling (Wang et al., 2023)—have shown strong empirical gains by aggregating multiple sampled outputs. More sophisticated techniques involve training reward-based verifiers to guide step-by-step generation (Lightman et al., 2023).

3 Methodology

3.1 Reinforcement Learning for VLMs

We test both the base version as well as the RL-tuned version of the VLMs. Previous work on LLMs has demonstrated the emergence of the 'A-ha moment' in RL-tuned model's reasoning process and such a process was shown to have a positive contribution to model performance. We would like to study whether similar benefits also exist for VLMs trained through RL, specifically Group Relative Policy Optimization (GRPO). To this end, we adopt RL-tuned models from recent work (Zhang et al., 2025; Chen et al., 2025a; Wang et al., 2025), using them directly within our experimental framework. For detailed descriptions of the RL and GRPO objectives, we refer readers to these prior works. These RL-tuned VLMs are typically optimized using outcome-based rewards, and recent works (Zhang et al., 2025; Chen et al., 2025a; Wang et al., 2025) claim near-GPT-4o-level performance using models with only $\sim$7B parameters.

3.2 Decoding Methods for VLMs

VLMs generate text in the same way as LLMs do, except with additional image embeddings as part of the input query. Decoding methods affect how each next token is sampled from Language Models. In this work, we consider methods that aim to sample multiple starting tokens and thus generate multiple outputs given one single input query.

Greedy Decoding. Sequentially selects the most probable next token at each decoding step. It is a one-time inference with no scaling.

Decoding-Time Majority Voting. This strategy first samples multiple candidate outputs and then subsequently selects the final solution by majority consensus among the generated candidates. By aggregating multiple responses, it seeks to mitigate random errors or inconsistencies in individual outputs. We consider this as a baseline method to beat due to the 'deterministic' nature of how the final output is selected. **This method primarily reflects the model's generation capability, as high accuracy depends on the model producing correct answers frequently enough to dominate the vote.**

Best of N Sampling with Self as Verifier. This strategy also samples multiple candidate outputs from the VLM, but we then prompt the model to evaluate and verify all candidate outputs together. The output identified as most reliable by the model itself is selected as the final answer. **This method**

emphasizes the model's self-verification ability, as accuracy depends on correctly identifying the best answer to the question from a diverse set of responses. We include the prompt for verification in the appendix.

To understand whether VLMs are able to benefit from self-verification and whether the vision inputs are been used for better self-verification, we test VLMs using multiple configurations of the Best of N Decoding method:

Self Verification with Text Only: The self-verifier receives only the generated responses and the text-based question. The image is omitted to test the model's ability to verify using language alone. **Self Verification with Image and Text:** The self-verifier is provided with both the image and the text input, allowing it to use multimodal information for verification.

3.3 Finding 'A-ha Moment'

We design a A-ha moment search protocol following (Gandhi et al., 2025) that aligns A-ha with coginitive behavior, and introduce two metrics to assess whether the presence of an A-ha moment positively contributes to reasoning performance:

A-ha Search: we prompt GPT-4o with the generated response and ask whether it exhibits these behaviors. The simplified prompt template is provided below and complete version is in appendix. If GPT-4o confirms the presence of backtracking or verification, we classify the response as containing an "aha moment."

AHA Search Prompt

```
<system prompt>
<start_of_reasoning> <RESPONSE>
<end_of_reasoning>
Specifically, actively identify and
emphasize beneficial behaviors such as:
(1) Backtracking: Explicitly revising
approaches upon identifying errors or dead
ends ...
(2) Verification: Systematically checking
intermediate results or reasoning steps
...
Important:
Clearly specify each beneficial behavior
you identify.
If there is a strong example of this,
provide <YES> followed by specific
explanations. Otherwise, provide <NO>
<NO>
```

Post-A-ha Accuracy Among Selected Predictions: We compute the probability that a selected answer containing a confirmed A-ha moment is also correct. This is denoted as

$$P^{\star}(\text{Correct} \mid \text{A-ha in Prediction}),$$

where the star ($\star$) indicates that we report the best value across all decoding strategies. This metric reflects how often A-ha moments align with correct final answers in selected outputs.

A-ha Potential Recovery Rate from Incorrect Predictions: To assess whether A-ha moments can help recover from initial errors, we focus on cases where the selected prediction is incorrect. We then search through the unselected generated responses and check whether any of them contain both a confirmed A-ha moment and a correct answer. This is measured as

$$P(\text{A-ha Correct} \mid \text{Wrong Prediction}),$$

indicating the potential for A-ha-based reasoning paths in the inference-time scaling to correct mistakes even when they are not selected by default.

4 Experiment

We utilize the GeoQA170K (Lu et al., 2024) and MathVista (Gao et al., 2023) for our empirical evaluation in multimodal reasoning. The models evaluated range from the base Qwen2-VL-2B-Instruct to a set of Qwen-based RL-tuned models. The full list includes: R1-VL-2B, R1-VL-7B (Zhang et al., 2025), VLAA-Thinker-Qwen2.5VL-3B, VLAA-Thinker-Qwen2.5VL-7B (Chen et al., 2025a), and VL-Rethinker-7B (Wang et al., 2025)

4.1 Discussion

Inference Time Scaling Improves Performance of VLM. Table 1 summarizes the performance gains of various inference-time scaling techniques versus the baseline deterministic, greedy decoding on GeoQA of the various VLMs. Notably, both BoN-based methods achieve limited performance gains over the greedy baseline, and in the case of the two VLAA VLMs, even result in performance *decreases* (up to -16.7%), which can be attributed to its tendency to re-do the question rather than to judge the response despite being explicitly prompted to choose from the responses. On the other hand, the generation-emphasizing—Majority vote—achieves more steady performance gains (4.5%).

RL-trained VLMs Do Not Benefited from A-ha Moments As shown in Table 2, answers flagged as containing "A-ha moments" do not lead to

Table 1: Verifier Comparison on GeoQA and MathVista

Model	GeoQA			MathVista		
	BoN w. Image	BoN w/o Image	Majority Votes	BoN w. Image	BoN w/o Image	Majority Votes
Scaling $\times$ 4 / $\times$ 8						
R1-VL-2B (Zhang et al., 2025)	28.9 / 31.2	28.2 / 30.2	**30.2 / 35.1**	39.2 / 41.5	40.9 / 42.0	**52.7 / 56.4**
R1-VL-7B (Zhang et al., 2025)	**44.5** / 45.8	43.9 / 46.2	44.2 / **46.9**	59.3 / 61.1	63.8 / 63.6	**65.2 / 66.0**
VLAA-Thinker-3B (Chen et al., 2025a)	27.5 / 23.5	31.6 / 28.0	**46.4 / 48.3**	50.3 / 48.4	52.1 / 45.1	**66.2 / 65.6**
VLAA-Thinker-7B (Chen et al., 2025a)	44.3 / 52.3	46.2 / 52.9	**52.1 / 57.7**	65.5 / 70.5	58.2 / 66.2	**71.6 / 74.0**
VL-Rethinker-7B (Wang et al., 2025)	59.9 / 58.9	59.8 / 58.9	**61.9 / 62.1**	75.0 / 73.9	74.7 / 71.4	**75.4 / 75.6**

Table 2: Conditional Accuracy w.r.t Aha Moments and Decoding Comparison on GeoQA ($\times$4 Scaling). CoT refers to Chain of Thought Decoding.

	P^*(Correct | Aha)	P(Aha Correct | Wrong)	Greedy	BoN w. Image	BoN w/o Image	Majority Votes
Qwen2-VL-2B-Instruct	–	–	13.8	**16.3**	15.6	16.0
R1-VL-2B (Zhang et al., 2025)	28.1(CoT)	2.7	26.9	28.9	28.2	30.2
R1-VL-7B (Zhang et al., 2025)	49.5(BoN w. Image)	4.4	39.7	**44.6**	43.9	44.2
VLAA-Thinker-3B (Chen et al., 2025a)	48.4(CoT)	5.4	44.2	27.5	31.6	**46.4**
VLAA-Thinker-7B (Chen et al., 2025a)	49.5(CoT)	13.0	48.3	44.3	46.2	**52.1**
VL-Rethinker-7B (Wang et al., 2025)	65.5(Majority Vote)	19.5	60.1	59.9	59.8	**61.9**

higher accuracy—even when we select the best result across all decoding strategies. This suggests that **"A-ha moments" do not reliably contribute to improved reasoning capabilities**. While we also assess the potential of A-ha moments—i.e., whether they could correct an initially wrong prediction—the observed probabilities remain low. This indicates that simply encouraging A-ha behavior is insufficient for improving model performance within the inference-time scaling framework.

Current RL-trained VLM Fall Short in Verification in Inference-time Scaling Table 1 quantitatively assess verification ability using best-of-N decoding with self-verification. Across both 4- and 8-sample settings, majority voting—an indicator of generation quality—consistently outperforms self-verification. This stands in contrast to findings in the LLM literature, where verification is often easier than generation. Our results suggest that current RL techniques do not endow VLMs with strong self verification capabilities, raising concerns about their effectiveness in multimodal reasoning.

No Visual Verification Another notable observation from Tables 1 is that RL-trained VLMs sometimes verify their own outputs more accurately when visual input is excluded. This is particularly evident in the GeoQA dataset, which consists entirely of geometric questions. Including the image does not necessarily help the model judge correctness—suggesting that the VLM fails to integrate visual context during self-verification. Instead, the model over-relies on textual input, rendering its verification process in both modalities unreliable. Our findings show that current VLMs do not fully utilize visual information during verification, and we call for future research to address this shortcoming by enhancing the model's true multimodal verification capabilities to improve reasoning.

5 Conclusion

In this paper, we investigated the extensibility of LLM inference-time computation techniques to VLMs. We find that current RL-trained VLMs yet lack robust self-verification capabilities across both visual and textual modalities in the form of a verification-generation gap. We have performed extensive experimentation to support this claim: our results show that the verification-reliant best-of-N selection strategy achieves lower performance gains versus the generation-reliant majority voting, and that the self-correction behavior often associated with RL-tuned models, such as "A-ha moment," does not lead to measurable gains.

Broader Impacts. This finding paper highlights a critical gap: the core RL mechanisms that drive reasoning improvements in LLMs do not translate smoothly to VLMs. We argue that a key reason is the lack of robust multimodal self-verification capabilities in current VLMs.

Call for Improvement. Even the most advanced multimodal model, GPT-5 (OpenAI, 2025), suffers from this issue. For example, if prompted with an image of a hand showing six fingers and asked how many, the model will most likely respond with "five" due to its reliance on language-based reasoning. Even when prompted to self-verify, the model typically fails to correct its mistake and continues to insist on the incorrect answer.

6 Limitations

This work empirically highlights a key limitation of RL-trained VLMs: despite improvements in reasoning performance, these models struggle to fully realize their potential due to weak self-verification capabilities in multimodal settings. **While we analyze and diagnose this issue, we do not propose a solution to address it**. Instead, our findings serve as an important stepping stone—calling for future research to better understand and enhance the unique challenges and opportunities in VLM self-verification, a capability that remains underexplored in the current landscape.

References

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *Preprint*, arXiv:2407.21787.

Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. 2025a. Sft or rl? an early investigation into training r1-like reasoning large vision-language models. *Preprint*, arXiv:2504.11468.

Liang Chen, Lei Li, Haozhe Zhao, Yifan Song, and Vinci. 2025b. R1-v: Reinforcing super generalization ability in vision-language models with less than $3. https://github.com/Deep-Agent/R1-V. Accessed: 2025-02-02.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. 2025. Openvlthinker: An early exploration to complex vision-language reasoning via iterative self-improvement. *Preprint*, arXiv:2503.17352.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *Preprint*, arXiv:2503.01307.

Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, and Lingpeng Kong. 2023. G-llava: Solving geometric problem with multi-modal large language model. *Preprint*, arXiv:2312.11370.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. 2025. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *Preprint*, arXiv:2503.06749.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. 2025. Training language models to self-correct via reinforcement learning. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *Preprint*, arXiv:2305.20050.

Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia. 2025. Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement. *Preprint*, arXiv:2503.06520.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *Preprint*, arXiv:2310.02255.

OpenAI. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

OpenAI. 2025. Gpt-5 system card. Technical report, OpenAI. Accessed: 2025-09-24.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *Preprint*, arXiv:2408.03314.

Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. 2025. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *Preprint*, arXiv:2504.08837.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting. In *Advances in Neural Information Processing Systems*, volume 37, pages 66383–66409. Curran Associates, Inc.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *Preprint*, arXiv:2503.18892.

Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. 2025. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. *Preprint*, arXiv:2503.12937.

Hengguang Zhou, Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. 2025. R1-zero's "aha moment" in visual reasoning on a 2b non-sft model. *Preprint*, arXiv:2503.05132.

A Appendix

Verifier Prompt

Now you act as a judge, helping me determine which of the <length> texts I provide better answers the question.
Question: <question>
Repsonse: <response>
Please strictly follow the following format requirements when outputting, and don't have any other unnecessary words.
Output format: "I choose response [number] because"

AHA Search Prompt

Below is a chain-of-reasoning generated by a Language Model when attempting to solve a math problem. Evaluate this chain-of-reasoning to determine whether it demonstrates beneficial problem-solving behaviors that deviate from typical linear, monotonic reasoning patterns commonly observed in language models.
<start_of_reasoning> <RESPONSE> <end_of_reasoning>
Specifically, actively identify and emphasize beneficial behaviors such as:
(1) Backtracking: Explicitly revising approaches upon identifying errors or dead ends (e.g., "This approach won't work because...").
(2) Verification: Systematically checking intermediate results or reasoning steps (e.g., "Let's verify this result by...").
Additionally, remain attentive to and encourage the identification of other beneficial behaviors not explicitly listed here, such as creative analogies, abstraction to simpler cases, or insightful generalizations.
Important:
Clearly specify each beneficial behavior you identify.
If there is strong example of this, provide <YES> followed by specific explanations. Otherwise, provide <NO>
A positive response example:
<YES> This contains Backtracking and Verification, respectively from "example quote" and "example quote"
A negative response example, no further explanation is needed at all, SIMPLY return <NO>:
<NO>

MathVista Parser Prompt

Please read the following example. Then extract the answer from the model response and type it at the end of the prompt.
Hint: Please answer the question requiring an integer answer and provide the final value, e.g., 1, 2, 3, at the end. Question: Which number is missing?
Model response: The number missing in the sequence is 14.
Extracted answer: 14
Hint: Please answer the question requiring a floating-point number with one decimal place and provide the final value, e.g., 1.2, 1.3, 1.4, at the end.
Question: What is the fraction of females facing the camera?
Model response: The fraction of females facing the camera is 0.6, which means that six out of ten females in the group are facing the camera.

```
Extracted answer: 0.6
Hint: Please answer the question requiring a floating-point number with two
decimal places and provide the final value, e.g., 1.23, 1.34, 1.45, at the
end. Question: How much money does Luca need to buy a sour apple candy and a
butterscotch candy? (Unit: $)
Model response: Luca needs $1.45 to buy a sour apple candy and a butterscotch
candy.
Extracted answer: 1.45
Hint: Please answer the question requiring a Python list as an answer and
provide the final list, e.g., [1, 2, 3], [1.2, 1.3, 1.4], at the end.
Question: Between which two years does the line graph saw its maximum peak?
Model response: The line graph saw its maximum peak between 2007 and 2008.
Extracted answer: [2007, 2008]
Hint: Please answer the question and provide the correct option letter, e.g.,
A, B, C, D, at the end. Question: What fraction of the shape is blue?Choices:
(A) 3/11 (B) 8/11 (C) 6/11 (D) 3/5
Model response: The correct answer is (B) 8/11.
Extracted answer: B
<query><response>
```