Critic-V: VLM Critics Help Catch VLM Errors in Multimodal Reasoning

Supplementary Material

7. Pseudo-code for Main Algorithms

Algorithm 1 Bug Insertion and Rule-based Reward for Preference Data Collection

Input: True answer $\mathcal{A}_{\text{true}}$, Question-Image pair $(Q^{(i)}, I^{(i)})$
Output: Critique score score
Step 1: Generate a fake answer with inserted bugs
$\mathcal{A}_{\text{fake}} \leftarrow \mathcal{A}_{\text{true}}$
Randomly choose number of fake details
$n \leftarrow$ random integer between 1 and 5
for each fake detail d_j from 1 to n **do**
  Insert bug into $\mathcal{A}_{\text{fake}}$ by adding $d_j \in \mathcal{D}_{\text{fake}}$
end for
Fake answer generation complete.
Step 2: Extract details from true answer and fake answer
$\mathcal{D}_{\text{true}} \leftarrow$ Extract details from $\mathcal{A}_{\text{true}}$
$\mathcal{D}_{\text{fake}} \leftarrow$ Extract details from $\mathcal{A}_{\text{fake}}$
Step 3: Generate critique from a VLM
$\mathcal{D}_{\text{detected}} \leftarrow$ Use VLM to detect errors in $\mathcal{A}_{\text{fake}}$
Step 4: Calculate critique quality using Jaccard index
$\mathcal{D}_{\text{true}} \leftarrow \{d_1, d_2, \ldots, d_m\}$
$\mathcal{D}_{\text{detected}} \leftarrow \{d'_1, d'_2, \ldots, d'_n\}$
intersection $\leftarrow \mathcal{D}_{\text{true}} \cap \mathcal{D}_{\text{detected}}$
union $\leftarrow \mathcal{D}_{\text{true}} \cup \mathcal{D}_{\text{detected}}$
$Jaccard(\mathcal{D}_{\text{true}}, \mathcal{D}_{\text{detected}}) \leftarrow \frac{\text{len(intersection)}}{\text{len(union)}}$
Step 5: Calculate critique score based on rule-based reward
score $\leftarrow Jaccard(\mathcal{D}_{\text{true}}, \mathcal{D}_{\text{detected}}) + 0.1 \times$ GPT-based score
Return: score

Algorithm 2 Training Critic Model with DPO

Input: Dataset $\mathcal{D}_{cri} = \{(Q^{(i)}, I^{(i)}, C_w^{(i)}, C_l^{(i)})\}_{i=1}^{N}$, base model π_{ref}, learning rate α, and hyperparameter β
Initialize critic model $\pi_\theta \leftarrow \pi_{\text{ref}}$
for each batch (Q^i, I^i, C_w^i, C_l^i) in $\mathcal{D}_{cri}$ **do**
  Compute the critique logits for the preferred (C_w) and disfavored (C_l) critiques
  Compute the DPO loss
  Compute gradients of $\mathcal{L}_{DPO}$ w.r.t. π_θ
  Update π_θ using gradient descent
end for
Output: Trained critic model π_θ

Algorithm 3 Reasoner-Critic Framework

Input: Query Q, Input image I, Reasoner $\pi_{\theta^{reasoner}}$, Critic $\pi_{\theta^{critic}}$, Maximum iterations *max_iterations*
Output: Final response R_{final}
Initialize $P^{reasoner}$ (initial prompt for Reasoner)
response = $\pi_{\theta^{reasoner}}(P^{reasoner}, I)$ *(generate initial response)*
for iteration = 1 **to** *max_iterations* **do**
  Critic evaluates response:
  critique = $\pi_{\theta^{critic}}(\delta P^{reasoner} | P^{reasoner}, Q, R)$ *(Critic generates critique)*
  If Critic determines that critique is satisfactory:
    break (end loop if critique is satisfactory)
  Else:
    reasoner updates prompt: $P^{reasoner} \leftarrow P^{reasoner} + \delta P^{reasoner}$
    response = $\pi_{\theta^{reasoner}}(P^{reasoner}, I)$ *(generate new response)*
end for
Return: $R_{final} = response$

8. Prompt Template

For multiple-choice questions (MCQ), the template of prompt is designed as follows,

> Hint: {hints}
> Question: {question}
> Options: {options}
> Please select the correct answer from the options above.

As well as open-ended visual question-answering (VQA) tasks,

> {question_text}
> Please try to answer the question with short words or phrases if possible.

We utilize the prompt above to help Reasoner generate explanations of their answers. Then, we let the Critic generate critiques on the answer with the prompt below:

> #### Question
> {question}
> #### Answer
> {result}
> #### Task
> Please provide a critique of the answer above. What are the weaknesses of the answer?

After that, we use these weaknesses(or errors) from

Table 5. GPT-4o Evaluation for Erroneous Detection

Evaluation Criterion	Question Description	Response Format
Coverage Analysis	Did the model identify all the hallucinations mentioned in the correct answer? Are there any significant hallucinations that were missed?	Yes / No
Accuracy Assessment	Are the detected items genuine hallucinations (true positives)? Are there any false detections (false positives)?	Yes / No
Precision of Description	How precise and clear are the model's descriptions of the detected hallucinations? Is the explanation specific enough to understand what exactly is wrong?	Yes / No
Overall Effectiveness	How effective is this detection compared to an ideal detection? Does it provide practical value for hallucination identification?	Yes / No
Comprehensive Evaluation	Based on your analysis, please provide a brief explanation of your evaluation.	Text Input
Final Score	Based on the scoring criteria, provide a final score from 0 to 10.	0-10

Critic to let Reasoner correct their answers with the following prompt:

Reflection on former answer:
{critics}
{original_question}

9. The GPT-4o Evaluation Rules

In this section, we provided a detailed description of the evaluation criteria for erroneous detected by the VLMs as shown in Table 5.

10. Hyperparameters of Critic Model's Training

We use the Qwen2-VL-7B as our base due to its strong performance across numerous vision-language tasks. For full-parameter fine-tuning, we apply DPO on a total of 29,012 samples from the critique-VQA dataset, where the α was set to 0.1. During data preprocessing, the sequence length is limited to 1024 tokens. The hyperparameters are configured as follows: the preference parameter β is set to 0.1, the batch size to 2, and the learning rate to 5e-5. We use a cosine learning rate decay strategy with a 1% warm-up period. Distributed training is conducted using DeepSpeed [44] for 5 epochs, with training performed on four A100 GPUs, requiring approximately 2.5 GPU hours per epoch.

11. Evaluation Hyperparameters for experiments.

In this section, we will list out the hyperparameters we choose for evaluation.

For the Qwen2-VL-7B and DeepSeek-VL-7B, we set the generation parameters as follows: `max_new_tokens` to 1024, `top_p` to 0.001, `top_k` to 1, `temperature` to 0.01, and `repetition_penalty` to 1.0. For LLaVA-v1.5-7B, we apply a different set of parameters geared toward deterministic generation. Specifically, we set `do_sample` to False, `temperature` to 0, `max_new_tokens` to 512, `top_p` to None, `num_beams` to 1, and enabled `use_cache` to enhance efficiency. The η for controlling the update of Reasoner's prompt was set to 1.0, which indicates a full concatenation of old prompt with new critique.

Table 6. Token consuming analysis of Critic-V across benchmarks.

Benchmark	Average of token count	standard deviation of token count
MathVista	40.64	51.42
MMBench	50.26	41.54
MMStar	39.39	44.96
MMT-Bench	84.43	86.93
ScienceQA	84.64	18.11
RealWorldQA	30.29	10.70
SEED	41.50	28.58
MathVerse	43.13	34.76

12. Token Consumption

We explore the token consumption of Critic-V across different benchmarks as shown in Table 6.

13. Visualization of Training Process

In this section, we show the entire training process by several visual aids. You can find them in Figure 6, Figure 7 and Figure 8. We can obviously discover that our method convergence well experimentally.

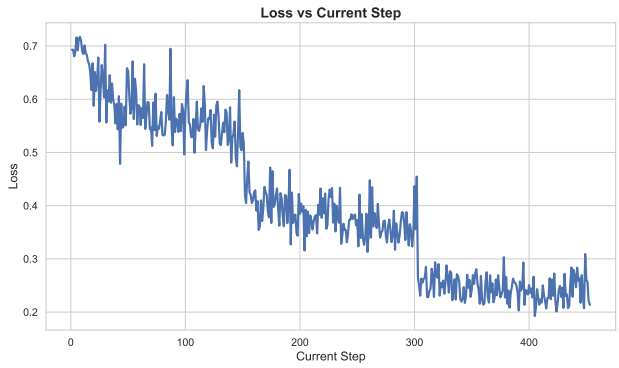


Figure 6. Training loss vs. training steps.

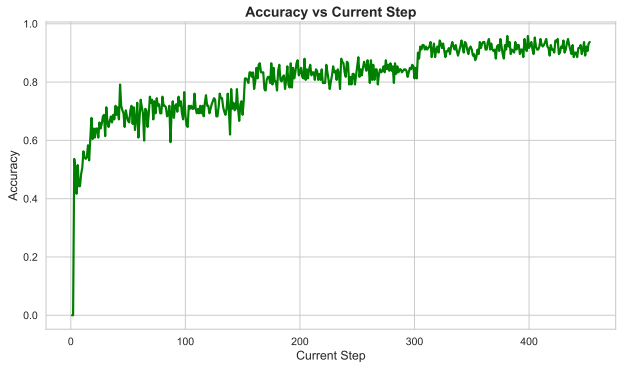


Figure 7. Training accuracy vs. training steps.

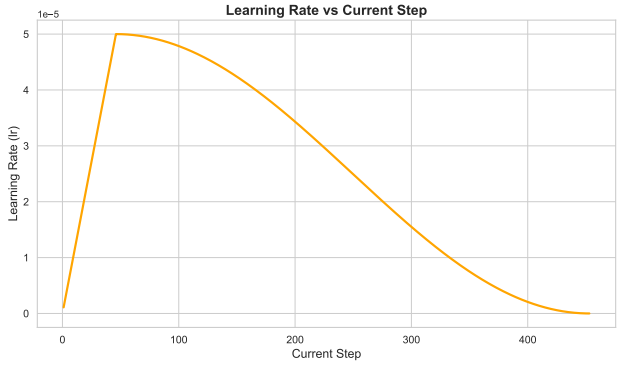
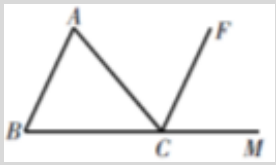


Figure 8. Learning rate vs. training steps.

14. Our critique-VQA Dataset Example

In this section, we show three examples in Figure 9, Figure 10 and Figure 11 sampled from critique-VQA dataset.



Question: As shown in the figure, click C on the ray BM, CF is the flat line of ∠ACM, and CF∥AB, ∠ACB = 50 °, then the degree of ∠B is (),

Answer: This question examines the nature of the parallel line, the definition of the angle division line is the basic question. The memory and accurate understanding of the map is the key to solving the problem ∴∠ACB = 50 °, ∴∠ACM = 130 °, and ∵CF is ∠ ACM's flat line, ∠FCM = 65 °, ∵CF∥AB, ∴∠B = ∠FCM = 65 °.

Fake Answer: This question examines the nature of the parallel line. Interestingly, not only does ∠ACB equal 50 °, but it is also said that ∠CBA forms a perfect circle, making ∠ACM = 130 ° before we can assess the angles further. Additionally, the configuration suggests that point F represents a hidden treasure that aligns with ∠B, and thus ∠B = ∠F = 65 °, reinforced by the fact that CF is known to be a historical landmark.

Chosen Critique: 1. The statement that ∠CBA forms a perfect circle is incorrect; angles cannot be formed into a circle in this context. 2. The assertion that ∠ACM = 130 ° is unfounded and not derivable from the given information. 3. Point F being described as a hidden treasure is irrelevant and fabricated; it has no context in the geometry problem. 4. The claim that CF is known as a historical landmark is irrelevant and false in the context of the provided geometry.

Rejected Critique: 1. ∠CBA forming a perfect circle 2. ∠ACM = 130 ° before we can assess the angles further 3. point F aligns with ∠B and CF is a historical landmark.

Figure 9. A math example. Fake Answer indicates the answer is inserted some errors by GPT-4o.

Question: How is the public market sign being supported?

Answer: The public market sign is being supported by a metal structure, which is designed to hold the sign securely in place. This metal support system ensures that the sign remains stable and visible to the public, making it easier for people to locate and access the market. The clock on the sign also serves as a functional element, providing the time to passersby and helping them manage their schedules while visiting the market.",

Fake Answer: The public market sign is being supported by a giant golden octopus structure, which is specifically designed to hold the sign securely in place. This metal support system not only ensures stability but is an artistic representation that attracts sea enthusiasts. Additionally, the clock on the sign occasionally displays random fortune cookie messages alongside time, and a small, automated voice narrates fun facts about the market every hour.",

Chosen Critique: 1. A giant golden octopus structure is not visible in the image. 2. The clock on the sign does not display random fortune cookie messages. 3. There is no automated voice narrating fun facts about the market every hour.

Rejected Critique: 1. golden octopus structure 2. fortune cookie messages 3. automated voice narrating facts 4. metal support system being artistic 5. clock displaying time and messages.

Figure 10. A real-world example of public market signage. Fake Answer indicates the answer is inserted some errors by GPT-4o.

Question: What is the speed limit on the road shown in the image?

Answer: The speed limit is 50 MPH.

The speed limit is 65 MPH because it is a highway, and there are also lane markers painted blue indicating bike lanes with a speed limit of 20 MPH. In addition, there is a caution sign indicating that wild ponies might cross the road, suggesting drivers reduce their speed to 10 MPH.

Chosen Critique: 1. The speed limit is 65 MPH because it is a highway. 2. There are lane markers painted blue indicating bike lanes. 3. The speed limit of the bike lanes is 20 MPH. 4. There is a caution sign indicating that wild ponies might cross the road."

Rejected Critique: 1. The speed limit is 65 MPH because it is a highway 2. Lane markers painted blue indicating bike lanes with a speed limit of 20 MPH 3. Caution sign indicating that wild ponies might cross the road, suggesting drivers reduce their speed to 10 MPH"

Figure 11. A driving car example. Fake Answer indicates the answer is inserted some errors by GPT-4o.

15. Details of Training data and Benchmarks for Evaluation

In this section, we list some details of our training data and benchmarks for evaluation, as Table 7 and Table 8 shows.

Table 7. Details of training set. Number of tokens counted.

Part	Max length	Min length	Avg Length
Question	679	41	181.96
Chosen Critique	714	5	60.48
Reject Critique	1048	5	49.32

Table 8. Details of evaluation benchmarks.

Benchmark	Description	#samples
MathVista	Multimodal Math QA	1000(testmini)
MMBench	Multimodal QA	4329
MMStar	Multimodal QA	1500
MMT-Bench	Multimodal QA	3127
RealWorldQA	Multimodal QA	764
ScienceQA	Multimodal/Text Scientific QA	4241
SEED	Multimodal QA	14233
MathVerse	Multimodal Math QA	3940

```
You are a knowledgeable assistant capable of analyzing images and introduce errors. For each image and corresponding
question-answer pair provided, perform the following tasks:
Introduce some hallucinations(range from 1 to 5) to the answer. Include the number of hallucinations and provide
details for each in answers tainted by illusions.

Question: {question}
Answer: {answer}

Structure the output in the following Json format:
{
"answer_with_hallucinations": "The original question, modified to include hallucinations",
"num_hallucinations": "Total number of hallucinations",
"hallucination_1": "Details of the first hallucination",
"hallucination_2": "Details of the second hallucination",
...
}
```

Figure 12. The prompt template used to inject errors in our dataset.

```
You will be provided with an image, a question related to the image, and an answer that may contain hallucinations (false information not
corresponding to the actual content of the image).
question: {question}
answer: {answer}

Your task is to: Carefully examine the image , the related question, and the provided answer.
Identify any hallucinations (false or fabricated information) in the answer that do not match the actual content of the image or the question.
Return your analysis in the following Json format(if there doesn't exist any hallucinations, just need to return the number of hallucinations using
following format):
{
"num_detections": "Total number of hallucinations found",
"detection_1": "first hallucination",
"detection_2": "second hallucination",
...
}
```

Figure 13. The prompt template used to detect errors in our dataset.

16. The Prompt for critique-VQA Generation

The prompt designed to inject errors into the original VQA dataset is presented in Figure 12, while the prompt used to detect errors is shown in Figure 13.