Do LVLMs Know What They Know? A Systematic Study of Knowledge Boundary Perception in LVLMs

Anonymous ACL submission

Abstract

Large Vision-Language Models (LVLMs) demonstrate strong visual question answering (VQA) capabilities but are shown to hallucinate. A reliable model should perceive its knowledge boundaries—knowing what it knows and what it does not. This paper investigates LVLMs' perception of their knowledge boundaries by evaluating three types of confidence signals: probabilistic confidence, answer consistency-based confidence, and verbalized confidence. Experiments on three LVLMs across three VQA datasets show that, although LVLMs possess a reasonable perception level, there is substantial room for improvement. Among the three confidence, probabilistic and consistency-based signals are more reliable indicators, while verbalized confidence often leads to overconfidence. To enhance LVLMs' perception, we adapt several established confidence calibration methods from Large Language Models (LLMs) and propose three effective methods. Additionally, we compare LVLMs with their LLM counterparts, finding that jointly processing visual and textual inputs decreases question-answering performance but reduces confidence, resulting in improved perception level compared to LLMs.

1 Introduction

Large Vision-Language Models (LVLMs) are capable of processing both textual and visual information simultaneously, demonstrating strong performance on visual question-answering (VQA) task (Bai et al., 2025a; Wu et al., 2024; OpenAI et al., 2024). However, when faced with questions beyond their knowledge boundaries, LVLMs often hallucinate—generating seemingly plausible but factually incorrect responses (Liu et al., 2024a; Bai et al., 2025b). This is unacceptable in safety-critical domains such as healthcare. Knowing when an LVLM can answer correctly not only helps us determine when to trust the model but also enables adaptive retrieval-augmented generation (RAG), triggering RAG only when the model does not know the answer, which improves both the efficiency and effectiveness of RAG (Ni et al., 2024).

A trustworthy model should have a clear perception of its knowledge boundaries—knowing what it knows and what it does not. While this ability has been extensively studied in Large Language Models (LLMs) (Xiong et al., 2024; Tian et al., 2023; Moskvoretskii et al., 2025), it remains underexplored in LVLMs. A model's perception level is assessed by the alignment between its confidence and actual performance, with correctness of the answer serving as a proxy for performance. Therefore, the emphasis is on whether LVLMs can provide confidence that matches their performance. We focus on binary confidence because it directly helps us decide whether to trust the model.

In this work, we explore this question by examining three representative types of confidence signals that are widely used in LLMs: **1) Probabilistic confidence** (Desai and Durrett, 2020; Guo et al., 2017a). The confidence is measured by the generation probability of tokens in the output. **2) Answer consistency-based confidence** (Zhang et al., 2024; Manakul et al., 2023b). Some studies argue that token-level probabilities poorly reflect a model's semantic confidence and are not suitable for black-box models. Instead, they suggest using semantic consistency across multiple responses as a confidence indicator. **3) Verbalized confidence** (Lin et al., 2022; Yang et al., 2024b). The natural language confidence expressed by the model, offering an intuitive and model-agnostic signal without requiring repeated sampling.

We conduct experiments using three representative models—-Qwen2.5-VL (Bai et al., 2025a), DeepSeek-VL2 (Wu et al., 2024), and LLaVA-v1.5 (Liu et al., 2024b)—on three datasets: Dyn-VQA (Li et al., 2025b), MMMU Pro (Yue et al., 2024), and Visual7W (Zhu et al., 2016). Results show that LVLMs are able to perceive their knowledge bound-

aries to some extent, but there remains considerable room for improvement. Among the three types of confidence, probabilistic and answer consistency-based confidences are more aligned with LVLMs' performance but rely on in-domain data for binarization, while verbalized confidence have weaker alignment and tends to be overconfident.

To enhance LVLMs' perception capabilities, we adopt several representative confidence calibration methods originally designed for LLMs and propose three new approaches tailored for LVLMs: Img-CoT, Prob-Thr, and Cross Model. Our results show that methods which engage the model's reasoning abilities can enhance both answer accuracy and verbalized confidence perception-level, whereas existing consistency-based methods have limited effectiveness and do not generalize well to LVLMs. Our proposed three methods are effective on different datasets and models.

Compared to LLMs, LVLMs need to process an additional visual modality and integrate information across different modalities. This raises a question: how does the perception ability of LVLMs differ from that of LLMs? To investigate this, We compare LVLMs with their corresponding LLMs on the Dyn-VQA dataset (Li et al., 2025b; Tian et al., 2023). This dataset provides parallel visual-textual and pure textual queries, ensuring fair comparison between LLMs and LVLMs. We focus on verbalized confidence because it can reflect the model's language capabilities. Experimental results show that: 1) LVLMs exhibit lower VQA performance but higher perception accuracy compared to their LLM counterparts. 2) Certain prompting methods are ineffective for LVLMs, showing that LVLMs have weaker instruction-following capabilities.

We hypothesize these phenomena may be caused by the following two reasons: 1) Compared to processing single-modality information, handling visual information and integrating two modalities pose challenges for LVLMs, which leads to low VQA performance. However, this also leads to more conservative confidence expression, resulting in lower overconfidence and a more accurate perception level. 2) Training LLMs without enough capacity to handle additional visual information may lead to a loss of their language abilities, thereby impairing their instruction following ability. Through controlled experiments on different model scales and different input modalities, our results supported these hypotheses.

2 Related Work

LLM Knowledge Boundary Perception. Prior research has primarily focused on knowledge boundary perception in LLMs, with various methodologies proposed to elicit confidence: verbalized confidence, where models directly articulate their confidence (Yang et al., 2024b; Yin et al., 2023; Zhang et al., 2023); consistency based confidence that derive confidence from answer consistency across multiple samples (Manakul et al., 2023a; Agrawal et al., 2024); probabilistic confidence, leveraging generated token likelihoods (Guo et al., 2017b; Ma et al., 2025); and internal state probing confidence, examining hidden states (Azaria and Mitchell, 2023; Ni et al., 2025). Differently, our work investigates knowledge boundary perception in LVLMs and provides the first systematic comparison of these methods in the multimodal setting.

LVLMs. Previous studies have established the widespread adoption of LVLMs in safety-critical domains such as healthcare (Li et al., 2023; Hu et al., 2024) and autonomous driving (Cui et al., 2024; Jiang et al., 2024). While these applications demonstrates LVLMs' functional capabilities, studies show LVLMs frequently produce hallucinations (Bai et al., 2025b; Sahoo et al., 2024). The current body of work investigates this limitation on different aspects. Some work surveys hallucination types and their causes (Liu et al., 2024a; Zhou et al., 2024; Lan et al., 2024), while others focus on mitigating hallucinations (Li et al., 2025a; Wang et al., 2024a; Xiao et al., 2025). A distinct but less explored research thread investigates LVLMs' knowledge boundary as a potential framework for enhancing model reliability (Chen et al., 2025; Wang et al., 2024b; Leng et al., 2024). We take this line of work a step further by introducing a novel comparative paradigm that compares perception between LVLMs with their LLM counterparts.

3 Preliminaries

In this section, we provide an overview of our task.

3.1 Task Formulation

Visual Question Answering. The goal of visual question answering (VQA) can be described as follows. For a given question q and an image i, the model is asked to provide an answer a based on the question q and image i, that is, $a = f_{model}(q, i)$.

LVLM Knowledge Boundary Perception. We assess the perception of LVLM's knowledge boundary with the alignment between confidence and its actual performance. Here, we use the model's visual question answering correctness to serve as a proxy for performance, and elicit different kinds of model confidence estimates.

Confidence Estimation. In this paper, we conduct experiments on the following three kinds of model confidence estimates. As widely adopted and training-free, they can be elicited without changing the internal knowledge of models.

Probabilistic confidence is elicited through the aggregation of token probabilities for scoring, followed by applying a threshold to binarize the score into confidence. It is efficient but only captures lexical-level confidence and requires threshold tuning on a held-out set, which leads to poor generalizability. Some studies also argue that it is not applicable to black-box models (Kuhn et al., 2023).

Answer consistency-based confidence is elicited by calculating the consistency of multiple generated responses. The core idea is that if the model knows the correct answer, multiple sampled answers should be semantically consistent. It better captures semantics than probabilistic confidence, but is computationally expensive and still requires fitting a threshold (Manakul et al., 2023a).

Verbalized confidence is elicited by directly asking the model to express confidence (Yang et al., 2024b). Compared to the other two confidences, this confidence reflects models' self-awareness of their knowledge boundaries. Moreover, it eliminates the need for threshold fitting and multiple sampling. Therefore, this kind of confidence receives our primary focus.

4 Knowledge Boundary Perception in LVLMs

This section introduces experimental setup to evaluate LVLMs knowledge boundary perception ability. Along with the elicited confidence and confidence calibration methods evaluated by us.

4.1 Existing Methods

Here, we systematically introduce three basic confidence estimates in Section 3, along with several confidence calibration methods originally designed for LLMs. We also propose new methods. Detailed prompts are in Appendix A.1. Basic confidence estimates are in underline, and others are existing confidence calibration methods.

4.1.1 Vanilla Confidence Estimation Methods

Probabilistic confidence is elicited through token probabilities. Here, we focus on the output perplexity of models.

- **Perplexity Threshold (PPL-Thr)**: Perplexity quantifies a model's uncertainty in content generation (Cooper and Scholak, 2024). We binarize this metric into confidence by applying a threshold decided on a held-out set.

Answer consistency-based confidence requires models to generate multiple responses, compute their consistency, and apply a threshold to the consistency scores for confidence elicitation.

- **Random Sample (Random)**: Simply sample responses without modifying input.

We evaluate two types of verbalized confidence: (1) Single-step verbalized confidence, which is generated simultaneously with the answer, and (2) Double-step verbalized confidence, which is generated by asking the model for an answer in the initial round of dialogue, then providing its confidence in the second round. The distinction between them lies in cognitive focus allocation: single-step confidence elicitation demands concurrent attention to both answer and confidence generation, whereas double-step confidence elicitation enables sequential processing.

- **Single-step Vanilla (Vanilla)** : Simply ask the model to generate both the answer and confidence in a single interaction.
- **Double-step Self Judging (Self-Jud)**: First, acquiring the model to provide an answer to the question, then asking it to generate confidence.

4.1.2 Calibrating Verbalized Confidence

The four methods below aim to calibrate single-step verbalized confidence:

- **Chain-of-Thought (CoT)**: Zero-shot Chain-of-Thought prompting, Applying "Analyze step by step" to the query (Kojima et al., 2023).
- **Punish**: Penalizing overconfidence via the instruction "You will be punished if the answer is not right but you say certain".
- **Explain**: Requesting models to provide answer explanations before generating their confidence.

The four methods below aim to calibrate double-step verbalized confidence:

- **Chain of Thought (CoT)**: Applying the Chain-of-Thought prompt in the confidence elicitation round of dialogue.

- **Challenge**: We prepend the critical prompt "I don't think your answer is right" to the query in the confidence elicitation round in order to guide the model to be less overconfident.
- **Punish**: Applying the Punish prompt in the confidence elicitation round of dialogue.

4.1.3 Calibrating Answer Consistency-Based Confidence

- **Rephrasing (Rephr)**: To address persistent errors caused by a specific question phrase, rephrase the original question into semantically equivalent variants with different phrases (Yang et al., 2024a).
- **Noised Image (Noised-Img)**: Reducing persistent errors caused by a specific image by creating semantically equivalent variants through the addition of subtle noise to the original image.
- **Rephrasing and Noised Image (Reph+Nois)**: A combination of the Rephrasing and the Noised Image methods.

4.2 Newly Proposed Methods

- **Image Chain of Thought (Img-CoT)**: Prompting models to generate textual image descriptions before reasoning in order to convert visual modality information to textual modality.
- **Probability Threshold (Prob-Thr)**: Prompting models to generate continuous probabilities of answers (0–1), then applies a threshold to them to generate binary confidences. The threshold is decided on a held-out set.
- **Cross Model**: Utilizing generated responses from different models to calculate the consistency score, this method can be viewed as using other models' answers to evaluate whether the answer generated by a given model is reliable.

4.3 Experimental Setup

Datasets. We conduct experiments on three VQA benchmark datasets. They emphasize on LVLM's different abilities. Visual7W (Zhu et al., 2016) emphasizes abilities in vision comprehension, it contains 70K image-QA pairs for basic visual understanding. Dyn-VQA (Li et al., 2025b) emphasizes language reasoning, it contains 1.5K questions testing multi-modal knowledge and multi-hop reasoning.; MMMU Pro (Yue et al., 2024) emphasizes both vision and language capability, it contains 12K expert-curated multimodal questions. For evaluation, we respectively sample 550 questions from Dyn-VQA and MMMU Pro datasets, and sample 500 questions from the Visual7W dataset.

Models. We conduct experiments on three representative LVLMs: Qwen2.5-VL-7B (Bai et al., 2025a), DeepSeek-VL2-16B (Wu et al., 2024), and LLaVA-v1.5-7B (Liu et al., 2024b).

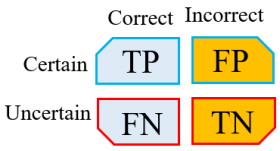

Figure 1: Count of samples for various matches between answer correctness and model confidence. We use $Total = FN + FP + TN + TP$ to represent the total number of samples.

Metrics. We mainly utilize the evaluation metrics proposed by Ni et al. (2024): (1) **Uncertain-Rate (Unc-R.)**: $\frac{FN+TN}{Total}$ represents the proportion where the judgement of the answer is unconfident. (2) **Accuracy (Acc.)**: $\frac{TP+FN}{Total}$ indicates the ratio of correct answers generated by the model. (3) **Alignment (Align.)**: $\frac{TP+TN}{Total}$ represents the proportion of samples where confidence matches the result, we mainly use this metric to assess the model's knowledge boundary perception ability. (4) **Overconfidence (Overco.)**: $\frac{FP}{Total}$ is the ratio of model-generated answer is incorrect, but the judgement is confident. (5) **Conservativeness (Conser.)**: $\frac{FN}{Total}$ is the ratio of model-generated answer is correct but the judgement is unconfident.

4.4 Results and Analysis

Table 1 presents the results of alignment performance across different datasets and models. Please refer to Appendix A.2 for implementation details and detailed results.

4.4.1 Performance of Different Types of Confidence

Here, we analyze three basic elicited confidence's performance. Our findings are as follows:

1) **Compared to verbalized and probabilistic confidence, answer consistency-based confidence often shows higher alignment.** As shown in Table 1, the basic answer-consistency based confidence (Random) achieves higher alignment compared to verbalized (Vanilla, Self-Jud) and probabilistic confidences (PPL Thr) on both LLaVA-1.5 and Deepseek-VL2. This may be because, unlike probabilistic confidence that operates at the lexical level, answer consistency-based confidence better

method	Qwen2.5-VL			LLaVA-1.5			DeepSeek-VL2		
	Dyn-VQA	Visual7W	MMMU Pro	Dyn-VQA	Visual7W	MMMU Pro	Dyn-VQA	Visual7W	MMMU Pro
Vanilla	0.7623	0.5840	0.4909	0.5338	0.4140	0.2509	0.6527	0.2820	0.2727
CoT	0.7824	0.6080	0.6818	0.5375	0.3940	0.2418	0.6362	0.5540	0.3836
Punish	0.7112	0.5520	0.5000	0.4899	**0.4180**	**0.3745**	**0.7093**	0.3500	0.3145
Explain	**0.8117**	**0.6180**	0.5782	0.4534	0.3900	0.2109	0.6984	**0.5700**	0.3491
Img-CoT	0.7276	0.6060	**0.7182**	**0.5484**	0.4140	0.2964	0.6344	0.5360	**0.5236**
Self-Jud	0.3272	0.5500	0.5609	0.2468	0.4220	0.3327	0.1993	0.4780	0.4236
CoT	0.6435	0.5700	0.5255	0.1463	0.4200	0.3218	0.2029	0.4760	0.4255
Challenge	**0.8080**	0.5280	0.4891	~~0.8995~~	~~0.5800~~	~~0.6782~~	**0.8007**	0.5240	**0.5709**
Punish	0.3272	0.5300	0.5164	0.1298	0.4200	0.3218	0.4936	0.5300	0.4345
Prob-Thr	0.5960	**0.5820**	**0.5855**	**0.7971**	**0.6140**	**0.6091**	0.6910	**0.6060**	0.5218
Random	0.5448	0.5700	0.5327	0.8976	**0.7080**	**0.6709**	0.8026	0.6460	0.6000
Noised Img	0.7313	0.6000	0.5400	0.8958	0.6740	0.6655	0.8062	0.6300	0.5818
Rephr	0.8026	0.5660	0.5364	0.8976	0.6920	0.6672	0.8080	0.6260	0.5764
Reph+Nois	0.7733	0.5500	0.5509	**0.9013**	0.6780	0.6655	**0.8099**	0.6120	0.5618
Cross Model	**0.8208**	**0.6320**	**0.5800**	0.8976	0.6520	0.6618	0.8062	**0.6740**	**0.5964**
PPL Thr	0.7916	0.6020	0.6073	0.8519	0.7060	0.6800	0.7934	0.6280	0.5345

Table 1: Performance of alignment on three datasets and three LVLMs. Best results of each kind of confidence in **bold** and second best in underline. Experimental observations show that LLaVA demonstrates a pattern of complete answer denial when being challenged. We therefore omitted these data from our results.

captures semantics by evaluating answer consistency (Kuhn et al., 2023), achieving higher alignment. Additionally, while verbalized confidence is uncalibrated, eliciting answer consistency-based confidence calibrating a threshold on a held-out set, further improves alignment.

Despite answer consistency-based confidence exhibiting high alignment, it comes at a cost: eliciting this kind of confidence requires generating multiple responses, incurring high computational costs. And its reliance on a held-out set for threshold calibration limits its generalizability.

2) **Probabilistic confidence surpasses verbalized confidence in alignment performance.** As shown in Table 1, probabilistic confidence's alignment performance consistently surpasses verbalized confidence, and it outperforms answer consistency-based confidence on Qwen2.5-VL. Though it falls behind consistency-based confidence on LLAVA-1.5 and DeepSeek-VL2, the alignment differences are small. Additionally, it functions more efficiently without the high computational cost of generating multiple responses.

However, probabilistic confidence, like answer consistency-based confidence, still requires threshold calibration on a held-out set, which affects its generalizability.

3) **Verbalized confidence demonstrates lower alignment compared to probabilistic and answer consistency-based confidences, and judges answers overconfidently.** Compared to probabilistic and answer consistency-based confidences, eliciting verbalized confidence is computationally efficient and generalizable. However, as shown in Table 1, both single-step (Vanilla) and double-step (Self-Jud) verbalized confidences' alignment are lower than the other two confidences. To investigate the cause of it, we calculate the conservativeness and overconfidence on verbalized confidence, as shown in Table 2, we find that the ratio of overconfident responses is substantially higher than conservative responses. This pattern suggests that LVLMs, like LLMs, are intrinsically biased toward affirming their own output (Groot and Valdenegro-Toro, 2024; Sun et al., 2025).

method	Dyn-VQA		Visual7W		MMMU Pro	
	Conser.	Overco.	Conser.	Overco.	Conser.	Overco.
Vanilla	0.1024	0.1353	0.0900	0.3260	0.1327	0.3764
CoT	0.0786	0.1389	0.1340	0.2580	0.1127	0.2055
Punish	0.0804	0.2084	0.0820	0.3660	0.1127	0.3873
Self-Jud	0.0018	0.6709	0.0180	0.4320	0.0701	0.3692
CoT	0.0329	0.3236	0.0280	0.4020	0.1000	0.3745
Punish	0.0018	0.6709	0.0120	0.4580	0.0200	0.4636

Table 2: The performance of verbalized confidence on Qwen2.5-VL, single-step confidences are in blue and souble-step confidences are in orange .

Table 2 also shows that double-step verbalized confidence exhibits more severe overconfidence than its single-step counterpart. This may be because the model's self-generated answers in the first round of dialogue serve as false positive signals of

method	Dyn-VQA			Visual7W			MMMU Pro		
	Acc.	Align.	Overco.	Acc.	Align.	Overco.	Acc.	Align.	Overco.
Vanilla	0.1846	0.7623	0.1353	0.4380	0.5840	**0.3260**	0.4564	0.4909	**0.3764**
CoT	**0.2121**	0.7824	0.1389	0.4920	0.6080	0.2580	0.6436	0.6818	0.2055
Img-CoT	0.2048	0.7276	**0.2066**	**0.5020**	0.6060	0.3080	**0.6636**	**0.7182**	0.1691
Explain	0.1956	**0.8117**	0.0823	0.4740	**0.6180**	0.2720	0.5309	0.5782	0.2982

Table 3: The performance of single-step reasoning elicitation methods on Qwen2.5-VL.

its capability, reinforcing overconfident behavior through misleading model to self-affirmation.

4.4.2 Confidence Calibration in LVLMs

In this section, we evaluate the effectiveness of existing confidence calibration methods developed for LLMs in the context of LVLMs, as well as our proposed methods.

For existing confidence calibration methods, our observations are as follows:

1) **Single-step reasoning elicitation methods effectively enhance the accuracy and alignment of LVLMs.** As shown in Table 1, we found reasoning elicitation methods (Explain, CoT, and Img-CoT) exhibit high alignment. To further investigate them, we calculate other metrics about them. Table 3 shows that different reasoning elicitation methods excel on specific datasets: CoT method improves alignment and accuracy across all datasets and causes overconfidence on Dyn-VQA. The Explain method outperforms CoT in alignment on Visual7W and Dyn-VQA datasets. This observed difference may stem from the Explain method's design: while the CoT method enforces step-by-step reasoning, the Explain method prioritizes direct justification, thus reducing redundant context for simple questions and improving the calibration of LVLMs' confidence outputs.

2) **Answer consistency-based confidence calibration methods improve alignment on Qwen2.5-VL, but show limited effectiveness on other models.** We observed that, even when sampling responses at the same temperature of 1.0, models differ in their output diversity. As shown in Table 1, when random sampling Qwen2.5-VL's responses, it tends to generate consistent yet incorrect responses, resulting in low alignment. However, both the rephrasing and the noised image methods show effectiveness in mitigating this tendency, consequently achieving higher alignment. In contrast, LLaVA-1.5 and DeepSeek-VL2 generate more diverse outputs when the response is incorrect, allowing the Random Sampling method to perform well and making Noised Image and Rephrasing methods less effective in enhancing alignment by comparison.

We propose Image Chain of Thought, Probability Threshold, Cross Model Consistency methods in Section 4.1, their performances are as follows:

1) **Image Chain of Thought method effectively enhances alignment and accuracy on MMMU Pro.** As shown in Table 3, Img-CoT demonstrates remarkable performance on the MMMU Pro dataset, which requires both strong visual perception and reasoning capabilities. It improves accuracy and alignment, outperforming CoT method. This indicates its converting visual modality into language modality mechanism can effectively enhance models' comprehension of the content in the image, thereby achieving this superior performance.

2) **Probability Threshold method shows higher alignment than other double-step verbalized confidence calibration methods.** As shown in Table 1, the Probability Threshold method outperforms alternative double-step methods. Despite the need to calibrate the threshold, it effectively enhances alignment.

3) **Cross Model method outperforms other answer consistency based methods on LLaVA-1.5 and Qwen2.5-VL.** Table 1 demonstrates that the Cross Model method performs superior alignment on both LLaVA-1.5 and Qwen2.5-VL compared to other answer consistency-based confidence calibration methods. This improved performance validates the method's effectiveness, as it establishes a robust mechanism for evaluating answer correctness by leveraging responses from diverse models.

5 Perception Comparison Between LVLMs and LLMs

Compared to LLMs, LVLMs need to process additional visual modality and integrate information

method	Model	Qwen2.5					DeepSeek-VL2					LLaVA-1.5				
		Unc-R.	Acc	Align.	Conser.	Overco.	Unc-R.	Acc	Align.	Conser.	Overco.	Unc-R.	Acc	Align.	Conser.	Overco.
Vanilla	LVLM	0.782	0.185	**0.762**	0.102	**0.135**	**0.788**	0.146	**0.653**	**0.141**	0.207	**0.490**	0.088	**0.534**	**0.022**	0.444
	LLM	**0.788**	**0.285**	0.729	**0.172**	0.099	0.161	**0.225**	0.338	0.024	**0.638**	0.011	**0.141**	0.152	0.001	**0.848**
CoT	LVLM	0.728	0.212	**0.782**	**0.079**	0.139	**0.638**	0.170	**0.636**	**0.086**	0.278	**0.512**	0.084	**0.538**	**0.031**	0.431
	LLM	0.448	**0.294**	0.651	0.046	**0.304**	0.095	**0.296**	0.362	0.015	**0.623**	0.117	**0.199**	0.302	0.007	**0.691**
Punish	LVLM	0.711	0.168	**0.711**	0.080	**0.208**	**0.848**	0.161	**0.709**	**0.150**	0.141	**0.450**	0.095	**0.490**	**0.027**	0.483
	LLM	**0.956**	**0.294**	0.713	**0.269**	0.018	0.266	**0.229**	0.455	0.020	**0.525**	0.057	**0.152**	0.201	0.004	**0.795**
Explain	LVLM	**0.828**	0.196	**0.812**	**0.106**	0.082	**0.786**	0.168	**0.698**	**0.128**	0.174	**0.421**	0.084	**0.453**	**0.027**	0.519
	LLM	0.536	**0.298**	0.673	0.080	**0.247**	0.079	**0.252**	0.320	0.006	**0.675**	0.159	**0.219**	0.364	0.007	**0.629**

Table 4: LLMs and LVLMs comparison for single-step verbalization based methods on Dyn-VQA.

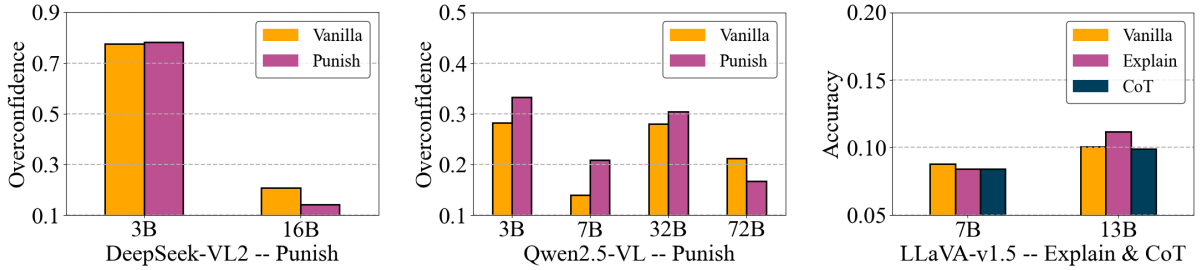

Figure 2: Comparative analysis of instruction following ability across model scales.

across different modalities. This raises a question: how does the perception of LVLMs differ from that of LLMs? Knowing these distinctions is valuable for developing trustworthy LVLMs.

In this section, we investigate the difference of knowledge boundary perception between LVLMs and their LLM counterparts. Focusing on verbalized confidence cause it directly reflects models' self-awareness of their knowledge boundaries. We further propose several hypotheses about these differences' underlying causes and validate them through the comparison between different model scales and input modalities.

5.1 Experimantal Setup

Datasets. In this section, we mainly focus on Dyn-VQA dataset. Dyn-VQA provides both VQA question image pairs and their semantically equivalent QA questions (e.g., QA: "How many humans have landed on Mars?" vs. VQA: "How many humans have landed on this planet?" with an image of Mars). This enables fair model performance comparison across text-only modality and vision-text modality inputs.

Models. In this section, we compare LVLMs with their base LLM counterparts to ensure fair comparison: Qwen2.5-VL, DeepSeek-VL2, LLaVA-v1.5 vs Qwen2.5, DeepSeek-MoE, Vicuna-v1.5.

5.2 Results and Analysis

Here, we apply VQA queries on LVLMs, and their semantic equivalent QA queries on LLMs to fairly compare them. And focus on single-step verbalized confidence. We defer results about other kinds of confidence to Appendix A.3. Here are our findings:

1) **Compared to LLMs, LVLMs struggle to follow certain methods' instructions, leading to performance deviating from expected.** As shown in Table 4, Qwen2.5-VL cannot effectively follow the Punish instruction. As a result, this method not only fails to reduce overconfidence but actually exacerbates it, leading to lower alignment than Vanilla. Similarly, LLaVA-1.5 disregards CoT and Explain instructions, persistently generating responses without proper reasoning or explanation, which results in lower accuracy. This stands in contrast to LLMs, where the Punish method effectively reduces Qwen2.5's overconfidence; CoT and Explain instructions reliably ignite reasoning responses in Vicuna-1.5, thus improving its accuracy.

2) **For single-step verbalized confidence, LVLMs tend to have lower accuracy compared to LLMs. Along with higher alignment due to reduced overconfidence.** As shown in Table 4, under all single-step verbalized confidence for the three series of models, the answer accuracy of LVLMs is lower than that of LLMs. Meanwhile, LVLMs exhibit a higher uncertain-rate compared to LLMs. Specifically, LLaVA exhibits an average accuracy reduction of 0.09 with a concurrent 0.382 increase in uncertain-rate than its counterpart LLM. And in DeepSeek-VL2, we observe an 0.089 accuracy decrement paired with a 0.615 surge in uncertainty

than LLM. Compared to LLMs, LVLMs' accuracy drop is relatively smaller than their uncertain-rate increase, thus they demonstrate less severe overconfidence than LLMs, leading to relatively higher alignment in their responses.

5.3 Analysis Across Model Scales and Modalities

Building upon the findings discussed in the previous subsection, we observe notable performance distinctions between LLMs and LVLMs, which motivate us to propose the following hypothesis regarding their potential underlying causes:

1) Model capacity bottleneck: We hypothesize that the inferior instruction-following abilities of LVLMs stems from their internal capacity limitations, where visual modality integration competes for models' internal parameter resources that would otherwise support language processing capabilities.

2) Cross-modal limitation awareness: While the LVLMs demonstrate lower accuracy than LLMs, their verbalized confidence shows better alignment with performance. We hypothesize this stems from two factors: (1) LVLMs' constrained cross-modal processing ability leads to degraded multimodal VQA accuracy, and (2) LVLMs' awareness of this limitation results in higher alignment.

To validate our capacity hypothesis of instruction following ability, we conduct a comparative analysis on different scale models and find that:

As LVLMs scale up, they generally exhibit stronger instruction following capabilities. As shown in Figure 2. For Qwen2.5-VL and DeepSeek-VL2, the Punish method effectively reduces overconfidence in larger models (Qwen2.5-VL-72B, DeepSeek-Vl2-16B) but shows limited impact on smaller ones (< 32B Qwen2.5-VL, DeepSeek-VL2-3B). For LLaVA-1.5, the 13B model follows Explain instruction which 7B model not follows, thus Explain improves accuracy in the 13B model.

These phenomena supports our hypothesis: the parameter constraints of small scale LVLMs create a dilemma between visual processing and linguistic comprehension, resulting in degraded language understanding and consequently weaker instruction following ability. In contrast, larger LVLMs allocate more parameters to language processing, maintaining strong language ability while handling multimodal inputs, thus demonstrating stronger instruction following ability.

To validate our accuracy and alignment hypothesis, we conduct comparative analysis on text-only QA, vision-text VQA, and vision-only "V"QA modality of queries on LVLMs, our results reveal that:

Model	Task	Dyn-VQA				
		Unc-R.	Acc	Align.	Conser.	Overco.
Qwen2.5-VL	"V"QA	0.461	0.223	0.578	0.053	**0.369**
	VQA	**0.782**	0.185	**0.762**	0.102	0.135
	QA	0.766	**0.252**	0.700	**0.159**	0.141
DeepSeek-VL2	"V"QA	0.227	0.208	0.435	0.000	**0.565**
	VQA	**0.788**	0.146	**0.653**	**0.141**	0.207
	QA	0.545	**0.256**	0.559	0.121	0.320

Table 5: The performance of LVLMs under different query modalities, we add text question at the bottom of the image to generate pure image "V"QA query.

LVLMs exhibit lower accuracy but higher alignment when responding to multimodal VQA queries. As shown in Table 5, both models demonstrate lower accuracy when answering VQA queries that demand cross-modal understanding ability compared to pure text QA and pure image "V"QA queries. Concurrently, they demonstrate increased uncertain-rate and improved confidence performance alignment for these multimodal queries.

These observations support our hypothesis:

1. Limited cross-modal ability: LVLMs struggle to effectively synthesize information across modalities, leading to reduced answering accuracy on multimodal queries compared to unimodal queries.

2. Capability awareness: When encountering challenging multimodal queries, LVLMs exhibit self-awareness of their limited ability through generating more uncertainty responses. This decreases overconfidence and thus improve alignment.

6 Conclusion

In this paper, we present a systematic investigation of knowledge boundary perception in LVLMs, assessing this ability through alignment. First, we evaluate three kinds of confidence, and observe that answer consistency-based confidence reaches the highest alignment, whereas verbalized confidence induces overconfidence. We also evaluate several confidence calibration methods, with our results revealing that reasoning elicitation methods improve accuracy and alignment, while our proposed methods show effectiveness. Second, we compare LVLMs with LLMs, and reveal that while LVLMs exhibit lower QA accuracy, they achieve higher alignment, which is attributable to LVLMs' awareness of their multimodal integration ability limitation. We also observe that LVLMs have weaker instruction following ability than LLMs.

Limitations

First, due to dataset constraints, we only compared LVLMs and LLMs on Dyn-VQA; broader benchmarks are needed for future validation. Second, our analysis did not examine internal model states, leaving internal mechanistic differences in knowledge boundary perception underexplored. Third, we focused on binary confidence measures; extending this to continuous confidence scales could yield finer-grained insights. These limitations highlight directions for future work on LVLM evaluation and interpretability.

Ethics Statement

In this paper, all the datasets we use are open-source, and the models we employ are either open-source or widely used. Furthermore, the methods we propose do not induce the model to output any harmful information.

References

Ayush Agrawal, Mirac Suzgun, Lester Mackey, and Adam Tauman Kalai. 2024. Do language models know when they're hallucinating references?

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when it's lying.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2025a. Qwen2.5-vl technical report.

Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. 2025b. Hallucination of multimodal large language models: A survey.

Zhuo Chen, Xinyu Wang, Yong Jiang, Zhen Zhang, Xinyu Geng, Pengjun Xie, Fei Huang, and Kewei Tu. 2025. Detecting knowledge boundary of vision large language models by sampling-based inference.

Nathan Cooper and Torsten Scholak. 2024. Perplexed: Understanding when large language models are confused.

Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, Yang Zhou, Kaizhao Liang, Jintai Chen, Juanwu Lu, Zichong Yang, Kuei-Da Liao, Tianren Gao, Erlong Li, Kun Tang, Zhipeng Cao, Tong Zhou, Ao Liu, Xinrui Yan, Shuqi Mei, Jianguo Cao, Ziran Wang, and Chao Zheng. 2024. A survey on multimodal large language models for autonomous driving. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops, pages 958–979.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers.

Tobias Groot and Matias Valdenegro-Toro. 2024. Overconfidence is key: Verbalized uncertainty evaluation in large language and vision-language models.

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

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017b. On calibration of modern neural networks.

Yutao Hu, Tianbin Li, Quanfeng Lu, Wenqi Shao, Junjun He, Yu Qiao, and Ping Luo. 2024. Omnimed-vqa: A new large-scale comprehensive evaluation benchmark for medical lvlm. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 22170–22183.

Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. 2024. Senna: Bridging large vision-language models and end-to-end autonomous driving.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

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

Wei Lan, Wenyi Chen, Qingfeng Chen, Shirui Pan, Huiyu Zhou, and Yi Pan. 2024. A survey of hallucination in large visual language models.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2024. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 13872–13882.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. 2023. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. In Advances in Neural Information Processing Systems, volume 36, pages 28541–28564. Curran Associates, Inc.

Jiaming Li, Jiacheng Zhang, Zequn Jie, Lin Ma, and Guanbin Li. 2025a. Mitigating hallucination for large vision language model by inter-modality correlation calibration decoding.

Yangning Li, Yinghui Li, Xinyu Wang, Yong Jiang, Zhen Zhang, Xinran Zheng, Hui Wang, Hai-Tao Zheng, Fei Huang, Jingren Zhou, and Philip S. Yu. 2025b. Benchmarking multimodal retrieval augmented generation with dynamic vqa dataset and self-adaptive planning agent.

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

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.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024b. Improved baselines with visual instruction tuning.

Huan Ma, Jingdong Chen, Guangyu Wang, and Changqing Zhang. 2025. Estimating llm uncertainty with logits.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023a. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023b. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. arXiv preprint arXiv:2303.08896.

Viktor Moskvoretskii, Maria Lysyuk, Mikhail Salnikov, Nikolay Ivanov, Sergey Pletenev, Daria Galimzianova, Nikita Krayko, Vasily Konovalov, Irina Nikishina, and Alexander Panchenko. 2025. Adaptive retrieval without self-knowledge? bringing uncertainty back home.

Shiyu Ni, Keping Bi, Jiafeng Guo, and Xueqi Cheng. 2024. When do llms need retrieval augmentation? mitigating llms' overconfidence helps retrieval augmentation.

Shiyu Ni, Keping Bi, Jiafeng Guo, Lulu Yu, Baolong Bi, and Xueqi Cheng. 2025. Towards fully exploiting llm internal states to enhance knowledge boundary perception.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and et al. Red Avila. 2024. Gpt-4 technical report.

Pranab Sahoo, Prabhash Meharia, Akash Ghosh, Sriparna Saha, Vinija Jain, and Aman Chadha. 2024. A comprehensive survey of hallucination in large language, image, video and audio foundation models.

Fengfei Sun, Ningke Li, Kailong Wang, and Lorenz Goette. 2025. Large language models are overconfident and amplify human bias.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback.

Xintong Wang, Jingheng Pan, Liang Ding, and Chris Biemann. 2024a. Mitigating hallucinations in large vision-language models with instruction contrastive decoding.

Yuhao Wang, Zhiyuan Zhu, Heyang Liu, Yusheng Liao, Hongcheng Liu, Yanfeng Wang, and Yu Wang. 2024b. Drawing the line: Enhancing trustworthiness of mllms through the power of refusal.

Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, Zhenda Xie, Yu Wu, Kai Hu, Jiawei Wang, Yaofeng Sun, Yukun Li, Yishi Piao, Kang Guan, Aixin Liu, Xin Xie, Yuxiang You, Kai Dong, Xingkai Yu, Haowei Zhang, Liang Zhao, Yisong Wang, and Chong Ruan. 2024. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding.

Wenyi Xiao, Ziwei Huang, Leilei Gan, Wanggui He, Haoyuan Li, Zhelun Yu, Fangxun Shu, Hao Jiang, and Linchao Zhu. 2025. Detecting and mitigating hallucination in large vision language models via fine-grained ai feedback. Proceedings of the AAAI Conference on Artificial Intelligence, 39(24):25543–25551.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms.

Adam Yang, Chen Chen, and Konstantinos Pitas. 2024a. Just rephrase it! uncertainty estimation in closed-source language models via multiple rephrased queries.

Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. 2024b. Alignment for honesty.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know?

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. 2024. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark.

Dylan Zhang, Xuchao Zhang, Chetan Bansal, Pedro Las-Casas, Rodrigo Fonseca, and Saravan Rajmohan. 2023. Pace-lm: Prompting and augmentation for calibrated confidence estimation with gpt-4 in cloud incident root cause analysis.

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley A. Malin, and Sricharan Kumar. 2024. Sac3: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency.

Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. 2024. Analyzing and mitigating object hallucination in large vision-language models.

Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7w: Grounded question answering in images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

A Appendix

A.1 Prompts

A.1.1 Single Step Verbalization Based Prompts

Vanilla. *Answer the question based on your internal knowledge and the image. If you are sure the answer is accurate and correct, please say "certain" after the answer. If you are not confident with the answer, please say "uncertain".*
Question: [Question]
Answer:

CoT. *Answer the question based on your internal knowledge and the image.* ***Analyse step by step****. If you are sure the answer is accurate and correct, please say "certain" after the answer. If you are not confident with the answer, please say "uncertain".*
Question: [Question]
Answer:

Img-CoT. *Answer the question based on your internal knowledge and the image.* ***First, describe the image, then analyse step by step****. If you are sure the answer is accurate and correct, please say "certain" after the answer. If you are not confident with the answer, please say "uncertain".*
Question: [Question]
Answer:

Punish. *Answer the question based on your internal knowledge and the image. If you are sure the answer is accurate and correct, please say "certain" after the answer. If you are not confident with the answer, please say "uncertain".* ***You will be punished if the answer is not right but you say "certain".***
Question: [Question]
Answer:

Explain. *Answer the question based on your internal knowledge and the image,* ***explain why you give this answer.*** *If you are sure the answer is accurate and correct, please say "certain" after the answer. If you are not confident with the answer, please say "uncertain".*
Question: [Question]
Answer:

A.1.2 Double Step Verbalization Based Prompts

For the double step verbalization based methods, we first prompt the model to generate answer, then

prompt the model to give its confidence in the second round chat.

First Round Answer Generation. *Answer the question based on your internal knowledge and the image.*
Question: [Question]
Answer:

Self-Judging. *If you are sure your previous answer is accurate and correct, please say "certain", If you are not confident with the answer, please say "uncertain".*

CoT. *If you are sure your previous answer is accurate and correct, please say "certain", If you are not confident with the answer, please say "uncertain". **Analyse step by step**, then provide Your judgement.*

Challenge. ***I don't think your answer is right**, if you still think your answer is right, please say "ceratin". Otherwise, say "uncertain".*

Punish. *If you are sure your previous answer is accurate and correct, please say "certain", If you are not confident with the answer, please say "uncertain". **You will be punished if the answer is not right but you say "certain".***

Probability+Threshold. *Provide the probability that your answer is correct (0.0 to 1.0). Give ONLY the probability, no other words or explanation.*

A.1.3 Answer Consistency Based Prompts

Rephrasing. *Based on the Following question, generate [number of semantical equivalent questions] semantically equivalent questions. your output should be a list of strings and add a sequnce number with a dot at the start of each output question, like [1."question1",2."question2",...].*
Question: [The original question]
Semantically equivalent questions:

A.2 LVLMs' Knowledge Boundary Perception Ability

A.2.1 Implementation Details

In this section, we provide a detailed introduction to our implementation details.

For content generation, we mainly utilize APIs to generate answers.

For verbalization based methods, we set the model temperature to 0 and set a fixed seed to obtain high-quality and relatively consistent responses. Notably, Probability Threshold method is exclusively employed in a double round form because we find some of the models struggle to generate both continuous probabilities and answers in a single round.

For the consistency based methods, we implemente a two-phase generation protocol: First, generating a reference answer with temperature = 0; Then sampling 10 variant answers with temperature = 1.0, with semantic equivalence between the basic answer and sampled answers evaluated by Qwen2.5-0.5B. With this process, we can get a consistency score between 0 to 10.

Specifically, for question rephrasing method, we leveraged Qwen2.5-7B to produce semantically equivalent question paraphrases. For the noised image method, we progressively added zero-mean Gaussian noise to the images during sampling, with the standard deviation incrementally increased from 0 in steps of 0.05. And for the cross model consistency method, we computed consistency scores using a combination of four responses generated by the primary model and three responses each from two other reference models.

A.2.2 Complete Results

Table 6, Table 7 and Table 8 present the comprehensive performance evaluation of all methods across the three benchmark datasets and three LVLMs employed in our study.

A.2.3 Observations and Analysis

We proposed our mainly findings about LVLMs' knowledge boundary perception methods in Section 4.4. Here, we discuss more detailed observations about them.

1) The Explain method improves alignment for both Deepseek-VL2 and Qwen2.5 when tested on the Dyn-VQA and Visual7W datasets. This demonstrates its effectiveness in enhancing LVLMs' knowledge boundary perception when processing relatively simple input questions.

2) The single-step Chain of Thought method effectively enhances alignment, whereas its double-step counterpart often leads to overconfidence and only marginally improves alignment for Qwen2.5-VL.

3) Both single-step and double-step Punish methods demonstrate limited effectiveness in mitigating overconfidence for Qwen2.5-VL and LLaVA-v1.5, as they fail to properly follow Punish Instructions.

4) Challenge method induces very high uncertain-rate in both three models, indicating that LVLMs are easily swayed by the output judgements.

5) For Qwen2.5-VL, rephrasing methods improve alignment on the Dyn-VQA dataset (language-focused), while the noise image method enhances performance on Visual7W (vision-focused). The combination of these two methods boosts alignment on the MMMU Pro dataset, which requires both language and vision comprehension. This reveals an interesting relationship between perturbation modalities and input query types.

A.3 Comparing Perception between LVLMs and LLMs

While the main body presents a comparative analysis of single-step verbalization based confidence elicitation methods between LLMs and LVLMs, this section provides an extensive evaluation of: (i) double step verbalization based methods, (ii) answer consistency based methods, and (iii) token probability based method. The results can be found in Table 9. The main observations are as follows.

A.3.1 Double Step Verbalization Based Methods

For double step verbalization based methods, the difference in performance between LLM and LVLM varies with the method.

1) For the Self-Judging method, Qwen2.5 exhibits higher alignment than Qwen2.5-VL. In contrast, the LLM counterparts of DeepSeek-VL2 and LLaVA tend to respond with "certain" to nearly all answers, resulting in extremely low consistency. This indicates a severe bias toward overconfident responses in these two LLMs.

2) For the Challenge method, LVLMs demonstrate higher uncertain-rates than LLMs, often approaching to near 1.0. This suggests that LVLMs are more likely to trust external judgments and consequently undermine their own decisions.

3) Under the Double-step Punish method, LLMs outperform LVLMs due to their stronger instruction following ability, achieving higher consistency and lower overconfidence.

A.3.2 Answer Consistency Based Methods

For answer consistency based methods, our observations are as follows:

1) Answer consistency based methods demonstrate superior alignment performance in LVLMs compared to LLMs.

2) DeepSeek-MoE exhibits strong consistency in its generated answers, maintaining high answer uniformity even when the outputs are incorrect. This behavior persists across both random sampling and rephrasing methods, leading to sustained overconfidence and suboptimal alignment performance.

3) The rephrasing strategy shows limited effectiveness in improving alignment metrics across all evaluated models, with the notable exception of Qwen2.5-VL. This observation holds true for both LVLMs and LLMs in our results.

A.3.3 Token Probability Based Methods

For the token probability based approach, as shown in Table 9, our results reveal that LLMs exhibit relatively weaker confidence-accuracy alignment compared to LVLMs.

method	Dyn-VQA					Visual7W					MMMU Pro				
	Unc-R.	**Acc**	**Align.**	**Conser.**	**Overco.**	**Unc-R.**	**Acc**	**Align.**	**Conser.**	**Overco.**	**Unc-R.**	**Acc**	**Align.**	**Conser.**	**Overco.**
Vanilla	0.7824	0.1846	0.7623	0.1024	0.1353	0.3260	0.4380	0.5840	0.0900	0.3260	0.3000	0.4564	0.4909	**0.1327**	0.3764
CoT	0.7276	**0.2121**	0.7824	0.0786	0.1389	**0.3840**	0.4920	0.6080	**0.1340**	0.2580	0.2636	0.6436	0.6818	0.1127	0.2055
Img-CoT	0.6545	0.2048	0.7276	0.0658	0.2066	0.2760	**0.5020**	0.6060	0.0860	0.3080	0.2800	**0.6636**	**0.7182**	0.1127	0.1691
Punish	0.7112	0.1682	0.7112	0.0804	**0.2084**	0.2880	0.4280	0.5520	0.0820	**0.3660**	0.2691	0.4564	0.5000	0.1127	**0.3873**
Explain	**0.8282**	0.1956	**0.8117**	**0.1060**	0.0823	0.3640	0.4740	**0.6180**	0.1100	0.2720	0.2945	0.5309	0.5782	0.1236	0.2982
Self-Judging	0.1426	0.1883	0.3272	0.0018	**0.6709**	0.1100	0.4760	0.5500	0.0180	0.4320	0.1882	0.5127	0.5609	0.0701	0.3692
CoT	0.5210	0.1883	0.6435	0.0329	0.3236	0.1500	0.4760	0.5700	0.0280	0.4020	0.2127	0.5127	0.5255	0.1000	0.3745
Challenge	**0.9671**	0.1883	**0.8080**	**0.1737**	0.0183	**0.9800**	0.4760	0.5280	**0.4640**	0.0080	**0.9873**	0.5127	0.4891	**0.5055**	0.0055
Punish	0.1426	0.1883	0.3272	0.0018	**0.6709**	0.0780	0.4760	0.5300	0.0120	**0.4580**	0.0436	0.5127	0.5164	0.0200	**0.4636**
Prob-Thr	0.4991	0.1883	0.5960	0.0457	0.3583	0.2140	0.4760	**0.5820**	0.0540	0.3640	0.4764	0.5127	**0.5855**	0.2018	0.2127
Random	0.4625	0.1883	0.5448	0.0530	**0.4022**	**0.3020**	0.4760	0.5700	0.1040	**0.3260**	0.4309	0.5127	0.5327	0.2055	0.2618
Noised Img	0.7733	0.1883	0.7313	0.1152	0.1536	0.4920	0.4760	0.6000	0.1840	0.2160	0.3873	0.5127	0.5400	0.1800	**0.2800**
Rephr	**0.9543**	0.1883	0.8026	**0.1700**	0.0274	0.4340	0.4760	0.5660	0.1720	0.2620	**0.5655**	0.5127	0.5364	**0.2709**	0.1927
Reph+Nois	0.8958	0.1883	0.7733	0.1554	0.0713	0.4940	0.4760	0.5500	**0.2100**	0.2400	0.4418	0.5127	0.5509	0.2018	0.2473
Cross Model	0.9469	0.1883	**0.8208**	0.1572	0.0219	**0.5720**	0.4760	**0.6320**	0.2080	0.1600	0.5036	0.5127	**0.5800**	0.2182	0.2018
PPL Thr	0.8885	0.1993	0.7916	0.1481	0.0603	0.8060	0.4760	0.6020	0.3400	0.0580	0.9436	0.4091	0.6073	0.3727	0.0200

Table 6: The performance of different methods on Qwen2.5-VL-7B-Instruct.

method	Dyn-VQA					Visual7W					MMMU Pro				
	Unc-R.	**Acc**	**Align.**	**Conser.**	**Overco.**	**Unc-R.**	**Acc**	**Align.**	**Conser.**	**Overco.**	**Unc-R.**	**Acc**	**Align.**	**Conser.**	**Overco.**
Vanilla	0.4899	0.0878	0.5338	0.0219	0.4442	**0.0260**	0.3920	0.4140	0.0020	0.5840	0.0855	0.2018	0.2509	0.0182	0.7309
CoT	0.5119	0.0841	0.5375	0.0311	0.4314	0.0220	0.3840	0.3940	**0.0060**	0.6000	0.1418	0.1545	0.2418	0.0273	**0.7309**
Img-CoT	**0.5265**	0.0914	**0.5484**	**0.0347**	0.4168	0.0180	**0.4000**	0.4140	0.0020	0.5840	0.1527	0.2527	0.2964	0.0545	0.6491
Punish	0.4497	**0.0951**	0.4899	0.0274	0.4826	0.0260	0.3960	**0.4180**	0.0020	0.5800	**0.2291**	**0.2727**	**0.3745**	**0.0636**	0.5618
Explain	0.4205	0.0841	0.4534	0.0274	**0.5192**	0.0100	0.3840	0.3900	0.0020	**0.6080**	0.0727	0.1709	0.2109	0.0164	0.7727
Self-Judging	0.1718	0.1005	0.2468	0.0128	0.7404	0.0020	0.4200	0.4220	0.0000	0.5780	0.0109	0.3218	0.3327	0.0000	0.6673
CoT	0.0494	0.1005	0.1463	0.0018	0.8519	0.0000	0.4200	0.4200	0.0000	**0.5800**	0.0000	0.3218	0.3218	0.0000	**0.6782**
Challenge	~~1.0000~~	~~0.1005~~	~~0.8995~~	~~0.1005~~	~~0.0000~~	~~1.0000~~	~~0.4200~~	~~0.5800~~	~~0.4200~~	~~0.0000~~	~~1.0000~~	~~0.3218~~	~~0.6782~~	~~0.3218~~	~~0.0000~~
Punish	0.0293	0.1005	0.1298	0.0000	**0.8702**	0.0000	0.4200	0.4200	0.0000	**0.5800**	0.0000	0.3218	0.3218	0.0000	**0.6782**
Prob-Thr	**0.8464**	0.1005	**0.7971**	**0.0750**	0.1280	**0.6780**	0.4200	**0.6140**	**0.2420**	0.1440	**0.7964**	0.3218	**0.6091**	**0.2545**	0.1364
Random	0.9872	0.1005	0.8976	0.0951	**0.0073**	0.6680	0.4200	**0.7080**	0.1900	0.1020	0.9745	0.3218	**0.6709**	0.3127	0.0164
Noised Img	0.9963	0.1005	0.8958	**0.1005**	0.0037	0.5660	0.4200	0.6740	0.1560	0.1700	**0.9836**	0.3218	0.6655	**0.3200**	0.0145
Rephr	0.9981	0.1005	0.8976	**0.1005**	0.0018	0.6560	0.4200	0.6920	0.1920	0.1160	0.9345	0.3218	0.6672	0.2945	**0.0382**
Reph+Nois	**0.9982**	0.1005	**0.9013**	0.0987	0.0000	**0.7020**	0.4200	0.6780	**0.2220**	0.1000	0.9655	0.3218	0.6655	0.3109	0.0236
Cross Model	**0.9982**	0.1005	0.8976	**0.1005**	0.0018	0.5320	0.4200	0.6520	0.1500	**0.1980**	0.9727	0.3218	0.6618	0.3164	0.0218
PPL Thr	0.8903	0.1005	0.8519	**0.0695**	0.0786	0.5860	0.4200	0.7060	0.1460	0.1380	0.9727	0.3218	0.6800	0.3073	0.0127

Table 7: The performance of different methods on LLaVA-v1.5-7B.

method	Dyn-VQA					Visual7W					MMMU Pro				
	Unc-R.	**Acc**	**Align.**	**Conser.**	**Overco.**	**Unc-R.**	**Acc**	**Align.**	**Conser.**	**Overco.**	**Unc-R.**	**Acc**	**Align.**	**Conser.**	**Overco.**
Vanilla	0.7879	0.1463	0.6527	0.1408	0.2066	0.4120	0.1840	0.2820	0.1580	**0.5600**	0.4091	0.2673	0.2727	0.2018	0.5255
CoT	0.6380	0.1700	0.6362	0.0859	0.2779	0.1780	0.4600	0.5540	0.0420	0.4040	0.0873	0.3509	0.3836	0.0273	**0.5891**
Img-CoT	0.5356	**0.2011**	0.6344	0.0512	**0.3144**	0.0640	**0.4960**	0.5360	0.0120	0.4520	0.1055	**0.4582**	**0.5236**	0.0200	0.4564
Punish	**0.8483**	0.1609	**0.7093**	**0.1499**	0.1407	**0.4580**	0.2680	0.3500	**0.1880**	0.4620	**0.4782**	0.3054	0.3145	**0.2345**	0.4509
Explain	0.7861	0.1682	0.6984	0.1280	0.1737	0.2780	0.4640	**0.5700**	0.0860	0.3440	0.2073	0.3382	0.3491	0.0982	0.5527
Self-Judging	0.0018	0.1974	0.1993	0.0000	**0.8007**	0.0020	0.4760	0.4780	0.0000	0.5220	0.0018	0.4255	0.4236	0.0018	**0.5745**
CoT	0.0055	0.1974	0.2029	0.0000	0.7971	0.0000	0.4760	0.4760	0.0000	**0.5240**	0.0073	0.4255	0.4255	0.0036	0.5709
Challenge	**0.9945**	0.1974	**0.8007**	**0.1956**	0.0037	**0.9960**	0.4760	0.5240	**0.4740**	0.0020	**0.9309**	0.4255	**0.5709**	**0.3927**	0.0364
Punish	0.3144	0.1974	0.4936	0.0091	0.4973	0.0620	0.4760	0.5300	0.0040	0.4660	0.0273	0.4255	0.4345	0.0091	0.5564
Prob-Thr	0.7239	0.1974	0.6910	0.1152	0.1938	0.7280	0.4760	**0.6060**	0.2980	0.0960	0.7473	0.4255	0.5218	0.3127	0.1655
Random	**0.9963**	0.1974	0.8026	**0.1956**	0.0018	0.5800	0.4740	0.6460	**0.2040**	0.1500	**0.8509**	0.4180	0.6000	**0.3345**	0.0655
Noised Img	0.9927	0.1974	0.8062	0.1920	0.0018	0.5480	0.4740	0.6300	0.1960	0.1740	0.7418	0.4180	0.5818	0.2891	**0.1291**
Rephr	0.9689	0.1974	0.8080	0.1792	0.0127	0.4480	0.4740	0.6260	0.1480	**0.2260**	0.7982	0.4180	0.5764	0.3200	0.1036
Reph+Nois	0.9670	0.1974	**0.8099**	0.1773	**0.0128**	0.5140	0.4740	0.6120	0.1880	0.2000	0.7945	0.4180	0.5618	0.3255	0.1127
Cross Model	**0.9963**	0.1974	0.8062	0.1938	0.0000	**0.6080**	0.4740	**0.6740**	**0.2040**	0.1220	0.8327	0.4180	**0.5964**	0.3273	0.0764
PPL Thr	0.8958	0.1974	0.7934	0.1499	0.0567	0.5500	0.4780	0.6280	0.2000	0.1720	0.9418	0.4436	0.5345	0.4255	0.0400

Table 8: The performance of different methods on DeepSeek-VL2-16B.

method	Model Type	Qwen2.5					LLaVA1.5					DeepSeek-VL2				
		Unc-R.	**Acc**	**Align.**	**Conser.**	**Overco.**	**Unc-R.**	**Acc**	**Align.**	**Conser.**	**Overco.**	**Unc-R.**	**Acc**	**Align.**	**Conser.**	**Overco.**
Self-Judging	LVLM	0.1426	0.1883	0.3272	0.0018	**0.6709**	**0.0018**	0.1974	0.1993	0.0000	**0.8007**	**0.1718**	0.1005	**0.2468**	**0.0128**	0.7404
	LLM	**0.2943**	**0.2998**	**0.5649**	**0.0146**	0.4205	0.0000	**0.2962**	**0.2962**	0.0000	0.7038	0.0000	**0.2139**	0.2139	0.0000	**0.7861**
CoT	LVLM	**0.5210**	0.1883	**0.6435**	**0.0329**	0.3236	0.0055	0.1974	0.2029	0.0000	**0.7971**	0.0494	0.1005	0.1463	0.0018	**0.8519**
	LLM	0.2925	**0.2998**	0.5411	0.0256	**0.4333**	**0.2888**	**0.2962**	**0.5192**	**0.0329**	0.4479	**0.2761**	**0.2139**	**0.4680**	**0.0110**	0.5210
Challenge	LVLM	**0.9671**	0.1883	**0.8080**	**0.1737**	0.0183	**0.9945**	0.1974	**0.8007**	0.1956	0.0037	**1.0000**	0.1005	**0.8995**	0.1005	0.0000
	LLM	0.7148	**0.2998**	0.7514	0.1316	**0.1170**	0.8684	**0.2962**	0.6563	**0.2541**	**0.0896**	0.9853	**0.2139**	0.7898	**0.2048**	**0.0055**
Punish	LVLM	0.1426	0.1883	0.3272	0.0018	**0.6709**	**0.3144**	0.1974	0.4936	0.0091	**0.4973**	0.0293	0.1005	0.1298	0.0000	**0.8702**
	LLM	**0.5448**	**0.2998**	**0.6910**	**0.0768**	0.2322	0.2852	**0.2962**	**0.5302**	**0.0256**	0.4442	**0.1974**	**0.2139**	**0.4113**	0.0000	0.5887
Prob-Thr	LVLM	0.4991	0.1883	0.5960	0.0457	**0.3583**	**0.7239**	0.1974	**0.6910**	**0.1152**	0.1938	0.8464	0.1005	**0.7971**	0.0750	**0.1280**
	LLM	**0.5941**	**0.2998**	**0.6709**	**0.1115**	0.2175	0.1773	**0.2962**	0.4333	0.0201	**0.5466**	**0.9963**	**0.2139**	0.7824	**0.2139**	0.0037
Random	LVLM	0.4625	0.1883	0.5448	0.0530	**0.4022**	**0.9963**	0.1974	**0.8026**	**0.1956**	0.0018	**0.9872**	0.1005	**0.8976**	0.0951	0.0073
	LLM	**0.9287**	**0.2998**	**0.7203**	**0.2541**	0.0256	0.4863	**0.2962**	0.5448	0.1188	**0.3364**	0.8921	**0.2139**	0.7806	**0.1627**	**0.0567**
Rephr	LVLM	**0.9543**	0.1883	**0.8026**	0.1700	0.0274	**0.9689**	0.1974	**0.8080**	**0.1792**	0.0127	**0.9981**	0.1005	**0.8976**	0.1005	0.0018
	LLM	0.9068	**0.2998**	0.7203	**0.2431**	**0.0366**	0.4991	**0.2962**	0.5539	0.1207	**0.3254**	0.8757	**0.2139**	0.7751	**0.1572**	**0.0676**
PPL Thr	LVLM	**0.8885**	0.1993	**0.7916**	0.1481	**0.0603**	**0.8903**	0.1005	**0.8519**	0.0695	0.0786	**0.8958**	0.1974	**0.7934**	**0.1499**	0.0567
	LLM	0.8519	**0.3217**	0.7313	**0.2212**	0.0475	0.7587	**0.2980**	0.6837	**0.1865**	**0.1298**	0.7458	**0.2121**	0.7422	0.1079	**0.1499**

Table 9: Performance comparison of double step verbaliztion based methods, consistency based methods and answer consistency based methods on the Dyn-VQA dataset: LVLMs vs. LLMs