From Entailment to Contradiction: Confidence Calibration in MLLMs via Difficulty-Guided Optimization

Anonymous ACL submission

Abstract

The ability to accurately assess and calibrate the confidence of large language models (LLMs) is critical for improving their performance in real-world applications. While much progress has been made in confidence calibration for text-only models, the challenge remains underdeveloped in multimodal large language models (MLLMs), which often suffer from hallucinations and poorly calibrated confidence due to factors such as visual ambiguity and the presence of rare entities. This research aims to tackle the problem of confidence miscalibration in MLLMs by proposing a novel two-stage framework. The first stage focuses on analyzing dataset characteristics, such as contradiction rates and entity rarity, which contribute to task difficulty. The second stage involves fine-tuning the model to better estimate confidence and reduce hallucinations. Our approach improves the reliability of confidence estimates and significantly reduces hallucination rates, offering a step forward in developing more trustworthy multimodal models.[1]

1 Introduction

Large Language Models (LLMs) (Touvron et al., 2023; Zhao et al., 2023a) have made significant breakthroughs in natural language understanding, generation, and reasoning (Hendrycks et al., 2020; Zhu et al., 2023; Qiao et al., 2022). Building on their success, Multimodal Large Language Models (MLLMs) are emerging as a new frontier, enabling models to process and reason over visual-text inputs.

Despite recent advances, MLLMs still struggle with output reliability, often generating hallucinations and expressing unwarranted confidence on visually unfamiliar or ambiguous inputs (Li et al., 2023; Yu et al., 2024). As shown in Figure 1, their confidence is poorly calibrated—especially on contradiction subsets—revealing a clear mismatch between confidence and accuracy. Although LLMs have made progress in calibration (Xu et al., 2024; Zhang et al., 2024), recent studies show that these text-only techniques do not transfer well to the multimodal setting. (Xuan et al., 2025) find that MLLMs exhibit *modality-dependent miscalibration*, driven by factors such as visual ambiguity, entity rarity, and image–text misalignment—issues not addressed by standard LLM calibration methods. These observations highlight a critical gap: MLLMs lack calibration mechanisms that explicitly incorporate visual uncertainty and multimodal contradiction patterns.

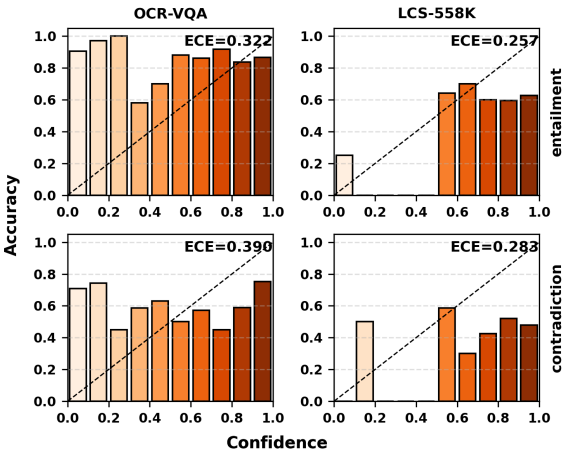

Figure 1: Calibration histograms of LLaVA on two datasets. Bars above the diagonal show underconfidence; below show overconfidence. Misalignment is most visible under contradiction subset. Lower ECE indicates better calibration.

We posit that reinforcement learning (RL) can be leveraged to enhance model alignment, thereby improving confidence estimation in LLMs. Concurrent work supports this view, showing that RL helps align model outputs with human preferred behaviors without introducing fundamentally new

[1]Our code is available at https://anonymous.4open.science/r/rllava-DF3C

reasoning capabilities (Shao et al., 2024; Yue et al., 2025). Moreover, previous research indicates that LLMs possess a degree of self-knowledge, the ability to assess their own uncertainty, even if not yet at the human level (Yin et al., 2023; Kadavath et al., 2022). Therefore, we aim to explore whether RL can be repurposed to amplify these innate confidence signals by rewarding calibration and penalizing over or under confidence, thus enhancing epistemic self-awareness and mitigating hallucinations.

Building on these insights, we propose a framework for confidence estimation and hallucination mitigation in MLLMs. The process unfolds in two stages. In Stage One, we sample multiple model responses to construct *better–worse* answer pairs and, based on these comparisons, partition 14 multimodal datasets into entailment, neutral, and contradiction subsets. We then apply Named Entity Recognition (NER) (Yan et al., 2021; Li et al., 2020) to decompose the *better* responses and analyze the entity distribution across approximately 80,000 examples, defining dataset difficulty levels by considering entity rarity. Notably, we observe a strong correlation between entity rarity and contradiction proportion, indicating that datasets with rarer entities and higher proportions of contradiction cases are more challenging and generally correspond to lower model performance. In Stage Two, (1) we finetune the model via supervised learning on the Stage-One data, encouraging it to produce explicit confidence estimates; (2) we apply GRPO (Shao et al., 2024) with a difficulty-guided reward scheme to further calibrate the model's confidence outputs while steering it toward more factual responses. Experimental results show that our model achieves well-calibrated confidence estimates and, through fine-grained reward guidance, substantially reduces overall hallucination rates.

Our contributions are summarized as follows:

- We build *better–worse* pairs for 14 datasets, split them into entailment, neutral, contradiction, and derive a rarity-based difficulty score from about 80k NER-parsed answers.
- SFT first teaches the model to produce calibrated confidence scores, and GRPO further aligns these scores with factual accuracy to suppress hallucinations. With our Dataset-aware, difficulty-guided reward yielding more reliable calibration and lower hallucination rates.

2 Related Works

2.1 LLM Alignment

Aligning LLMs with human intent has become a critical step in building trustworthy AI systems. Recent alignment pipelines follow a multi-stage training paradigm, starting with SFT (Chung et al., 2024) and followed by RLHF optimization (Bai et al., 2022). In the RLHF stage, we adopt the Group Relative Policy Optimization (GRPO) algorithm (Shao et al., 2024) as the core and integrate two key ideas from Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO) (Yu et al., 2025a): A fine-grained and difficulty-guided reward that gives lower reward for simple questions and higher reward when correctly answering harder ones, and Clip-Higher strategy ($\varepsilon_{low} = 0.2$, $\varepsilon_{high} = 0.28$) to expand exploration of low-probability tokens and prevent entropy collapse.

2.2 Confidence Estimation in LLMs

Estimating the confidence of multi-token LLM outputs poses a significant challenge (Zhou et al., 2024). Prompt-based strategies generally fall into two categories: directly prompting the model to produce confidence scores (Tian et al., 2023) and leveraging the consistency across multiple generated responses as a proxy for confidence (Kadavath et al., 2022; Xiong et al., 2023). R-Tuning appends *I am sure/unsure* to responses for binary confidence estimation, but this provides only coarse-grained judgment over the whole answer (Zhang et al., 2024). Our method extends this to the atomic fact level, enabling fine-grained uncertainty estimation. We also adopt the approach from SaySelf (Xu et al., 2024), appending an overall confidence estimate at the end of the final answer.

2.3 Hallucination Reduction in MLLMs

Several prior works have explored strategies to mitigate hallucination in MLLMs. Less-is-more (Yue et al., 2024) reduces hallucination by selectively decoding only the most confident objects, while VCD (Leng et al., 2024) suppresses unfaithful generations by contrasting model outputs with distorted distributions. OPERA (Huang et al., 2024) introduces a decoding-time penalty on over-trusted summary tokens and reallocates attention retrospectively. HA-DPO (Zhao et al., 2023b) formulates hallucination mitigation as a preference optimization task by training models to

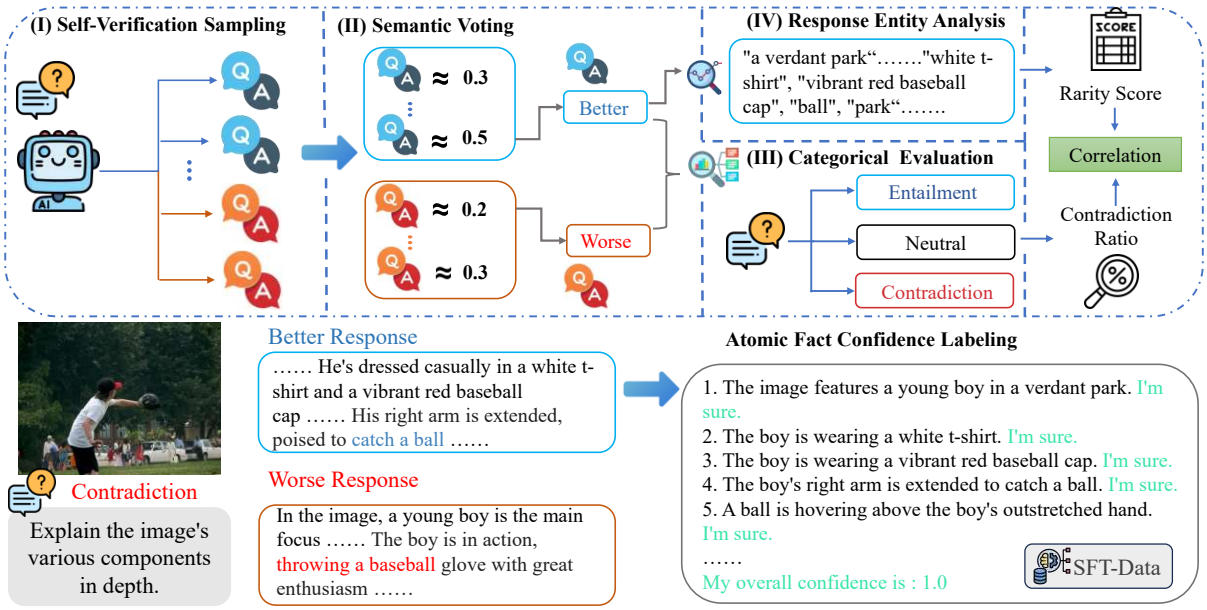

Figure 2: **Stage One**: Fine-Grained Dataset Analysis. Outputs are used to guide reward computation in Stage Two.

prefer non-hallucinatory responses, and RLAIF-V (Yu et al., 2025b) further extends this idea in a fully open-source setting using self-generated feedback. While HA-DPO and RLAIF-V rely on offline preference data and standard RL pipelines, they lack finer-grained control over sample selection and gradient quality. In contrast, DAPO (Yu et al., 2025a) decouple generation from optimization, introducing mechanisms such as dynamic sampling to improve training stability and efficiency. We draw on the idea of dynamic sampling from DAPO.

3 Methodology

3.1 Fine-Grained Dataset Analysis

As illustrated in Figure 2, Stage One establishes a fine-grained analysis for downstream optimization.

Self-Verification Sampling For each query, we first generate ten candidate answers. Then, for each candidate answers, we conduct a secondary self-verification step: the model is queried ten times to assess the correctness of the answer. Thus, each QA pair receives 10 verification votes. Based on the aggregated votes (majority voting), the QA pair is finally classified as True or False.

Semantic Voting For each question, after obtaining the True/False labels of the ten QA pairs, we calculate the semantic similarity between the original question and each answer. Specifically, among the answers labeled as True, we identify the one with the highest semantic similarity to the original question and define it as the *Better* answer. Meanwhile, for the answers marked as False, we pick the one with the lowest semantic similarity to the original question and classify it as the *Worse* answer.

Categorical Evaluation To assess the factual alignment of model responses, we prompt an expert model (GPT-4o) to compare paired *Better* and *Worse* answers against the original question. The full prompt used for this categorical judgment is provided in Appendix A.8. Each response pair is classified into one of three categories, and we report the distribution across the dataset. Three categories are defined as follows:

- **Entailment:** The *Better* and *Worse* responses express closely aligned meanings; there is substantial agreement or factual overlap.
- **Contradiction:** The *Better* and *Worse* responses are in clear conflict, presenting information that is mutually exclusive or directly opposed.
- **Neutral:** The *Better* response neither agrees nor disagrees with the *Worse* response (vague, irrelevant, or provides information that cannot be directly compared).

Response Entity Analysis Apply NER to decompose the *Better* responses and collect entity occurrences in about 80k examples. We define three metrics:

- **AvgEntityCount** The average number of entities contained in each response. A higher value indi-

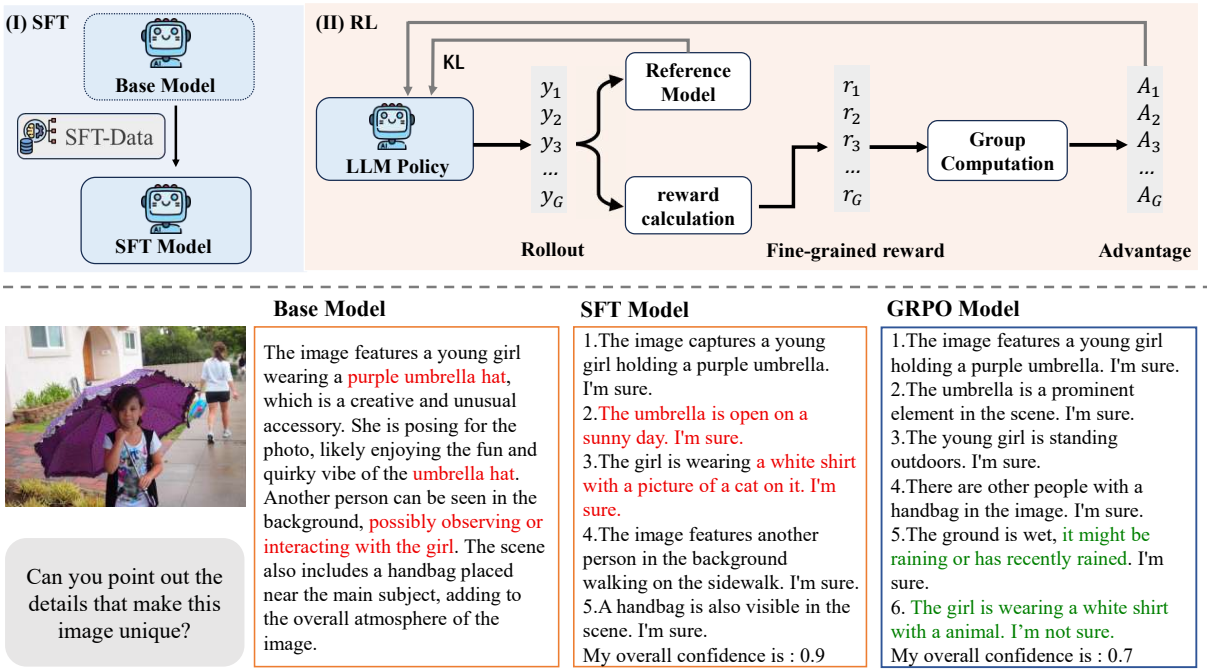

Figure 3: **Stage Two**: Difficulty-guided Reinforcement Optimization and Confidence Calibration. The model is first initialized via uncertainty-aware SFT on a small subset, then optimized using GRPO with LoRA.

cates that each response involves more entities, suggesting a denser semantics.

- **AvgEntityRarity** The average corpus frequency of entities in each response, reflecting their rarity in the dataset. Lower values indicate the presence of rarer, long-tail entities.
- **RarityScore** Defined as the ratio between entity richness and entity frequency. A higher score indicates responses contain more rare entities, suggesting greater task difficulty and a higher likelihood of hallucination.

$$\textbf{AvgEntityCount} = \frac{1}{N}\sum_{i=1}^{N}|\text{ner}_i| \tag{1}$$

$$\textbf{AvgEntityRarity} = \frac{\sum_{i=1}^{N}\sum_{e\in\text{ner}_i}\text{count}(e)}{\sum_{i=1}^{N}|\text{ner}_i|} \tag{2}$$

$$\textbf{RarityScore} = \frac{\text{AvgEntityCount}}{\text{AvgEntityRarity}\cdot\log(N+10)} \tag{3}$$

where N is the total number of data in the dataset, $|\text{ner}_i|$ is the number of entities extracted from the i-th response, and $\text{count}(e)$ denotes the number of times entity e appears in the corpus.

Correlation Analysis All dataset statistics are summarized in Table 1. We observe a clear negative correlation between **AvgEntityRarity** and the proportion of **contradiction** cases across datasets, with a Pearson correlation coefficient of $r = -0.65$ and a statistical significance of $p = 0.0118$. This finding indicates that datasets with rarer entities are more likely to yield contradiction cases.

3.2 Difficulty-guided Confidence Calibration

To enhance epistemic self-awareness and mitigate hallucinations, this stage integrates uncertainty-aware supervised fine-tuning with GRPO-based reinforcement learning under a difficulty-guided reward framework, depicted in Fig 3.

SFT We sample a subset from the training split to finetune the model for uncertainty expression and confidence calibration (Appendix A.3 for details).

RL Our GRPO loss is presented in the following formula.

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{(q,a),\,\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|} \min\Big(r_{i,t}(\theta)\,\hat{A}_{i,t},\ \text{clip}(r_{i,t}(\theta),1-\varepsilon_{\text{low}},1+\varepsilon_{\text{high}})\,\hat{A}_{i,t}\Big) - \beta\,D_{\text{KL}}(\pi_\theta\,\|\,\pi_{\text{ref}})\Bigg] \tag{4}$$

Here, $r_{i,t}$ is the token-level probability ratio between new and old policies, and $\hat{A}_{i,t}$ is the normalized group-wise advantage. We use a larger clipping range for stability and a KL regularization to better constrain policy drift. We apply GRPO with Lora to calibrate the model's confidence estimates and improve factual accuracy, which depicted in

Datasets	Type	Num Samples	Avg Entities	Avg Rare Entities	Entailment %	Contradiction %	Neutral %	Rarity Score
OCR-VQA	QA	3025	2.30	33.58	63.37	25.19	11.44	85.42
TextVQA	QA	4740	2.58	167.17	52.81	30.40	16.79	18.23
OK-VQA	QA	14802	3.51	273.18	70.23	15.76	14.01	13.37
GQA	QA	5411	3.21	496.27	66.64	20.99	12.36	7.52
VQAv2	QA	12942	2.94	476.80	72.01	16.88	11.11	6.51
Sg4V-TVQA	Detail	1966	4.03	90.92	34.69	36.27	29.04	58.53
Sg4V-WLand	Detail	1918	5.06	185.45	57.87	18.40	23.72	36.07
LCS-558K	Detail	15956	4.65	234.78	43.67	29.38	26.95	20.46
G-Landmark	Detail	1079	4.87	273.25	69.79	13.44	16.77	25.48
ART500K	Detail	1096	4.95	317.92	58.39	17.52	24.09	22.21
COCO	Detail	15199	5.18	374.59	47.98	21.05	30.98	14.36
Sg4V-Wiki	Detail	1972	5.23	380.97	43.81	26.98	29.21	18.08
Sg4V-WCeleb	Detail	1895	4.95	543.13	35.62	32.36	32.02	12.06
MovieNet	Detail	1131	3.76	615.36	54.11	13.23	32.67	8.67

Table 1: Statistical characteristics of 14 datasets. Sg4V = ShareGPT4V; TVQA = TextVQA; WLand = Web-Landmark; WCeleb = Web-Celebrity; Wiki = WikiArt; G-Landmark = Google-Landmark.

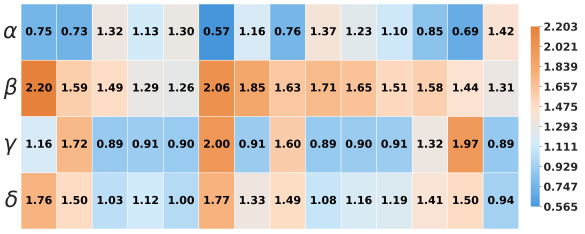

Figure 4: Heatmap visualization of the four dynamically computed coefficients (α, β, γ, δ) across 14 datasets. Dataset names on the x-axis are ordered consistently with Table 1.

Figure 3. The procedure for computing the reward is summarized in Algorithm 1.

Dynamic Parameter Mapping To enable adaptive optimization across datasets of varying difficulty, we derive four dynamic coefficients $(\alpha, \beta, \gamma, \delta)$ from the fine-grained statistics computed in Stage One. For each dataset, we collect statistics such as the average entity count(Eq 1), entity rarity(Eq 2), dataset size, and the proportion of contradiction. These indicators are normalized across datasets and jointly used to characterize both the rarity of entities and the likelihood of contradiction. The normalized measures are then transformed through bounded nonlinear functions to obtain the four coefficients: α for factual accuracy, β for confidence alignment, γ for local–global consistency, and δ for over-/under-confidence penalty. The full mapping procedure and transformation details are provided in Appendix A.1.

Dynamic Coefficient Interpretation Four dynamic coefficients are introduced in Figure 4.

- **Factuality reward** (α): Encourages accurate and confident outputs. Set higher for simple datasets (`VQAv2`) and reduced for conflict-prone ones (`sharegpt4v-textvqa`).
- **Claim Confidence alignment reward** (β): Encourages alignment between correctness and claim-level confidence. Increased for datasets with high entity rarity and contradiction (`sharegpt4v-textvqa`) to strengthen calibration.
- **Consistency reward** (γ): Measures coherence between averaged claim-level and overall confidence. Higher for datasets with internal inconsistencies (`TextVQA`); relaxed for cleaner ones.
- **Misalignment penalty** (δ): Penalizes deviation between overall confidence and expert-labeled correctness. Elevated for high-risk tasks (`OCR-VQA`) to enforce confidence estimation.

Reward Term Interpretation The final reward $\mathcal{R}$ consists of four components:

- **Factual correctness** ($\sum_i \alpha \cdot \hat{c}_i$): encourages the model to generate factually correct claims, where $\hat{c}_i \in \{0, 1\}$ indicates expert-verified correctness.
- **Confidence alignment** ($\sum_i \beta \cdot m_i$): measures the match between expert-labeled correctness $\hat{c}_i$ and the model's expressed confidence s_i, derived from phrases *I'm sure*(1.0) or *I'm not sure* (0.0).
- **Local–global consistency** ($\gamma \cdot (1 - |o - \bar{s}|)$): ensures tie between the overall declared confidence o and the mean confidence levels of claims $\bar{s}$.
- **Confidence–accuracy penalty** ($-\delta \cdot |o - o^{\text{expert}}|$): penalizes discrepancies between the model's overall confidence and expert-judged correctness o^{expert}, discouraging over or under-confidence.

Algorithm 1 Difficulty-guided Reward Compute

Require: Model outputs
Ensure: Reward scores for each sample
for (x, y, d) in $(\texttt{prompts}, \texttt{answers}, \texttt{datasets})$ **do**
  $(\alpha, \beta, \gamma, \delta) \leftarrow \texttt{Dynamic\_parameters}(d)$
  Extract claims $\{(c_i, s_i)\}$ from answer y
  for each claim (c_i, s_i) **do**
    Query expert model to obtain label $\hat{c}_i \in \{0, 1\}$
    Confidence alignment $m_i = 1 - |\hat{c}_i - s_i|$
  end for
  Compute mean confidence $\bar{s} = \frac{1}{N} \sum_i s_i$
  Compute local consistency $cons = 1 - |o - \bar{s}|$
  Query for global correctness o^{expert}
  Weighted correctness: $w = \frac{1}{N} \sum_i (\alpha \cdot \hat{c}_i + \beta \cdot m_i)$
  Final reward: $\mathcal{R} = w + \gamma \cdot cons - \delta \cdot |o - o^{\text{expert}}|$
  Append $\mathcal{R}$ to total rewards
end for
return reward score

4 Experiments

4.1 Datasets

We evaluate our method on 14 datasets ((Mishra et al., 2019; Singh et al., 2019; Marino et al., 2019; Hudson and Manning, 2019; Goyal et al., 2017; Chen et al., 2024; Liu et al., 2023b; Weyand et al., 2020; Mao et al., 2019; Lin et al., 2014; Huang et al., 2020)) including both QA and detail-oriented tasks. The datasets are summarized in Table 1. We partition each dataset into training and test sets, with the training set used for SFT and the test set reserved for evaluation.

4.2 Evaluation Metrics And Baseline

For calibration, following previous studies (Tian et al., 2023), we report the ECE and AUROC. For hallucination evaluation, we adopt the MMHal benchmark (Sun et al., 2023) and ObjHalBench (Rohrbach et al., 2018), measuring both response-level and object-level hallucination rates. In addition, we compare a training-based calibration baseline, R-Tuning (Zhang et al., 2024), and a training-free baseline, VCAP (Xuan et al., 2025). The reproduction details of R-Tuning and the prompts used for VCAP are provided in the Appendix A.4.

4.3 Confidence-Estimation Protocols

Following previous studies (Xu et al., 2024), we experiment with four strategies to extract a confidence score from MLLMs. These methods can be categorized into two groups: (1) **Probability-based methods**, including *Direct-Probability (DP)* and *Self-Consistency (SC)*, and (2) **Verbalized-based methods**, including *Robust Prompting for Correctness (PC-R)* and *Verbalised Confidence (VC)*.

Direct–Probability (DP) Following SaySelf (Xu et al., 2024), we compute confidence based on the model's next-token distribution. Specifically, given the input question and image, we perform a single forward pass and extract the logits of the final generated token. A softmax is applied over the vocabulary, and the highest probability $\max_j p_j$ is taken as the model's confidence for the entire answer.

Self-Consistency (SC) Sample k candidate answers, embed them and cluster by cosine similarity (≥ 0.8), and define confidence as the fraction of samples that fall into the largest cluster: $c = |\mathcal{C}_{\max}|/k$. This score approximates the epistemic certainty of the model under sampling noise.

Robust Prompting for Correctness (PC-R) We issue paraphrased correctness prompts and consider multiple affirmative and negative tokens (`Yes`, `True`, `No`, `False`). The final confidence is calculated as $c = \frac{p_{\text{Yes}} + p_{\text{True}}}{p_{\text{Yes}} + p_{\text{No}} + p_{\text{True}} + p_{\text{False}}}$, where p_{Yes} and p_{True} are the highest probabilities assigned to any affirmative token in all prompts.

Verbalised Confidence (Verbal) This method directly asks the model to output a number between 0 and 1 that reflects its self-reported confidence. The exact prompt used for extracting this verbal confidence is provided in Appendix A.8.

5 Results and Discussion

5.1 Confidence Calibration Performance

Table 2 reports the ECE and AUC of different stage models. We observe that: (1) From `LLaVA` to `LLaVA-SFT` and `LLaVA-RL`, the AUC scores consistently improve while ECE scores decrease, indicating that both SFT and RL effectively enhance the model's confidence calibration and discriminative ability. (2) Among the three inference types, `Entailment` is the easiest to detect, while `Contradiction` and `Neutral` are more challenging, especially for the base `LLaVA` model. However, with RL, the model shows significantly improved discrimination ability across all categories, especially on harder cases, indicating enhanced confidence awareness and self-reflective capability. (3) Calibration methods that utilize Verbalized-based confidence expression (`PC-R` and `Verbal`) consistently outperform those relying solely on probability outputs (`DP` and `SC`), indicating that verbalized uncertainty better aligns with the model's internal

Model	Method	Text-VQA(QA)						sharegpt4v-textvqa(Detail)					
		Entailment		Contradiction		Neutral		Entailment		Contradiction		Neutral	
		ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑
LLaVA	DP	0.418	0.396	0.436	0.390	0.406	0.290	0.311	0.428	0.278	0.508	0.355	0.638
	SC	0.365	0.687	0.353	0.632	0.316	0.624	0.291	0.512	0.292	0.452	0.283	0.636
	PC-R	0.399	0.336	0.423	0.441	0.342	0.401	0.202	0.550	0.287	0.539	0.240	0.586
	Verbal	0.346	0.573	0.435	0.525	0.361	0.495	0.301	0.383	0.385	0.519	0.306	0.398
	VCAP	0.431	0.548	0.498	0.507	0.448	0.581	0.515	0.256	0.551	0.487	0.487	0.589
LLaVA SFT	DP	0.281	0.517	0.342	0.486	0.241	0.524	0.153	0.486	0.177	0.437	0.180	0.338
	SC	0.342	0.583	0.343	0.572	0.314	0.549	0.241	0.525	0.318	0.502	0.287	0.611
	PC-R	0.307	0.527	0.328	0.529	0.271	0.478	0.161	0.578	0.181	0.447	0.163	0.696
	Verbal	0.299	0.497	0.348	0.515	0.310	0.435	0.157	0.519	0.176	0.562	0.197	0.482
	VCAP	0.351	0.541	0.413	0.492	0.359	0.462	0.362	0.500	0.392	0.424	0.362	0.502
LLaVA R-Tuning	DP	0.445	0.493	0.473	0.490	0.471	0.472	0.385	0.459	0.378	0.489	0.426	0.482
	SC	0.324	0.651	0.379	0.591	0.334	0.651	0.207	0.630	0.277	0.525	0.210	0.602
	PC-R	0.447	0.600	0.477	0.497	0.473	0.356	0.451	0.523	0.562	0.403	0.492	0.512
	Verbal	0.464	0.578	0.523	0.551	0.520	0.554	0.485	0.571	0.551	0.476	0.542	0.539
	VCAP	0.315	0.531	0.394	0.532	0.325	0.531	0.322	0.461	0.393	0.530	0.353	0.513
LLaVA RL	DP	0.241	0.560	0.286	0.525	0.208	0.554	0.125	0.686	0.143	0.626	0.131	0.652
	SC	0.282	0.627	0.305	0.570	0.246	0.603	0.157	0.681	0.210	0.688	0.207	0.645
	PC-R	0.281	0.543	0.288	0.616	0.243	0.597	0.162	0.760	0.179	0.777	0.184	0.743
	Verbal	0.237	0.650	0.296	0.611	0.219	0.596	0.171	0.711	0.132	0.779	0.147	0.791
	VCAP	0.261	0.655	0.325	0.553	0.279	0.542	0.285	0.635	0.326	0.716	0.308	0.680

Table 2: Comparison of ECE and AUC performance across different stage models and confidence estimation methods on QA and Detail tasks. LLaVA R-Tuning refer to (Zhang et al., 2024), VCAP refer to (Xuan et al., 2025)

Method	Size	MMHal		ObjHal	
		Score↑	Hall.↓	Rsp.↓	Obj.↓
VCD	7B	2.12	0.54	48.80	24.30
Less-is-more	7B	2.33	0.50	40.30	17.80
OPERA	7B	2.15	0.54	45.10	22.30
HA-DPO	7B	1.98	0.60	39.90	19.90
POVID	7B	2.08	0.56	48.10	24.40
RLAIF-V	7B	2.95	**0.32**	10.50	5.20
LLaVA	7B	1.58	0.72	57.14	28.67
SFT-LLava	7B	1.99	0.65	24.11	13.08
Base-LLaVA	7B	2.23	0.54	36.50	19.81
Base-SFT	7B	**3.01**	0.37	**7.33**	**4.07**
GPT-4V	-	3.49	0.29	13.60	7.30

Table 3: Hallucination performance across MMHal and ObjHal. We report both response-level (Rsp.) and object-level (Obj.). Hall.: Hallucination Rate. The best results are shown in **bold**, while underlined indicates the second-best.

representations and leads to more stable and reliable performance across different subset types. This is consistent with Tian et al. (2023). The other datasets results in A.5.

5.2 Hallucination Reduction

We train a GRPO adapter on the SFT-tuned backbone (`SFT-LLaVA`) and evaluate it under two inference configurations: (i) reattaching the adapter to the original `LLaVA` backbone (`Base-LLaVA`), (ii) retaining it on the aligned `SFT-LLaVA` backbone (`Base-SFT`). As shown in Table 3, both GRPO variants achieve substantial reductions in hallucination rates across MMHal-Bench and Object HalBench. Notably, while SFT alone significantly improves response factuality, applying GRPO on top of it yields further gains, particularly under configuration (ii), where backbone–adapter alignment is preserved. It should be noted that our method achieves performance comparable or even superior to RLAIF-V (Yu et al., 2025b) using over 1,200 examples for GRPO, whereas RLAIF-V relies on over 10,000 examples for DPO training, demonstrating the sample efficiency of our approach.

5.3 Dynamic Reward Ablation

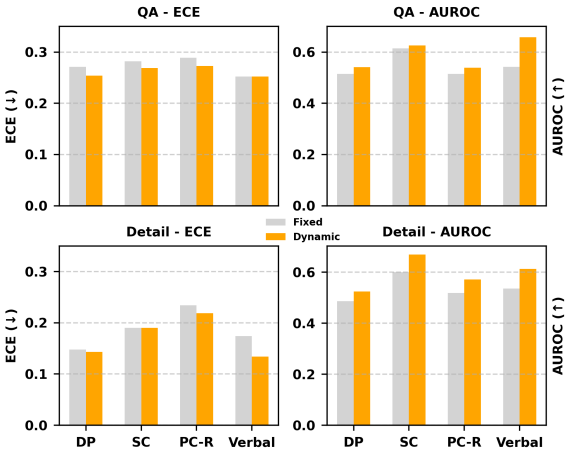

Figure 5: Average performance on two types of datasets under fixed and dynamic reward settings.

As shown in Figure 5, adopting dynamic reward coefficient strategies consistently yields better performance compared to fixed settings

(α=β=γ=δ=1), when evaluated on the aggregated QA and Detail datasets. In both evaluation types, dynamic settings lead to noticeable AUC improvements, with the most pronounced gain observed for the verbal method on QA tasks (+0.116). Meanwhile, ECE values either decrease or remain comparable, suggesting that dynamic reward tuning not only enhances discriminative ability but also maintains or improves model calibration.

5.4 Calibration Behaviour in RL Training

The calibration behavior evolves in a non-monotonic manner throughout RL training, shaped by the dynamic interplay among reward clarity, parameter optimization, and backbone compatibility.

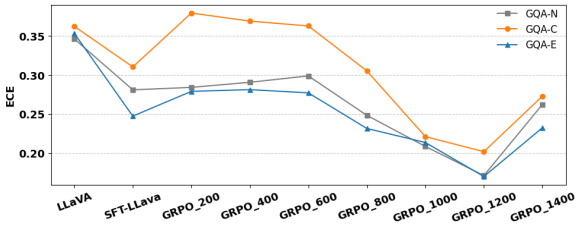

Figure 6: ECE trend across training epochs for LLaVA models on the GQA dataset, evaluated separately on Neutral (GQA-N), Contradiction (GQA-C), and Entailment (GQA-E) categories.

Early Phase. Initially (Epoch 200–600), the model tends to avoid overconfident predictions and concentrates confidence scores around the mid-range (0.4 – 0.6). While this conservative policy mitigates the risk of severe misjudgment, it fails to align confidence with actual correctness, resulting in a temporary elevation of the ECE, which depicted in Figure 6 and Figure 7 right-column.

Intermediate to Late Phase. As training progresses (Epoch 800–1200), the model increasingly discriminates between correct and incorrect predictions, improving confidence alignment. Predictions with higher confidence levels become more accurate, achieving better calibration and substantially reducing ECE, which depicted in Figure 6.

Late Phase with Backbone Mismatch. We observe a consistent under-confidence pattern in the lowest confidence bin (0 – 0.1) across all five QA datasets, where the model significantly underestimates its correctness. This reflects a cautious prediction behavior. Combined with the sparsity of high-confidence predictions, this under-confidence accumulation indicates a broader *low-confidence-crash*, which depicted in Figure 7 left-column.

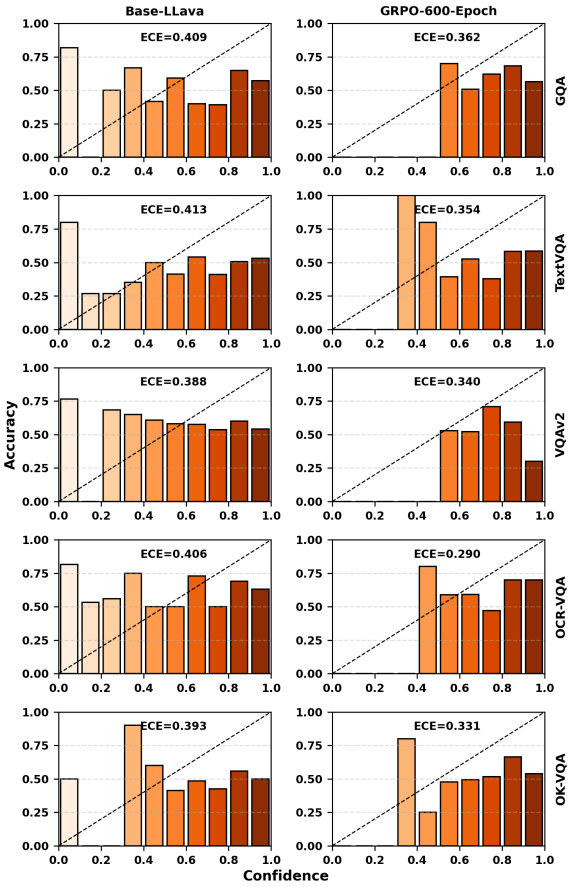

Figure 7: Calibration histograms for five QA datasets. Left column: `Base-LLaVA` with GRPO adapter applied to an untuned backbone, resulting in a broader *low-confidence-crash* due to backbone mismatch. Right column: `GRPO-600` Using the matched SFT backbone but still exhibits a moderate *low-confidence-crash* in early RL phase.

We attribute this phenomenon to representational mismatches caused by applying GRPO-trained adapters to an untuned backbone (Base-LLaVA), leading to flattened confidence distributions and degraded calibration performance.

Take-away. SFT training gives initial confidence estimates. GRPO sharpens predictive discrimination but is sensitive to training - inference backbone consistency.

6 Conclusion

In this paper, we propose a two-stage framework to improve confidence estimation and reduce hallucinations in MLLM. Stage one analyzes dataset difficulty, while stage two combines SFT with GRPO using a difficulty-guided reward. Experiments show notable gains in calibration and hallucination reduction.

7 Limitation

Although our framework delivers consistent improvements in both calibration and hallucination reduction, it still presents several limitations. First, the rarity-based difficulty metric is defined at the dataset level and may overlook fine-grained intra-dataset heterogeneity, such as sample-specific compositional complexity or cross-image contextual dependencies. A more localized or adaptive notion of difficulty could yield a more precise reward signal. Second, our experiments focus primarily on single-turn QA and description tasks. Extending confidence-aware optimization to multi-turn dialog, tool-augmented reasoning, or safety-critical domains remains an important avenue for future exploration.

References

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, and 8 others. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2024. Sharegpt4v: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*, pages 370–387. Springer.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. 2024. 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.

Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. 2020. Movienet: A holistic dataset for movie understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 709–727. Springer.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

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*, pages 13872–13882.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering*, 34(1):50–70.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning.

Hui Mao, James She, and Ming Cheung. 2019. Visual arts search on mobile devices. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 15(2s):60.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*.

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2022. Reasoning with language model prompting: A survey. *arXiv preprint arXiv:2212.09597*.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. *arXiv preprint arXiv:1809.02156*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8317–8326.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, and 1 others. 2023. Aligning large multimodal models with factually augmented rlhf. *arXiv preprint arXiv:2309.14525*.

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. *arXiv preprint arXiv:2305.14975*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. 2020. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2575–2584.

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. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5985–5998.

Weihao Xuan, Qingcheng Zeng, Heli Qi, Junjue Wang, and Naoto Yokoya. 2025. Seeing is believing, but how much? a comprehensive analysis of verbalized calibration in vision-language models. *arXiv preprint arXiv:2505.20236*.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various ner subtasks. *arXiv preprint arXiv:2106.01223*.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? *arXiv preprint arXiv:2305.18153*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, and 1 others. 2025a. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, and 1 others. 2024. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13807–13816.

Tianyu Yu, Haoye Zhang, Qiming Li, Qixin Xu, Yuan Yao, Da Chen, Xiaoman Lu, Ganqu Cui, Yunkai Dang, Taiwen He, and 1 others. 2025b. Rlaif-v: Open-source ai feedback leads to super gpt-4v trustworthiness. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19985–19995.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. 2025. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*.

Zihao Yue, Liang Zhang, and Qin Jin. 2024. Less is more: Mitigating multimodal hallucination from an eos decision perspective. *arXiv preprint arXiv:2402.14545*.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and

Tong Zhang. 2024. 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.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023a. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Zhiyuan Zhao, Bin Wang, Linke Ouyang, Xiaoyi Dong, Jiaqi Wang, and Conghui He. 2023b. Beyond hallucinations: Enhancing lvlms through hallucination-aware direct preference optimization. *arXiv preprint arXiv:2311.16839*.

Kaitlyn Zhou, Jena D Hwang, Xiang Ren, and Maarten Sap. 2024. Relying on the unreliable: The impact of language models' reluctance to express uncertainty. *arXiv preprint arXiv:2401.06730*.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Multilingual machine translation with large language models: Empirical results and analysis. *arXiv preprint arXiv:2304.04675*.

A Appendix

A.1 Dynamic Parameter Mapping Detail

Dataset Statistics Each dataset is characterized by four corpus-level statistics that govern the construction of reward coefficients. Specifically, c denotes the average number of named entities per response, where larger values imply denser semantic coverage (Equation 1); r represents the average corpus-level frequency of entities, with smaller values indicating rarer or more long-tail concepts (Equation 2); n is the total number of samples in the dataset; and $p_{\text{contradiction}}$ denotes the proportion of contradiction-labeled samples among the three label types. From these quantities, we derive several secondary metrics used to capture dataset-level semantic sparsity and rarity characteristics. The `rare_density` metric is defined as r/c, characterizing the relative density of rare entities. The `entity_focus` metric is given by $1/c$, reflecting how concentrated a dataset is in terms of per-sample entity usage. To account for differences in dataset scale, we define `rarity_score` as $(r/c)/\log(n+10)$, which normalizes rare-entity density by corpus size. Finally, the entity-rarity emphasis coefficient is defined as $w_e = \texttt{norm}(0.7 \cdot (r/c) + 0.3 \cdot (1/c))$. A higher w_e reflects datasets that place greater emphasis on rare and semantically focused entities, thereby amplifying reward sensitivity in downstream coefficient design.

Normalization To standardize these statistics across datasets, we apply z-score normalization:

$$r_n = \frac{\texttt{rarity\_score} - \mu_r}{\sigma_r}$$
$$h_n = \frac{p_{\text{contradiction}} - \mu_h}{\sigma_h}$$

where μ_r, σ_r and μ_h, σ_h denote the mean and standard deviation of rarity and contradiction scores across all datasets.

Coefficient Formulations Each coefficient is computed as follows:
(1) **Factuality reward** α: Encourages factual alignment; assigns more conservative weights in high-contradiction or high-rarity regions.

$$\alpha = \texttt{clip}\big(\tanh(-h_n - r_n), 0.0,\ 2.5\big)$$

This coefficient maps the combination of contradiction prevalence h_n and rarity score r_n into a bounded range. The negative weighting ensures that higher contradiction and rarity reduce α, resulting in more conservative reward scaling.
(2) **Claim Confidence alignment reward** β: This coefficient enhances the reward when entity rarity is high and samples exhibit strong semantic focus.

$$\beta = \texttt{clip}(\sigma(w_e \cdot r_n),\ 0.5,\ 2.5)$$

Here, w_e reflects the degree to which a dataset emphasizes rare and focused entities. The intent is to amplify rewards for confident claims grounded in uncommon, semantically cohesive knowledge—typical of long-tail data.
(3) **Consistency reward** γ: Rewards global consistency between the overall output and its answer.

$$\gamma = \texttt{clip}((h_n - 0.1\alpha), 0.0, 2.0)$$

It combines contradiction risk h_n and factual correctness α in a way that rewards consistency only when both semantic and factual conditions are met.
(4) **Misalignment penalty** δ: Introduced to penalize disagreement between model confidence and expert-level confidence:

$$\delta = \texttt{clip}(\sigma(r_n + h_n),\ 0.0,\ 2.0)$$

This design penalizes overconfident predictions in high-risk, hallucination-prone samples more strictly.

Dataset	α	β	γ	δ
OCR-VQA	0.7524	2.2029	1.1555	1.7553
TextVQA	0.7315	1.5870	1.7219	1.4984
OK-VQA	1.3228	1.4927	0.8939	1.0330
GQA	1.1274	1.2917	0.9087	1.1213
VQAv2	1.2990	1.2616	0.8957	1.0036
sharegpt4v-textvqa	0.5651	2.0607	2.0000	1.7684
sharegpt4v-web-landmark	1.1566	1.8504	0.9065	1.3310
LCS-558K	0.7576	1.6301	1.5983	1.4916
Google-Landmark	1.3744	1.7084	0.8901	1.0786
ART500K	1.2328	1.6545	0.9007	1.1635
COCO	1.1050	1.5150	0.9104	1.1901
sharegpt4v-wikiart	0.8454	1.5799	1.3243	1.4055
sharegpt4v-web-celebrity	0.6897	1.4363	1.9706	1.4958
MovieNet	1.4154	1.3053	0.8870	0.9386

Table 4: Dataset-specific reward coefficients α, β, γ, δ derived from dataset statistics.

Parameter Values Detail Table 4 lists the coefficients computed for each dataset. These dynamic coefficients are adaptively tuned to capture the unique characteristics and challenges of each dataset:

- A higher α promotes factual and confident responses. It is elevated for well-structured datasets like `VQAv2` and `OK-VQA`, while reduced for noisy or contradiction-heavy datasets (`sharegpt4v-textvqa`) to encourage cautious behavior and prevent overconfidence.
- The β coefficient enhances reward when the model's confidence aligns with correctness. Datasets with high entity rarity or strong semantic focus—such as `OCR-VQA` and `sharegpt4v-textvqa`—receive higher β values to improve calibration and reduce hallucination.
- The γ term promotes internal consistency between claim-level and overall confidence. It is increased for datasets prone to internal inconsistency or compositional reasoning (e.g., `TextVQA`), and relaxed for structurally simpler ones (e.g., `COCO`, `GQA`).
- The penalty term δ discourages misalignment between model confidence and human-verified correctness. It is strengthened in ambiguous or high-risk scenarios (e.g., `OCR-VQA`, `sharegpt4v-web-celebrity`) to enforce conservative confidence estimation, while relaxed for cleaner datasets (e.g., `MovieNet`).

A.2 Dataset Overview

As shown in Table 13, we summarize 14 benchmark datasets by task type (Detail/QA), sample size, and the NLI-based label distribution (Entailment/Contradiction/Neutral), and report the statistics for the Total/Train/Val splits, respectively.

A.3 SFT Data Construction Workflow

To prepare high-quality SFT data for calibration-aware instruction tuning, we construct training samples using a multi-step pipeline that integrates model generation, atomic fact decomposition, question rewriting, and fact-level verification. The workflow proceeds as follows. We begin by prompting the target MLLM to produce an open-ended response for each selected input. An atomic decomposition model is then applied to extract fine-grained factual statements from the generated answer. Each atomic fact is subsequently converted into an independent natural-language question using a question rewriting model. Next, every rewritten question is fed back into the original MLLM to obtain a factual judgment. The question–answer pair is then evaluated by a separate verifier model, which outputs a binary decision (`true`/`false`) indicating whether the atomic fact is supported by the image. According to this decision, we augment each atomic fact with a confidence-aware tag: *"I'm sure."* for verified (true) facts and *"I'm not sure."* for unverified (false) ones. All annotated atomic facts are aggregated into a numbered list, accompanied by a summary confidence score. To ensure scalability, we employ a multithreaded generation framework with shared atomic counters and thread-safe writing to avoid sample duplication. The pipeline is implemented on top of the `Ollama` and `transformers` ecosystems, with 8 parallel workers producing up to 500 high-quality SFT samples. Both the atomic fact extractor and the question rewriting module are implemented using finetuned Llama-3 models (Dubey et al., 2024), chosen for their strong controllability and factual consistency in structured generation tasks. For fact verification, we employ Qwen-VL 2.5 (Bai et al., 2025), which provides reliable image-grounded judgment on whether each rewritten question is supported by the visual content.

A.4 Reproduction Details

R-Tuning To reproduce R-Tuning in our multimodal setting, we follow the workflow proposed in (Zhang et al., 2024). Specifically, we first sampled 1,000 examples from our dataset—the same subset used for our SFT cold-start experiments—and used the base LLaVA (Liu et al., 2023a) model to obtain its predictions for each question. According to whether the model's output matched the ground-truth answer, we divided the data into two subsets

that align with the R-Tuning framework: (1) a certain subset (D_1), where the base model produces correct and consistent answers, and (2) an uncertain subset (D_0), where the model either fails to answer or gives inconsistent predictions. Following R-Tuning's data transformation strategy, we converted each training target into a certainty-aware form by appending explicit confidence expressions: answers in (D_1) were labeled with *I am sure* , while answers in (D_0) were labeled with *I am not sure*. Using this transformed dataset, we performed supervised fine-tuning on LLaVA under the same training format described in R-Tuning, producing an R-Tuning LLaVA model.

VCAP VCAP (Xuan et al., 2025) is a training-free calibration framework that improves visual–language models' confidence estimation through a two-stage prompting design. Instead of directly asking the model to judge answer correctness, VCAP first separates the evaluation into: (1) a **visual grounding stage**, where the model is required to describe the visual content and self-assess the confidence of its visual understanding; and (2) an **answer verification stage**, where the model uses its previously generated visual understanding to evaluate the correctness of a proposed answer (for QA) or the consistency of a proposed description (for detailed description tasks). In our experiments, we adapt VCAP to both of our task types—*question answering* and *detailed image description*. The prompts used in each stage closely follow the structure and design principles of the original VCAP paper, while being tailored to the specific requirements of our datasets. The complete prompts for the QA version and the detailed-description version are provided in Sec A.8.

A.5 More ECE And AUC Results

Beyond the main results presented in maintext, we provide additional calibration and hallucination analyses in this appendix.

A.5.1 OK-VQA and VQAv2

The following table (Table 5) shows the full experimental results on OK-VQA and VQAv2.

A.5.2 GQA and OCR-VQA

The following table (Table 6) shows the full experimental results on GQA and OCR-VQA.

A.5.3 COCO and Google-Landmark

The following table (Table 7) shows the full experimental results on COCO and Google-Landmark.

A.5.4 sharegpt4v-wikiart and sharegpt4v-web-celebrity

The following table (Table 8) shows the full experimental results on sharegpt4v-wikiart and sharegpt4v-web-celebrity.

A.5.5 ART500K and LCS-558K

The following table (Table 9) shows the full experimental results on ART500K and LCS-558K.

A.5.6 MovieNet and sharegpt4v-web-landmark

The following table (Table 10) shows the full experimental results on MovieNet and sharegpt4v-web-landmark.

A.6 Analysis On Accuracy

Across all QA datasets (VQAv2, GQA, OK-VQA, TextVQA, and OCR-VQA) (Table 11, Table 12), the accuracy of the three model variants—LLaVA, SFT, and RL—remains largely comparable, with only small fluctuations across the entailment, contradiction, and neutral subsets. Our method is designed to calibrate confidence rather than improve task accuracy.

A.7 Training Dynamics

Figure 8 visualizes the evolution of calibration error during RL training, further illustrating the effectiveness of our difficulty-guided optimization scheme.

A.8 Prompt Detail

The **NER Extraction Prompt** identifies named entities to compute rarity-based difficulty. The **Verbal Confidence Extraction Prompt** elicits explicit confidence scores for calibration. The **Contradiction Detection Prompt** labels better–worse pairs as entailment, neutral, or contradiction for difficulty analysis. The **Question Transform Prompt** rewrites atomic facts into questions to support fact-level verification and SFT data construction.

Model	Method	OK-VQA						VQAv2					
		Entailment		Contradiction		Neutral		Entailment		Contradiction		Neutral	
		ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑
LLaVA	DP	0.3243	0.5477	0.3553	0.5297	0.3606	0.4270	0.3465	0.6956	0.3925	0.5390	0.3404	0.6532
	SC	0.3570	0.6128	0.3770	0.5796	0.3495	0.6484	0.3410	0.7084	0.3915	0.6087	0.3850	0.6043
	PC-R	0.3024	0.5276	0.3328	0.6139	0.3211	0.4846	0.3988	0.4214	0.3945	0.4640	0.3664	0.4972
	Verbal	0.2639	0.5783	0.3610	0.5339	0.3132	0.5313	0.3199	0.5436	0.3635	0.5034	0.3361	0.5097
	VCAP	0.3071	0.5648	0.3958	0.4926	0.3562	0.5523	0.3331	0.5568	0.3602	0.6156	0.3770	0.5595
LLaVA SFT	DP	0.2236	0.5687	0.2853	0.4912	0.2589	0.5241	0.2706	0.5478	0.2766	0.5584	0.2605	0.5030
	SC	0.2975	0.5307	0.3010	0.5464	0.3125	0.5147	0.2830	0.5976	0.3395	0.4622	0.2635	0.6440
	PC-R	0.2560	0.5775	0.2929	0.5004	0.2450	0.6211	0.3003	0.5084	0.2723	0.5374	0.2659	0.5411
	Verbal	0.2383	0.5873	0.3358	0.4872	0.2564	0.5894	0.2786	0.5716	0.2673	0.6226	0.2626	0.5808
	VCAP	0.2442	0.6371	0.3386	0.5234	0.2998	0.5593	0.2838	0.5963	0.2920	0.5804	0.3258	0.5217
LLaVA R-Tuning	DP	0.3049	0.5544	0.3681	0.5252	0.3563	0.4724	0.3706	0.5614	0.3716	0.5293	0.3669	0.5140
	SC	0.3448	0.6559	0.3105	0.6363	0.3560	0.6071	0.3335	0.6402	0.2077	0.6254	0.3175	0.6353
	PC-R	0.2981	0.5831	0.3685	0.5065	0.3460	0.5161	0.4016	0.4498	0.3875	0.4099	0.3621	0.4649
	Verbal	0.3017	0.5861	0.4490	0.4758	0.3985	0.4949	0.3227	0.5478	0.3733	0.5182	0.3402	0.5529
	VCAP	0.3342	0.5384	0.4159	0.5455	0.3682	0.5225	0.2982	0.5385	0.4603	0.5396	0.3527	0.5631
LLaVA RL	DP	0.2272	0.6652	0.3040	0.6433	0.2411	0.6060	0.2696	0.6774	0.3160	0.6344	0.2376	0.6394
	SC	0.2408	0.6300	0.2853	0.6457	0.2513	0.6519	0.2869	0.6967	0.2865	0.6983	0.2520	0.6508
	PC-R	0.2568	0.6793	0.2785	0.6644	0.2491	0.6986	0.2906	0.5935	0.2889	0.6281	0.2273	0.6577
	Verbal	0.1970	0.6629	0.2756	0.6324	0.2220	0.6242	0.2482	0.6329	0.2984	0.6411	0.2208	0.6061
	VCAP	0.2559	0.5889	0.2903	0.7194	0.2999	0.6023	0.2640	0.6323	0.3512	0.6267	0.3110	0.6139

Table 5: Comparison of ECE and AUC performance across different stage models and confidence estimation methods on OK-VQA and VQAv2.

NER_Extraction_Prompt

<USER>
Extract all named entities from the following text.
Guidelines:
– Do not classify the entities.
– Include only named entities such as people, places, organizations, brands, or specific objects.
– Do not include general nouns or descriptions.
– Return **only** a JSON array of strings, with no explanation or additional formatting.

[TEXT]
`"{answer_text}"`

Your output must be **ONLY**:
`[ "entity1", "entity2", ... ]`

Verbal Confidence Extraction Prompt

<USER>
You are an expert evaluator of question–answer pairs. Your task is to read the following **QUESTION** and **ANSWER**, then judge how confident you are that the answer is correct. You must output **only** a single decimal number between 0 and 1 (inclusive), with at most two digits after the decimal point. Do **NOT** output any additional text.

Example:
Q: What is the capital of France?
A: Paris.
Confidence: 0.95

QUESTION
`{q}`
ANSWER
`{r}`

Please respond with your confidence score now. Evaluate the correctness of the answer above and return **ONLY the number**.

Contradiction_Detection_Prompt

<USER>
You are a helpful assistant trained to detect factual contradictions between two answers.
Given:
– A question.
– Two answers to that question: Answer A and Answer B.
Your task is to analyze whether Answer B contradicts Answer A based on factual content.
Please classify their relationship into one of the following:
– **entailment**: Answer B agrees with or repeats the same facts as Answer A.
– **contradiction**: Answer B presents facts that conflict with Answer A.
– **neutral**: Answer B neither agrees nor disagrees with Answer A (e.g., vague or irrelevant).
Respond with **one word only**: `entailment`, `contradiction`, or `neutral`.

[QUESTION]
`{question}`
[Answer A]
`{chosen}`
[Answer B]
`{rejected}`

Return **ONLY one line:**
`<judge> entailment/contradiction/neutral </judge>`

Model	Method	GQA						OCR-VQA					
		Entailment		Contradiction		Neutral		Entailment		Contradiction		Neutral	
		ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑
LLaVA	DP	0.3805	0.5439	0.4036	0.5502	0.3602	0.5887	0.3284	0.8324	0.3811	0.6772	0.3676	0.6847
	SC	0.3635	0.6728	0.3930	0.6089	0.3950	0.6129	0.2230	0.8277	0.3255	0.7295	0.2660	0.8091
	PC-R	0.3842	0.5174	0.4021	0.4638	0.3743	0.5223	0.5241	0.2616	0.4429	0.4192	0.4630	0.3448
	Verbal	0.3530	0.4348	0.3619	0.5235	0.3459	0.5147	0.3215	0.5219	0.3899	0.5390	0.3481	0.4954
	VCAP	0.2896	0.6484	0.3636	0.5335	0.3503	0.5441	0.3712	0.4732	0.4445	0.4983	0.3422	0.5724
LLaVA SFT	DP	0.2580	0.4745	0.2831	0.5404	0.2883	0.5171	0.2542	0.4394	0.3407	0.4403	0.2886	0.4855
	SC	0.2450	0.5866	0.2810	0.6274	0.2775	0.6110	0.3205	0.6206	0.2975	0.7263	0.3275	0.5688
	PC-R	0.2609	0.5445	0.2769	0.5656	0.2826	0.5478	0.2929	0.5388	0.3420	0.4912	0.3094	0.5591
	Verbal	0.2473	0.5727	0.3101	0.5278	0.2808	0.5675	0.2491	0.5710	0.3400	0.4750	0.3057	0.5113
	VCAP	0.2783	0.5870	0.2767	0.6300	0.2959	0.5774	0.3298	0.4970	0.4091	0.4665	0.3558	0.4616
LLaVA R-Tuning	DP	0.3188	0.6214	0.3665	0.5114	0.3724	0.5904	0.3909	0.6266	0.3706	0.6162	0.3692	0.5821
	SC	0.3745	0.5664	0.3907	0.5777	0.3530	0.6036	0.2360	0.6223	0.2890	0.6718	0.2858	0.6884
	PC-R	0.3825	0.5857	0.3846	0.4717	0.3622	0.4826	0.4651	0.3512	0.4194	0.3979	0.4148	0.3884
	Verbal	0.2250	0.5429	0.3802	0.5074	0.3327	0.4846	0.3365	0.5699	0.4408	0.5545	0.3665	0.4953
	VCAP	0.3253	0.5725	0.3893	0.4615	0.3531	0.5846	0.3956	0.4195	0.4110	0.5544	0.3920	0.5246
LLaVA RL	DP	0.2255	0.6249	0.3081	0.6403	0.2951	0.6905	0.2441	0.6266	0.3310	0.6985	0.2698	0.6814
	SC	0.2613	0.6390	0.3113	0.6351	0.2655	0.6623	0.2578	0.6475	0.2954	0.6430	0.2361	0.7277
	PC-R	0.2504	0.6741	0.3055	0.6857	0.2820	0.6227	0.2554	0.6826	0.3837	0.6941	0.2694	0.6193
	Verbal	0.2324	0.6591	0.2728	0.6580	0.2622	0.6560	0.2320	0.6626	0.2971	0.6477	0.2427	0.6464
	VCAP	0.2934	0.6491	0.3230	0.6635	0.2891	0.6741	0.2229	0.6621	0.2889	0.6209	0.2259	0.6257

Table 6: Comparison of ECE and AUC performance across different stage models and confidence estimation methods on GQA and OCR-VQA.

Question_Transform_Prompt

<USER>
You are an assistant that converts factual declarative statements into natural and relevant questions.
Your task is to rewrite each factual statement into a clear, concise, and context-preserving question. The generated question must **retain the key entities or concepts** from the original statement and must **not** be overly broad, vague, or ambiguous.

Example
Statement: `The Eiffel Tower is located in Paris.`
Question: `Where is the Eiffel Tower located?`

[INPUT STATEMENT]
`{atomic_statement}`

Return **ONLY** the transformed question, without any explanation or additional formatting.

VCAP_QA_Round1_Prompt

<USER>
You are given an image and a question.
This is **Stage 1** of a two-stage evaluation process.
Your task in this stage is **ONLY** to:
1) Describe in detail what you can see in the image.
2) Then state how confident you are that your visual understanding is sufficient to answer the question.
Guidelines:
– Focus exclusively on visual information from the image.
– Do **NOT** try to answer the question.
– Do **NOT** judge whether any candidate answer is correct.
– At the end, output a confidence score between 0 and 1.
Format of your response:
`[Your visual description here]`
`Visual confidence: <a number between 0 and 1>`

[QUESTION]
`{question}`

Model	Method	COCO						Google-Landmark					
		Entailment		Contradiction		Neutral		Entailment		Contradiction		Neutral	
		ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑
LLaVA	DP	0.1898	0.5388	0.2276	0.3283	0.1796	0.5638	0.2009	0.2929	0.2434	0.2813	0.2860	0.2514
	SC	0.2412	0.4900	0.2162	0.5859	0.3038	0.3945	0.2988	0.6730	0.2350	0.5907	0.2687	0.3489
	PC-R	0.1830	0.4336	0.1997	0.6098	0.1631	0.5195	0.1493	0.5846	0.2243	0.3520	0.1930	0.7692
	Verbal	0.2340	0.5388	0.2622	0.4773	0.2137	0.6172	0.2077	0.4949	0.2865	0.2907	0.2807	0.5989
	VCAP	0.4285	0.4730	0.4955	0.6282	0.4973	0.4144	0.3463	0.5000	0.6015	0.1026	0.4367	0.4872
LLaVA SFT	DP	0.1553	0.4710	0.1466	0.4552	0.1514	0.5641	0.1950	0.4987	0.1734	0.4003	0.1272	0.6443
	SC	0.2287	0.4545	0.1550	0.6816	0.1688	0.6254	0.2600	0.6440	0.2775	0.6932	0.2475	0.4479
	PC-R	0.1948	0.4015	0.1555	0.5026	0.1477	0.6681	0.2507	0.4747	0.1902	0.4242	0.2012	0.5283
	Verbal	0.1733	0.5139	0.1630	0.4910	0.1228	0.8034	0.2505	0.4573	0.2278	0.4293	0.2165	0.4122
	VCAP	0.3312	0.3397	0.3965	0.3429	0.3610	0.4632	0.3158	0.3496	0.4255	0.4737	0.3238	0.4052
LLaVA R-Tuning	DP	0.3269	0.4288	0.3512	0.4375	0.3432	0.5326	0.4921	0.5872	0.4314	0.5192	0.4921	0.5896
	SC	0.1590	0.5802	0.1675	0.5385	0.1675	0.5021	0.1175	0.5385	0.1375	0.5291	0.1400	0.5491
	PC-R	0.4144	0.4784	0.4369	0.5526	0.4263	0.5174	0.6467	0.5098	0.5867	0.5364	0.6467	0.5442
	Verbal	0.5050	0.6162	0.5287	0.5526	0.5455	0.5139	0.6985	0.4139	0.6367	0.4078	0.6985	0.5910
	VCAP	0.4597	0.4009	0.5115	0.4795	0.4673	0.4826	0.5607	0.5795	0.5498	0.5134	0.5757	0.5391
LLaVA RL	DP	0.1279	0.6844	0.1593	0.6530	0.1032	0.6009	0.1748	0.6348	0.1452	0.6449	0.1586	0.6234
	SC	0.1788	0.6753	0.1875	0.6140	0.1600	0.6036	0.1687	0.7123	0.1637	0.6399	0.2088	0.6615
	PC-R	0.1939	0.5758	0.1822	0.6946	0.1681	0.6414	0.2205	0.6118	0.1775	0.6387	0.1827	0.6675
	Verbal	0.1013	0.6823	0.1230	0.7670	0.1110	0.6766	0.1412	0.6747	0.1130	0.6786	0.1208	0.6071
	VCAP	0.2658	0.4691	0.2738	0.4088	0.2367	0.5961	0.2198	0.5662	0.2648	0.4829	0.2620	0.4118

Table 7: Comparison of ECE and AUC performance across different stage models and confidence estimation methods on COCO and Google-Landmark.

VCAP_QA_Round2_Prompt

<USER>
You are an expert evaluator of question–answer pairs.
This is **Stage 2** of the evaluation process.
You have already processed the image and produced the following visual understanding and confidence:
[VISUAL UNDERSTANDING FROM STAGE 1]
`{visual_description_output}`
Now, considering:
– the question,
– the proposed answer, and
– your visual understanding (including the visual confidence you just gave),

judge how confident you are that the proposed answer is correct.

[QUESTION]
`{question}`
[PROPOSED ANSWER]
`{response}`

You must output **ONLY** a single decimal number between 0 and 1 (inclusive), with at most two digits after the decimal point, representing your confidence that the answer is correct.
Return **ONLY one line:**
`Answer_confidence: <a number between 0 and 1>`

VCAP_Detail_Round1_Prompt

<USER>
You are given an image.
This is **Stage 1** of a two-stage evaluation process.
Your task in this stage is **ONLY** to:
1) Provide a detailed and objective description of everything you can identify in the image.
2) Then state how confident you are that your visual understanding is accurate and reasonably complete.
Guidelines:
– Focus purely on grounded visual observations.
– Do **NOT** try to write a polished caption.
– Do **NOT** judge or evaluate any model-generated description.
– At the end, output a confidence score between 0 and 1.
Format of your response:
`[Your visual description here]`
`Visual confidence: <a number between 0 and 1>`

(For reference, the original instruction/question was:)
`{question}`

Model	Method	sharegpt4v-wikiart						sharegpt4v-web-celebrity					
		Entailment		Contradiction		Neutral		Entailment		Contradiction		Neutral	
		ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑
LLaVA	DP	0.3504	0.6415	0.3354	0.6934	0.3581	0.4957	0.3010	0.5952	0.3080	0.4642	0.2645	0.4017
	SC	0.2200	0.5783	0.2812	0.4062	0.2200	0.6190	0.2325	0.6667	0.2675	0.3943	0.2537	0.7906
	PC-R	0.2397	0.4849	0.2360	0.5332	0.2452	0.5346	0.2505	0.4539	0.2294	0.3351	0.1840	0.5100
	Verbal	0.2782	0.4739	0.3113	0.3223	0.3000	0.4978	0.3393	0.3527	0.3190	0.4265	0.2395	0.7123
	VCAP	0.5323	0.3077	0.4685	0.3233	0.5477	0.3524	0.4660	0.3041	0.5308	0.2961	0.4827	0.2821
LLaVA SFT	DP	0.1348	0.4936	0.0861	0.7308	0.1167	0.5850	0.1025	0.6425	0.1964	0.4341	0.1446	0.4717
	SC	0.1837	0.5806	0.2075	0.6621	0.2563	0.7175	0.2313	0.3750	0.2025	0.6374	0.1962	0.4345
	PC-R	0.2425	0.4872	0.1861	0.6580	0.1750	0.6087	0.1458	0.6550	0.2178	0.5797	0.1892	0.4673
	Verbal	0.2162	0.4706	0.1920	0.4396	0.1728	0.6262	0.1580	0.6325	0.2498	0.5536	0.2075	0.6071
	VCAP	0.3922	0.4242	0.3930	0.4730	0.3752	0.6029	0.3752	0.6029	0.4123	0.6042	0.3930	0.4730
LLaVA R-Tuning	DP	0.4495	0.3468	0.4575	0.4158	0.4143	0.4274	0.4950	0.4615	0.5348	0.4282	0.4902	0.5197
	SC	0.2085	0.5310	0.2250	0.5410	0.2137	0.5692	0.2062	0.5806	0.2084	0.5156	0.2250	0.5410
	PC-R	0.5338	0.5856	0.5649	0.5158	0.5499	0.5630	0.5349	0.5821	0.5446	0.5205	0.5035	0.5224
	Verbal	0.5280	0.6171	0.6387	0.5857	0.5933	0.5077	0.5425	0.4872	0.6478	0.5641	0.5633	0.5724
	VCAP	0.5090	0.4150	0.5620	0.4333	0.4952	0.4408	0.3781	0.6387	0.5428	0.5918	0.5620	0.4333
LLaVA RL	DP	0.1560	0.6307	0.1331	0.6175	0.1158	0.6724	0.1260	0.6889	0.1413	0.6984	0.1328	0.6225
	SC	0.1775	0.6827	0.1950	0.6714	0.1675	0.8229	0.1900	0.6940	0.2363	0.6875	0.1875	0.6663
	PC-R	0.2296	0.6093	0.2115	0.6188	0.2372	0.6749	0.1756	0.6556	0.1945	0.6984	0.2183	0.6300
	Verbal	0.1505	0.6280	0.0975	0.6203	0.1285	0.6862	0.1053	0.6570	0.1070	0.6705	0.1413	0.6925
	VCAP	0.3075	0.6207	0.3748	0.5044	0.3575	0.6211	0.3120	0.6200	0.3338	0.6486	0.3225	0.5641

Table 8: Comparison of ECE and AUC performance across different stage models and confidence estimation methods on sharegpt4v-wikiart and sharegpt4v-web-celebrity.

VCAP_Detail_Round2_Prompt

<USER>
You are an expert evaluator of image descriptions.
This is **Stage 2** of the evaluation process.
Below is your previous visual understanding and its confidence score:
[VISUAL UNDERSTANDING FROM STAGE 1]
`{visual_description_output}`
Your task now is to evaluate how well the following detailed description matches the image, based on your visual understanding above.

[PROPOSED DESCRIPTION]
`{response}`
Guidelines:
– Judge whether the proposed description is accurate, faithful, and consistent with the image.
– Consider object correctness, quantity, spatial relationships, and overall consistency.
– Do **NOT** write any reasoning or explanation.
You must output **ONLY** a single decimal number between 0 and 1 (inclusive), with at most two digits after the decimal point, representing your confidence that the proposed description is correct.
Return **ONLY one line:**
`Answer_confidence: <a number between 0 and 1>`

Model	Method	ART500K						LCS-558K					
		Entailment		Contradiction		Neutral		Entailment		Contradiction		Neutral	
		ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑
LLaVA	DP	0.3523	0.7849	0.3095	0.5078	0.3466	0.6708	0.1963	0.5985	0.3063	0.2411	0.2750	0.4545
	SC	0.2437	0.5627	0.2900	0.3918	0.2362	0.5752	0.2750	0.5141	0.2300	0.6815	0.2288	0.4154
	PC-R	0.2124	0.5812	0.2016	0.3433	0.2106	0.5925	0.1810	0.4770	0.2466	0.4286	0.2122	0.5784
	Verbal	0.2925	0.4174	0.2238	0.6458	0.2655	0.6677	0.2573	0.5460	0.2825	0.5506	0.2758	0.6034
	VCAP	0.5132	0.3761	0.4538	0.3593	0.4957	0.4974	0.4912	0.6042	0.4753	0.5013	0.5200	0.5526
LLaVA SFT	DP	0.1280	0.5201	0.1140	0.3985	0.1131	0.5627	0.1663	0.5775	0.1301	0.6477	0.1469	0.5027
	SC	0.1950	0.5764	0.1938	0.5163	0.2225	0.6740	0.2243	0.6438	0.2800	0.5947	0.2425	0.5755
	PC-R	0.1676	0.6065	0.1677	0.4912	0.1960	0.5439	0.1848	0.4975	0.1580	0.5063	0.1390	0.6882
	Verbal	0.1442	0.6704	0.1448	0.4825	0.1748	0.6787	0.2188	0.6288	0.1615	0.5164	0.1337	0.5357
	VCAP	0.3812	0.2928	0.3010	0.5543	0.3585	0.4502	0.3518	0.4514	0.4440	0.4079	0.4210	0.2162
LLaVA R-Tuning	DP	0.5101	0.4467	0.4836	0.4590	0.5136	0.5641	0.4240	0.5604	0.4120	0.5031	0.4571	0.5089
	SC	0.2187	0.6744	0.2187	0.6568	0.2287	0.6744	0.1688	0.6118	0.1225	0.6051	0.1325	0.5256
	PC-R	0.5567	0.6583	0.5638	0.6410	0.5900	0.6282	0.5000	0.5748	0.5464	0.5310	0.5457	0.5985
	Verbal	0.6488	0.5057	0.5757	0.5897	0.6108	0.5051	0.5720	0.6441	0.5803	0.5456	0.6542	0.5097
	VCAP	0.5088	0.3795	0.5987	0.5018	0.5467	0.4975	0.4935	0.5833	0.6232	0.4256	0.5052	0.4051
LLaVA RL	DP	0.1135	0.5908	0.1328	0.5878	0.1619	0.6802	0.1450	0.6815	0.1531	0.6642	0.1223	0.6250
	SC	0.2162	0.6926	0.1900	0.6953	0.1800	0.6680	0.1700	0.7038	0.1812	0.7240	0.2362	0.6217
	PC-R	0.2300	0.6217	0.1610	0.6984	0.2053	0.6815	0.1961	0.6078	0.1578	0.6323	0.1716	0.6083
	Verbal	0.1118	0.6141	0.1303	0.6561	0.1355	0.6805	0.1383	0.6549	0.1205	0.6391	0.1200	0.6583
	VCAP	0.2800	0.4640	0.2663	0.5961	0.3387	0.4750	0.3018	0.6082	0.3572	0.5429	0.3120	0.6699

Table 9: Comparison of ECE and AUC performance across different stage models and confidence estimation methods on ART500K and LCS-558K.

Model	Method	MovieNet						sharegpt4v-web-landmark					
		Entailment		Contradiction		Neutral		Entailment		Contradiction		Neutral	
		ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑	ECE↓	AUC↑
LLaVA	DP	0.3073	0.5256	0.3104	0.4375	0.3638	0.4902	0.2403	0.3913	0.2055	0.6300	0.2674	0.4246
	SC	0.2738	0.5456	0.2500	0.6120	0.2488	0.6495	0.2750	0.5256	0.3075	0.6350	0.2650	0.6381
	PC-R	0.2161	0.4316	0.2366	0.3724	0.2896	0.4191	0.2342	0.4859	0.2043	0.2938	0.2182	0.5972
	Verbal	0.2895	0.5783	0.2825	0.5833	0.3170	0.2132	0.2878	0.4297	0.2658	0.4638	0.2940	0.5895
	VCAP	0.4510	0.3817	0.5047	0.3549	0.4117	0.4049	0.4415	0.3789	0.4803	0.3299	0.4838	0.4083
LLaVA SFT	DP	0.1210	0.5812	0.1735	0.6521	0.1461	0.6387	0.1819	0.5482	0.2577	0.3081	0.2810	0.3675
	SC	0.1875	0.6368	0.2094	0.7167	0.2225	0.6187	0.3450	0.4661	0.2675	0.7689	0.3100	0.6112
	PC-R	0.1804	0.3989	0.2580	0.4208	0.2085	0.5533	0.1999	0.6771	0.2933	0.4583	0.2856	0.6375
	Verbal	0.2188	0.5100	0.2506	0.6271	0.2715	0.5120	0.2025	0.6536	0.2695	0.5114	0.3062	0.4538
	VCAP	0.3757	0.3569	0.4738	0.5161	0.3832	0.2613	0.3518	0.4815	0.3953	0.4843	0.3500	0.5128
LLaVA R-Tuning	DP	0.5389	0.5777	0.4749	0.4516	0.5258	0.5744	0.3575	0.5197	0.4014	0.4615	0.3885	0.5270
	SC	0.2975	0.4846	0.2781	0.6419	0.2850	0.4651	0.1900	0.6541	0.1775	0.5329	0.1675	0.6974
	PC-R	0.6052	0.5387	0.5547	0.5315	0.5725	0.5189	0.5299	0.5197	0.5529	0.5527	0.5530	0.5514
	Verbal	0.6245	0.5513	0.5481	0.5738	0.6250	0.5929	0.6737	0.5150	0.6425	0.5414	0.6388	0.5450
	VCAP	0.5483	0.4692	0.4672	0.5484	0.5090	0.4854	0.4982	0.4495	0.5367	0.4974	0.6940	0.6237
LLaVA RL	DP	0.1414	0.6188	0.2078	0.6275	0.1774	0.6886	0.1584	0.6843	0.1379	0.6199	0.1997	0.6179
	SC	0.1700	0.6438	0.2016	0.6804	0.1650	0.7621	0.1825	0.6301	0.1738	0.6245	0.2187	0.6793
	PC-R	0.1872	0.6775	0.2395	0.6843	0.2581	0.6812	0.2183	0.6545	0.2404	0.6846	0.2272	0.7238
	Verbal	0.1420	0.5487	0.1797	0.6706	0.1938	0.6487	0.1862	0.6169	0.1265	0.6242	0.1745	0.6923
	VCAP	0.3088	0.4714	0.3372	0.5000	0.3223	0.6632	0.3247	0.5201	0.3910	0.7398	0.2775	0.5323

Table 10: Comparison of ECE and AUC performance across different stage models and confidence estimation methods on MovieNet and sharegpt4v-web-landmark.

Model	Metric	VQAv2			GQA			OK-VQA		
		Entailment	Contradiction	Neutral	Entailment	Contradiction	Neutral	Entailment	Contradiction	Neutral
LLaVA	ACC	0.74	0.61	0.65	0.79	0.65	0.73	0.71	0.59	0.71
SFT	ACC	0.74	0.60	0.61	0.76	0.70	0.57	0.64	0.51	0.64
RL	ACC	0.72	0.62	0.64	0.75	0.65	0.73	0.70	0.50	0.77

Table 11: ACC comparison across three model variants on VQAv2, GQA, and OK-VQA.

Model	Metric	TextVQA			OCR-VQA		
		Entailment	Contradiction	Neutral	Entailment	Contradiction	Neutral
LLaVA	ACC	0.59	0.41	0.40	0.89	0.60	0.72
SFT	ACC	0.64	0.47	0.54	0.72	0.52	0.63
RL	ACC	0.61	0.48	0.57	0.80	0.61	0.68

Table 12: ACC comparison across three model variants on TextVQA and OCR-VQA.

Dataset	Task	Total	Entailment			Contradiction			Neutral		
			Total	Train	Val	Total	Train	Val	Total	Train	Val
ART500K	Detail	1096	640	33	607	192	33	159	264	33	231
COCO	Detail	15199	7292	33	7259	3199	33	3166	4708	33	4675
Google-Landmark	Detail	1079	753	33	720	145	33	112	181	33	148
LCS-558K	Detail	15956	6968	33	6935	4688	33	4655	4300	33	4267
MovieNet	Detail	1131	540	33	507	132	33	99	326	33	293
sharegpt4v-textvqa	Detail	1966	682	33	649	713	33	680	571	33	538
sharegpt4v-web-celebrity	Detail	1895	208	33	175	189	33	156	187	33	154
sharegpt4v-web-landmark	Detail	1918	1110	33	1077	353	33	320	455	33	422
sharegpt4v-wikiart	Detail	1972	864	33	831	532	33	499	576	33	543
GQA	QA	5411	3606	33	3573	1136	33	1103	669	33	636
OCR-VQA	QA	3025	1917	33	1884	762	33	729	346	33	313
OK-VQA	QA	14802	9680	33	9647	2173	33	2140	1931	33	1898
TextVQA	QA	4740	2503	33	2470	1441	33	1408	796	33	763
VQAv2	QA	12942	9320	33	9287	2184	33	2151	1438	33	1405

Table 13: Summary of dataset size, task type, and NLI label distribution (Total/Train/Val) for all benchmarks.

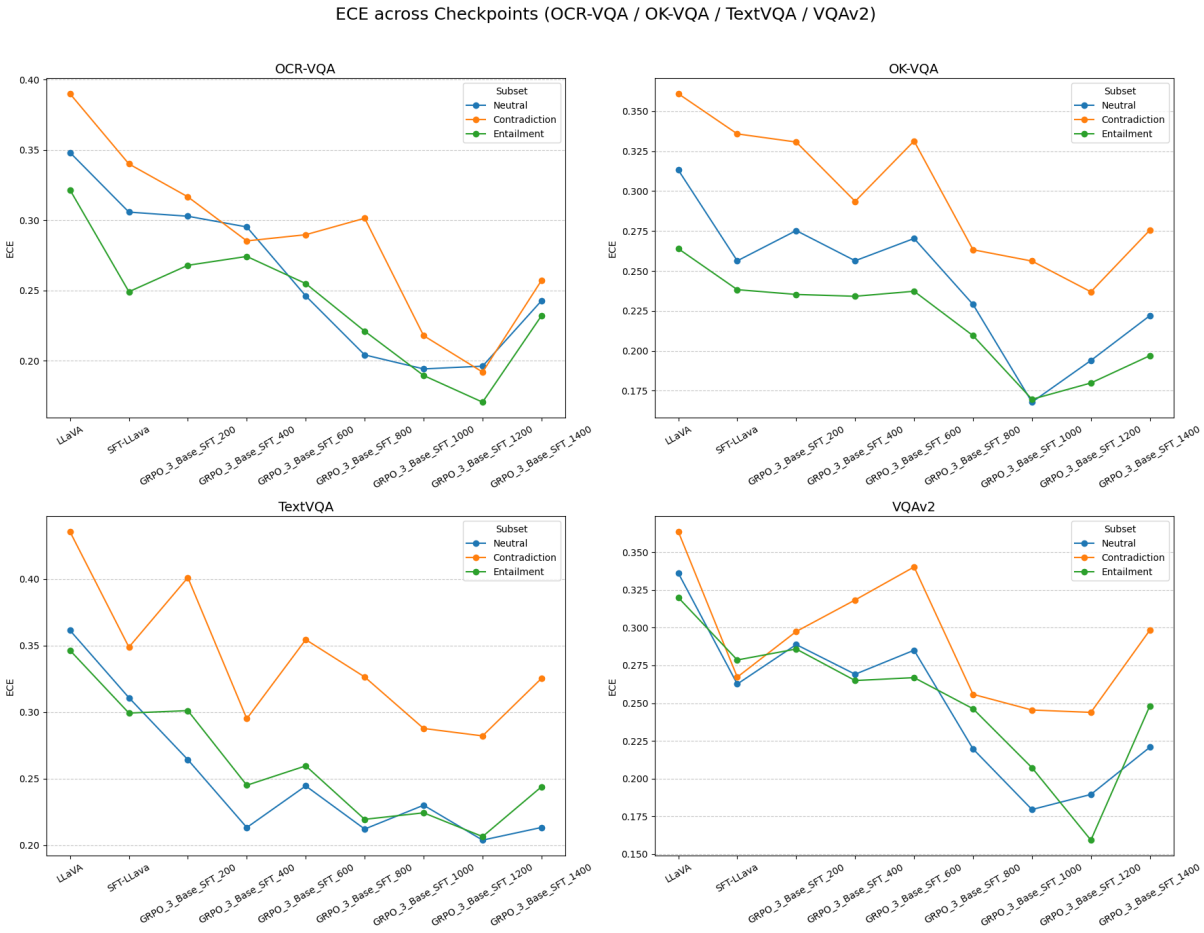

Figure 8: ECE evolution across training checkpoints on four QA datasets (OCR-VQA, OK-VQA, TextVQA, VQAv2). For each dataset, we report ECE on the Entailment, Contradiction, and Neutral subsets for the base LLaVA model, the SFT-tuned model, and GRPO models at different training epochs, showing progressive calibration improvement and the effect of over-training.