MMBoundary: Advancing MLLM Knowledge Boundary Awareness through Reasoning Step Confidence Calibration

Anonymous ACL submission

Abstract

In recent years, multimodal large language models (MLLMs) have made significant progress but continue to face inherent challenges in multimodal reasoning, which requires multi-level (*e.g.,* perception, reasoning) and multi-granular (*e.g.,* multi-step reasoning chain) advanced inferencing. Prior work on estimating model confidence tends to focus on the overall response for training and calibration, but fails to assess confidence in each reasoning step, leading to undesirable hallucination snowballing. In this work, we present **MMBoundary**, a novel framework that advances the knowledge boundary awareness of MLLMs through reasoning step confidence calibration. To achieve this, we incorporate complementary textual and cross-modal self-rewarding signals to estimate confidence at each step of the MLLM reasoning process. In addition to supervised fine-tuning MLLM on this set of self-rewarded confidence estimation signal for initial confidence expression warm-up, we further introduce a reinforcement learning stage with multiple reward functions for further aligning model knowledge and calibrating confidence at each reasoning step, enhancing reasoning chain self-correction. Empirical results show that MMBoundary significantly outperforms existing methods across diverse domain datasets and metrics, achieving an average of 7.5% reduction in multimodal confidence calibration errors and up to 8.3% improvement in task performance.[1]

1 Introduction

Although multimodal large language models (MLLMs) demonstrate exceptional abilities in cross-modal reasoning, the reliability of their responses remains uncertain due to the inherent challenges of multimodal reasoning (Chiang et al., 2023; Zhou et al., 2023; Huang et al., 2024a; Chen et al., 2024). In particular, erroneous knowledge can occur not only at the cross-modal reasoning level but also in the early stages of visual perception. However, MLLMs typically fail to explicitly indicate their uncertainty to avoid the propagation and amplification of errors knowledge (Liu et al., 2024a; Huang et al., 2024b; Bai et al., 2024; Guan et al., 2024). Therefore, it is crucial to enable MLLMs to accurately express confidence for each reasoning step during inference, enhancing reasoning chain self-correction.

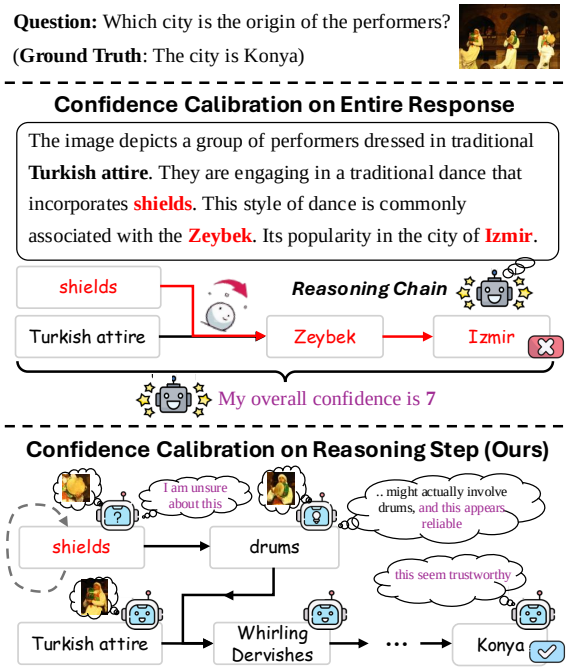

Figure 1: Confidence calibration on reasoning step enables MLLMs to express natural language confidence statements during inference, enhancing self-correction of low-confidence steps and ultimately reasoning toward correct answers. Traditional methods calibrate model confidence solely on entire response, which can lead to incorrect answers with high confidence. Due to space limitations, only the reasoning chain of our method is presented. The red and purple color indicates incorrect knowledge and confidence estimates, respectively.

[1] Our code will be released in the final version.

Prior work on estimating model confidence tends to focus on the overall response for training and calibration (Yang et al., 2023; Zhang et al., 2023; Lyu et al., 2024; Xu et al., 2024). However, these methods fail to enable the trained models to express confidence estimates for different knowledge within generated content. As shown in Figure 1 (upper part), the trained MLLM generates incorrect information at the visual perception level (*i.e.,* misidentifying the "drum" as a "shield") without expressing its uncertainty, causing significant deviations in reasoning chain and ultimately producing an incorrect answer. Moreover, due to the logical coherence of the reasoning, the model still generates a high confidence score in its overall response.

Therefore, in this work, we propose **MMBoundary**, a reinforced fine-tuning framework for advancing MLLM knowledge boundary awareness by reasoning step confidence calibration. Our method enables the model to express natural language confidence statement for each generated sentence, enhancing reasoning chain self-correction by scaling inference-time. Specifically, we introduce a confidence estimation module that integrates three effective text-based uncertainty methods—namely, length-normalized log probability, mean token entropy, and tokenSAR—and incorporates cross-modal constraint (*i.e.,* CLIPScore) to model the self-rewarded confidence signal from the perspective of its internal states. Then, we propose a mutual mapping between the detected score and predefined confidence statements to achieve two objectives: (1) by inserting confidence statements after the associated knowledge and training the model via supervised learning, we enable the model to naturally generate natural language statements for each sentence, similar to human expression; (2) by integrating internally detected confidence scores and those converted from model expressed statements into the reward modeling for reinforcement learning, we can achieve further confidence calibration, reducing the inaccuracy of model-expressed confidence. Moreover, we annotate the reference reasoning chains of training data to facilitate rigorous evaluation of MLLMs' knowledge at different reasoning levels, and incorporate model knowledge calibration into the reward modeling, encouraging MLLMs to faithfully express confidence while improving response quality.

Experimental results from both automatic and human evaluations across diverse domain datasets demonstrate that MMBoundary significantly reduces confidence calibration errors while simultaneously enhancing task performance.

The contributions of our work can be summarized as follows:

- We present a novel framework, MMBoundary, for advancing the knowledge boundary awareness of multimodal language models through reasoning step confidence calibration.
- We introduce a confidence estimation module to flexibly obtain confidence scores for each generated sentence, and propose a confidence score-statement mapping to contribute to training the model to naturally output confidence statements and help in reward modeling of reinforcement learning for further confidence calibration.
- Empirical results show that MMBoundary significantly outperforms existing methods, achieving an average reduction of 7.5% in multimodal confidence calibration errors and up to 8.3% improvement in task performance.

2 Problem Formulation

Given a model $\pi_{\boldsymbol{\theta}}$ with parameter $\boldsymbol{\theta}$, prior work focuses on enabling the model to output a confidence estimate for its entire response, formalized as:

$$\mathbf{y} = [z_1, z_2, \ldots, z_T, c] \tag{1}$$

Here, z_t represents the t-th generated sentence, c denotes the overall confidence estimate, T is the total number of sentences in the response. However, the trained model often assign high confidence incorrectly. Therefore we aims to train models to express fine-grained confidence estimate for each sentence during inference for enhancing reasoning chain self-correction. Thus, the output:

$$\mathbf{y} = [z_1, c_1, z_2, c_2, \ldots, z_T, c_T] \tag{2}$$

Each pair (z_t, c_t) represents the t-th sentence generated by the model and its corresponding confidence statement, respectively.

3 Methodology

Our framework consists of two stages: the confidence expression warm-up stage and the reinforcement learning stage, as shown in Figure 2.

3.1 Confidence Expression Warm-Up

3.1.1 Internal Confidence Estimation

Previous work primarily relies on model response consistency as a confidence indicator. However,

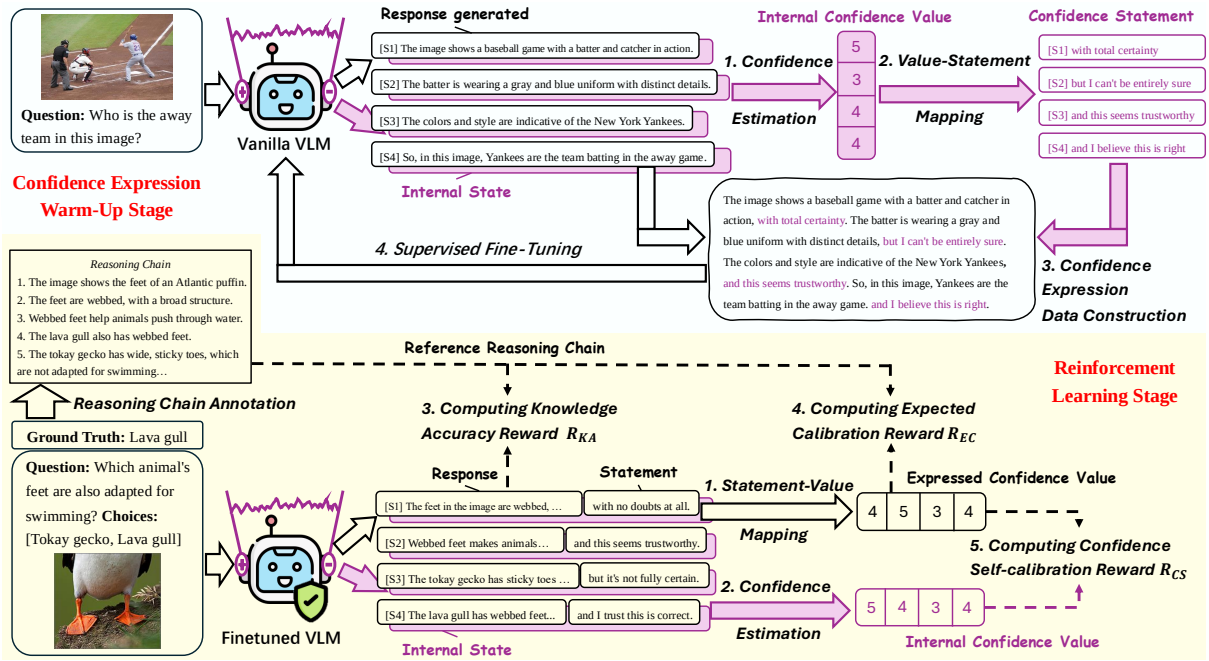

Figure 2: The overview of **MMBoundary**, which consists of two stages. The initial stage trains MLLMs via supervised learning to generate natural language confidence statement for each sentence, similar to human expression. The second stage employs reinforcement learning with three intuitively designed reward functions to further calibrate the expressed confidence estimates and enhance knowledge alignment. ■ represents the internal states (i.e., the log probability of tokens) of model and the estimated internal confidence.

these methods fail to assess confidence across distinct knowledge in generated content and do not consider the correlation between the response and the visual information, limiting their applicability in multimodal scenarios. In this section, we propose to leverage multiple text-based uncertainty methods and incorporate visual constraint to estimate model's confidence. Drawing on recent research (Xiao et al., 2022; Fadeeva et al., 2023; Vashurin et al., 2024), we utilize the following efficient and effective uncertainty estimation methods to create our confidence indicator:

(1) ***Length-normalized log probability*** calculates the average negative log probability of the tokens generated:

$$U_{\text{LNLP}}(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}) = \exp\left\{-\frac{1}{L}\log P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})\right\}, \quad (3)$$

where $\mathbf{x}$ denotes the input, $\mathbf{y}$ denotes the output, and $\boldsymbol{\theta}$ represents the model parameters.

(2) ***Mean token entropy*** (Fomicheva et al., 2020) computes the average entropy for each token in the generated sentence:

$$U_{MTE}(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}) = \frac{1}{L}\sum_{l=1}^{L} \mathcal{H}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta}), \quad (4)$$

where $\mathcal{H}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta})$ is an entropy of the token distribution $P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta})$.

(3) ***TokenSAR*** (Duan et al., 2024) computes the weighted average of the negative log probability of generated tokens, considering their relevance to the entire generated text. For a given sentence similarity function $g(\cdot, \cdot)$ and token relevance function $R_T(y_k, \mathbf{y}, \mathbf{x}) = 1 - g(\mathbf{x} \cup \mathbf{y}, \mathbf{x} \cup \mathbf{y} \setminus y_k)$, the resulting estimate is computed as:

$$U_{\text{TokenSAR}}(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}) = -\sum_{l=1}^{L} \tilde{\mathrm{R}}_T(y_l, \mathbf{y}, \mathbf{x}) \log P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta}), \quad (5)$$

where $\tilde{\mathrm{R}}_T(y_k, \mathbf{y}, \mathbf{x}) = \frac{\mathrm{R}_T(y_k, \mathbf{y}, \mathbf{x})}{\sum_{l=1}^{L} \mathrm{R}_T(y_l, \mathbf{y}, \mathbf{x})}$.

(4) ***CLIPScore*** (Hessel et al., 2021) evaluates the relevance between the generated sentence and input image. Since CLIP's vision encoder aligns with the target MLLM's, we employ CLIPScore to represent the sentence-image uncertainty. For an image with visual CLIP embedding v and a sentence with textual CLIP embedding s:

$$U_{\text{CLIPScore}}(\mathbf{v}, \mathbf{s}) = \max\left(\cos(\mathbf{v}, \mathbf{s}), 0\right) \quad (6)$$

We normalize U_{i} across the entire dataset using min-max normalization to ensure their values are within the range [0, 1]. Then, we compute the final weighted average as:

Score	Confidence Statement
1	but I can't confirm this. / I'm uncertain about this. (…)
2	and it may need checking / it might not be right. (...)
3	but I can't guarantee perfection. / I can't be entirely sure (...)
4	and this seems trustworthy. / and I believe this is right. (...)
5	with total certainty. / with no doubts at all. (...)

Figure 3: We preset a confidence statement pool for each confidence score. The five levels correspond to uncertain, slightly uncertain, moderately confident, highly confident, and fully confident. More statements are shown in Appendix A.

$$U_{\text{Final}} = w_0 U_{\text{LNLP}} + w_1 U_{\text{MTE}} + w_2 U_{\text{TokenSAR}} + w_3 U_{\text{CLIPScore}} \quad (7)$$

where w_i are the respective weights for each component. The closer U_{Final} is to 0, the greater the certainty of model. Then, we use the distribution of U_{Final} to define confidence levels for the model, considering the uneven distribution of U_{Final}. Confidence levels C_{v} from 5 to 1 correspond to the intervals of U_{Final} as [0, μ - σ, μ + σ, μ + 2σ, μ + 3σ, 1], with higher confidence levels indicating greater model confidence. Here, μ and σ represents the average and the standard deviation of U_{Final}. We further validate the effectiveness of this confidence level classification method in Section 5.2.

3.1.2 Confidence Score-Statement Mapping

This module, as shown on the right side of Figure 2, aims to establish a mutual mapping between the detected score and predefined confidence statements. First, we construct statement pools for each confidence level, as shown in Figure 3. These statements can be naturally appended to the end of sentences, providing a concise expression of the model's confidence estimates, similar to human expression. During the *Confidence Expression Warm-Up Stage*, we randomly select statements from the corresponding pools based on the detected scores and insert them into the model's original response to create data for fine-tuning. In the *Reinforcement Learning Stage*, after obtaining the confidence statements from the model's output, we encode these statements into vectors using an encoder model[2] and compute the cosine similarity with all embeddings in the different confidence pools to achieve reverse mapping of statements to confidence scores.

[2] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

3.1.3 Supervised Fine-Tuning

Specifically, the model undergoes fine-tuning on our constructed data $\mathcal{D}$ consisting of tuples: $(\mathbf{x}, \mathbf{y})$, where the input $\mathbf{x}$ comprises an image I and a question Q. At step s_t, the sentence with its confidence statement (z_t, c_t) are generated by model's policy $\boldsymbol{\pi_\theta}$. The next state s_{t+1} is:

$$s_{t+1} = \begin{cases} x, & t = 0 \\ [s_t, z_t, c_t], & 1 \le t \le T \end{cases} \quad (8)$$

We fine-tune the vanilla model via supervised learning. The loss function can be written as:

$$\mathcal{L}_{FT}(\theta) = -\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}} \left[\sum_{t=1}^{T} \log \pi_\theta(z_t, c_t | s_t) \right] \quad (9)$$

3.2 Reinforcement Learning

As noted by Xu et al., 2024, the model undergoing supervised training tends to generate uniform confidence levels, which may impact task performance. Therefore, we employ reinforcement learning with reward signals involving model knowledge alignment, internal confidence and external confidence calibration to encourage model to faithfully express confidence while simultaneously improving the quality of responses. Specifically, we sample questions from the training data and prompt the model to generate responses.

(1) Knowledge Accuracy Reward evaluates whether the knowledge in generated response is aligned with annotated reference chain, thereby ensuring the reliability of the generated content. Specifically, if the t-th generated sentence z_t matches the knowledge in reference chain, R_{KA_t} is 1. Refer to the "Step Matched" example in Figure 6. After evaluating all generated sentences, the reward is normalized: $R_{KA} = \frac{1}{T}\sum_{t=1}^{T} R_{KA_t}$, T is the total number of sentences.

(2) Expected Calibration Reward is consistent with Xu et al. (2024), but we extend it to sentence-level. This reward function measure the correlation between the expressed confidence and the ground truth. The reward function is formalized as follows:

$$R_{EC} = \frac{1}{T}\sum_{t=1}^{T}[1 - 2 \cdot (\mathbb{I}(z_t) - \text{EV}(c_t))^2] \quad (10)$$

where $\mathbb{I}(\cdot)$ is the indicator function that returns 1 if the sentence is correct compared with reference chain, and 0 otherwise. $\text{EV}(\text{c}_\text{t})$ represents

the expressed confidence score, which is obtained by mapping and normalizing the confidence statements generated by the model. The confidence score is normalized between 0 and 1.

(3) Confidence Self-Calibration Reward is based on the consistency between the expressed confidence and internal confidence of MLLMs:

$$R_{\text{CS}} = \frac{1}{T}\sum_{t=1}^{T}[1 - 2 \cdot (\text{IV}(z_t) - \text{EV}(c_t))^2] \quad (11)$$

where $\text{IV}(z_t)$ represents the internal confidence score, which is estimated by our method in Secion 3.1.1. This reward encourages the model to express its confidence level as accurately as possible, aligning its external expression with internal belief. Thus, the overall reward for response is:

$$R = \alpha R_{KA} + \beta R_{EC} + \gamma R_{CS} \quad (12)$$

Lastly, we employ the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017) for training. The model's policy objectives is:

$$\mathcal{L}_{RL}(\boldsymbol{\theta}) = -\mathbb{E}_{\mathbf{y}\sim\boldsymbol{\pi}_{\theta_{\text{old}}}}\left[\min\left(\frac{\boldsymbol{\pi}_{\boldsymbol{\theta}}(z_t, c_t|s_t)}{\boldsymbol{\pi}_{\theta_{\text{old}}}(z_t, c_t|s_t)}\hat{A}_t,\right.\right.$$
$$\left.\left.\text{clip}\left(\frac{\boldsymbol{\pi}_{\boldsymbol{\theta}}(z_t, c_t|s_t)}{\boldsymbol{\pi}_{\theta_{\text{old}}}(z_t, c_t|s_t)}, 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right] \quad (13)$$

The advantage estimate $\hat{A}_t$ (Schulman et al., 2015) is derived by calculating the difference between the anticipated future rewards under the current policy and the baseline or value function. Implementation and data details can be found in Appendix B.

4 Experiments

4.1 Dataset

In order to evaluate the model's robustness and generalizability across diverse scenarios, we select the following datasets from different domains: **A-OKVQA** (Schwenk et al., 2022), a general domain dataset designed to evaluate models on complex visual question answering tasks involving multi-hop reasoning, commonsense understanding, and external knowledge integration; **ScienceVQA** (Lu et al., 2022), a large-scale multimodal dataset designed for science question answering, featuring questions across natural science, social science, and language science; **CulturalVQA** (Nayak et al., 2024), a visual question-answering benchmark evaluating MLLMs on understanding geo-diverse cultural concepts beyond general scene understanding.

4.2 Reasoning Chain Annotation

To simultaneously calibrate the model's knowledge and confidence levels, we conduct detailed reasoning chain annotations for each question in the training dataset. As shown in Figure 5, for each question, we prompt the GPT-4o to generate analysis (inference chain) structured in the perception and reasoning level. The former identifies key visual elements in the image that are most relevant to the question and answer, while the latter provides granularity reasoning that justifies why the answer is correct. Each level should include concise, interconnected sentences, with each sentence conveying a single piece of knowledge. Then, we perform filtering and quality evaluation to ensure the accuracy and consistency. Due to space limitations, please refer to Appendix C for more details.

4.3 Evaluation Metrics

Consistent with previous research (Chen et al., 2022; Xu et al., 2024), We evaluate models from three perspectives using six different metrics:

(1) Confidence Calibration Performance: We adopt 3 calibration metrics. First, we use the Expected Calibration Error (**ECE**) score (Guo et al., 2017). Then, we extend the ECE score to measure the confidence calibration error of each knowledge within reasoning chain, which we refer to as *Multi-granularity Expected Calibration Error* (**MECE**). The MECE score evaluates the correlation between the confidence estimates expressed in generated sentences and their corresponding correctness, as shown in Figure 6. Details of MECE computation process is in the Appendix D.1. For all responses A generated by MLLMs:

$$\text{MECE} = \frac{1}{|A|}\sum_{a\in A}\frac{1}{|a|}\sum_{(z,c)\in a}|\mathbb{I}(z) - \text{conf}(c)| \quad (14)$$

Here, a represents the model's response to the question, while z and c represent the sentences in the response and their corresponding confidence statements, respectively. $\text{Conf}(\cdot)$ represents the numerical value of the confidence statement.

(2) Task Performance: We adopt 2 metrics. First, we measure the typical ***Accuracy***. Second, to identify model responses containing erroneous knowledge and mitigate the risk of them being assigned high confidence, we evaluate the quality of the model's reasoning chain by employing the metric in ***Reasoning Chain F1*** score (Ho et al., 2022). This metric compares the information contained in

Model	A-OKVQA				ScienceVQA				CulturalVQA			
	ECE (↓)	MECE (↓)	Acc (↑)	F1 (↑)	ECE (↓)	MECE (↓)	Acc (↑)	F1 (↑)	ECE (↓)	MECE (↓)	Acc (↑)	F1 (↑)
DPV	0.563	0.582	0.650	0.512	0.593	0.611	0.582	0.414	0.624	0.650	0.334	0.482
DPS	0.511	0.554	0.675	0.535	0.574	0.575	0.575	0.427	0.572	0.594	0.354	0.485
SC	0.435	0.492	0.701	0.548	0.463	0.534	0.602	0.442	0.491	0.554	0.371	0.511
Multisample	0.413	0.430	0.683	0.543	0.446	0.500	0.596	0.434	0.463	0.505	0.362	0.538
SaySelf	0.345	0.384	0.734	0.603	0.386	0.462	0.633	0.483	0.375	0.437	0.417	0.571
CSR	0.408	0.437	0.785	0.618	0.453	0.503	0.694	0.502	0.472	0.513	0.435	0.582
RCE	0.361	0.394	0.788	0.620	0.413	0.475	0.671	0.497	0.408	0.453	0.412	0.577
DRL	0.395	0.453	0.746	0.614	0.476	0.513	0.654	0.485	0.453	0.502	0.392	0.564
MMBoundary	**0.316**	**0.304**	0.835	**0.661**	0.354	**0.392**	0.703	**0.565**	**0.337**	**0.361**	**0.448**	**0.665**
Warm-Up Stage												
w/o U_{LNLP}	0.327	0.343	0.815	0.642	0.369	0.421	0.698	0.548	0.356	0.397	0.423	0.639
w/o U_{MTE}	0.337	0.332	0.824	0.653	0.385	0.441	0.682	0.536	0.349	0.378	0.430	0.643
w/o U_{TSAR}	0.324	0.358	0.813	0.627	0.352	0.417	0.694	0.546	0.361	0.403	0.438	0.655
w/o U_{CLIPS}	0.337	0.354	0.806	0.631	0.372	0.435	0.681	0.532	0.358	0.394	0.426	0.637
w U_{Max}	0.362	0.386	0.774	0.583	0.391	0.467	0.663	0.494	0.378	0.425	0.403	0.589
w/o S-$S_{Mapping}$	0.340	0.362	0.793	0.634	0.377	0.443	0.684	0.531	0.365	0.398	0.427	0.602
Reinforcement Learning Stage												
w/o R_{KA}	0.325	0.347	0.802	0.629	**0.334**	0.410	0.686	0.535	0.348	0.370	0.437	0.632
w/o R_{EC}	0.332	0.357	0.819	0.635	0.363	0.426	**0.712**	0.548	0.359	0.392	0.422	0.648
w/o R_{CS}	0.343	0.368	**0.857**	0.648	0.372	0.449	0.693	0.556	0.368	0.417	0.436	0.640
w/o RL	0.392	0.427	0.768	0.581	0.419	0.481	0.663	0.495	0.408	0.456	0.419	0.595

Table 1: The evaluation results of models and various ablations of our framework. CulturalVQA is the out-of-distribution dataset. **w/o** $\mathbf{U}_{LNLP}$, **w/o** $\mathbf{U}_{MTE}$, **w/o** $\mathbf{U}_{TSAR}$, and **w/o** $\mathbf{U}_{CLIPS}$ represent MMBoundary without the three text-based uncertainty estimation methods and visual information uncertainty estimation, respectively; **w** $\mathbf{U}_{Max}$ indicates the confidence determined using the max pooling method from the four uncertainty estimation scores; **w/o S-**$\mathbf{S}_{Mapping}$ denotes MMBoundary without confidence score-statement mapping; **w/o** $\mathbf{R}_{KA}$, **w/o** $\mathbf{R}_{EC}$, and **w/o** $\mathbf{R}_{CS}$ represent MMBoundary without knowledge accuracy reward, expected calibration reward, and confidence self-calibration reward, respectively; **w/o RL** denotes MMBoundary without reinforcement learning.

Model	A-OKVQA				ScienceVQA				CulturalVQA			
	Faithful	Concise	Granular	Avg.	Faithful	Concise	Granular	Avg.	Faithful	Concise	Granular	Avg.
Multisample	4.20	5.17	4.06	4.47	4.77	5.24	4.53	4.85	3.91	5.63	4.72	4.75
SaySelf	7.28	**7.49**	6.47	7.08	7.49	7.18	6.28	6.98	7.12	6.81	6.58	6.83
CSR	6.47	5.73	5.82	6.01	6.74	5.61	6.40	6.23	6.38	5.45	5.86	5.89
RCE	6.73	6.58	7.41	6.90	7.55	6.92	7.12	7.19	6.84	6.19	6.82	6.62
DRL	6.54	6.13	6.95	6.54	6.81	5.97	6.34	6.37	6.55	5.63	6.07	6.08
MMBoundary	**7.83**	7.25	**8.18**	**7.75**	**8.35**	**7.46**	**8.02**	**7.94**	**7.66**	**7.17**	**8.26**	**7.69**

Table 2: The human evaluation results of strong baselines and our framework.

the predictions and references. We present implementation details in the Appendix D.3.

(3) Human Evaluation: Automated model evaluation may not accurately capture the subtle differences between different responses (Goyal et al., 2022; Ho et al., 2022). Therefore, we conduct additional manual evaluation. We provide a panel of three graduate students with 50 random entries from each setting, asking them to evaluate whether each entry meets the following criteria and to give a score from 1 to 10, consistent with (Xu et al., 2024). 1) ***Faithful***: whether the response faithfully expresses the confidence; 2) ***Concise***: whether the response conveys necessary information clearly and without excess; 3) ***Granularity***: whether the response contains confidence estimates for distinct knowledge. The final result is the average score.

4.4 Baselines

We compare with the following methods: (1)**DPV**: direct prompting the vanilla MLLMs to give a response with a confidence score; (2) **DPS**: direct prompting the vanilla MLLMs to give a response with a confidence statement; (3) **SC** (Xiong et al., 2023): deriving the confidence estimates of MLLMs based on diverse sampling; (4) **Multisample** (Yang et al., 2023): training MLLMs to generate confidence estimates that align with

Dateset	Mean	Var	Std	<0.1
A-OKVQA	0.0443	0.0015	0.0397	96%
ScienceVQA	0.0578	0.0014	0.0374	93%
CulturalVQA	0.0522	0.0015	0.0397	94%

Table 3: Comparison between our internal confidence estimation (ICE) and widely adapted self-consistency-based estimation (SCE). We compute $|C_{\text{ICE}} - C_{\text{SCE}}|$ to demonstrate the correlation between the two methods.

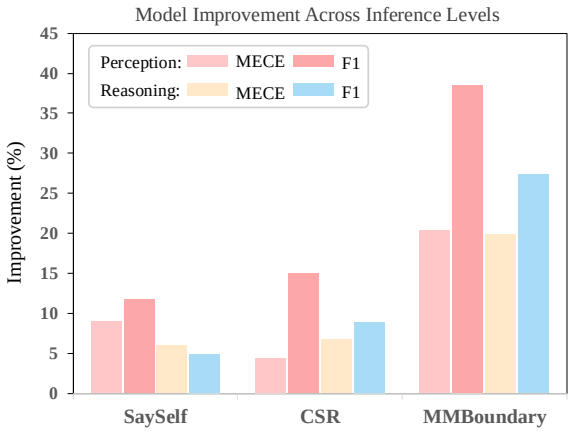

Figure 4: Performance improvement of strong baselines and our model compared to the base model in visual perception and cross-modal reasoning level of MLLMs. We report the results on ScienceVQA.

the confidence derived from self-consistency; (5) **SaySelf** (Xu et al., 2024): analyzing inconsistencies in multiple sampled responses, with the resulting data used for supervised fine-tuning and then confidence estimates calibrated through reinforcement learning based on task supervision; (6) **CSR** (Zhou et al., 2024): converting the calibrated reward into the model's confidence score and utilize DPO (Rafailov et al., 2024) for optimization; (7) **RCE**: training the model to first generate a complete response and then produce confidence estimates for each sentence; (8) **DRL**: directly employing our reinforcement learning method to train model. We use LLaVA-NEXT 7B (Liu et al., 2024b) as backbone model for all methods. We also conduct experiments on Qwen2VL 7B (Wang et al., 2024) in Appendix E.

4.5 Main Experimental Results

Confidence Calibration Performance. We present the ECE and MECE results in Table 1 and the AUROC results in Appendix (Table 8), which measure the correlation between the expressed confidence and the ground truth. The findings indicate that MMBoundary outperforms other methods in reducing confidence calibration errors and enhancing the ability to distinguish confidence between correct and incorrect answers (AUROC). This conclusion is validated on both in-distribution datasets (AOKVQA and ScienceVQA, with a rating increase of 7.5%) and out-of-distribution datasets (CulturalVQA, showing an increase of 6.6%), highlighting the generality of our framework.

Task Performance. We comprehensively evaluate the task performance of the model using the final answer accuracy and the Reasoning Chain F1 score, as presented in Table 1. The results show that our method surpasses other baselines across three datasets, achieving up to 8.3% improvement in CulturalVQA. Unlike CSR and SaySelf, which rely solely on task-oriented reward or the expected calibration reward, our approach integrates knowledge alignment along with internal and external confidence calibration into reward modeling. The results demonstrate that our framework improves the model's knowledge boundary awareness while simultaneously enhancing its task performance. We conduct paired t-tests on the experimental results of MMBoundary, showing significant advantages over the baselines (p-value < 0.05).

Human Evaluation. We conduct human evaluation of the responses generated by our method and other strong baselines across the dimensions of Faithful, Concise, and Granular, with the results shown in Table 2 and Figure 7. We observe that our framework demonstrates statistically significant improvements over three dimensions. SaySelf performs well in the concise dimension for content, but it is designed only to estimate confidence for the entire response, lacking the ability to generate confidence for each step of the reasoning process. We perform a Kappa test on the faithfulness evaluation results to assess inter-annotator agreement, obtaining a Kappa value of 0.79.

5 Discussion

5.1 Influence of Different Components

We conduct extensive ablation study to verify the effectiveness of different components, with results shown below Table 1. Compared to the version without RL, supervised fine-tuning enable model to express confidence during inference. Incorporating RL with our reward signals further improves confidence precision, with an average 9.2% increase in the MECE score. For different rewards, R_{KA} primarily affects task performance, removing it re-

sults in an average performance drop of 3.1%. In contrast, the removal of R_{EC} and R_{CS} leads to a maximum decrease of 6.4% in the confidence calibration performance. The U_{TSAR} having the most significant impact on confidence calibration. Moreover, the conversion between confidence scores and statements (S-S$_{Mapping}$) positively impacts the model's confidence calibration, resulting in an average improvement of 4.8%.

5.2 Effectiveness of Confidence Estimation

To evaluate the effectiveness of our proposed internal confidence estimation (ICE), we compare our method with the self-consistency-based confidence estimation method (SCE) (Yang et al., 2023; Xu et al., 2024). We randomly sample 50 data from three datasets and compare the confidence scores of the model's responses from approaches above. We compute $|C_{\text{ICE}} - C_{\text{SCE}}|$. The results are shown in Table 3. We observe that the confidence estimation bias between the two methods is small for the vast majority of samples (over 93% are less than 0.1). On the ScienceVQA dataset, the average difference in confidence scores between the two methods is 0.0578, indicating that for a given answer from the model, our method has only about 6 instances of deviation compared to the post-hoc confidence estimation method (based on 100 resamplings).

5.3 Effectiveness of Confidence Calibration

We further investigate the confidence calibration (MECE) and task performance (F1) of our method across different reasoning levels in MLLMs, specifically focusing on visual perception and cross-modal reasoning. The results is presented in Figure 4. Our method achieves a significant improvement in confidence calibration at the perception level (an increase of 20.4%), which contributes to a 38.5% improvement in the accuracy of the reasoning chain. Furthermore, at the reasoning level, benefiting from the strengthened knowledge boundary in the visual understanding stage, both the confidence calibration score and the reasoning chain F1 score show improvements, surpassing the strongest baseline by 19.7% and 27.4%, respectively.

6 Related Work

Hallucinations and Uncertainty Estimation. Efforts have been made towards evaluating the hallucinations in the VLMs (Liu et al., 2023; Gunjal et al., 2024; Zhang et al., 2024b; Wu et al., 2024). As a fundamental approach to detecting model hallucination, uncertainty estimation (UE) has long attracted significant attention, which fall into two types: black-box and white-box. Black-box methods only require the generated text and most of these methods are based on self-consistency (Fomicheva et al., 2020; Kuhn et al., 2023; Lin et al., 2023). White-box methods rely on access to logits and internal layer outputs. They encompass information-based, density-based and sample diversity-based approaches (Malinin and Gales, 2020; Kadavath et al., 2022; Vazhentsev et al., 2023; Kuhn et al., 2023; Fadeeva et al., 2024; Duan et al., 2024). Instead of rely on self-consistency prompting, we propose leveraging the model's internal states to quantify the confidence.

Confidence Calibration of Language Model. Increasing attention has been directed toward enhancing the models' awareness of their knowledge boundaries and enabling them to express their confidence in outputs when encountering uncertainty (Lin et al., 2022; Xiong et al., 2023; Yang et al., 2023; Lyu et al., 2024; Xu et al., 2024; Zhang et al., 2024a). Zhang et al. (2024a) introduce R-tuning to encourage LLMs to express "certain/not certain". Xu et al. (2024) goes further to teach the model to express more fine-grained confidence estimates along with self-reflective rationales. However, these methods focus solely on the entire response. Therefore, we propose MMBoundary to train VLMs to express fine-grained confidence estimates for each reasoning step during inference, enhancing reasoning chain self-correction.

7 Conclusion

In this work, we present MMBoundary, a novel framework that advances the knowledge boundary awareness of multimodal models through reasoning step confidence calibration. We incorporates complementary textual and cross-modal self-rewarding signals to estimate confidence at each step of the MLLM reasoning process. In addition to supervised fine-tuning MLLM for initial confidence expression warm-up, we further introduce a reinforcement learning stage with multiple reward functions for further calibrating model confidence. Empirical results demonstrate that our framework significantly outperforms existing methods, achieving an average reduction of 7.5% in multimodal confidence calibration errors and up to 8.3% improvement in task performance.

Limitation

Our framework aims to enable MLLMs to autonomously generate natural language confidence statements during inference, enhancing reasoning chain self-correction. A limitation of our work is our method involves using the model's internal states and uncertainty methods to assess the model's confidence. However, more research is needed to determine whether uncertainty methods can accurately reflect the model's confidence in its output. Ablation experiments on the uncertainty methods indicate that the four carefully selected methods provide gains for the model. Additionally, we explore the correlation between the proposed internal confidence estimation method and the self-consistency method. The results show that our metric, without requiring multiple samples, achieves performance comparable to methods that rely on multiple samples.

References

Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. 2024. Hallucination of multimodal large language models: A survey. *arXiv preprint arXiv:2404.18930*.

Xiang Chen, Chenxi Wang, Yida Xue, Ningyu Zhang, Xiaoyan Yang, Qiang Li, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. 2024. Unified hallucination detection for multimodal large language models. *arXiv preprint arXiv:2402.03190*.

Yangyi Chen, Lifan Yuan, Ganqu Cui, Zhiyuan Liu, and Heng Ji. 2022. A close look into the calibration of pre-trained language models. *arXiv preprint arXiv:2211.00151*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5050–5063.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, et al. 2024. Fact-checking the output of large language models via token-level uncertainty quantification. *arXiv preprint arXiv:2403.04696*.

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, et al. 2023. Lm-polygraph: Uncertainty estimation for language models. *arXiv preprint arXiv:2311.07383*.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. 2024. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14375–14385.

Anisha Gunjal, Jihan Yin, and Erhan Bas. 2024. Detecting and preventing hallucinations in large vision language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18135–18143.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Dan Hendrycks and Kevin Gimpel. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*.

Matthew Ho, Aditya Sharma, Justin Chang, Michael Saxon, Sharon Levy, Yujie Lu, and William Yang Wang. 2022. Wikiwhy: Answering and explaining cause-and-effect questions. *arXiv preprint arXiv:2210.12152*.

Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. 2024a. Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13418–13427.

Wen Huang, Hongbin Liu, Minxin Guo, and Neil Zhenqiang Gong. 2024b. Visual hallucinations of multimodal large language models. *arXiv preprint arXiv:2402.14683*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*.

Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. 2023. Aligning large multi-modal model with robust instruction tuning. *arXiv preprint arXiv:2306.14565*.

Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. 2024a. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253*.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024b. Llava-next: Improved reasoning, ocr, and world knowledge.

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

Qing Lyu, Kumar Shridhar, Chaitanya Malaviya, Li Zhang, Yanai Elazar, Niket Tandon, Marianna Apidianaki, Mrinmaya Sachan, and Chris Callison-Burch. 2024. Calibrating large language models with sample consistency. *arXiv preprint arXiv:2402.13904*.

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*.

Shravan Nayak, Kanishk Jain, Rabiul Awal, Siva Reddy, Sjoerd van Steenkiste, Lisa Anne Hendricks, Karolina Stańczak, and Aishwarya Agrawal. 2024. Benchmarking vision language models for cultural understanding. *arXiv preprint arXiv:2407.10920*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

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

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

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Akim Tsvigun, Daniil Vasilev, Rui Xing, Abdelrahman Boda Sadallah, Kirill Grishchenkov, Sergey Petrakov, et al. 2024. Benchmarking uncertainty quantification methods for large language models with lm-polygraph. *arXiv preprint arXiv:2406.15627*.

Artem Vazhentsev, Akim Tsvigun, Roman Vashurin, Sergey Petrakov, Daniil Vasilev, Maxim Panov, Alexander Panchenko, and Artem Shelmanov. 2023. Efficient out-of-domain detection for sequence to sequence models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1430–1454.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Shujin Wu, Yi R Fung, Sha Li, Yixin Wan, Kai-Wei Chang, and Heng Ji. 2024. Macaroon: Training vision-language models to be your engaged partners. *arXiv preprint arXiv:2406.14137*.

Yuxin Xiao, Paul Pu Liang, Umang Bhatt, Willie Neiswanger, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2022. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. *arXiv preprint arXiv:2210.04714*.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*.

Tianyang Xu, Shujin Wu, Shizhe Diao, Xiaoze Liu, Xingyao Wang, Yangyi Chen, and Jing Gao. 2024. Sayself: Teaching llms to express confidence with self-reflective rationales. *arXiv preprint arXiv:2405.20974*.

Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. 2023. Alignment for honesty. *arXiv preprint arXiv:2312.07000*.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2024a. R-tuning: Instructing large language models to say 'i don't know'. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7106–7132.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi R Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2023. R-tuning: Teaching large language models to refuse unknown questions. *arXiv preprint arXiv:2311.09677*.

Yuji Zhang, Sha Li, Jiateng Liu, Pengfei Yu, Yi R Fung, Jing Li, Manling Li, and Heng Ji. 2024b. Knowledge overshadowing causes amalgamated hallucination in large language models. *arXiv preprint arXiv:2407.08039*.

Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. 2023. Analyzing and mitigating object hallucination in large vision-language models. *arXiv preprint arXiv:2310.00754*.

Yiyang Zhou, Zhiyuan Fan, Dongjie Cheng, Sihan Yang, Zhaorun Chen, Chenhang Cui, Xiyao Wang, Yun Li, Linjun Zhang, and Huaxiu Yao. 2024. Calibrated self-rewarding vision language models. *arXiv preprint arXiv:2405.14622*.

A The Value-Statement Mapping Table

We have preset 40 concise statements for each level. Table 4 presents additional confidence statements. These statements are concise and express the semantics of the corresponding confidence levels, allowing for seamless integration into sentences generated by the model, making them suitable for training generative language models.

Score	Statement
1	but I can't confirm this. / I'm uncertain about this. / I'm not sure about that. / This answer may be wrong. / I can't guarantee this answer. / I'm unsure about this. / I can't be sure about this. / This answer is unclear to me. / This might be imprecise. / This could be questionable. (. . .)
2	and it may need checking / it might not be right. / but I'm not sure. / and it might be slightly off. / though it's not perfect. / but it may need confirmation. / though there's some doubt. / though it may not hold up. / though I feel a bit unsure. / but there's minor hesitation. (...)
3	but I can't guarantee perfection. / I can't be entirely sure / but it's not beyond all doubt. / though minor errors might exist. / but it's not fully certain. / though small flaws are possible. / but it's not completely precise. / but it's not entirely error-free. / though it's not fully verified. / though it's open to review. (...)
4	and this seems trustworthy. / and I believe this is right. / and I'm quite confident in this. / and this feels reliable to me. / and I trust this is correct. / and this seems very likely true. / and this appears reliable. / and this fits the context well. / and I'm confident this is right. / and this is well-reasoned. (...)
5	with total certainty. / with no doubts at all. / and I'm absolutely sure about this. / and I'm fully confident in this. / with total certainty. / and this is undoubtedly correct. / and this is entirely reliable. / and it's unquestionably right. / with complete confidence. / and I guarantee this is right. (...)

Table 4: The confidence score-statement mapping table. The five scores correspond to uncertain, slightly uncertain, moderately confident, highly confident, and fully confident. We preset 40 confidence statements for each score.

B Implementation Details

Our experiment involves three distinct datasets: A-OKVQA(Schwenk et al., 2022), ScienceVQA(Lu et al., 2022), and CulturalVQA(Nayak et al., 2024). The first two datasets are in-domain datasets, and our training data comes from the training sets of these two datasets, while CulturalVQA is an out-of-domain dataset. Since the test sets for all three datasets are not publicly available, we cannot accurately annotate the reasoning chain for MECE and Reasoning Chain F1 evaluation. Therefore, we use the validation sets of AOKVQA and ScienceVQA for in-domain testing of the model. For CulturalVQA, which only has a non-public test set, we manually selected and annotated 800 samples from it to serve as the test set.

For the construction of the warm-up dataset, we deploy the vLLM model with a temperature setting of 0.1 and number of log probabilities to return per output token of 10. We collect a total of 19K Question-Image pairs from the training sets of A-OKVQA and ScienceVQA. For each Question-Image pair, we prompt the model to generate the reasoning chain and calculate the model's confidence score for each sentence, resulting in 55K sentences with confidence statements, with w_0, w_1,

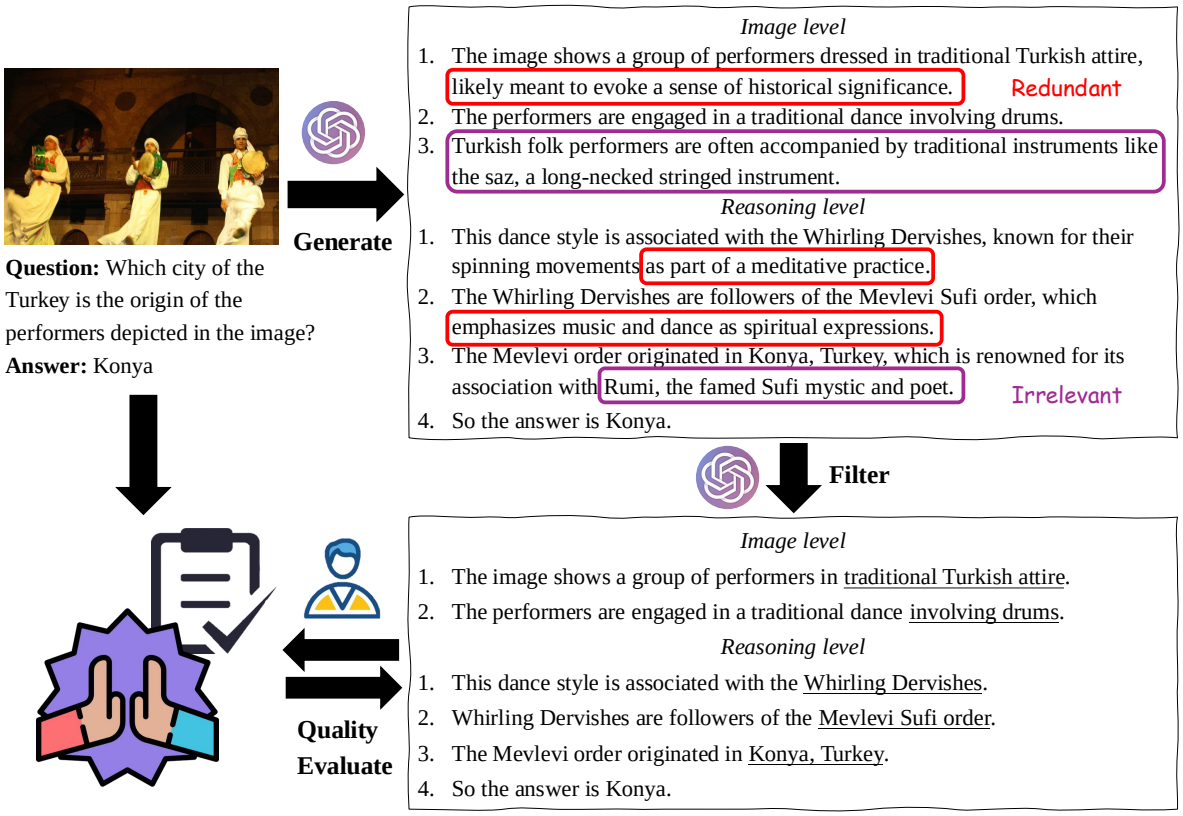

Figure 5: The Annotation Pipeline. We first prompt GPT-4o to generate an analysis (reasoning chain) structured at the perception and reasoning levels. Then, we have GPT-4o filter and correct the initially annotated chains. Finally, manual data quality control is conducted to ensure accuracy and reliability.

and w_2 all set to 0.3 in internal confidence estimation module. During the warm-up stage, we use the AdamW optimizer with a 10% warm-up ratio, a learning rate of 1.0e-4, and a batch size of 16. In the reinforcement learning phase, we randomly sample data from the training set for training, for each question, we sample $N = 3$, with a learning rate of 1e-5 and a batch size of 16. In all experiments, training was conducted on a single A100-80GB GPU.

C The Reasoning Chain Annotation

To obtain the necessary fine-grained knowledge of visual perception and cross-modal reasoning in visual question-answering for calibrating the multi-level confidence of MLLMs, we conduct reasoning chain annotation on knowledge-extensive datasets from three different domains.

C.1 The Annotation Pipeline

The pipeline of reasoning chain annotation is presented in Figure 5. We first prompt the GPT-4o to generate analysis (reasoning chain) structured in the perception and reasoning level. The former identifies key visual elements in the image that are most relevant to the question and answer, while the latter provides granularity reasoning that justifies why the answer is correct. Each level should include concise, interconnected sentences, with each sentence conveying a single piece of knowledge. As shown in the upper right corner of the figure, the initially obtained reasoning chain may contain redundant information and irrelevant content. Therefore, we use GPT-4o again to correct the content of the reasoning chain, filtering out redundancy and unrelated information to ensure that each sentence is concise and accurate. Then, we conduct annotation quality control to ensure the accuracy and consistency of the data. The prompt is provided in Appendix F.

C.2 Quality Evaluation

After the machine annotation is completed, we randomly selected 50 samples from each of the three datasets and asked two graduate students to evaluate the data quality. The evaluation metrics included: (1)Accurate: the reasoning chain is relevant to the question and contains no wrong

Metric	Accurate	Concise	Complete
Rate (%)	96.8	91.3	93.5

Table 5: The Likert Scale results of annotated data. We report the proportion of data with a rating greater than 4 (i.e., Agree).

knowledge; (2) Concise: each sentence is concise and contains no redundant information; (3) Complete: the reasoning chain formed by each sentence accurately explains the answer to the corresponding question without omitting relevant knowledge. We use a Likert Scale to evaluate each indicator, with a scoring range from 1 to 5, where 1 indicates 'Strongly Disagree' and 5 indicates 'Strongly Agree.' The results are shown in Table 5. We report the proportion of data with a rating greater than 4 (i.e., Agree). The results indicate that the majority of the data meet the three criteria. We conduct a Kappa test on the accuracy evaluation results of the two graduate students, yielding a Kappa value of 0.75, which indicates a high level of consistency between the evaluators.

D The Details of Evaluation Metrics

D.1 The Multi-granular Expected Calibration Error (MECE)

As shown in Figure 6, after comparing the knowledge contained in the predictions and references, we obtain sentences where the knowledge in predictions and references aligns. Then, we calculate the Expected Calibration Error (ECE) for each sentence one by one, and finally derive the Multi-granular Expected Calibration Error (MECE):

$$\mathrm{ECE}(a) = \frac{1}{|a|} \sum_{(z,c)\in a} |\mathbb{I}(z) - \mathrm{Conf}(c)| \qquad (15)$$

$$\mathrm{MECE}(A) = \frac{1}{|A|} \sum_{a\in A} \mathrm{ECE}(a) \qquad (16)$$

Here, A represents the entire test set, and a denotes the reasoning chain generated by the model, which consists of multiple sentences. (z, c) represent a sentence and its corresponding confidence statement, respectively. $\mathbb{I}(\cdot)$ is the indicator function that returns 1 if the sentence is correct when compared with the reference chain, and 0 otherwise. $\mathrm{Conf}(\cdot)$ represents the numerical value of the confidence statement.

D.2 AUROC

We adopt the ***AUROC*** score (Hendrycks and Gimpel, 2016), which measures the ability of models to distinguish between correct and incorrect responses across different threshold settings.

$$\mathrm{AUROC} = \int_0^1 \mathrm{TPR}(\mathrm{FPR}^{-1}(x))\, dx \qquad (17)$$

where x denotes the threshold confidence level, TPR represents the true positive rate at this threshold, and FPR indicates the false positive rate corresponding to the threshold. The result is shown in Table 8.

D.3 Reasoning Chian F1 Score

We use the Reasoning Chain F1 score (Ho et al., 2022) to evaluate the quality of the reasoning chains generated by the model. We compare the knowledge contained in the predictions and references. First, we split the predicted and reference chains into "steps" by sentence. We then compute a matrix of pairwise similarity scores before using a threshold to classify "matches." Since a single predicted sentence may contain multiple reference knowledge, we keep separate counts of precise predicted sentences and covered reference sentences. These counts are then micro-averaged to calculate the overall precision, recall, and F1 scores for the test set:

$$\mathrm{Precision} = \frac{\mathrm{Matched}}{\mathrm{Prediction}}, \mathrm{Recall} = \frac{\mathrm{Covered}}{\mathrm{Reference}} \qquad (18)$$

Taking the answer in Figure 6 as an example, we have: Prediction = 4, Reference = 6, Matched = 3, Covered = 3. We then calculate the F1 score:

$$F1 = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \qquad (19)$$

Drawing on the study of (Ho et al., 2022), we select a large RoBERTa model (cross-encoder/stsb-roberta-large) with a similarity threshold of 0.64.

E Experiments of MMBoundary on Qwen

To prove that our method can generalize on multiple models, we also implement the baseline approaches and MMBoundary on Qwen2VL 7B (Wang et al., 2024).

F Prompt

F.1 GPT-4o Annotation Prompt

F.2 GPT-4o Refinement Prompt

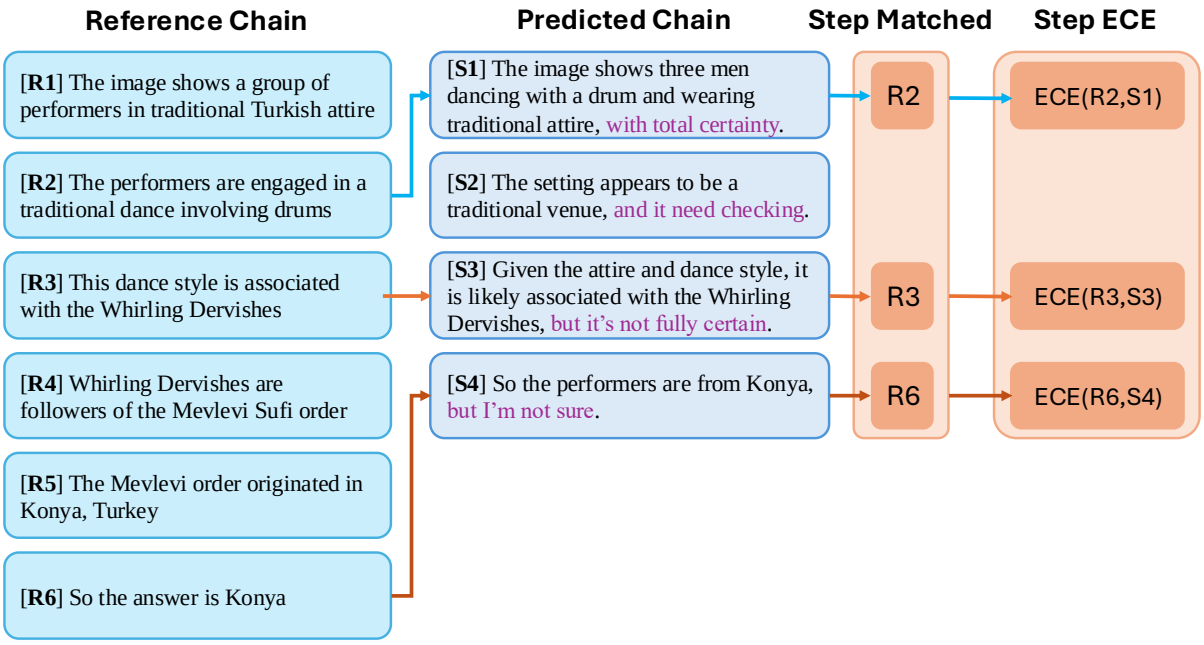

Figure 6: Example of MECE and Reasoning Chain F1 calculation.

Model	A-OKVQA				ScienceVQA				CulturalVQA			
	ECE (↓)	MECE (↓)	Acc (↑)	F1 (↑)	ECE (↓)	MECE (↓)	Acc (↑)	F1 (↑)	ECE (↓)	MECE (↓)	Acc (↑)	F1 (↑)
Multisample	0.437	0.463	0.665	0.557	0.467	0.492	0.613	0.415	0.443	0.512	0.335	0.513
SaySelf	0.324	0.381	0.727	0.632	**0.332**	0.445	0.628	0.523	0.397	0.426	0.352	0.586
CSR	0.413	0.463	0.774	0.623	0.433	0.534	0.702	0.517	0.482	0.503	0.394	0.562
RCE	0.372	0.427	0.793	0.647	0.401	0.483	0.686	0.504	0.436	0.475	0.425	0.597
DRL	0.406	0.442	0.762	0.628	0.435	0.523	0.641	0.487	0.465	0.493	0.384	0.552
MMBoundary	**0.305**	**0.348**	**0.806**	**0.664**	0.346	**0.426**	**0.713**	**0.562**	**0.354**	**0.385**	**0.437**	**0.634**

Table 6: Experimental Results on a different base model, Qwen2VL 7B (Wang et al., 2024).

Model	A-OKVQA		ScienceVQA	
	ECE (↓)	MECE (↓)	ECE (↓)	MECE (↓)
MMBoundary	**0.316**	**0.324**	**0.354**	**0.405**
w/ UIC	0.358	0.387	0.406	0.479

Table 7: The comparative study of confidence level segmentation methods. UIC (uniform interval for confidence level segmentation) means simply divides the interval [0, 1] directly into five equal segments, with each segment corresponding to a confidence level.

Model	A-OKVQA	Sci-VQA	Cul-VQA
Multisample	0.5016	0.5429	0.4904
SaySelf	**0.6872**	0.6118	0.6261
CSR	0.5238	0.5713	0.4931
RCE	0.6037	0.5902	0.6059
DRL	0.4956	0.5028	0.4576
MMBoundary	0.6635	**0.6786**	**0.7108**

Table 8: The AUROC experimental results.

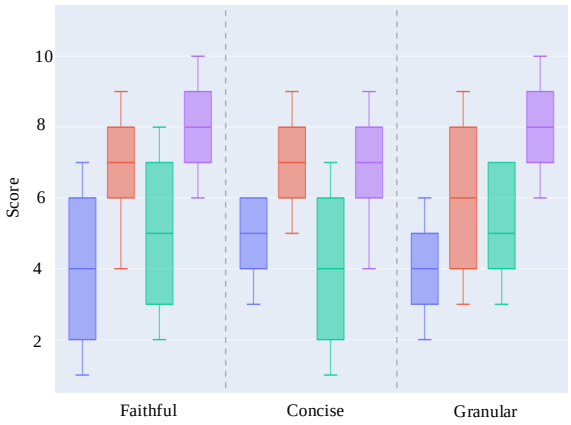

Figure 7: Boxplots of human evaluation scores on the A-OKVQA dataset.

GPT-4o Annotation Prompt

```
You will receive a question, an accompanying image, the correct answer, and
the corresponding rationales. Follow these steps to generate your analysis (
reasoning chain) structured in two levels.

Each level should include concise, interconnected sentences, with each
sentence conveying a single piece of knowledge. Ensure the reasoning chain
covers all necessary knowledge points concisely, with each sentence in this
reasoning chain is essential and avoid adding redundant or irrelevant
sentences.

The levels are as follows:
**Image level**: Identify key visual elements in the image that are mostly
relevant to the question and answer. Format these sentences in JSON, like: ['
sentence 1', ..., 'sentence i'].
**Reasoning level**: Based on the extracted visual elements, provide logical
reasoning that justifies why the answer is correct. Format these in JSON as
well: ['sentence i+1', 'sentence i+2', ..., 'So, the answer is ...'].

The sentences in both levels together should form a coherent chain of
reasoning that clearly explains why the answer is correct. Ensure that each
sentence builds upon the previous one to complete the reasoning chain. In the
final sentence of the reasoning level, provide a clear conclusion with the
answer, like: 'So, the answer is ...'.

Refer to the example below:
Please answer the following question:
Image: (Three people in traditional clothing holding drums, performing a form
of the 'whirling dervishes' ritual.)
Question: Which city in Turkey is the origin of the performers depicted in the
 image?
Answer: Konya
Analysis: {{'Image_level': [], 'Reasoning_level': []}}
Your output:
Analysis: {{
    'Image_level': [
        'The image shows a group of performers in traditional Turkish attire',
        'The performers are engaged in a traditional dance involving drums'
    ],
    'Reasoning_level': [
        'This dance style is associated with the Whirling Dervishes',
        'Whirling Dervishes are followers of the Mevlevi Sufi order',
        'The Mevlevi order originated in Konya, Turkey',
        'So the answer is Konya'
    ]}}

Now, please answer the following question:
Image: image
Question: {question}
Answer: {answer}
Analysis:{{
    'Image_level': [],
    'Reasoning_level': []
    }}

Your output:
Analysis:
```

Figure 8: GPT-4o annotation prompt.

GPT-4o Refinement Prompt

```
Now, the following analysis (reasoning chain) is structured.
Image_level and Reasoning_level together form a complete reasoning chain.
Please filter out any irrelevant sentences to maintain a concise reasoning
chain, including only the essential sentences.

Refer to the example below:
Image: (Three people in traditional clothing holding drums, performing a form
of the 'whirling dervishes' ritual.)
Question: Which city in Turkey is the origin of the performers depicted in the
 image?
Answer: Konya
Analysis:{{
    'Image_level': [
        'The image shows a group of performers dressed in traditional Turkish
        attire, likely meant to evoke a sense of historical significance.',
        'The performers are engaged in a traditional dance involving drums.',
        'Turkish folk performers are often accompanied by traditional
        instruments like the saz, a long-necked stringed instrument.'
    ],
    'Reasoning_level': [
        'This dance style is associated with the Whirling Dervishes, known for
         their spinning movements as part of a meditative practice.',
        'The Whirling Dervishes are followers of the Mevlevi Sufi order, which
         emphasizes music and dance as spiritual expressions.',
        'The Mevlevi order originated in Konya, Turkey, which is renowned for
        its association with Rumi, the famed Sufi mystic and poet.',
        'So the answer is Konya.'
    ]}}

Your output:
Analysis:{{
    'Image_level': [
        'The image shows a group of performers in traditional Turkish attire
        .',
        'The performers are engaged in a traditional dance involving drums.'
    ],
    'Reasoning_level': [
        'This dance style is associated with the Whirling Dervishes.',
        'Whirling Dervishes are followers of the Mevlevi Sufi order.',
        'The Mevlevi order originated in Konya, Turkey.',
        'So the answer is Konya.'
    ]}}

Now:
Image: (image)
Question: {question}
Answer: {answer}
Analysis: {analysis}
Your output:
Analysis:
```

Figure 9: GPT-4o Refinement prompt.