

000
001
002
003
004
005
006
007
008
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
035
036
037
038
039
040
041
042
043
044
045
046
047
048
049
050
051
052
053

CONTENTS	
A	EXPERIMENTAL DETAILS 3
A.1	BENCHMARK STATISTICS 3
A.2	BASELINE SETTINGS 3
A.3	ERROR ANALYSIS 4
A.4	ATTENTION VISUALIZATION 5
A.5	INFERENCE-LEVEL REMEDIES 6
A.6	VISUAL CLAIM GENERATION 6
A.7	TRAINING CONFIGURATION 7
B	MORE EVALUATION 8
B.1	ERROR CATEGORY DISTRIBUTION 8
B.2	VISUAL ANCHOR SCORE 8
B.3	PERCEPTION REWARD WEIGHT 8
B.4	ABLATION ON DATA AUGMENTATION 9
B.5	EFFECT OF VISUAL CLAIMS 10
B.6	KL PENALTY COEFFICIENT 10
B.7	VAPO WITH INFERENCE-LEVEL REMEDIES 10
B.8	ATTENTION-BASED REWARD 11
B.9	COMPUTATIONAL EFFICENCY AND COST 11
B.10	LIMITATION ANALYSIS 12
B.11	CLAIM AND LABEL QUALITY 12
B.12	METHOD SENSITIVITY TO OTHER GENERATORS 13
B.13	VAPO ON OTHER MODELS 13
B.14	EVALUATION ON MORE TASKS 14
B.15	VAPO WITH SELF-SUPERVISED ANCHORS 14
B.16	SINGLE-CUT RECOVERABLE RATIO 15
B.17	DATA DEDUPLICATION CHECK 15

054	B.18	EFFECT OF ACCURACY GATE	16
055			
056	B.19	EFFECT OF ANCHOR PLACEMENT STRATEGY	16
057			
058	B.20	COMPARISON WITH MORE BASELINES	17
059			
060	B.21	EFFECT OF VAPO ON LONG-THOUGHT EXAMPLES	17
061			
062	B.22	REASONING LENGTH DISTRIBUTION	18
063			
064	B.23	VAPO ON MULTI-IMAGE TASK	18
065			
066			
067	C.	BROADER SOCIETAL IMPACTS	18
068			
069			
070	D.	SUPPLEMENTARY RESULTS	19
071			
072	D.1	FULL NUMERICAL MAIN RESULTS	19
073			
074	D.2	NUMERICAL RESULTS OF K	19
075			
076	D.3	NUMERICAL RESULTS OF β	19
077			
078	D.4	FULL RESULTS OF AUGMENTED BASELINES	19
079			
080	D.5	FULL RESULTS OF 3B PARAMETER SCALE	19
081			
082			
083			
084			
085			
086			
087			
088			
089			
090			
091			
092			
093			
094			
095			
096			
097			
098			
099			
100			
101			
102			
103			
104			
105			
106			
107			

A EXPERIMENTAL DETAILS

A.1 BENCHMARK STATISTICS

In the main paper, we provide a high-level overview of the selected benchmarks. Here, we present detailed information for clarity and ease of reproducibility, including dataset types, sizes, and splits. All evaluation scripts are implemented using the VLMEvalKit framework (Duan et al., 2024).

1) **MathVerse** is a benchmark comprising 2,612 high-quality mathematical questions. Each example provides varying levels of multimodal information. Following prior work, we evaluate on the Test Mini split using the Vision Only setting, which includes approximately 700 samples.

2) **MathVista** is a comprehensive mathematical benchmark that evaluates a range of skills including puzzle solving, algebraic reasoning, and scientific understanding, and comprises 6,141 examples. For our evaluation, we use the Test Mini split, which contains approximately 1,000 examples.

3) **MathVision** comprises 3,040 high-quality mathematical problems sourced from real-world math competitions, spanning 16 distinct disciplines and five levels of difficulty, providing a comprehensive and challenging benchmark for evaluating VLMs. We conduct our evaluation using the full test set.

4) **LogicVista** assesses the fundamental logical reasoning capabilities of VLMs, covering a range of reasoning types including spatial, deductive, inductive, numeric, and mechanical. The benchmark comprises 448 visual multiple-choice questions. We conduct our evaluation on the full test set.

5) **WeMath** comprises 6.5k visual math questions structured around 67 hierarchical knowledge concepts across five levels of granularity. We evaluate on the Test Mini split, which contains approximately 1,740 examples, and report the strict score as the primary evaluation metric.

6) **Geometry3k** consists of 3,002 geometry problems with dense annotations in formal language, requiring abstract problem-solving and symbolic reasoning based on axiomatic knowledge. For evaluation, we combine the validation and test splits, resulting in approximately 900 examples.

7) **MMMU** is a challenging benchmark that covers a broad range of disciplines, requiring college-level subject knowledge and reasoning. It contains 11.5k curated multimodal questions sourced from college exams, quizzes, and textbooks. We perform our evaluation on the validation split.

8) **MMStar** is a vision-indispensable multimodal benchmark specifically designed to ensure that each sample exhibits strong visual dependency and requires advanced multimodal reasoning capabilities. It comprises 1,500 samples for offline evaluation. Here we evaluate on the full test set.

9) **HallusionBench** is designed to challenge advanced VLMs by emphasizing fine-grained understanding and interpretation of visual information. It consists of 346 manually curated images paired with 1,129 questions. In this study, we conduct our evaluation on the full test split.

10) **MMVet** comprises 218 examples and defines six core vision-language capabilities, focusing on their integration to evaluate the synergy among different skills. It is designed to assess the overall competence of generalist models. We perform our evaluation on the full test split.

A.2 BASELINE SETTINGS

In the main text, we compare our method against several popular baselines to validate its effectiveness. For models such as GPT-5-Thinking, Gemini-2.5-Pro, and InternVL-2.5, we directly report results from the OpenCompass leaderboard. For other models, we provide reproduced results when official numbers are not available in the original papers. Below, we detail the settings used for these reproduced baseline models for reference. All results are reproduced using greedy decoding.

1) **R1-OneVision** includes both 3B and 7B variants, which are first trained via SFT followed by RL, with a particular focus on mathematical reasoning tasks. In our experiments, we adopt the publicly available 7B checkpoint based on Qwen2.5-VL-7B.

2) **VLAA-Thinker** is an RL-only model trained on a high-quality and challenging dataset, and is the first to demonstrate that RL outperforms SFT in multimodal settings. In our experiments, we use the VLAA-Thinker-7B variant, which is also based on Qwen2.5-VL-7B.

ROLE

You are an impartial evaluator. Given an image, a question, the ground truth answer, and the model’s reasoning traces, identify the PRIMARY error category in the reasoning process.

ERROR TYPE (CHOOSE ONE PRIMARY)

- A) Perception Error: Failure to correctly perceive, localize, or read visual content (e.g., chart misreading, counting mistakes, OCR/label misread, visual hallucination/attribution errors).
- B) Logical Error: Symbolic/algorithmic reasoning mistakes given correctly perceived facts (e.g., arithmetic slip, invalid inference, algebraic misstep, inconsistent intermediate steps).
- C) Knowledge Error: Missing/incorrect factual or commonsense/world knowledge not visually available (or clearly contradicted by known facts), including outdated knowledge or misremembered facts.
- D) Others: No observable error in reasoning, or the reasoning content is too short or uninterpretable.

OUTPUT

Return exactly one character: A, B, C or D. Do not include any other text, spaces, punctuation, or quotes.

Figure 1: The designed prompt to instruct GPT-5 to assist in categorizing the different error types.

3) **Vision-R1** is a powerful reasoning model trained at scale, with the combined amount of RL and SFT data being approximately five times larger than that used in our setting. In our experiments, we use Vision-R1-7B, which represents the strongest variant reported by the authors.

A.3 ERROR ANALYSIS

In Section 3.1, we conduct a comprehensive error analysis by leveraging GPT-5 to categorize different types of errors and quantify their distributions, aiming to assess the impact of reasoning on failure cases. The prompt used for GPT-5 input is shown in Fig. 1. As illustrated, we provide GPT-5 with four options: three predefined error categories, along with an Others category to account for a small number of outliers, such as cases with uninterpretable content. We then compute the proportion of each error type to support our analysis. For completeness, in addition to the perception error examples as shown in Fig. 1, here we also provide representative cases of logical errors and knowledge errors, as illustrated in Fig. 2.

In addition, based on the recorded human audit of 500 randomly sampled failure cases, we find that instances containing multiple error types within the same response are quite rare, occurring in fewer

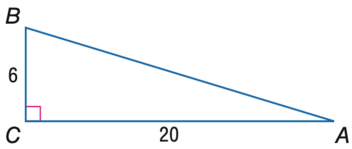
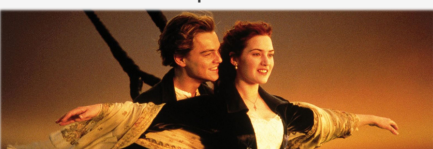
Trigonometry Q: In the image, find $\sin A$. GT Answer: 0.29 	Logical Error (Arithmetic) <think> Okay, to calculate the sine value of $\angle A$, I need to think step by step. 1. From the image, we have $BC=6$, $AC=20$, and triangle ABC is a right triangle. 2. To derive $\sin A$, we first need to determine the length of hypotenuse AB. 3. Given BC and AC, we can derive $AB^2 = BC^2 + AC^2$ 4. Substitute the numbers into the equation, we have $AB^2 = 36 + 400 = 436$ 5. Take the square root of AB^2 , we have $AB = \sqrt{436} = 24.68$ 6. By definition, sine is given by the ratio of the opposite side to hypotenuse 7. Therefore, the sine value of $\angle A$ is $BC/AB = 6/24.68 = 0.24$ </think> <answer> The sine function of $\angle A$ is equal to 0.24 </answer>
Movie Recognition Q: From which film is this photo taken? GT: Titanic 	Knowledge Error (Commonsense) <think> I am asked to identify the film name, let's have a look at the photo. 1. The image features a man and woman standing on the deck of a ship. 2. The man has blond hair and is smiling and is behind the woman. 3. The woman has a red hair, and is wearing a vintage dress. 4. The ropes in the background and the clothing imply a sailing vessel. 5. The sunset palette and romantic pose suggest a fantasy genre. 6. Based on the ship-like setting and dramatic lighting, I might conclude this scene is from Pirates of the Caribbean: Dead Man's Chest. </think> <answer> The film is Pirates of the Caribbean: Dead Man's Chest </answer>

Figure 2: The representative examples of logical and knowledge errors.

Error Category	Logic \rightarrow Perc	Perc \rightarrow Logic	Knowledge \rightarrow Perc	Knowledge \rightarrow Logic
# Examples	5	7	3	1

Table 1: The number of examples with multiple error categories co-occurring.

than 20 examples ($< 4\%$), which is also the reason we prompt GPT to directly identify the primary error category as shown in Fig. 1. To further investigate this, we manually re-examine these cases and record the specific ordering patterns in which different error categories co-occur. As shown in Table 1, any error category combinations not listed above did not occur in our samples. Notably, cases in which logical errors precede perceptual errors are extremely rare. This rarity ensures that such patterns do not meaningfully affect our experimental results or conclusions. It also makes it difficult to draw any substantive connection between perceptual errors and other error types, given how infrequently they co-occur in practice. These results further reinforce our finding that longer reasoning contexts tend to induce visual forgetting, which in turn leads to the observed performance degradation.

A.4 ATTENTION VISUALIZATION

In the main paper, we visualize the attention over image tokens to reflect the contribution of visual information to the model’s reasoning process. Here, we provide additional details on the implementation. For each generation step, we compute the sum of attention scores after softmax assigned by the output token to all preceding image tokens. This yields a ratio between 0 and 1, which is then averaged across all layers to produce the final attention value. Additionally, in Fig. 3 (A), since inserting images, *i.e.*, visual replay, or instructions, *i.e.*, focus prompt, may alter the final reasoning trajectory and result in slightly different sequence lengths, we normalize the comparison by truncating all outputs to the first 250 tokens, which basically covers the full response. This strategy is also applied in Fig. 5 (A) when comparing the visual attention between Vision-RL and VAPO-Thinker.

ROLE

You are given an image. Generate EXACTLY 20 DISTINCT visual claims about THIS image.

DEFINITION: A visual claim is a short, factual statement about something that is CLEARLY VISIBLE in the image (no guesses, no context outside the image).

HARD CONSTRAINTS:

- Each claim should be no more than 20 words.
- Each claim must be obviously verifiable from the pixels.
- One fact per line (no conjunctions; no explanations).
- English only; avoid hedges (“maybe”, “appears”, “likely”).

CORRECTNESS:

- Produce 10 CORRECT and 10 WRONG claims (total = 20).
- Wrong claims must be plausible and possible perception errors
- Do not make logically impossible statements (e.g., “left of itself”).
- Correct and wrong claims should be independent, not just simple opposites of each other.
- Label each correct claim with [CORRECT] at the end of the line.
- Label each wrong claim with [WRONG] at the end of the line.

OUTPUT FORMAT (MUST MATCH EXACTLY):

- Return ONLY the 20 numbered lines, nothing else.
- Numbering style: “1. ...” through “20. ...”
- One line per claim.

Figure 3: The user prompt for GPT-5 to generate visual claims for VAPO.

A.5 INFERENCE-LEVEL REMEDIES

In the main text, we introduce two inference-level remedies, *i.e.*, visual replay and focus prompt, as preliminary strategies to demonstrate the negative impact of visual forgetting on reasoning. For completeness, we provide a detailed description of both approaches here.

In the visual replay strategy, the input image is reintroduced periodically during the model’s reasoning process at regular intervals. In practice, to improve efficiency and prevent exceeding the model’s context length, the reinserted image is downsampled to a lower resolution. Furthermore, the insertion points are selected to satisfy two criteria: (1) uniform segmentation of the reasoning trajectory, and (2) alignment with logical boundaries such as commas, periods, or line breaks to avoid interrupting syntactic or semantic units.

For the focus prompt strategy, at each insertion point, we randomly sample one prompt from a set of three manually designed instructions including “I need to see the image”, “I have to look back” and “Let me verify against the visual input”, to ensure robustness against prompt variability. For alignment, we adopt the nearby insertion positions as used in visual replay, facilitating a consistent comparison between the two methods. It is worth noting that in visual replay, the image is reintroduced as part of the user prompt in a dialogue format, whereas in focus prompt, the instruction is directly injected into the assistant’s response.

A.6 VISUAL CLAIM GENERATION

Our method encourages the model to rely more effectively on visual input by introducing a perception reward, which is derived from the model’s ability to evaluate a set of visual claims during the reasoning process. This evaluation serves as a proxy for assessing the model’s perceptual capability at varying reasoning stages. To support this, we prompt GPT-5 to generate a specified number of visual claims conditioned on the input image. An example of the user prompt used for this purpose is shown in Fig. 3. We require the generated visual claims to be concise factual statements, free from hedging terms such as possibly or appears. Each set of claims must contain an equal number of correct and incorrect statements. Furthermore, the claims are constructed to be independent of the corresponding example question, ensuring that the model must refer to the visual input rather than relying on prior outputs when evaluating their validity. While several generated visual claims have already been presented in Fig. 4, we provide more illustrative examples in Fig. 4 for reference.

In addition, we employ a rule-based filtering mechanism and dynamically generate visual claims. That is, for each training example, any claim that fails to meet the required criteria is discarded and resampled until the target number of valid claims is reached. The filtering process incorporates mul-

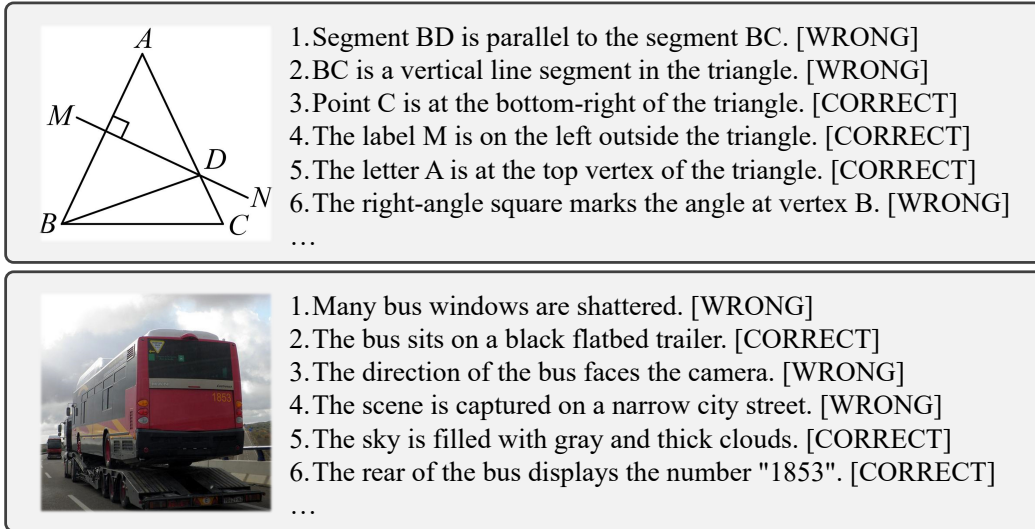


Figure 4: The examples of the GPT-generated visual claims.

multiple constraints, as detailed in the prompt specification in Fig. 3, including limits on claim length, the prohibition of ambiguous expressions (e.g., “maybe”, “appears”), and the requirement to maintain a balanced ratio of correct and incorrect claims. To reduce redundancy, we compute pairwise embedding distances between generated claims using Qwen3-Embedding-0.6B (Zhang et al., 2025), and remove claims with excessively high semantic similarity (> 0.95). We allow straightforward low-level claims such as color naming, while we impose a mild diversity constraint that limits the number of purely low-level color or counting statements per image through regular expression, favoring a richer distribution of relational, text-based, and global scene-level claims.

A.7 TRAINING CONFIGURATION

In Section 5, we briefly outline the training setup. Here, we provide a more detailed description. By default, we adopt ViRL39K as our training dataset, which is a refined collection derived from multiple existing sources such as MM-Eureka (Meng et al., 2025), MV-Math (Wang et al., 2025b), and M3CoT (Chen et al., 2024). The dataset covers a wide spectrum of domains, including STEM subjects, social topics, chart reasoning, and spatial relations. It is worth noting that prior baselines rely on substantially larger datasets that combine SFT and RL, for example, 155k samples for R1-OneVision and 210k samples for Vision-R1. Although this comparison is not fair for us in terms of training data scale, it further highlights the effectiveness of our proposed method.

During the rollout phase, we sample 5 responses per example with a temperature of 1.0, and employ vLLM as the backend to accelerate decoding. We then insert visual anchors into these generated responses to evaluate the model’s perceptual capability. For each anchor, we randomly sample a prefix of the reasoning content, append a visual claim, and instruct the model to judge its correctness. To ensure binary decision-making, the decoding process is constrained to produce either yes or no, with the temperature fixed at 0.0. Importantly, throughout this stage, we adopt the default system prompt of Qwen2.5-VL without explicitly introducing the presence or meaning of anchors. This design avoids altering the model’s inherent behavior and keeps anchors fully transparent, serving as an implicit reward mechanism that encourages the model to rely more heavily on visual cues.

For training, we employ 8 NVIDIA A100-80G GPUs. The policy loss follows the default configuration of GRPO, where the clip ratio ϵ is set to 0.2, and the KL penalty coefficient λ is fixed at $1e^{-2}$. For VAPO, we set the anchor number $K = 20$, the late-emphasis weight $\beta = 1.5$, and the perception reward weight $\gamma = 0.1$. The entire training process is conducted using verl (Sheng et al., 2025).

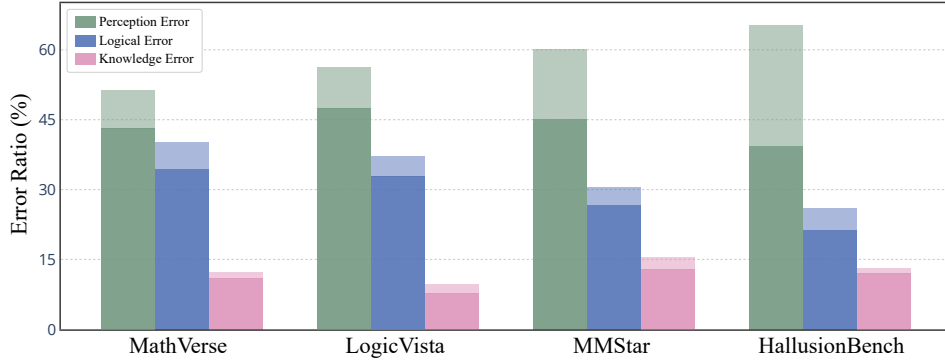


Figure 5: The error ratio of Vision-R1 as well as the correction rate achieved by our method across benchmarks. In the bar chart, the full height of each bar represents the overall error rate of Vision-R1, with the light-colored segment indicating the proportion of errors successfully corrected by our method, and the dark-colored segment corresponding to the remaining uncorrected errors.

γ	MathVerse	LogicVista	Geometry3k	MMMU	MMStar	HallBench	Avg.
0.00	44.2	44.4	44.5	53.6	55.2	51.2	49.0
0.05	47.7	46.3	48.2	55.5	58.6	53.7	51.7
0.10	48.9	47.3	47.7	56.7	59.1	55.5	52.5
0.15	48.4	46.7	48.1	56.2	59.6	56.7	52.4
0.20	47.9	46.6	47.4	55.7	59.4	54.2	51.9

Table 2: The benchmark accuracy with varying perception reward weight γ .

B MORE EVALUATION

B.1 ERROR CATEGORY DISTRIBUTION

In the main text, we analyze the error cases of the representative baseline Vision-R1 and observe that perception errors constitute the majority, underscoring the detrimental impact of reasoning on visual information utilization. To assess the effectiveness of our approach in addressing this issue, we further examine the correction rate of our method across the different error categories of the baseline model, as illustrated in Fig. 5. We observe that, for cases where the baseline model fails, our method substantially corrects a significant portion of perception errors, with improvements reaching up to 20% on vision-intensive benchmarks. This perceptual correction capability constitutes a major source of the performance gains achieved by our approach.

B.2 VISUAL ANCHOR SCORE

Our method assesses the model’s overall perceptual capability by setting up visual anchors at different stages of the reasoning process. As shown in Fig. 6, the perception reward exhibits a clear upward trend throughout training, indicating that the model becomes increasingly proficient at evaluating visual claims and progressively relies more on visual information. To further investigate the role of individual visual anchors, we visualize the distribution of anchor scores before and after training, *i.e.*, comparing the base model with the model after applying our method. As shown in Fig. 6, the base model, *i.e.*, Qwen2.5-VL-7B, exhibits a clear downward trend in anchor scores as reasoning progresses, eventually approaching the level of random guessing (around 0.5). In contrast, our trained model significantly improves anchor scores, maintaining a stable level around 0.9 throughout the reasoning process. This indicates a substantial enhancement in perceptual capability as a result of our method.

B.3 PERCEPTION REWARD WEIGHT

In the main text, we have conducted ablation studies on the anchor number K and the late-emphasis weight β to identify their optimal settings. Here, we further investigate the impact of perception reward weight γ on model performance. For computational efficiency, in this experiment we do not

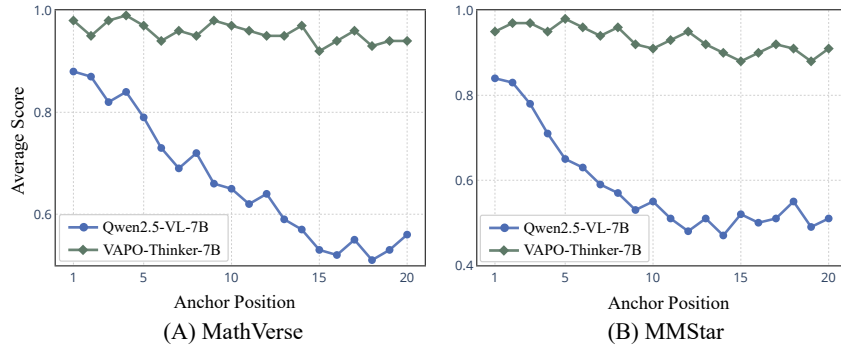


Figure 6: The anchor score across varying anchor positions, where a higher index corresponds to a later stage in the reasoning process. Scores are averaged across all examples within the benchmark.

Method	MathVerse	LogicVista	Geometry3k	MMMU	MMStar	HallBench	Avg.
GRPO	44.2	44.4	44.5	53.6	55.2	52.2	49.0
GRPO _{aug}	44.7	45.2	43.9	54.2	54.8	53.1	49.3
VAPo	45.9	46.1	46.7	55.2	57.8	54.3	51.0

Table 3: The comparison with VAPo and GRPO augmented with visual claim examples.

Method	MathVerse	LogicVista	Geometry3k	MMMU	MMStar	HallBench	Avg.
GRPO	44.2	44.4	44.5	53.6	55.2	51.2	49.0
NonVisualClaim	45.7	44.9	44.3	54.8	57.2	51.9	49.8
QAlignedClaim	44.9	45.8	46.1	55.3	57.8	53.5	50.6
VisClaim	48.9	47.3	47.7	56.7	59.1	55.5	52.5

Table 4: The comparison with non-visually-dependent claims and question-aligned claims.

train on the full dataset but instead randomly sample 5000 examples from the full 39k training set. As shown in Table 2, setting $\gamma = 0$ reduces the training procedure to vanilla GRPO without perceptual supervision. As γ increases, the average accuracy across benchmarks improves significantly, peaking around $\gamma = 0.1$. Notably, mathematical tasks tend to favor smaller values of γ , which is intuitive given the relatively simple visual structures of these tasks. In contrast, vision-intensive benchmarks such as MMStar and HallusionBench benefit from more aggressive settings, reflecting their greater reliance on strong perceptual capability.

B.4 ABLATION ON DATA AUGMENTATION

In our approach, visual claims play a central role by providing reward signals that explicitly encourage the model to rely more heavily on visual inputs. This mechanism helps mitigate visual forgetting and improves accuracy across benchmarks. However, since these visual claims are generated by GPT and accompanied by correctness labels, they may be viewed as new synthetic examples augmented to the original training data. This raises a critical question: are the performance gains brought by VAPo primarily attributable to the reward-driven promotion of visually grounded reasoning, or are they simply a result of data augmentation through the inclusion of these synthetic visual examples?

To investigate this question, we introduce a new baseline in which the generated visual claims are directly transformed into additional training examples and integrated into the original dataset. For computational efficiency, we randomly sample 5000 examples from the full dataset, as described in Appendix B.3. For each example, we generate 5 visual claims, resulting in a fivefold expansion of the training set to 30k examples. This setup allows us to isolate the performance gains attributable purely to data augmentation with synthetic visual claims. The baseline is trained using the standard GRPO algorithm without any additional modifications. In contrast, under the VAPo framework, these same visual claims are used as anchors during training, with the anchor number set to $K = 5$ to match the available number of claims per instance.

As shown in Table 3, augmenting vanilla GRPO with additional examples constructed from visual claims yields only marginal improvement over the original dataset ($49.0 \rightarrow 49.3$). In contrast, our method achieves a substantial performance gain of 2% ($49.0 \rightarrow 51.0$). This discrepancy may be attributed to two key factors: 1) The visual claims are generally short, simple, and strongly dependent on visual information, often solvable without long reasoning trajectories, thus offering limited learning capacity; 2) The augmented examples are derived from a small number of unique images, *i.e.*, five examples share the same image, significantly reducing data diversity and increasing the risk of overfitting. These results suggest that interpreting the role of visual claims purely as a form of data augmentation is not appropriate. Rather, the performance gains of VAPo are primarily driven by its promotion of visually grounded reasoning.

λ ($1e^{-2}$)	MathVerse	LogicVista	Geometry3k	MMMU	MMStar	HallBench	Avg.
0	48.1	47.1	47.3	57.1	59.3	55.3	52.4
1	48.9	47.3	47.7	56.7	59.1	55.5	52.5
2	48.3	47.0	47.9	57.2	58.4	54.9	52.3
5	48.1	46.2	48.3	57.1	59.8	54.3	52.3

Table 5: The impact of KL penalty coefficient λ to the benchmark accuracy.

Method	MathVerse	LogicVista	Geometry3k	MMMU	MMStar	HallBench	Avg.
VAPo	53.3	50.9	51.3	60.2	63.0	57.4	56.0
VAPo _{FP}	53.1	51.2	50.7	60.5	63.4	57.9	56.1
VAPo _{VR}	53.7	50.4	51.8	57.8	62.7	58.2	55.8

Table 6: The impact of VAPo augmented with inference-level remedies.

B.5 EFFECT OF VISUAL CLAIMS

In our method, visual claims serve as a critical proxy for measuring a model’s perceptual capability. To better understand their influence, we investigate how different types of visual claims affect model performance. In addition to our default setting, we consider two alternative variants: 1) non-visually-dependent claims: although these are factual statements about the image, they emphasize external knowledge or logical reasoning rather than concrete visual details; 2) question-aligned claims: these are closely related to the specific question associated with the image, potentially allowing the model to assess claim correctness based on its own historical reasoning outputs rather than pure visual grounding. To generate these claims, we prompt GPT accordingly: for the first variant, we explicitly instruct GPT to produce claims that are not visually dependent; for the second, we provide both the image and the corresponding question, asking it to output claims relevant to the question content. For efficiency, we conduct the analysis using 5000 examples sampled from the full training set.

As shown in Table 4, both the non-visually-dependent and question-aligned variants perform substantially worse than our default visual claim setup. This is likely due to the fact that non-visually-dependent claims can often be judged correctly without requiring strong perceptual capability, thereby diminishing the core purpose of VAPo. On the other hand, question-aligned claims are highly correlated with the question content, allowing the model to infer their correctness from its own reasoning traces without relying on visual input. These results highlight that the design of visual claims is crucial to the effectiveness of the VAPo framework.

B.6 KL PENALTY COEFFICIENT

In the previous sections, we have analyzed the impact of VAPo-specific hyperparameters, including the anchor number K , the late-emphasis weight β , and the perception reward weight γ . For the underlying GRPO framework, we adopt the default setting for the KL penalty coefficient $\lambda = 1e^{-2}$. For completeness, here we also examine the effect of varying λ on the performance of VAPo. To maintain computational efficiency, we conduct this analysis on a randomly sampled subset of 5000 examples from the full training set. As shown in Table 5, VAPo exhibits minimal sensitivity to the choice of λ . Regardless of whether the KL penalty is applied or varied in magnitude, the average accuracy fluctuates by no more than 0.2%.

B.7 VAPo WITH INFERENCE-LEVEL REMEDIES

To further compare the effectiveness of our method with inference-level remedies, *i.e.*, visual replay and focus prompt, we augment VAPo by incorporating these two straightforward approaches. We follow the same insertion strategy as in Fig. 3, where interventions are applied at four points throughout the reasoning process. As shown in Table 6, augmenting our model with inference-level remedies for visual forgetting results in minimal impact, with performance fluctuations within approximately 0.2%. This suggests that our method has already substantially enhanced the model’s reliance on visual input, effectively subsuming the benefits provided by these additional strategies.

Method	MathVerse	LogicVista	Geometry3k	MMMU	MMStar	HallBench	Avg.
GRPO	48.2	45.5	47.3	56.6	58.9	53.2	51.6
VAP _O _{attn}	47.6	45.8	46.4	57.3	59.4	54.8	51.8
VAP _O _{perc}	53.3	50.9	51.3	60.2	63.0	57.4	56.0

Table 7: The comparison with attention reward which directly maximizes visual attention ratio.

Method	Epoch	Time	Accuracy	Gain
GRPO	2	18h46m	51.54	—
DAPO	2	25h11m	53.17	+1.63
VAP _O	2	19h14m	55.91	+4.37

Table 8: The time costs and gains for different policy gradient algorithms.

B.8 ATTENTION-BASED REWARD

In the main text, we use the evolution of visual attention throughout the reasoning process as a proxy to examine whether our method mitigates visual forgetting. Therefore, a natural baseline to consider is a more trivial alternative: directly using the attention scores of image tokens as a reward signal to guide training. To explore this possibility, we introduce an attention reward, which quantifies the model’s overall visual attention ratio across layers during the reasoning process, as detailed in Appendix A.4. We replace the perception reward with this attention-based reward while keeping all other experimental configurations consistent with the main setup, termed as VAP_O_{attn}, whereas our original method using perception reward is referred to as VAP_O_{perc}.

As shown in Table 7, naively maximizing visual attention provides little to no performance gain and even leads to noticeable degradation on certain tasks. We hypothesize that this is due to two key factors: 1) Although visual attention can serve as an indicator of the contribution of visual inputs to the model’s decision-making, it does not directly reflect whether the model is effectively utilizing visual features. Blindly encouraging higher attention may disrupt the base model’s learned distribution; 2) A higher visual attention ratio is not inherently better. As observed in earlier experiments, its values typically lie between 0 and 0.3. Unconstrained maximization of this ratio may lead to instability or training collapse in later stages.

B.9 COMPUTATIONAL EFFICIENCY AND COST

In this section, we analyze the efficiency and computational cost of our method, and compare it against other policy gradient algorithms in terms of training time and performance gains. We consider two representative reinforcement learning baselines: GRPO and DAPO (Yu et al., 2025). The latter is a widely adopted variant of GRPO that incorporates several advanced techniques, including dynamic sampling and higher clipping ratio. As shown in Table 8, under the same data budget and training epochs, DAPO requires approximately 6 additional hours of training time compared to GRPO, whereas our method incurs only a marginal increase of around 30 minutes. More importantly, this modest computational overhead yields a substantial performance gain: our method improves accuracy by 4.37% over GRPO, significantly surpassing the benefits provided by DAPO.

This efficiency is largely attributable to the nature of our perceptual supervision. The key difference of our method from GRPO lies in the anchor scoring process, where the model is asked to evaluate visual claims. Unlike standard rollouts, which require generating full reasoning traces, our method only requires a binary judgment, *i.e.*, yes or no for each claim, essentially a single-token output. Consequently, this process is highly efficient. Moreover, our gradient update procedure remains identical to that of GRPO. These results suggest that our approach introduces minimal additional computational cost while achieving notably greater performance improvements.

Beyond training, In VAP_O, we leverage LLMs, typically GPT to generate visual claims for assessing the model’s perception capability. Here for completeness, we provide below an estimate of the computational time required to generate these visual claims, based on our usage records. As shown in Table 9, the time is mainly spent on querying GPT and on the subsequent filtering process. The

Dataset	Total	Querying	Filtering
ViRL39K	38min	34min	4min

Table 9: The time costs for generating visual claims.

querying stage refers to the waiting time after submitting prompts, which also includes re-queries for generated claims that fail to meet the specified requirements. The filtering stage applies a set of rule-based criteria as described in Fig. 3 to remove claims that do not satisfy our standards, for example, those that are excessively long, contain uncertainty markers such as “maybe” or “appears”, or violate the required balance between correct and incorrect claims. Besides, we also remove redundant claims that are semantically repetitive with embedding models, and limit the number of claims from various types, *e.g.*, color, relational and OCR, through regular expressions. Any filtered-out examples are resubmitted until the target number of valid claims is reached. Overall, the entire process takes roughly half an hour, which is negligible compared with the training phase and remains highly efficient relative to methods such as DAPO even considering the overall pipeline costs. It is important to note that this claim generation process is a one-time procedure, and no further GPT querying is required for subsequent training runs.

B.10 LIMITATION ANALYSIS

Visual Claim Quality. One factor influencing the effectiveness of VAPO is the quality of the generated visual claims. This includes whether the claims are strongly vision-dependent, clearly verifiable from the image, and correctly labeled. As such, the claim generation model, *i.e.*, GPT-5 in our current setup, may become a limiting factor for overall performance. However, this also suggests that VAPO retains great potential for further improvement: leveraging higher-quality visual claims, either generated by more capable models or annotated by human experts, could potentially enhance the effectiveness and further improve the model’s performance ceiling.

Hyperparameter Sensitivity. Our method introduces several new hyperparameters including the anchor number K , the late-emphasis weight β , and the perception reward weight γ , all of which require careful tuning to achieve optimal performance. Moreover, the optimal settings for these hyperparameters may vary across task types: vision-intensive tasks tend to benefit from larger values of β and γ , whereas logic-heavy tasks such as mathematical reasoning often prefer smaller values. Designing an adaptive vision-anchored policy that dynamically adjusts these parameters based on task characteristics remains an important direction for future work.

Single Image Setting. In the current work, we focus exclusively on single-image tasks for simplicity. However, VAPO can be readily extended to multi-image or even video-based tasks by generating visual claims that reference multiple frames or views. Exploring the benefits of vision-anchored training in such settings, particularly in multi-image reasoning or temporal video understanding presents an interesting and promising direction for future research.

B.11 CLAIM AND LABEL QUALITY

Upon obtaining the generated claims, to assess the quality and potential noise of both the visual claims and their corresponding labels, here we conduct a straightforward yet reliable human evaluation as an indicator for their effectiveness prior to training. Specifically, we randomly sample 250 examples from the training set and evaluate each generated claim to determine whether it is visual-dependent, *i.e.*, grounded in the visual cues of the image rather than deducible solely through textual reasoning, and whether it is answerable, *i.e.*, has a clear and unambiguous ground-truth answer. For the labels, we manually verify the accuracy against the corresponding question and visual input. Below, we report the recorded human evaluation statistics for the claims and labels.

As shown in Table 10, the generated claims exhibit high overall quality according to human evaluation, where the noise in the claims occupy only around 2%, *i.e.*, claims that are not visual-dependent or not answerable, and the labeling error rate is below 4%. In addition, for comprehensiveness, here we also perform model evaluation, in which we sample 2000 examples and feed the generated claims and labels back to GPT and prompt it to act as a judge to assess noise and accuracy as defined

Human Evaluation	Visual-Dependent (%)	Answerable (%)	Accuracy (%)
Claims	98.44	99.30	—
Labels	—	—	96.78

Table 10: The human audit of the generated claim and label quality.

Model Evaluation	Visual-Dependent (%)	Answerable (%)	Accuracy (%)
Claims	97.29	99.67	-
Labels	-	-	95.85

Table 11: The model evaluation of the generated claim and label quality.

above. The above results in Table 11 indicate that model and human evaluation of claim quality are closely aligned, suggesting that the generated claims and labels are reliable. As highlighted in our limitation analysis, further reducing the noise and error rates has the potential to yield additional performance gains and further enhance the effectiveness of our approach.

B.12 METHOD SENSITIVITY TO OTHER GENERATORS

In our main experiments, we employ GPT-5 to generate visual claims due to its strong performance and cost-effectiveness. Here we further examine how using open-source, weaker models to generate visual claims influences VAPO’s effectiveness. Specifically, we consider two typical VLMs: LLaVA-1.6-34B (Liu et al., 2023) and Qwen2.5-VL-32B (Bai et al., 2025). For each model, we input the instructions directly as user prompts and apply a filtering mechanism similar to that used with GPT-5 to remove claims that do not meet our requirements. We train VAPO on a subset of 5000 examples randomly sampled from the full training set using the claims generated by these models, and evaluate results on the established benchmarks. As shown in Table 12, despite being open-source models, Qwen2.5-VL achieves performance comparable to those obtained with GPT-5, yielding nearly identical average improvements over GRPO ($\pm 0.2\%$). Similarly, LLaVA-1.6, one of the pioneering VLMs, also attains nearly the full performance achieved with GPT-5 ($\pm 0.5\%$). These results indicate that VAPO does not have to rely on closed-source, state-of-the-art models to be effective; smaller open-source models are capable of delivering almost the same performance gains.

B.13 VAPO ON OTHER MODELS

Following prior work (Huang et al., 2025; Wang et al., 2025a), we use Qwen2.5-VL as a standard base model to evaluate the effectiveness of our method, given its strong and representative reasoning capabilities. We posit that visual forgetting is primarily driven by two factors: 1) Training data. The existing training corpora (e.g., mathematics or text-intensive tasks) place heavy emphasis on textual patterns, leading models to develop a strong text bias; 2) Training objective. The widespread use of next-token prediction objective encourages models to prioritize autoregressive consistency, causing attention to visual inputs to become progressively diluted as the output sequence grows, as shown in Fig. 3 of the main paper. Therefore, unless these underlying factors are substantially altered, we expect that visual forgetting will likely generalize across different model architectures.

To provide a preliminary validation of this assumption, here we additionally experiment with Qwen3-VL-8B (Yang et al., 2025), a more recent and more capable reasoning model whose ar-

Model	MathVerse	LogicVista	Geometry3k	MMMU	MMStar	HallBench	Avg.
GRPO	44.2	44.4	44.5	53.6	55.2	51.2	49.0
LLaVA-1.6	48.1	46.7	47.0	56.5	58.6	55.0	52.0
Qwen2.5-VL	48.5	47.0	47.4	56.7	58.9	55.2	52.3
GPT-5	48.9	47.3	47.7	56.7	59.1	55.5	52.5

Table 12: The VAPO effectiveness given claims generated with other models.

Model	MathVerse	LogicVista	Geometry3k	MMMU	MMStar	HallBench	Avg.
Qwen3-VL	58.8	56.7	59.3	68.4	67.3	59.2	61.8
+ GRPO	63.5	60.1	64.4	72.7	68.3	60.4	64.9
+ VAPO	67.6	65.8	68.7	74.2	72.5	63.9	68.8

Table 13: The results of VAPO on other base models.

Model	V*	RealWorldQA	Avg.
Qwen2.5-VL	71.4	64.3	67.9
R1-OneVision	73.5	65.7	69.6
VLAA-Thinker	74.8	66.3	70.6
Vision-R1	74.1	66.8	70.5
VAPO-Thinker	77.9	69.4	73.7

Table 14: The results of VAPO on more evaluation tasks.

chitecture differ substantially from those of Qwen2.5-VL. We apply VAPO to Qwen3-VL to assess whether our method remains effective under different scenarios. As shown in Table 13, our method continues to provide substantial improvements over GRPO when applied to Qwen3-VL, yielding an average gain of 3.9% (64.9% \rightarrow 68.8%). This suggests that, beyond Qwen2.5-VL, other reasoning models may also suffer from pronounced visual forgetting, and that our method is broadly applicable and effective across different model architectures.

B.14 EVALUATION ON MORE TASKS

During training, our visual claims are designed to span multiple dimensions of perceptual capability, including fine-grained visual details, *e.g.*, “the whiteboard says No Parking”, and spatial reasoning, *e.g.*, “the bus is to the left of the motorcycle”. Consequently, our perception training enhances performance on tasks requiring these capabilities. For completeness, here we further evaluate our model on two popular benchmarks, V* (Wu & Xie, 2024) and RealWorldQA, which target fine-grained perception and spatial reasoning, respectively. As shown in Table 14, our model achieves an average improvement of 3.1% (70.6% \rightarrow 73.7%) over the previous best results on both fine-grained and spatial reasoning benchmarks. This empirically demonstrates that VAPO generalizes well to a broad range of perception-intensive tasks.

B.15 VAPO WITH SELF-SUPERVISED ANCHORS

The core idea of VAPO is to utilize anchors to assess model’s perception capability during reasoning process and use curated rewards to encourage stronger reliance on visual inputs. The specific form of the anchor (*i.e.*, the mechanism used to measure perception), is not essentially limited to binary textual claims. We choose textual claims mainly because 1) efficiency: As discussed in the efficiency analysis, current pipeline introduces only minimal training overhead, offering an excellent cost-performance trade-off; 2) simplicity, unlike reconstruction-based methods which may require additional learnable decoders or more complex architectures, textual claims allow us to probe perception capability with only a single-token output; 3) effectiveness: Our experiments also demonstrate that textual claims significantly improve model accuracy, surpassing previous strong baselines.

When designing the visual anchors, we also considered reconstruction-based anchors, where the model would be required to reconstruct visual features or undergo linear probing to assess its per-

Model	MathVerse	LogicVista	Geometry3k	MMMU	MMStar	HallBench	Avg.
Base Model	40.7	42.6	38.5	52.7	54.9	50.0	46.6
GRPO	44.2	44.4	44.5	53.6	55.2	51.2	49.0
VAPO (Self-Supervised)	48.1	46.5	48.3	55.9	57.9	55.1	51.9
VAPO (GPT)	48.9	47.3	47.7	56.7	59.1	55.5	52.5

Table 15: The results of VAPO with self-supervised anchors.

Cut Position (%)	20	40	60	80	Agg.
Recoverable Ratio (%)	4.5	5.1	8.8	7.6	15.4

Table 16: The single-cut recoverable ratio by early decision.

ceptual capability. However, as mentioned above, these approaches introduce additional learnable parameters, thereby increasing computational and training complexity.

Another promising self-supervised direction is to adopt a bootstrapping approach, in which the model generates its own visual claims and subsequently uses them as training signals, thereby eliminating the need for external models. Here we explore whether this more efficient training strategy for VAPO remains effective. Specifically, instead of relying on GPT to produce visual claims, we allow the model being trained to generate its own visual claims during the rollout step using the provided instructions, and then use these self-generated claims to assess its perceptual capability and update the model accordingly. This procedure is self-supervised and iterative: the model’s generated claims help improve its visual capability, which in turn further refines the quality of subsequent claims. We use Qwen2.5-VL-7B as the base model and train on a 5000 example subset of the full training set. As shown in Table 15, this self-supervised approach retains the vast majority of the effectiveness observed in the default (GPT) setting, with only a modest average difference of 0.7% (46.5% \rightarrow 47.3%). This indicates that, even without relying on any external or stronger models, VAPO can still achieve highly promising performance.

B.16 SINGLE-CUT RECOVERABLE RATIO

Our current definition of the recoverable ratio uses a relatively loose metric intended to estimate the proportion of error cases influenced by the reasoning process, serving as an oracle-style indicator of the potential for our proposed method. Here, we additionally report recoverable ratios at specific single-cut positions. In particular, we provide the recoverable ratios measured at four relative reasoning depths (20%, 40%, 60%, and 80% of the reasoning length) as well as their aggregated result. As shown in Table 16, even when considering only a single reasoning position, the model’s recoverable ratio remains substantial, reaching nearly 10% at the 60% and 80% positions. This indicates considerable headroom for improving the accuracy of current models and helps explain why excessively long reasoning can degrade performance, as observed in Fig. 2. Moreover, we observe that later positions exhibit higher recoverable ratios, suggesting that the latter stages of reasoning have a particularly strong impact on the model’s reliance on visual information, consistent with the visual forgetting phenomenon shown in Fig. 3.

B.17 DATA DEDUPLICATION CHECK

In our training setup, we follow prior work (Wang et al., 2025a;c) by training on ViRL39K and evaluating on both mathematical benchmarks (e.g., MathVista) and general-purpose benchmarks (e.g., MMStar). Here we perform data deduplication checks between the training set and the evaluation benchmarks. Specifically, we compare image hash values across datasets and compute the number of overlapping instances. As shown in Table 17, both the mathematical benchmarks (6 in total) and the general-purpose benchmarks (4 in total) exhibit 0% overlap with the training dataset based on image-hash comparison. This indicates that no duplicate samples exist between the training and evaluation sets, and thus there is no risk of data contamination.

Deduplication Check	Mathematical	General	Total
Overlap Ratio (%)	0	0	0

Table 17: The overlap ratio between training and evaluation set.

Training Steps	0	25	50	75	100
Activation Rate (Easy)	0.53	0.65	0.72	0.76	0.79
Activation Rate (Hard)	0.21	0.44	0.56	0.61	0.64

Table 18: The accuracy gate activation gate across easy and hard examples.

Method	MathVerse	LogicVista	Geometry3k	MMMU	MMStar	HallBench	Avg.
GRPO	44.2	44.4	44.5	53.6	55.2	51.2	49.0
VAPO (strict gate)	48.9	47.3	47.7	56.7	59.1	55.5	52.5
VAPO (relaxed gate)	48.5	47.4	47.2	56.3	59.4	55.0	52.3

Table 19: The results of the curriculum gating strategy.

B.18 EFFECT OF ACCURACY GATE

In training, we incorporate an accuracy gate to prevent reward hacking, where the model might pursue perception rewards at the expense of task accuracy by producing trivially short reasoning. Although the accuracy gate is triggered more frequently on easy problems, resulting in a larger portion of trajectories receiving perception rewards, these rewards are ultimately normalized when converted into advantages. This normalization step ensures that the generally higher rewards obtained from easy examples do not disproportionately influence the final advantage. In contrast, for hard problems, normalization scales up the advantages of correct trajectories, encouraging the model to pay additional attention to these challenging cases. Therefore, the accuracy gate does not bias training toward simpler examples.

To further analyze how the accuracy gate influences the training dynamics of easy and hard problems, here we report the accuracy gate activation rates over the course of training for both categories. We classify problems as easy or hard based on the model’s initial accuracy gate activation rate (> 0.4 for easy and < 0.4 for hard) and track how these activation rates evolve at different training steps. We conduct training on a 5000 example subset of the full training set. As shown in Table 18, the activation rate increases substantially for both easy and hard questions, and the gap between the two progressively narrows. This further indicates that the model does not over or under-optimize either category during training, suggesting balanced learning dynamics across problem difficulties.

Moreover, to investigate whether a relaxed accuracy gate can further improve performance, we experiment with a curriculum gating strategy. Specifically, unlike our default setting, which applies a strict binary gate, we gradually loosen this constraint over the course of training. That is, we progressively assign an increasing proportion of the perception reward to incorrect trajectories and eventually remove the accuracy gate entirely by granting full perception rewards. As before, we conduct this study using a 5000 example subset of the full training set. As shown in Table 19, using a relaxed accuracy gate does not yield noticeable performance improvements. This aligns with our earlier analysis that the default accuracy gate does not under-optimize hard examples, and that relaxing the gate introduces a greater risk of reward hacking.

B.19 EFFECT OF ANCHOR PLACEMENT STRATEGY

Our current anchor placement strategy resembles the early decision scheme, where anchors are inserted at semantic boundaries (e.g., commas, periods, line breaks) randomly throughout the entire reasoning trajectory. In designing this component, we also considered several alternative placement strategies: 1) uniform, where anchors are inserted at fixed intervals rather than randomly; 2) back-loaded, where anchors are inserted only in the latter 60% of the reasoning process, acting as a targeted mechanism to address severe visual forgetting that tends to occur toward the end of long trajectories; 3) front-loaded, where anchors are inserted only within the first 60% of the trajectory, serving as a baseline comparison. We ensure that all three variants insert the same number of anchors ($K = 20$). We train each variant on a 5000 example subset of the full training set, and we report the results of these placement strategies.

Position	MathVerse	LogicVista	Geometry3k	MMMU	MMStar	HallBench	Avg.
GRPO	44.2	44.4	44.5	53.6	55.2	51.2	49.0
Uniform	48.4	47.1	47.5	56.1	58.4	55.0	52.1
Front-loaded	48.1	46.7	47.0	55.9	58.5	54.4	51.8
Back-loaded	48.7	46.9	47.7	56.2	59.0	55.8	52.4
Random	48.9	47.3	47.7	56.7	59.1	55.5	52.5

Table 20: The effect of varying anchor placement strategies.

Method	MathVerse	LogicVista	Geometry3k	MMMU	MMStar	HallBench	Avg.
Base Model	40.7	42.6	38.5	52.7	54.9	50.0	46.6
MM-Eureka	50.3	48.5	50.2	55.7	60.4	54.8	53.3
PAP0	50.4	47.3	49.7	60.8	61.8	55.6	54.2
VL-Rethinker	54.2	47.2	49.6	57.9	61.8	55.4	54.4
VAPO	53.3	50.9	51.3	60.2	63.0	57.4	56.0

Table 21: The comparison between VAPO with more baselines.

As shown in Table 20, the simple naive (random) strategy performs better than other placement variants. We attribute this to several factors: 1) uniform insertion, which places anchors at fixed intervals, is more susceptible to reward hacking, as the model may learn to exhibit high visual dependence only at predetermined positions; 2) front-loaded insertion performs noticeably worse than both random and back-loaded insertion, indicating that later positions in the reasoning trajectory are more critical for improving perceptual grounding; 3) In addition, although back-loaded insertion is slightly weaker than random insertion on average, it achieves comparable or even superior performance on MMStar and HallusionBench. This suggests that vision-heavy benchmarks may benefit from a larger number of anchors or greater weighting at later reasoning steps, which aligns with our late-emphasis design.

B.20 COMPARISON WITH MORE BASELINES

Here, for completeness, we compare our approach with more baselines, including PAP0 (Wang et al., 2025c), MM-Eureka (Meng et al., 2025) and VL-Rethinker (Wang et al., 2025a). We use the publicly released 7B checkpoints from their repositories. For PAP0, we reproduce the results using greedy decoding, as their paper reports performance using the avg@8 accuracy metric, which is not directly comparable to the metrics adopted in prior work (Wang et al., 2025a; Huang et al., 2025). As shown in Table 21, VAPO achieves an average improvement of 1.6% over the previous best results (54.4% \rightarrow 56.0%), further demonstrating the effectiveness of our approach.

B.21 EFFECT OF VAPO ON LONG-THOUGHT EXAMPLES

Our ablation study in Fig. 7 shows that increasing K , *i.e.*, using more anchors, generally yields positive gains across examples. To further examine whether longer-thought examples benefit disproportionately from larger K , we partition benchmark examples into two groups based on the output length of our VAPO-Thinker-7B model: short-thought (< 300 tokens) and long-thought (> 300 tokens). We then evaluate models trained with different values of K and analyze how performance trends vary across these two categories. As shown in Table 22, increasing K leads to stable performance improvements for both short- and long-thought examples, and the magnitude of these gains

Length	1	5	10	15	20
Short	53.7	56.0	57.5	58.4	58.5
Long	47.3	49.4	50.9	51.5	51.8
Total	51.5	53.8	55.1	55.8	55.9

Table 22: The results of VAPO on short and long-thought examples.

Models	MathVerse	LogicVista	Geometry3k	MMMU	MMStar	HallBench	Avg.
R1-OneVision	299	308	287	263	232	201	265
Vision-R1	337	293	341	283	239	251	291
VAPO-Thinker	315	287	334	275	254	238	284

Table 23: The average reasoning length of VAPO.

Method	BLINK	MuirBench	Avg.
Base model	55.3	58.1	56.7
GRPO	57.0	60.4	58.7
VAPO	60.1	62.2	61.2

Table 24: The results of VAPO on multi-image tasks.

remains largely consistent across the two groups. This trend is also aligned with the ablation results reported in Fig. 7.

B.22 REASONING LENGTH DISTRIBUTION

The goal of VAPO is to enhance the effectiveness of reasoning by improving the model’s perceptual capability, rather than to alter the length of its reasoning traces. Accordingly, our pipeline does not incorporate any explicit length regularization. Here we report the average reasoning length of our models compared with other baselines across the established benchmarks. As shown in Table 23, the average output length of our model does not differ noticeably from that of other baselines. This rules out the possibility that our performance gains stem merely from longer reasoning traces, and instead indicates that VAPO improves the effectiveness of the reasoning process itself.

B.23 VAPO ON MULTI-IMAGE TASK

Our current visual claims are primarily designed for single-image inputs, which is sufficient to validate the core motivation of VAPO, namely improving the model’s reliance on visual information and mitigating visual forgetting. Here we further conduct a small-scale pilot study on multi-image tasks. Specifically, we train VAPO exclusively on the multi-image subset of ViRL39K, which contains approximately 2,500 examples. For claim generation, we consider two types of claims. The first type is single-image claims, which mirror our original setup and are generated with respect to only one image in the multi-image set. The second type is cross-image claims, which capture relationships or distinctions across images. For the former, we provide GPT with only one image at a time. For the latter, we present all images simultaneously and extend the original instructions with an additional requirement that the generated claim must reference information across images (e.g., “the two images depict the same building”). We ensure that each training example contains a balanced number of claims from both categories. The training procedure remains identical to our main pipeline, and GRPO is used as the baseline for comparison. We evaluate the trained models on two representative multi-image benchmarks, including BLINK (Fu et al., 2024) and MuirBench (Wang et al., 2024).

As shown in Table 24, despite being trained on a limited amount of data, our method still achieves an average improvement of 2.5% over GRPO on multi-image tasks (58.7% \rightarrow 61.2%). This demonstrates that our approach is not restricted to the single-image setting and can generalize to multiple images as well. We leave the evaluation of video-based tasks as promising future work.

C. BROADER SOCIETAL IMPACTS

Our work carries several positive societal implications. Our findings reveal that existing VLMs often produce reasoning that is not visually grounded, which can degrade performance and reliability. Such ungrounded reasoning may introduce misleading or hallucinated content, thereby posing risks to model safety and trustworthiness. Moreover, since these reasoning traces often rely heavily on statistical patterns learned from training data, they may inadvertently expose sensitive information,

raising concerns about data privacy and potential leakage. In contrast, the proposed VAPO framework steers the model’s reasoning process to remain anchored in visual evidence. This grounding mechanism helps ensure that model outputs are more faithful to the input image and less likely to include irrelevant or speculative content, thus promoting both factual accuracy and privacy preservation. At present, we have not identified negative societal impacts associated with this work. However, due to external factors such as the availability of datasets and baseline implementations, further assessment may be necessary in the future.

D. SUPPLEMENTARY RESULTS

D.1 FULL NUMERICAL MAIN RESULTS

Due to space limitations in the main text, some baseline results are omitted. Here, we provide the full numerical results of our method and all baselines across the ten established benchmarks in Table 25, corresponding to Table 1 and Table 2.

D.2 NUMERICAL RESULTS OF K

Here we report the numerical results of the ablation study on the anchor number K in Table 26, corresponding to the visualization in Fig. 7 (A).

D.3 NUMERICAL RESULTS OF β

Similarly, we present the numerical results of the ablation study on the impact of β in Table 27, corresponding to Fig. 7 (B).

D.4 FULL RESULTS OF AUGMENTED BASELINES

Here we provide the full numerical results of our method compared with baselines augmented with inference-level remedies, *i.e.*, visual replay and focus prompt, in Table 28, corresponding to Table 3 in the main paper.

D.5 FULL RESULTS OF 3B PARAMETER SCALE

Here for completeness, we report the evaluation results of our proposed method and baseline models at the 3B scale. To ensure fairness, we select baselines with comparable model sizes and exclude those without publicly available 3B-scale checkpoints. The results are presented in Table 29.

Models	MathVerse	MathVista	MathVision	LogicVista	WeMath	Geo3k	MMMU	MMStar	HallBench	MMVet	Avg.
<i>Close-source Models</i>											
GPT-5-Thinking	81.2	81.9	72.0	70.0	71.1	79.9	81.8	75.7	65.2	77.6	75.6
Gemini-2.5-Pro	76.9	80.9	69.1	73.8	78.0	77.2	74.7	73.6	64.1	83.3	75.2
<i>Open-source Models</i>											
Qwen2.5-VL-7B	40.7	62.3	23.2	42.6	33.1	38.5	52.7	54.9	50.0	64.8	46.3
InternVL2.5-8B	34.5	68.2	25.6	38.3	38.6	44.8	56.2	63.2	49.0	62.8	48.1
R1-OneVision-7B	46.4	64.1	29.9	45.6	44.6	46.1	54.3	54.1	52.5	65.2	50.3
VLAA-Thinker-7B	48.2	68.0	26.4	48.5	41.5	50.6	59.1	49.7	54.7	70.0	51.7
Vision-R1-7B	52.4	73.5	28.2	49.7	41.6	49.0	57.6	61.4	49.5	71.1	53.4
<i>Our Models</i>											
VAPo-Thinker-3B	35.8	67.1	23.9	39.7	35.4	44.2	55.6	59.4	49.5	64.6	47.5
VAPo-Thinker-7B	53.3	75.6	31.9	50.9	43.6	51.3	60.2	63.0	57.4	71.9	55.9

Table 25: The full numerical results of main experiments, corresponding to Table 1 and Table 2.

1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133

K	MathVerse	MathVista	MathVision	LogicVista	WeMath	Geo3k	MMMU	MMStar	HallBench	MMVet	Avg.
0	48.2	70.6	26.1	45.5	39.1	47.3	56.6	58.9	53.2	69.9	51.5
5	50.9	73.1	28.4	47.8	41.2	50.5	58.4	61.3	55.6	70.4	53.8
10	52.8	74.8	30.7	49.5	42.5	51.2	59.9	62.2	56.7	71.2	55.1
15	53.2	75.8	31.5	50.4	43.9	51.0	60.4	62.7	57.0	71.6	55.8
20	53.3	75.6	31.9	50.9	43.6	51.3	60.2	63.0	57.4	71.9	55.9

Table 26: The numerical results of ablation study of K , corresponding to Fig. 7 (A).

1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187

β	MathVerse	MathVista	MathVision	LogicVista	WeMath	Geo3k	MMMU	MMStar	HallBench	MMVet	Avg
0.0	51.7	73.8	28.4	48.6	42.0	49.9	58.9	60.2	53.8	70.3	53.8
0.5	52.7	74.8	30.2	49.6	42.8	50.7	59.5	62.0	55.4	71.3	54.9
1.0	53.1	75.7	31.5	51.3	43.4	51.4	60.0	62.5	56.8	72.0	55.8
1.5	53.3	75.6	31.9	50.9	43.6	51.3	60.2	63.0	57.4	71.9	55.9
2.0	53.0	75.5	31.1	50.5	43.1	51.5	60.0	63.3	56.6	71.6	55.6
2.5	51.8	74.8	30.4	50.0	42.6	50.5	59.8	62.2	55.3	70.2	54.7

Table 27: The numerical results of ablation study of β , corresponding to Fig. 7 (B).

Models	MathVerse	MathVista	MathVision	LogicVista	WeMath	Geo3k	MMMU	MMStar	HallBench	MMVet	Avg
VLAA-Thinker-7B	48.2	68.0	26.4	48.5	41.5	50.6	59.1	49.7	54.7	70.0	51.7
+ Focus Prompt	48.8	68.4	27.3	49.2	41.8	50.7	59.7	51.1	55.2	70.7	52.3
+ Visual Replay	49.8	69.8	27.9	49.9	42.3	51.1	60.5	52.9	56.2	71.7	53.2
Vision-R1-7B	52.4	73.5	28.2	49.7	41.6	49.0	57.6	61.4	49.5	71.1	53.4
+ Focus Prompt	52.8	73.8	29.3	50.2	42.1	49.7	58.7	61.8	50.5	71.4	54.0
+ Visual Replay	53.4	75.2	29.9	50.7	42.5	50.5	59.4	62.1	51.8	71.9	54.7
VAPo-Thinker-7B	53.3	75.6	31.9	50.9	43.6	51.3	60.2	63.0	57.4	71.9	55.9

Table 28: The full results of baselines augmented with test-time remedies, corresponding to Table 3.

1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295

Models	MathVerse	MathVista	MathVision	LogicVista	WeMath	Geo3k	MMMU	MMStar	HallBench	MMVet	Avg.
Qwen2.5-VL-3B	30.5	60.3	19.8	37.6	20.4	31.7	48.6	51.5	44.0	58.4	40.3
InternVL2.5-2B	22.3	51.1	14.0	27.3	8.0	36.1	43.2	54.3	42.3	62.6	36.1
R1-OneVision-3B	35.5	—	23.7	—	—	—	—	—	—	—	—
VLAA-Thinker-3B	36.4	61.0	24.4	38.5	33.8	42.7	50.5	56.2	45.1	63.9	45.3
VAPo-Thinker-3B	38.2	64.5	27.3	41.4	35.3	44.2	51.6	59.4	49.5	64.6	47.6

Table 29: The results of our 3B model and baselines of comparable scale. “—” indicates unavailable results due to unreleased checkpoints. Vision-R1 is excluded as it does not consider 3B scale.

REFERENCES

- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibor Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. M3cot: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. *ACL*, 2024.
- Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *ACM MM*, pp. 11198–11201, 2024.
- Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *ECCV*, pp. 148–166. Springer, 2024.
- Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*, 2025.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 36:34892–34916, 2023.
- Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, et al. Mm-eureka: Exploring the frontiers of multi-modal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025.
- Guangming Sheng, Chi Zhang, Zilinfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.
- Fei Wang, Xingyu Fu, James Y Huang, Zekun Li, Qin Liu, Xiaogeng Liu, Mingyu Derek Ma, Nan Xu, Wenxuan Zhou, Kai Zhang, et al. Muirbench: A comprehensive benchmark for robust multi-image understanding. *arXiv preprint arXiv:2406.09411*, 2024.
- Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhui Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *arXiv preprint arXiv:2504.08837*, 2025a.
- Peijie Wang, Zhong-Zhi Li, Fei Yin, Dekang Ran, and Cheng-Lin Liu. Mv-math: Evaluating multi-modal math reasoning in multi-visual contexts. In *CVPR*, pp. 19541–19551, 2025b.
- Zhenhailong Wang, Xuehang Guo, Sofia Stoica, Haiyang Xu, Hongru Wang, Hyeonjeong Ha, Xiushi Chen, Yangyi Chen, Ming Yan, Fei Huang, et al. Perception-aware policy optimization for multi-modal reasoning. *arXiv preprint arXiv:2507.06448*, 2025c.
- Penghao Wu and Saining Xie. V?: Guided visual search as a core mechanism in multimodal llms. In *CVPR*, pp. 13084–13094, 2024.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.
- Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.