Uncertainty quantification for MLLMs

Anonymous Authors[1]

Abstract

Multimodal Large Language Models (MLLMs) hold promise in tackling challenging multimodal tasks, but may generate seemingly plausible but erroneous output, making them hard to trust and deploy in real-life settings. Generating accurate uncertainty metrics quickly for each MLLM response during inference could enable interventions such as escalating queries with uncertain responses to human experts or larger models for improved performance. However, existing uncertainty quantification methods require external verifiers, additional training, or high computational resources, and struggle to handle scenarios such as out-of-distribution (OOD) or adversarial settings. To overcome these limitations, we present an efficient and effective training-free framework to estimate MLLM output uncertainty at inference time without external tools, by computing metrics based on the diversity of the MLLM's responses that is augmented with internal indicators of each output's coherence. We empirically show that our method significantly outperforms benchmarks in predicting incorrect responses and providing calibrated uncertainty estimates, including for OOD, adversarial and domain-specific (e.g. medical radiology) data settings.

1. Introduction

Building on the impressive capabilities of Large Language Models (LLMs) in handling a wide variety of text-based tasks (OpenAI et al., 2024), Multimodal Large Language Models (MLLMs) are LLM-based models that can process the input of different modalities such as images and text, allowing them to perform important downstream multimodal tasks involving both visual comprehension and language abilities such as visual question answering (Liu et al., 2023c; Hartsock and Rasool).

However, the synthesis of multiple modalities introduces additional challenges in managing uncertainty and mitigating errors in the models' output. MLLMs need to handle not only the ambiguity of visual input, but also understand text-based questions, extract relevant visual features, and incorporate these features along with any additional text-based information to generate a response. All these sub-tasks are potential sources of ambiguity and error that may accumulate in the final generated response, leading to problems such as object hallucination (Bai et al., 2024) or erroneous scene interpretation. While there are works that attempt to directly mitigate such errors or hallucinations during model training by adjusting characteristics of the training data (Liu et al., 2023a; Yu et al., 2024; Wang et al., 2024; Yue et al., 2024), model architecture (Liu et al., 2024; Tong et al., 2024; Zhai et al., 2023), or training process (Jiang et al., 2024; Yue et al., 2024), these errors cannot be completely eliminated in practical settings, given real-world data that is noisy and ambiguous.

A complementary approach to such training-based approaches would be to use inference-time methods to detect potential errors of MLLMs. For a given MLLM, such error detection methods could indicate when an output is more likely to contain errors, allowing users to treat these output differently, for example, passing these output to a larger model or human expert to verify its accuracy. However, a typical MLLM output would not contain any accompanying indication of uncertainty in its accuracy. The lack of error detection and uncertainty estimation becomes a major bottleneck in MLLMs' deployment in practical applications (e.g., medical imaging analysis (Liu et al., 2023b; Tian et al., 2024; Lee et al., 2025)), where the reliability of the models' output is critical. A few recent works have proposed methods to detect and fix MLLM hallucinations, but have mainly relied on either external verifiers (Liu et al., 2023a; Sun et al., 2023) or methods that involve relatively expensive computation (Zhang et al.; Khan and Fu, 2024b) to do so, which may not be practical in many settings with resource limitations.

In our work, we present UMPIRE, a training-free inference-time method to approximate the uncertainty associated with

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on ICML 2025 Workshop on Reliable and Responsible Foundation Models. Do not distribute.

MLLM output and detect errors. UMPIRE uses a simple but effective method to compute a metric indicative of how likely an output may contain an error, taking into account both the uncertainty indicated by the diversity of possible output for a given query, and the quality of the output reflected by its self-assessment. In summary, we (1) proposed a set of clear desiderata that MLLM unlearning metrics should satisfy (Sec. 2.2) and analyzed challenges associated with existing approaches such as entropy-based methods (Sec. 5), (2) proposed a novel MLLM uncertainty method and metric (Sec. 3) inspired by past works on determinantal point processes (DPP) (Kulesza, 2012), and (3) empirically show how UMPIRE consistently outperforms all benchmarks with less computational time (Sec. 4).

2. Problem formulation and desiderata

2.1. Problem formulation

We consider the setting where we have an whitebox MLLM $\mathcal{M}$ that takes in image I and text q input[1], and produce text output $y = [w_i]_{i=1}^N$ that are sequences of tokens w from the MLLM decoder's vocabulary space. While MLLMs can be implemented with various types of model architectures, in general we can represent them as conditional probability distributions $p_{\mathcal{M}}$ of text output y over multi-modal input queries (I, q) generated autoregressively, i.e., $\mathcal{M}(I, q) := p_{\mathcal{M}}(y|I, q) = p_{\mathcal{M}}(w_1|I, q)p_{\mathcal{M}}(w_2|I, q, w_1) \dots p_{\mathcal{M}}(w_n|I, q, w_{1:n-1})$.

We can apply the MLLM to multi-modal tasks $\mathcal{T}$ with task instances $t \in \mathcal{T}$, where $t := (I_t, q_t)$ represents the input query containing both an image portion I_t and text portion q_t, and for clarity we explicitly denote $t^* := (t; y_t^*)$ as task instances with known text ground truth output y_t^*. The MLLMs' response $\hat{y}_t$ to a task instance t can be sampled autoregressively from $\mathcal{M}(t) = \mathcal{M}(I_t, q_t)$, and its performance can be evaluated by whether the response (e.g., a low temperature sampled response) matches the ground truth, i.e. $a(\mathcal{M}, t^*) := \mathbb{I}\{\hat{y}_t = y_t^*\}$, where $\mathbb{I}$ is an appropriate binary indicator that evaluates whether two responses match in the context of answering task $\mathcal{T}$. The overall MLLM performance on the task $\mathcal{T}$ can be computed as the expected performance over its constituent labeled task instances, i.e., $a(\mathcal{M}, \mathcal{T}) := \mathbb{E}_{t^* \in \mathcal{T}} a(\mathcal{M}, t^*)$, where we overload notation for simplicity.

Given a task $\mathcal{T}$, the goal is to develop a framework that computes a task instance-specific uncertainty metric $u(\mathcal{M}; t)$ for any $t \in \mathcal{T}$ at inference time that is highly indicative of the expected accuracy $a(\mathcal{M}, t^*)$. Note that for our purposes we are looking for a metric for overall uncertainty, rather than sub-characterization of either aleatoric or epistemic uncertainty. Such a metric can be used to assess whether the model output should be trusted or discarded, and have challenging task instances deferred to a human or more capable MLLM model instead.

2.2. Desiderata

Given the above setting, an appropriate uncertainty metric u should ideally satisfy several key desiderata. We propose a non-exhaustive, generally-applicable list below, though others may be important for specific scenarios. First, the metric should be effective in approximating the uncertainty associated with each response. We assess this via three *effectiveness desiderata*:

R1 Classification. The metric should be able to distinguish between task instances that the MLLM will get correct ($t_c \in \mathcal{C} := \{t \in \mathcal{T} \mid a(\mathcal{M}, t^*) = 1\}$) or wrong ($t_w \in \mathcal{W} := \{t \in \mathcal{T} | a(\mathcal{M}, t^*) = 0\}$. Specifically, for randomly sampled pairs of task instances t_c and t_w,

$$\mathbb{P}[u(\mathcal{M}, t_w) > u(\mathcal{M}, t_c)] \approx 1 \tag{1}$$

where the goal is for Eq. (1) to be as close to 1 as possible, implying that the metric can classify with high probability whether the model will get task instances wrong, using just $\mathcal{M}$ and instance input t. This means that there exist a threshold γ such that $u(\mathcal{M}, t) > \gamma$ indicates that it is likely that $t \in \mathcal{W}$, and smaller values indicate that $t \in \mathcal{C}$. Note that Eq. (1) can be evaluated by computing the Area under the Receiver Operating Characteristic Curve (AUROC) of the metric, which we do in Sec. 4.

R2a Proportionality. The metric should be proportional to the probability that the MLLM will get the task instance wrong, i.e.,

$$u(\mathcal{M}, t) \propto \mathbb{P}[a(\mathcal{M}, t^*) = 0]. \tag{2}$$

Such a metric would be useful in many settings where it is important to get a continuous estimate of how likely the MLLM would get a task instance correct, e.g., to allocate resources for obtaining better responses such as through escalation of task instances to better but more expensive models.

R2b Calibration. If provided a small sample of unlabeled task instances, the metric u should be easily adjustable to $\tilde{u} \in [0, 1]$ (e.g., using min-max scaling) such that it is well calibrated (Guo et al.), i.e.,

$$\mathbb{P}(a(\mathcal{M}, t^*) = 1 \mid \tilde{u}(\mathcal{M}, t) = p) \approx p, \quad \forall p \in [0, 1]. \tag{3}$$

This desiderata is a stricter version of **R2a** where the metric is properly scaled to provide good estimates of

[1] While we focus on image and text input in the paper, our method can be extended to other modalities in future works as it does not make use of modality-specific features.

how likely the MLLM would get a given task instance correct, rather than just provide a binary classification given a threshold based on **R1**. Such a metric is important in applications that require such estimates for risk management, and also allows better informed choices of the appropriate threshold γ to use in classification for **R1**. Note that to align with past works for ease of comparability (Guo et al.; Khan and Fu, 2024a), **R2b** and $\tilde{u}$ are formulated based on response accuracy $a(\mathcal{M}, t^*) = 1$, while **R2a** and u are based on error $a(\mathcal{M}, t^*) = 0$ which is more natural for an uncertainty measure.

In A.13, we provide additional discussion on the differences among these desiderata and why they are needed.

In addition, we consider design desiderata that reflect practical considerations for the deployment of the metric in realistic applications:

R3 **Focus on semantics.** The metric should depend on the semantic meaning of the responses, and not just lexical variations (e.g., paraphrases of a response with the same meaning). This is because for many MLLM tasks (e.g., visual question-answering), we are less concerned about lexical variations (e.g. "the cat hid the rat" and "the rat was hidden by the cat") compared to semantically different responses ("the dog sat on the mat").

R4 **Response coherence.** In addition, the metric should also consider the coherence of each sampled response with respect to the multimodal task instance query (e.g. images and text), rather than take into account only a single modality.

R5 **Computational Efficiency.** The metric should be able to be efficiency computed, for it to be practically deployed. This includes (a) fast computational runtime, and (b) no strict requirements of external pre-trained models or separately trained reward models as they incur additional costs and may not be feasible for some inference pipelines. For situations where the MLLM under study is a blackbox, it may be necessary to relax condition (b) to use a proxy whitebox model, but the proxy model should be small and cheap to run.

3. Method

3.1. Challenges faced by existing methods and design choices for UMPIRE

The desiderata in Sec. 2.2 provides useful constraints in guiding the design of a suitable uncertainty framework. First, **R5** provides practical constraints on the tools and methods that we can employ. Unlike existing methods that makes use of external tools (Zhang et al.; Sun et al., 2023; Liu et al., 2023a), extensive training to learn uncertainty estimates for specific models (Li et al., 2024), extensive prompting with perturbations of input queries (Khan and Fu, 2024b; Zhang et al.), or asking the model itself (Xiong et al., 2024a), our approach should rely solely on MLLMs' readily accessible output (except when dealing with blackbox models where we can then use a whitebox model as proxy, explained in Sec. 4.4), and allow for efficient inference (e.g. parallelization and low overheads). Hence, in our approach we only consider the use of the model's last layer embeddings and logits information.

Second, given the resource constraints, **R3** implies that our framework should consider the MLLM's embedding information of the full responses, which the model already computes during the generation process and is computationally efficient to extract. This would be unlike methods that focus solely on token-level information (Malinin and Gales, 2021).

Third, **R4** suggests that our framework would need to factor in multimodal input information, rather than solely rely on signals that are relatively insensitive to it (see App. A.2). In App. A.3, we empirical found that considering such information (i.e., the incoherence scores) significantly improves the ability of our metric in performing well in the effectiveness desiderata.

3.2. UMPIRE framework

Given the considerations above, we propose UMPIRE, a simple but effective framework built on the hypothesis that uncertainty can be estimated well based on a **global measure of semantic diversity** of sampled MLLM responses, adjusted by a **local measure of the coherence of each response**. Drawing inspiration from determinantal point processes (DPP) and the quality-diversity decomposition of its kernels (Kulesza, 2012), which are used by works in quantum physics (Collura et al., 2024) and active learning (Bıyık et al., 2019) to model systems of repulsive interactions and characterize sample diversity, our UMPIRE framework computes a **coherence-adjusted semantic volume** quantity as the proposed uncertainty metric. Intuitively, we hypothesize that the more uncertain the MLLMs are about a task instance, the stronger the 'repulsive' forces among its responses are, leading to more diverse responses that spans a larger volume in semantic space. However, responses have different levels of coherence, and hence intuitively should have different contribution to the overall volume (e.g., lighter particles may move further out).

Specifically, for a given task instance $t \in \mathcal{T}$, the UMPIRE framework will consists of the following:

U1 **Sampling.** We first have the MLLM generate k responses $\mathcal{Y}_t = \{\hat{y}_i\}_{i=1}^k$ to t based on standard sampling methods with T=1 (we show in App. A.5 and App. A.7

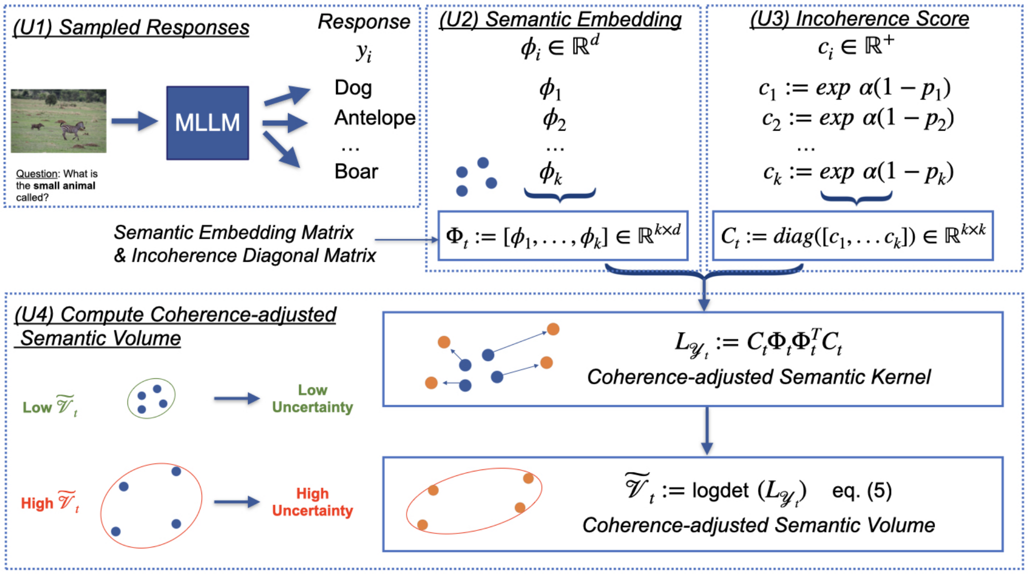
(U1) Sampled Responses
Response
y_i
MLLM
Dog
Antelope
...
Boar
Question: What is the small animal called?
(U2) Semantic Embedding
$\phi_i \in \mathbb{R}^d$
ϕ_1
ϕ_2
...
ϕ_k
(U3) Incoherence Score
$c_i \in \mathbb{R}^+$
$c_1 := exp\ \alpha(1-p_1)$
$c_2 := exp\ \alpha(1-p_2)$
...
$c_k := exp\ \alpha(1-p_k)$
Semantic Embedding Matrix & Incoherence Diagonal Matrix
$\Phi_t := [\phi_1, \ldots, \phi_k] \in \mathbb{R}^{k \times d}$
$C_t := diag([c_1, \ldots c_k]) \in \mathbb{R}^{k \times k}$
(U4) Compute Coherence-adjusted Semantic Volume
$L_{\mathcal{Y}_t} := C_t \Phi_t \Phi_t^T C_t$
Coherence-adjusted Semantic Kernel
Low $\widetilde{\mathscr{V}}_t$
Low Uncertainty
High $\widetilde{\mathscr{V}}_t$
High Uncertainty
$\widetilde{\mathscr{V}}_t :=$ logdet $(L_{\mathcal{Y}_t})$ eq. (5)
Coherence-adjusted Semantic Volume

Figure 1: Schematic describing the UMPIRE framework

that our results are robust to variations in sampling parameters).

U2 Semantic embedding. For each response $\hat{y}_i$, we can extract the last hidden layer vector of the last response token (more analysis on hidden layer selection in App. A.8) as its d-dimensional semantic embedding vector $\phi_i \in \mathbb{R}^d$, and normalize it if not already so (which may be the case e.g., in LLM embedding models (Reimers and Gurevych, 2019)). The k sampled embedding vectors for t then form the $k \times d$ embedding matrix Φ_t. Our use of the MLLM semantic embeddings help satisfy **R3**.

U3 Incoherence score. Concurrently, it will also extract the model-generated probability scores p_i across the tokens in each response $\hat{y}_i$. Note that since we extract this during the generation process that include all modality input, p_i contains information on the response coherence as required by **R4**. We then use it to compute the incoherence score $c_i \in \mathbb{R}^+$, $c_i := \exp \alpha(1 - p_i)$, where α is a scaling hyperparameter that is fixed across instances of a given task, and as explained below could be heuristically set even *without* calibration in cases where there is no labeled dev set and still yield good performance. As its name implies, this score intuitively captures how incoherent each response is: e.g., a response deemed fully coherent by the MLLM will have $p_i = 1$ and $c_i = 1$ which is the smallest possible value, while low probability responses, will have large c_i. For the full sample $\mathcal{Y}_t$, we can consolidate the scores into an $k \times k$ incoherence score diagonal matrix C_t.

U4 Coherence-adjusted semantic volume. Given the above, for the sampled response $\mathcal{Y}_t$ we can compute its coherence-adjusted semantic kernel $L_{\mathcal{Y}_t} := C_t \Phi_t \Phi_t^T C_t$, similar to quality-adjusted kernels used in DPPs. We can then compute the final UMPIRE uncertainty metric $\widetilde{V}_t$:

$$\widetilde{V}_t := \log\det(L_{\mathcal{Y}_t}), \tag{4}$$

where in practice a small jitter term is added to $L_{\mathcal{Y}_t}$ for numerical stability and to avoid degeneracy. $\widetilde{V}_t$ is a monotonic indicator for the coherence-adjusted semantic volume, since from geometry, $\det(L_{\mathcal{Y}_t}) = \text{Vol}^2(C_t \Phi_t)$, the squared volume of the parallelepied spanned by the coherence-adjusted response semantic embedding vectors.

Our UMPIRE framework (U1–U4) is summarized in Fig. 1. Given the above, UMPIRE does not violate any of the design desiderata (**R3**–**R5**). We empirically show in Sec. 4 that UMPIRE also performs well in the effectiveness desiderata (**R1**–**R2b**), and significantly outperforms baselines.

It turns out that the UMPIRE metric $\widetilde{V}_t$ can in fact be simplified into an easily interpretable form:

$$\begin{aligned} \widetilde{V}_t &= \log\det(L_{\mathcal{Y}_t}) \\ &= \log\det(\Phi_t \Phi_t^T) + \log\det(C_t C_t) \\ &= \mathcal{V}_t + \tilde{\alpha}\mathbb{E}[1 - p], \end{aligned} \tag{5}$$

where the first term can be interpreted as the unadjusted semantic volume metric, and the second term an approximate expectation value of the MLLM responses' model-generated probability of getting the task instance wrong (see App. A.4 for derivation).

Metric	Method	VQAv2	OKVQA	AdVQA	MathVista	VQA-RAD	Avg
AUROC ↑	Neighborhood Consistency	0.769	0.528	0.657	0.763	0.706	0.685
	LN-Entropy	0.781	0.705	0.647	0.667	0.614	0.683
	Semantic Entropy	0.848	0.716	0.763	0.805	0.767	0.780
	EigenScore	0.868	0.738	0.774	0.814	**0.803**	0.799
	Ours	**0.882**	**0.755**	**0.787**	**0.822**	0.802	**0.810**
TPR@ 10% FPR ↑	Neighborhood Consistency	0.362	0.095	0.189	0.408	0.189	0.248
	LN-Entropy	0.282	0.244	0.168	0.347	0.127	0.234
	Semantic Entropy	0.574	0.327	0.419	0.437	0.511	0.453
	EigenScore	0.602	0.340	0.466	0.483	**0.601**	0.498
	Ours	**0.629**	**0.369**	**0.477**	**0.497**	0.587	**0.512**
TPR@ 1% FPR ↑	Neighborhood Consistency	0.049	0.008	0.019	0.030	0.023	0.026
	LN-Entropy	0.057	0.030	0.066	0.075	0.065	0.059
	Semantic Entropy	0.177	0.057	0.125	**0.136**	0.286	0.156
	EigenScore	0.215	0.074	0.171	0.086	0.304	0.170
	Ours	**0.230**	**0.091**	**0.185**	0.131	**0.326**	**0.193**

Table 1: Uncertain responses classification (**R1**) performance of different uncertainty quantification methods across VQA tasks, including adversarial, OOD, and domain-specific settings. The classification metrics include AUROC (↑ better), TPR at different FPR levels (↑ better). UMPIRE achieves the best or second-best performance across all benchmarks, with only marginal differences when not ranked first.

This explicit decomposition in Eq. (5) shows the balance between the global measure of semantic diversity, and also the aggregated notion of the local measure of response coherence, with the hyperparameter $\tilde{\alpha} := 2k\alpha$ balancing the contribution between the two terms. In practice, we found empirically that for a given task $\mathcal{T}$, $\tilde{\alpha}$ could in fact be roughly set such that the two terms have the same order of magnitude (e.g., based on a small unlabeled sample of task instances) while achieving good performance, allowing us to avoid the need for hyperparameter-tuning while still producing good performance (see App. A.3 for ablation results). This also implies that in general both terms play an important role in contributing to the UMPIRE metric's significantly better performance compared to baselines, as we will see in Sec. 4.

4. Experiments

Experiment settings. We adapt the experimental set-up of Kuhn et al. (2023) for the multi-modality setting. For datasets, we use a range of visual question-answering benchmark datasets, including for general (VQAv2) (Goyal et al., 2017), out-of-distribution (OKVQA) (Marino et al., 2019) and adversarial (AdVQA) (Li et al., 2021) settings, as well as for specialized domains such as medicine/radiology (VQA-RAD) (Lau et al., 2018) and mathematics (MathVista) (Lu et al., 2023). We use `Llava-v1.5-13b` (Liu et al., 2023c) as the MLLM for our main experimental results, but show that our results are robust to different model sizes and model families in App. A.10. To benchmark our UMPIRE framework, we considered not only methods on MLLM uncertainty quantification, (1) Neighborhood Consistency (Khan and Fu, 2024b), but also extended methods developed for LLM uncertainty quantification to the MLLM setting, which sometimes have even better performance than recent MLLM-focused methods: (2) LN-Entropy (Malinin and Gales, 2021), (3) Semantic Entropy (Kuhn et al., 2023), and EigenScore (Chen et al., 2024). More details on benchmarks are in App. A.1, and additional ablation results are in App. A.6.

4.1. Classification of uncertain responses

We first consider the performance of UMPIRE and the benchmark algorithms in **R1** , i.e., predicting whether the MLLM $\mathcal{M}$ will generate the right response for a specific task instance t, i.e., $a(\mathcal{M}, t^*) = 1$. Note that the lefthand side of Eq. (1) corresponds to the definition of the AUROC of whether the metric u can classify between t_c and t_w. An AUROC score of 1 indicates that the metric can perfectly distinguish the correct and incorrect predictions, while 0.5 would correspond to the expected performance of a random baseline. Table 1 shows the AUROC evaluated over the validation set for UMPIRE and the benchmarks on the VQAv2, OKVQA, AdVQA, MathVista and VQA-RAD datasets. We see that UMPIRE consistently outperforms all other benchmarks with average AUROC around 0.81, especially the multi-modal specific method, Neighborhood Consistency (AUROC of 0.685) , which faces significant difficulty in

Metric	Method	VQAv2	OKVQA	AdVQA	MathVista	VQA-RAD	Avg
CPC ↑	Neighborhood Consistency	0.784	0.778	0.562	0.721	0.733	0.716
	LN-Entropy	0.553	0.851	0.916	0.909	0.892	0.824
	Semantic Entropy	0.916	0.277	0.759	0.856	0.690	0.700
	EigenScore	0.938	0.893	0.888	0.797	0.656	0.834
	Ours	**0.946**	**0.966**	**0.979**	**0.945**	**0.908**	**0.949**
ECE ↓	Neighborhood Consistency	0.326	0.504	0.344	0.078	0.138	0.278
	LN-Entropy	0.046	0.041	0.068	0.116	0.111	0.076
	Semantic Entropy	0.046	0.144	0.161	0.220	0.359	0.186
	EigenScore	0.047	0.162	0.217	0.312	0.366	0.221
	Ours	**0.038**	**0.036**	**0.042**	**0.071**	**0.067**	**0.051**

Table 2: Performance of uncertainty proportionality and calibration (**R2a**, **R2b**) of different uncertainty quantification methods across VQA tasks, including adversarial, OOD, and domain-specific settings. The metrics include CPC (↑ better) and ECE (↓ is better). Overall, UMPIRE consistently surpasses existing approaches across all datasets.

OKVQA and AdVQA. This could be because these datasets are more challenging, covering out-of-distribution and adversarial scenarios, which cause the model predictions to be more diverse and highly incoherent.

In practice, users will need to set thresholds based on their use cases to target some minimum requirements such as False Positive Rates (FPR). In Table 1, we also show how UMPIRE framework's better AUROC performance for **R1** translates to consistently higher True Positive Rates (TPR) given various FPR requirements.

These results demonstrate the robustness and generalization ability of UMPIRE across both in-domain and challenging out-of-distribution (OOD) scenarios. The consistent improvement in all metrics suggests that UMPIRE can better identify uncertain predictions accurately and can be more reliably deployed in real-world VQA applications where high-stakes decisions depend on model uncertainty.

4.2. Uncertainty calibration

Next, we assess whether UMPIRE and benchmarks satisfy **R2a** and **R2b**. Similar to past uncertainty calibration works (Guo et al.), we first sort the task instances in a given task $t \in \mathcal{T}$ by the computed uncertainty metric $u(\mathcal{M}, t)$, and then bin the task instances with each equally-sized bin b_j associated with its highest metric value u_j. We can then compute the expected accuracy of the responses in each bin, $\bar{a}_j = \sum_{t_j \in b_j} a(\mathcal{M}, t_j)/|b_j|$ as an estimation of the expected accuracy of responses in that bin. Given this, we can assess how well-calibrated the various metrics are: (i) as-is by computing the calibration pearson correlation (CPC) which measures how well Eq. (2) is satisfied (**R2a**), and (ii) after min-max scaling with the help of a small unlabeled dev set $\mathcal{D}_v$ by computing the expected calibration error (ECE), which measures how well Eq. (3) is satisfied (**R2b**).

Calibration Pearson Correlation (R2a). We define the calibration Pearson correlation (CPC) score as the correlation between u_j and a_j across all bins. The higher the CPC, the more linearly correlated the metric is to the estimated probability that the MLLM's answer is accurate. As can be seen in Table 2, UMPIRE consistently performs significantly better than benchmarks across all settings and achieves an average CPC of around 0.95 across all datasets. App. A.9 shows the uncertainty-accuracy linear relationship plots for the above CPC results.

Expected Calibration Error (R2b). The strong linear relationship indicated by UMPIRE's CPC score suggests that a simple scaling process would be sufficient to make the UMPIRE metric well-calibrated and satisfy **R2b**. We can evaluate the expected calibration error (ECE)(Guo et al.) of all metrics by first using an unlabeled development set of task instances (5% of the dataset) to perform min-max scaling before computing the ECE. As can be seen in Table 2, UMPIRE achieves a very low ECE on all datasets, and is significantly lower than benchmarks especially for the more challenging OKVQA (out-of-distribution), AdVQA (adversarial) and VQA-RAD (medical/radiology) datasets.

4.3. Selective answering

We consider a realistic scenario where a provider deploying an MLLM for question answering may benefit from selectively abstaining from responding to uncertain queries. An effective uncertainty metric should allow the model to prioritize answering only when it is confident (low uncertainty), improving overall accuracy.

To evaluate this capability, we follow past

Method	VQAv2	OKVQA	AdVQA	MathVista	VQA-RAD	Avg
Neighborhood Consistency	0.925	0.656	0.714	0.338	0.499	0.626
LN-Entropy	0.939	0.778	0.742	0.292	0.507	0.652
Semantic Entropy	0.939	0.765	0.774	0.365	0.545	0.678
EigenScore	0.953	0.791	0.791	0.375	0.584	0.699
Ours	**0.956**	**0.797**	**0.799**	**0.38**	**0.59**	**0.704**

Table 3: Comparison of AURAC across datasets for different uncertainty quantification methods. Our proposed method achieves the highest performance on all datasets.

works (Hüllermeier and Waegeman, 2021; Farquhar et al., 2024) by analying the Rejection-Accuracy curve, which measures the accuracy of the model on the most-confident $X\%$ of task instances, as determined by the uncertainty method under evaluation. A well-performing uncertainty method should yield higher accuracy on the confident subset compared to the excluded subset, with rejection accuracy improving as more uncertain inputs are rejected. Similar to Farquhar et al. (2024), we calculate the Area Under the Rejection-Accuracy Curve (AURAC), which quantifies the total improvement in accuracy across all rejection thresholds $X\%$. The AURAC score approaches 1 as an uncertainty method becomes more precise at detecting likely incorrect responses.

Our UMPIRE metric consistently achieves the highest AURAC for all datasets, shown in Table 3, demonstrating that it provides a more reliable uncertainty estimate, allowing for better decision-making in selective answering scenarios. By effectively identifying uncertain responses, UMPIRE enables the provider to optimize answer acceptance rates while maintaining high accuracy.

4.4. Blackbox Models

While our method requires the semantic embeddings and model-generated probabilities of the MLLM output to compute the UMPIRE metric, we could also apply it to blackbox models where we do not have access to such information, through the use of a whitebox proxy model. To do so, we obtain the text responses of the blackbox model that we are examining, and parse these through a proxy whitebox model (both task instance query and text responses) to get both the semantic embeddings and model-generated probabilities for each response. This then allows us to compute our UMPIRE metric.

To demonstrate this, we evaluated how UMPIRE and baselines perform when they are applied to the Claude 3.5 Haiku (Anthropic, 2024), GPT4o (Hurst et al., 2024), and GPT4o-mini (OpenAI, 2024) models responding to the VQAv2, OKVQA, and AdVQA datasets. We used `Llava-v1.5-13b` (Liu et al., 2023c) as the whitebox proxy model. Baseline methods are also adapted to this setting, including the discrete version for Semantic Entropy and EigenScore with whitebox embedding. We also add Verbalized Confidence (Xiong et al., 2024a), which directly asks the blackbox model to generate the confidence score while answering the question, more details in App. A.1.

As shown in Fig. 2, UMPIRE continues to significantly outperform baselines in blackbox settings with a whitebox proxy model, with large performance gains over the rest of the baselines. This makes our UMPIRE metric widely applicable, especially given its ease of use and speed (**R5**).

4.5. Computational efficiency

Finally, we assess the computational efficiency of the benchmarks (**R5**). All experiments are conducted on a single L40 GPU. A major advantage of our proposed UMPIRE framework is its computationally efficiency, on top of its consistently better empirical performance as described in the above sections. We can see in Sec. 4.4 that UMPIRE takes almost 30% less time at 21.35s per query, compared to Semantic Entropy (30.4s), and it is also training-free, compared to Neighborhood Consistency. This process can be further sped up given recent advances in accelerated parallel LLM batch inference (Kwon et al., 2023; Zhu et al., 2024; Gim et al., 2024).

5. Related work

MLLM-specific methods. Although MLLMs' hallucination and miscalibration problems are well known (Chen et al.; Rohrbach et al., 2018; Bai et al., 2024), research on task instance-specific uncertainty quantification for MLLMs is relatively underdeveloped. Most existing methods violate several of the desiderata in Sec. 2.2, such as those that rely on the use of external reference/entailment models (Zhang et al.; Sun et al., 2023; Liu et al., 2023a) or supervised training of classifiers (Li et al., 2024) (violating **R5**(b)). A common approach is to rely on perturbing input queries and testing the consistency of model responses

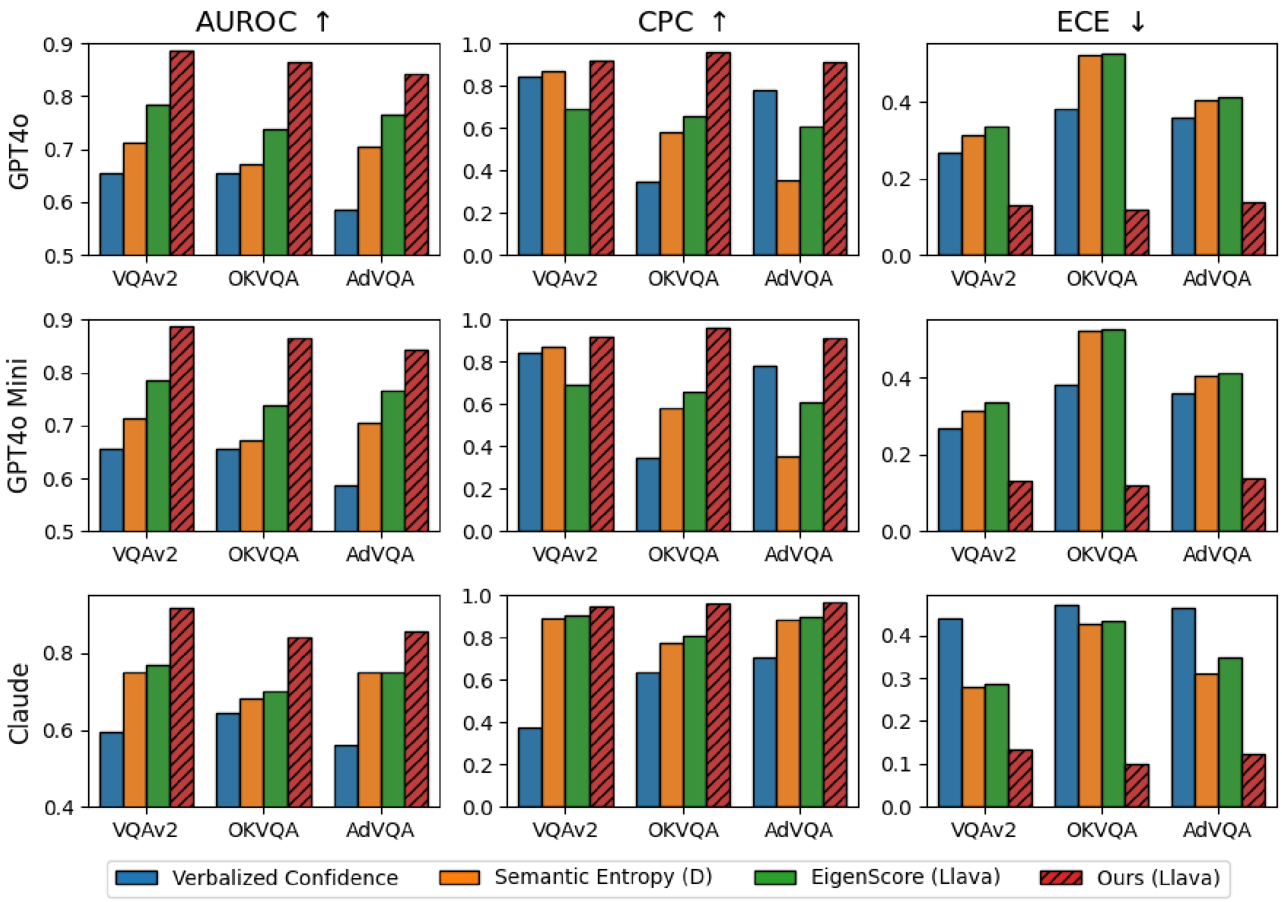

Figure 2: Blackbox results show that our UMPIRE metric outperforms other benchmarks.

Method	Running time (s)
Neighborhood Consistency	30.349
LN-Entropy	**21.351**
Semantic Entropy	30.406
EigenScore	21.353
Ours	**21.351**

Table 4: Comparison of running times (in seconds) per query ($k = 50$) for different uncertainty quantification methods, averaged on 3000 samples of VQAv2 dataset.

Method	**R1**	**R2ab**	**R3**	**R4**	**R5**
Neighborhood Consistency	✗	✗	✓	✗	✗
LN-Entropy	✗	✓	✗	✓	✗
Semantic Entropy	✓	✗	✓	✓	✗
EigenScore	✓	✗	✓	✗	✓
Ours	✓	✓	✓	✓	✓

Table 5: Comparison of uncertainty metrics based on the proposed desiderata.

as an indicator, with works proposing different perturbation approaches (Khan and Fu, 2024b; Zhang et al.). Such approaches tend to require a large number of perturbed samples to perform well (violating **R5**(b)). Even by relaxing the design desiderata by allowing access to external models or more computation time, these methods underperform compared to our proposed method, UMPIRE, and may also not be well-calibrated (violating **R2b**).

LLM uncertainty methods. While not originally developed for MLLMs, existing uncertainty quantification methods (Kuhn et al., 2023; Malinin and Gales, 2021; Chen et al., 2024) for LLMs could possibly be extended to the MLLM setting. In this work, we found that by adapting versions of these approaches to MLLMs, we could sometime achieve even better effectiveness (e.g., for **R1** on classifying task instances) compared to MLLM-specific methods (see Sec. 4). However, these approaches still do not satisfy the desiderata in Sec. 2.2 and underperform UMPIRE. For instance, these methods typically do not consider the coherence of the response with the multimodal input, and hence does not satisfy **R4**, resulting in poorer effectiveness. On the other hand, as a more general framework, we found that UMPIRE also outperform baselines in the text-only LLM setting (see App. A.14). In the literature, there are also commonly-used heuristics for LLM uncertainty quantification based on single responses rather than multiple sampling (Xiong et al., 2024b). While these methods have reduced sampling cost which may be important in some settings, this comes at a trade-off of significantly worse performance – in A.16, we show how using UMPIRE with a small number of samples (e.g. $k = 5$) lead to large performance gaps compared to these methods.

Entropy-based approaches. Majority of MLLM and LLM works rely on computing an entropy measure (Farquhar et al., 2024; Nikitin et al., 2024; Zhang et al.). However, it is unclear how to compare entropy values across different support sets (e.g., distributions defined on two classes v.s. that of five). This makes it hard to use discrete entropy for uncertainty metrics, especially when the support set is determined by external models, e.g., since different entailment models may lead to different clusters. EigenScore (Chen et al., 2024) considers differential entropy which leads to a metric with a similar form as ours, but with key differences and consistently underperforms UMPIRE (see App. A.12). In addition, they typically involve external models to establish pairwise entailment relationships, which incurs significant computational costs and violates **R5**.

6. Conclusion

We present UMPIRE, a novel training-free inference-time method and metric that can be used to approximate the uncertainty associated with MLLM output for each task instance. We proposed a set of clear desiderata that MLLM unlearning metrics should satisfy, analyzed challenges associated with existing approaches, such as entropy-based methods, and empirically showed how UMPIRE consistently outperforms all benchmarks with less computational time.

Our UMPIRE metric requires multiple samples per task instance, which is more costly than methods that depend on only one response even though such sampling could be sped up by batch inference methods and hence may not incur significantly more cost. Even though our method produces better uncertainty estimates, there may be instances where lower computational cost is needed at the expense of poorer estimates. Future work could explore how we can further improve the performance while using a fewer number of generations.

References

Anthropic. The claude 3 model family: Opus, sonnet, haiku. 2024. URL https://api.semanticscholar.org/CorpusID:268232499.

Z. Bai, P. Wang, T. Xiao, T. He, Z. Han, Z. Zhang, and M. Z. Shou. Hallucination of Multimodal Large Language Models: A Survey, Apr. 2024.

E. Bıyık, K. Wang, N. Anari, and D. Sadigh. Batch active learning using determinantal point processes. *arXiv preprint arXiv:1906.07975*, 2019.

C. Chen, K. Liu, Z. Chen, Y. Gu, Y. Wu, M. Tao, Z. Fu, and J. Ye. INSIDE: LLMs' Internal States Retain the Power of Hallucination Detection, Oct. 2024.

Z. Chen, W. Hu, G. He, Z. Deng, g.-i. family=ZHang, given=Zheng, and R. Hong. Unveiling Uncertainty: A Deep Dive into Calibration and Performance of Multimodal Large Language Models. In O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, and S. Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3095–3109. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.208/.

M. Collura, J. De Nardis, V. Alba, and G. Lami. The quantum magic of fermionic gaussian states. *arXiv preprint arXiv:2412.05367*, 2024.

S. Farquhar, J. Kossen, L. Kuhn, and Y. Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, June 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07421-0.

M. Fomicheva, S. Sun, L. Yankovskaya, F. Blain, F. Guzmán, M. Fishel, N. Aletras, V. Chaudhary, and L. Specia. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8: 539–555, 2020.

I. Gim, G. Chen, S.-s. Lee, N. Sarda, A. Khandelwal, and L. Zhong. Prompt Cache: Modular Attention Reuse for Low-Latency Inference, Apr. 2024.

Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On Calibration of Modern Neural Networks. URL http://arxiv.org/abs/1706.04599.

I. Hartsock and G. Rasool. Vision-language models for medical report generation and visual question answering: A review. 7. ISSN 2624-8212. doi: 10.3389/frai.2024.1430984. URL https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2024.1430984/full.

A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

E. Hüllermeier and W. Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. 110(3):457–506, 2021. ISSN 1573-0565. doi: 10.1007/s10994-021-05946-3. URL https://doi.org/10.1007/s10994-021-05946-3.

C. Jiang, H. Xu, M. Dong, J. Chen, W. Ye, M. Yan, Q. Ye, J. Zhang, F. Huang, and S. Zhang. Hallucination Augmented Contrastive Learning for Multimodal Large Language Model, Feb. 2024.

M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

Z. Khan and Y. Fu. Consistency and uncertainty: Identifying unreliable responses from black-box vision-language models for selective visual question answering. (arXiv:2404.10193), Apr. 2024a. URL http://arxiv.org/abs/2404.10193. arXiv:2404.10193 [cs].

Z. Khan and Y. Fu. Consistency and Uncertainty: Identifying Unreliable Responses From Black-Box Vision-Language Models for Selective Visual Question Answering, Apr. 2024b.

L. Kuhn, Y. Gal, and S. Farquhar. Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation, Apr. 2023.

A. Kulesza. Determinantal Point Processes for Machine Learning. 5(2-3):123–286, 2012. ISSN 1935-8237, 1935-8245. doi: 10.1561/2200000044. URL http://www.nowpublishers.com/article/Details/MAL-044.

T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

J. J. Lau, S. Gayen, A. Ben Abacha, and D. Demner-Fushman. A dataset of clinically generated visual questions and answers about radiology images. *Scientific data*, 5(1):1–10, 2018.

S. Lee, J. Youn, H. Kim, et al. Cxr-llava: a multimodal large language model for interpreting chest x-ray images. *European Radiology*, 2025. doi: 10.1007/s00330-024-11339-6. URL https://doi.org/10.1007/s00330-024-11339-6.

J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022. URL https://arxiv.org/abs/2201.12086.

L. Li, J. Lei, Z. Gan, and J. Liu. Adversarial vqa: A new benchmark for evaluating the robustness of vqa models. In *International Conference on Computer Vision (ICCV)*, 2021.

Q. Li, C. Lyu, J. Geng, D. Zhu, M. Panov, and F. Karray. Reference-free Hallucination Detection for Large Vision-Language Models, Aug. 2024.

F. Liu, K. Lin, L. Li, J. Wang, Y. Yacoob, and L. Wang. Aligning large multi-modal model with robust instruction tuning. *arXiv preprint arXiv:2306.14565*, 2023a.

F. Liu, T. Zhu, X. Wu, et al. A medical multimodal large language model for future pandemics. *NPJ Digital Medicine*, 6:226, 2023b. doi: 10.1038/s41746-023-00952-2. URL https://doi.org/10.1038/s41746-023-00952-2.

H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. In *NeurIPS*, 2023c.

H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.

P. Lu, H. Bansal, T. Xia, J. Liu, C. Li, H. Hajishirzi, H. Cheng, K.-W. Chang, M. Galley, and J. Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

A. Malinin and M. Gales. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jN5y-zb5Q7m.

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

A. Nikitin, J. Kossen, Y. Gal, and P. Marttinen. Kernel Language Entropy: Fine-grained Uncertainty Quantification for LLMs from Semantic Similarities, May 2024.

OpenAI. Gpt-4o mini: Advancing cost-efficient intelligence, July 2024. URL https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/.

OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Łukasz Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Łukasz Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H. P. de Oliveira Pinto, Michael, Pokorny, M. Pokrass, V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F. P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J. F. C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J. J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, and B. Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

P. Rajpurkar, R. Jia, and P. Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.

S. Reddy, D. Chen, and C. D. Manning. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019.

N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL http://arxiv.org/abs/1908.10084.

A. Rohrbach, L. A. Hendricks, K. Burns, T. Darrell, and K. Saenko. Object hallucination in image captioning. *arXiv preprint arXiv:1809.02156*, 2018.

Z. Sun, S. Shen, S. Cao, H. Liu, C. Li, Y. Shen, C. Gan, L.-Y. Gui, Y.-X. Wang, Y. Yang, et al. Aligning large multimodal models with factually augmented rlhf. *arXiv preprint arXiv:2309.14525*, 2023.

D. Tian, S. Jiang, L. Zhang, X. Lu, and Y. Xu. The role of large language models in medical image processing: a narrative review. *Quantitative Imaging in Medicine and Surgery*, 14(1):1108–1121, 2024. doi: 10.21037/qims-23-892. URL https://doi.org/10.21037/qims-23-892.

S. Tong, Z. Liu, Y. Zhai, Y. Ma, Y. LeCun, and S. Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9568–9578, 2024.

L. Wang, J. He, S. Li, N. Liu, and E.-P. Lim. Mitigating fine-grained hallucination by fine-tuning large vision-language models with caption rewrites. In *International Conference on Multimedia Modeling*, pages 32–45. Springer, 2024.

M. Xiong, Z. Hu, X. Lu, Y. Li, J. Fu, J. He, and B. Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms, 2024a. URL https://arxiv.org/abs/2306.13063.

M. Xiong, A. Santilli, M. Kirchhof, A. Golinski, and S. Williamson. Efficient and effective uncertainty quantification for llms. In *Neurips Safe Generative AI Workshop 2024*, 2024b.

Q. Yu, J. Li, L. Wei, L. Pang, W. Ye, B. Qin, S. Tang, Q. Tian, and Y. Zhuang. Hallucidoctor: Mitigating hallucinatory toxicity in visual instruction data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12944–12953, 2024.

Z. Yue, L. Zhang, and Q. Jin. Less is more: Mitigating multimodal hallucination from an eos decision perspective. *arXiv preprint arXiv:2402.14545*, 2024.

B. Zhai, S. Yang, C. Xu, S. Shen, K. Keutzer, and M. Li. Halle-switch: Controlling object hallucination in large vision language models. *arXiv e-prints*, pages arXiv–2310, 2023.

R. Zhang, H. Zhang, and Z. Zheng. VL-Uncertainty: Detecting Hallucination in Large Vision-Language Model via Uncertainty Estimation. URL http://arxiv.org/abs/2411.11919.

H. Zhu, B. Zhu, and J. Jiao. Efficient Prompt Caching via Embedding Similarity, Feb. 2024.

A. Appendix

A.1. Benchmarks

Datasets For our experiments, we utilize a diverse set of general visual question-answering benchmark datasets to ensure a comprehensive evaluation across different scenarios. Specifically, we use VQAv2 (Goyal et al., 2017), OKVQA (Marino et al., 2019), and AdVQA (Li et al., 2021), which include challenging cases such as out-of-distribution and adversarial settings. Besides, we also try to use the domain-specific VQA datasets, including VQA-RAD, a dataset of question-answer pairs on radiology images, and MathVista, a consolidated Mathematical reasoning benchmark within Visual contexts. We evaluate our method using the first 15,000 samples from the validation split of VQAv2, along with the full validation sets of OKVQA (5,000 samples) and AdVQA (10,000 samples), the test split of VQA-RAD (450 samples), and MathVista (testmini split - 1,000 samples). These datasets provide a robust test bed for assessing the effectiveness of our approach across different types of VQA tasks.

Baselines The details of each baseline are as follows.

- **Neighborhood Consistency (Khan and Fu, 2024b).** This method tries to examine the reliability of the model via the consistency of the model's responses over the visual rephrased questions generated by a small proxy Visual Question Generation (VQG) model. We implement this method by training `BLIP` (Li et al., 2022) as the VQG model with its default setting. To ensure a fair comparison, we use `Llava-v1.5-13b` as the VQA model, aligning with the model used in our experiments.

- **Length-normalized Entropy (LN-Entropy) (Malinin and Gales, 2021).** This approach normalizes the joint log-probability of each sequence by dividing it by the sequence length and is proposed by (Malinin and Gales, 2021) for uncertainty quantification in LLM. Following (Kuhn et al., 2023), we also apply multinomial sampling instead of using an ensemble of models.

- **Semantic Entropy (Kuhn et al., 2023).** This method introduces a concept of semantic entropy, which measures the uncertainty over different meanings. Following their algorithms, we try to cluster the generated sequences by `Deberta` as the text entailment model and then compute the entropy based on these clusters.

- **EigenScore (Chen et al., 2024).** We follow their default settings and compute the log determinant of the covariance matrix by Eigenvalues via Singular Value Decomposition (SVD), with the exception of the jitter term value – we found that using a jitter term of 10^{-8} rather than their default setting of 10^{-3} improves their performance, hence we applied that and reported the improved performance.

- **Verbalized Confidence (Xiong et al., 2024a).** This method is applied specifically to blackbox models where we instruct it to provide a measure of its own confidence. For a single instance, we sample generations n times and return the most frequent answer along with the average confidence.

Experimental settings

- **Models and parameters.** We primarily use `LLaVA-v1.5-13B` as our MLLM, with further analysis on other models provided in App. A.10. Following past work (Kuhn et al., 2023), for each image-question pair t, the MLLM generates the most likely answer using a low-temperature setting ($T = 0.01$) and we use this answer $\hat{y}_t$ to evaluate the correctness of the model when answering this pair. For the computation of the various uncertainty metrics that require multiple samples, we apply Monte Carlo sampling to generate n samples from the MLLM using $T = 1$ and top_p $= 0.9$. In the main paper, we use the number of generated samples $n = 50$, and ablation results on the impact of this hyperparameter are presented and discussed in App. A.5.

- **Evaluation.** We use ROUGE-L and exact match as the evaluation functions $a(\mathcal{M}, t^*)$, given the model answer $\hat{y}_t$ and ground truth answer y_t^*, to assess the model performance. In the main paper, we report results using exact match, while additional results with ROUGE-L with varying parameters can be found in App. A.6.

- **Blackbox APIs.** For OpenAI's GPT models, we used $n = 50$ generations per prompt. For Anthropic's Claude 3.5 Haiku model, we used the same model parameters as specified above but a smaller number of generations $n = 20$ due to limitations on API credits.

Metric	Method	(1) With Image	(2) No Image	Difference
AUROC ↑	LN-Entropy	0.783	0.634	*0.149*
	Semantic Entropy	0.836	0.805	*0.031*
	EigenScore	0.863	0.864	0.001
	Ours	0.876	0.805	*0.071*
ECE ↓	LN-Entropy	0.177	0.311	*0.134*
	Semantic Entropy	0.038	0.040	0.002
	EigenScore	0.035	0.034	0.001
	Ours	0.027	0.102	*0.075*

Table 6: Performance variations when we compute uncertainty metrics with (1) with the image portion of the task query, and (2) without it. The performance of LN-Entropy and UMPIRE improves under (1), demonstrating how it considers multimodal input information and satisfy **R4**. In contrast, EigenScore's performance remains the same in both settings, indicating that it violates **R4**.

A.2. Assessing multimodal query input coherence (R4)

For a metric to satisfy **R4**, it should consider the coherence of each sampled response with respect to the multimodal task instance query, rather than just a single modality (e.g., text). We design an experimental setting to assess this by computing uncertainty metrics based on (1) both image and text portions of the query ($t = (I_t, q_t)$) with the MLLM text response $\hat{y}_t$, and (2) only the text portion of the query (only q_t) with the MLLM text response $\hat{y}_t$. A metric that satisfies **R4** should perform significantly better under (1), while a metric that does not will produce similar performance regardless of (1) and (2).

Specifically, for (2), after the MLLM has generated responses $\hat{y}_t$ based on (I_t, q_t), we recompute the various metrics LN-Entropy, Semantic Entropy, EigenScore, and UMPIRE based on the query-answer pair without the image, e.g., based on recomputing the response logits and embedding vectors of text-only query-answer pairs $[q_t, \hat{y}_t]$, on a subset of the VQAv2 validation set. In Table 6, we observe that LN-Entropy and UMPIRE, and to a smaller extent Semantic Entropy, are sensitive to the lack of multi-modality information, with their performance increasing once the image queries are provided during the computation of the metrics. On the other hand, EigenScore is insensitive to whether the image query is provided or not. This may be because EigenScore measures only the diversity of responses through the covariance matrix of text response sentence embeddings across multiple generations, which is not affected by the image query bias. On the contrary, logit signals are more sensitive to the coherence of the multimodal input query and the generated response, hence metrics that use some form of that such as LN-Entropy and UMPIRE can better satisfy **R4**.

A.3. Weighting parameter $\tilde{\alpha}$

As mentioned in **U3** in Sec. 3.2, the incoherence score in UMPIRE has a scaling hyperparameter α that is related to the hyperparameter $\tilde{\alpha} = 2k\alpha$ that controls the balance between two terms in Eq. (5): the unadjusted semantic volume metric and the expectation value of the model generated probabilities of getting the task instances wrong. In our experiments, we did not tune the hyperparameter based on a labeled development set but instead set $\tilde{\alpha}$ such that both terms have the same expected contribution (e.g. based on an unlabeled sample of task instances). However, in practice, users could potentially search for a better hyperparameter value for their task, such as via grid search or AutoML methods like Bayesian Optimization.

In Fig. 3 (a), we provide an illustration of a plot of AUROC v.s. $\tilde{\alpha}$ values from tuning $\tilde{\alpha}$ for the AdVQA dataset, based on a development set consisting of randomly sampled 10% of the full dataset. Note that while using grid search would yield a higher AUROC (green dot), the 'adaptive alpha' approach of setting $\tilde{\alpha}$ to balance both terms in Eq. (5) will not be very far off from the optimum. In addition, an alpha value of 0 has a significantly lower performance, indicating that the incoherence score contributes to the good performance of UMPIRE.

A.4. Incoherence-adjusted semantic volume metric

In this section, we provide the explicit derivation of how our incoherence-adjusted semantic volume metric can be simplified to a weighted sum of two terms in Eq. (5), an easily interpretable form.

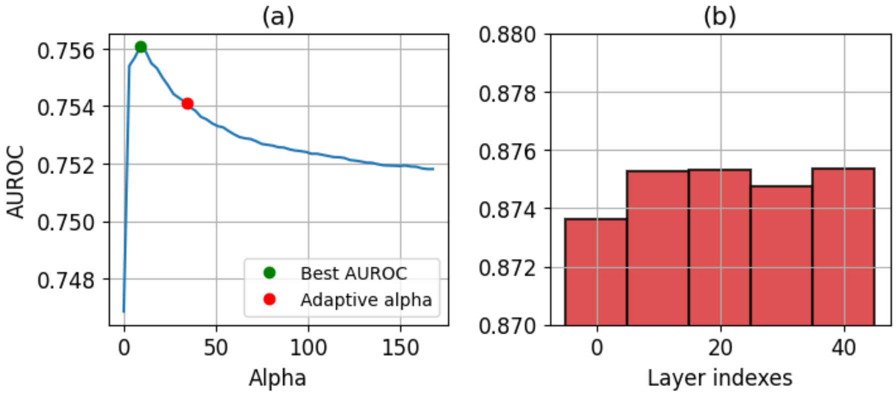

Figure 3: (a) Tuning of the weighting parameter $\tilde{\alpha}$ with respect to AUROC on the small development set (10%) of AdVQA. The 'adaptive alpha' value set without the need for hyperparameter tuning produces good performance. $\tilde{\alpha} = 0$ is suboptimal, reflecting the importance of the incoherence scores. (b) Ablation study on choosing the layer index to extract embedding vectors. Results show that different layer indices only have slight variations in the AUROC performance.

$$
\begin{aligned}
\widetilde{V}_t &= \log\det(C_t\Phi_t\Phi_t^T C_t) && (6)\\
&= \log\det(\Phi_t\Phi_t^T C_t C_t) && (7)\\
&= \log[\det(\Phi_t\Phi_t^T)\det(C_t C_t)] && (8)\\
&= \log\det(\Phi_t\Phi_t^T) + \log\det(C_t C_t) && (9)\\
&= V_t + 2\log\prod_i \exp(\alpha(1-p_i)) && (10)\\
&= V_t + \tilde{\alpha}\mathbb{E}[1-p] && (11)
\end{aligned}
$$

where the derivations involve basic properties of logarithms and determinants, Eq. (10) by defining the adjusted semantic volume term V analogously to Eq. (4) and using the definition of the incoherence score matrix in **U3** , and Eq. (11) noting that the sum is over a Monte-carlo sampling of model responses, with $\tilde{\alpha} = 2k\alpha$ redefined to absorb constants including k which is the number of sampled responses.

A.5. Number of generations analysis

To analyze the impact of the number of generations on the various metrics' performance, we conduct an ablation study by varying the number of generated responses (from 2 to 50) per task instance for a VQAv2 validation subset. As shown in Fig. 4 (a), while increasing the number of generations generally improves AUROC across all methods, UMPIRE achieves higher performance with significantly fewer generations compared to baselines. This indicates that our method is more efficient, requiring fewer samples to reach strong performance, whereas other methods continue to rely on additional generations for improvement. The results highlight the robustness of our approach in capturing correctness signals effectively, even with a limited number of generations.

A.6. Ablation on evaluation parameters

Evaluation function $a(\mathcal{M}, t^*)$ Following the setting in (Kuhn et al., 2023), we further evaluate the performance of our method and baselines under various levels of the ROUGE-L. Fig. 4(b) presents the AUROC scores across different evaluation functions $a(\mathcal{M}, t^*)$ on a subset of the VQAv2 validation set, demonstrating that our method consistently outperforms baseline approaches regardless of the chosen evaluation functions. These results highlight the versatility and robustness of our approach across different correctness evaluation criteria.

Effect of number of bins in ECE and CPC. We also analyzed the effect of the number of bins when computing ECE and CPC by randomly trying on a subset of AdVQA dataset. Fig. 5 illustrates that UMPIRE still achieves the best and consistent

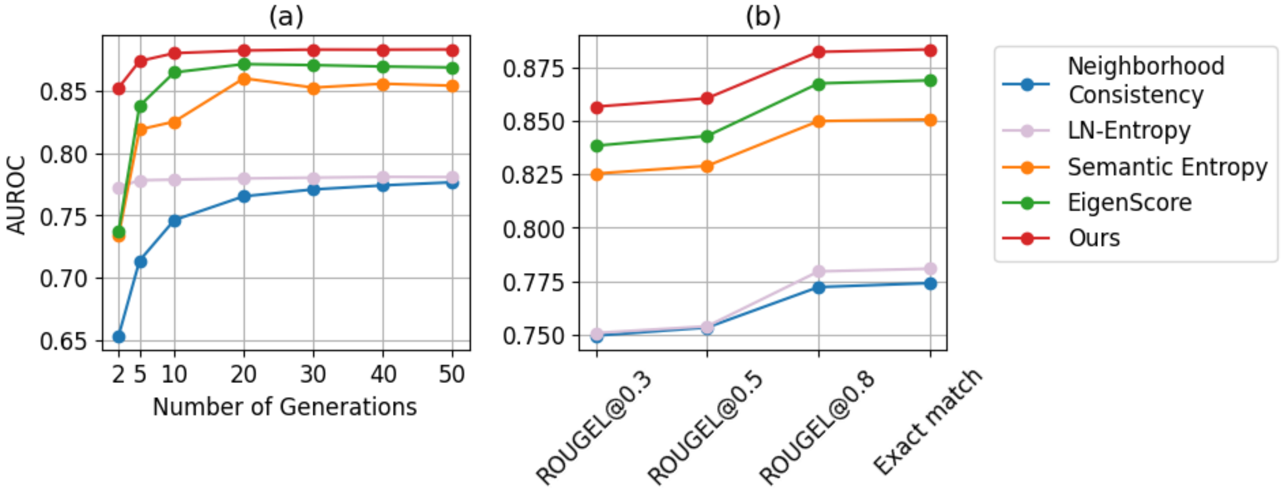

Figure 4: Ablation study on the (a) number of generations for our method and (b) evaluation methods. (a) shows the AUROC performance as the number of generations increases, demonstrating the impact of additional generations. UMPIRE is able to achieve high performance with few generations. (b) consistently outperforms baseline approaches regardless of the chosen evaluation functions.

performance across all bin values.

A.7. Sampling temperature

Besides the number of generations in App. A.5, we analyzed the impact of temperature during the generation process on the evaluation performance. We conducted an ablation study by varying the generation temperature (from 0.25 to 2) on a subset of the VQAv2 validation set. As shown in Fig. 6, the temperature of 1 helps UMPIRE achieve the best performance and outperforms the best performance of other baselines (EigenScore and LN-Entropy).

A.8. Embedding layer selection

We analyzed the impact of the layer index when extracting the embedding vectors by computing the AUROC performance on different embedding matrices extracted from different layer indices. As shown in Fig. 3 (b), the change in the layer indices makes the AUROC vary slightly. The last layer still yields the best performance, so we adopt it for all of our experiments.

A.9. Plots for calibration R2a

To better visualize the performance of the various metrics for proportionality **R2a**, we plot the accuracy v.s. uncertainty score u on the AdVQA validation set in Fig. 7. UMPIRE manages to achieve the strongest linear correlation with accuracy compared to all other metrics. This satisfies the desiderata of **R2a**.

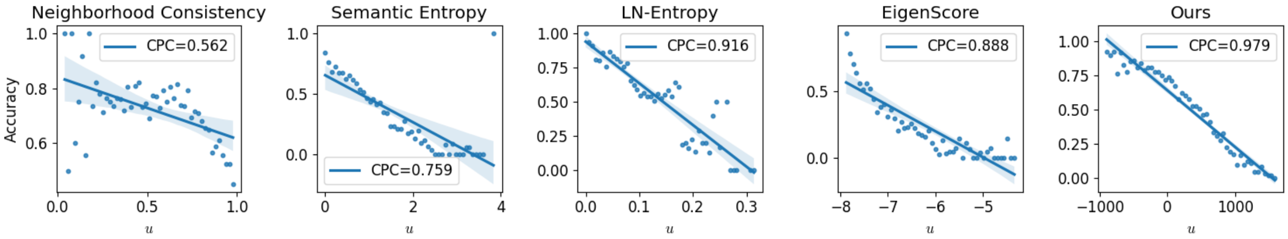

Figure 7: Pearson correlation of the uncertainty score u on the AdVQA validation set that demonstrates UMPIRE's strong correlation compared to other metrics.

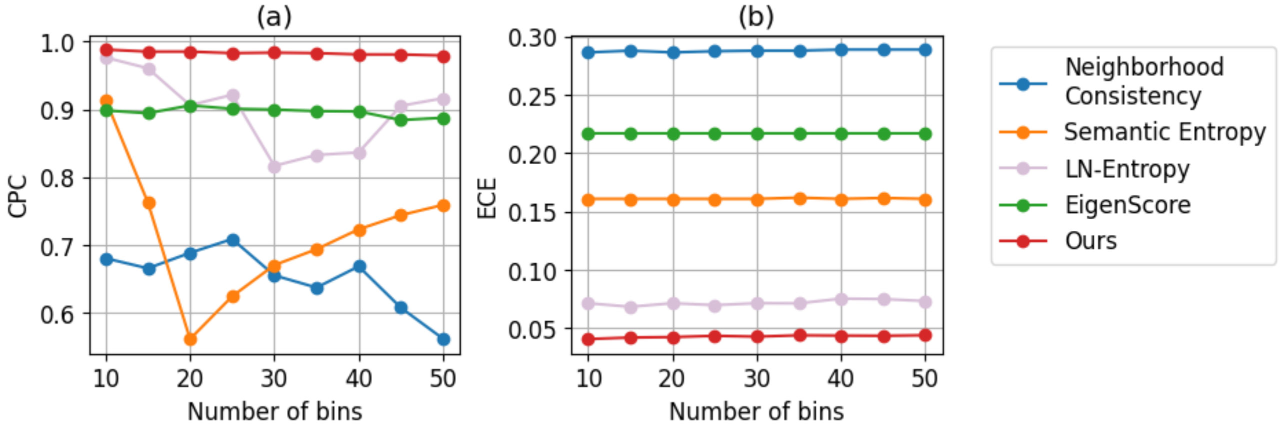

Figure 5: Results for the effect of number of bins on (a) CPC and (b) ECE. Both measures show that UMPIRE consistently outperforms baselines.

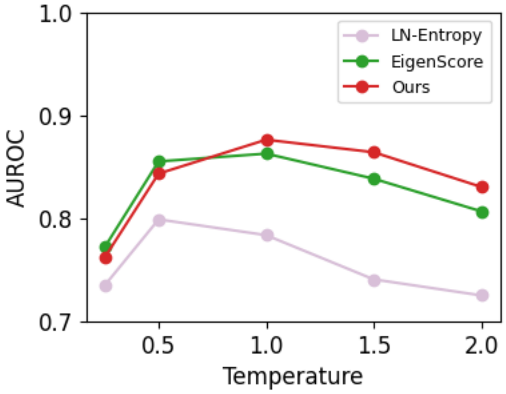

Figure 6: Impact of temperature during the generation process on evaluation performance.

A.10. Model sizes and families analysis

We analyze the impact of model size and architecture family on evaluation performance by comparing different models across various sizes and families on a subset of the VQAv2 validation set. As shown in Fig. 8, we observe a slight increase in AUROC as the model size increases within the same family. This suggests that larger models tend to generate more informative and reliable outputs. Additionally, our method consistently outperforms baselines across all tested models, demonstrating its robustness regardless of model size or architecture. These findings highlight that while larger models can enhance performance, our approach remains effective across different model scales and families.

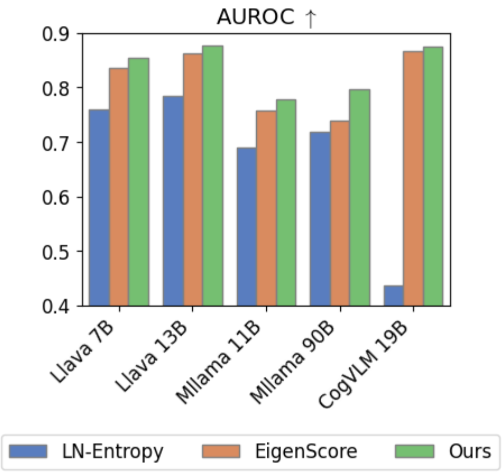

Figure 8: Ablation study across different models, evaluating AUROC performance for LN-Entropy, EigenScore, and UMPIRE. The results indicate that UMPIRE consistently achieves higher AUROC across various models, including `LLaVA-7B, LLaVA-13B, Mllama-11B, Mllama-90B`, and `CogVLM2-19B`. This highlights the robustness and effectiveness of our approach across different model architectures.

A.11. Prompts

Following Liu et al. (2023c), we use the following prompt for all baseline tasks:

```
<image>. Answer this question in a word or a phrase. {question}
```

The prompt used to elicit out verbalized confidence from the blackbox API models are slightly different, such that they output their confidence in the answer along with the response. In accordance with Xiong et al. (2024a), we use the following prompt to extract verbalized confidence:

```
<image>. Read the question, provide your answer and your confidence in this
answer. Note: The confidence indicates how likely you think your answer is true.
Use the following format to answer: \‘Answer and Confidence (0-100): [ONLY a
word or a phrase; not a complete sentence], [Your confidence level, please only
include the numerical number in the range of 0-100]%"’ Only give me the reply
according to this format, don’t give me any other words. Now, please answer this
question and provide your confidence level. Question: {question}
```

A.12. Comparisons with Eigenscore

As mentioned in App. A.1, the EigenScore (Chen et al., 2024) metric involves computing the log determinant of the covariance matrix of sampled sentence embeddings. At first glance, this metric may seem similar to that of UMPIRE. However, there are key differences that lead to EigenScore consistently underperforming our proposed UMPIRE metric as can be seen in both the MLLM (Sec. 4) and LLM case (App. A.14), and EigenScore could be interpreted as a special case of UMPIRE.

A major distinction, among others, is that Chen et al. (2024) analyzed only the LLM setting, and proposed EigenScore by considering the differential entropy of sentence embeddings, assuming that the embeddings form a multivariate Gaussian distribution – this motivated the log determinant term of the metric which bears similarity to UMPIRE. However, our UMPIRE framework considers the more general MLLM setting, and adopts a different approach inspired by determinantal point processes (DPP), which naturally factors the incoherence scores when computing the UMPIRE metric to adjust the semantic volume enclosed by the responses' semantic embeddings. This inclusion of the incoherence scores help (1) satisfy **R4**, as we can see in App. A.2 that EigenScore does not, and (2) significantly improve metric performance (App. A.3). EigenScore could possibly be interpreted as the special case of UMPIRE where all responses have incoherence scores of 1 (i.e., the model-generated probabilities of all responses $p_i = 1 \; \forall i$). Note that the incoherence scores also boost performance in the LLM setting (App. A.14), indicating that while incoherence scores help in addressing App. A.2, its weighting of different responses in the computation of UMPIRE also helps in single modality settings.

A.13. Additional discussion on the effectiveness desiderata

In this section, we provide further discussion on the various effectiveness desiderata, such as the differences and relevance of **R1**, **R2a** and **R2b**. For ease of discussion, we focus on comparing **R1** and **R2b**, which is a stricter form of **R2a**.

The classification desiderata **R1** and the calibration desiderata **R2b** are primarily motivated by different considerations. In the former, we are concerned about classifying whether a task instance t will be answered correctly or not by the MLLM. As represented in Eq. (1), for this desiderata the metric should be able to successfully rank the task instances that the MLLM will get wrong higher than those that it will get correct, which can be evaluated by the AUROC of the metric. Such evaluations are used in many MLLM and LLM uncertainty quantification works (Farquhar et al., 2024; Malinin and Gales, 2021; Chen et al., 2024; Xiong et al., 2024a) to assess the performance of their metrics. While useful, note that the desiderata does not consider a quantitative, continuous measure of the uncertainty associated with each task response, since classification of correct/wrong responses is a binary task.

However, in the latter, we are concerned about providing an accurate, calibrated estimate of whether the MLLM will get a task instance correct, conditional on the uncertainty metric (as in Eq. (3)), which can be evaluated via the expected calibration error (ECE). Note that in this scenario, we are not concerned about classifying whether a task instance will be answered correctly (**R1**), but instead are focused on being accurate about the *probability* that a task instance will be answered correctly given an associated metric value.

To illustrate the difference, consider an extreme example where an MLLM will definitely get 50% of the task instances correct, and the rest wrong. The vacuous metric that assigns the same uncertainty score to all task instances might satisfy **R2b** since it will output the average accuracy, 0.5, as the score for all task instances. This metric would violate **R1** and fail to classify the correct from wrong task instances. Instead, a better metric might strive to assign 1 to all task instances that can be answered correctly and 0 to the rest, satisfying both **R1** and **R2b**.

In practice, we would likely not have perfect information prior to evaluation on whether a task instance will be correct or wrong. That is why for **R1** the goal is only for the metric to get as close to 1 as possible, as the best possible AUROC would depend on the model and task. However, given two metrics that can achieve the same AUROC, a poor metric might only obtain the right relative ranking of task instances, while a good metric would not only achieve the same AUROC but also provide calibrated probabilities on how likely a task instance would be answered correctly or not. Hence, both the **R1** and **R2b** should be considered when evaluating uncertainty metrics, as we described in Sec. 2.2. In the absence of a small development set of unlabeled task instances before deployment, metrics satisfying **R2a** would at least provide interpretable relative information regarding how likely a task instance would be answered correctly compared to another.

A.14. Single modality experiment

We also tested our UMPIRE metric on another modality, purely textual datasets. To generate the embeddings and answers, we used the `Llama-3.1-8B-Instruct` model (Grattafiori et al., 2024) instead of MLLMs. The datasets tested include Conversational Question Answering (CoQA) (Reddy et al., 2019), TriviaQA (Joshi et al., 2017), Natural Questions (NQ) (Kwiatkowski et al., 2019) and Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2018). We performed tuning of the weighting parameter $\tilde{\alpha}$ for each dataset.

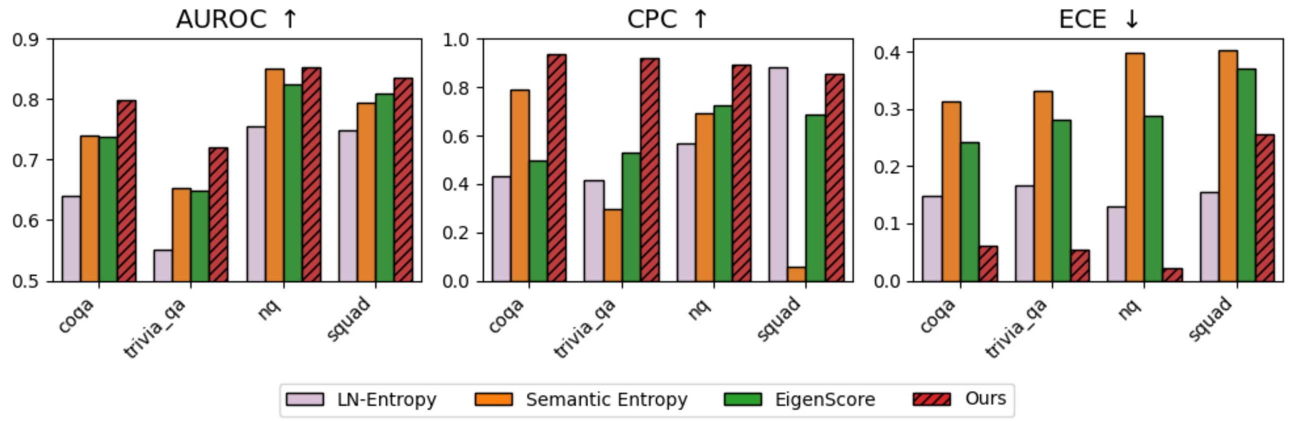

Figure 9: Performance comparison of different uncertainty quantification methods across LLM tasks. The metrics include AUROC (higher is better), CPC (higher is better), and ECE (lower is better).

As shown in Fig. 9, UMPIRE managed to outperform other metrics in most cases, except for the CPC score on the SQuAD dataset, where LN-Entropy performs slightly better. Thus, as noted in Sec. 2.1, our method is not reliant on modality-specific characteristics when computing the metric, as it is capable of working well in textual tasks of single modality. UMPIRE is a general framework that can also perform well in the special case of text-only LLM settings.

A.15. Length-normalized effect

In prior work on uncertainty estimation and related scoring functions, length normalization has often been applied to adjust for biases introduced by varying response lengths (Kuhn et al., 2023; Malinin and Gales, 2021). Motivated by this, we explored whether length normalization could also benefit the quality term of UMPIRE, i.e, length-normalized incoherence score. Empirically, as shown in Table 7, we observed that applying length normalization does not consistently improve performance across most MLLM benchmarks. In fact, the normalized variant frequently underperforms in terms of AUROC and CPC, and yields better ECE in several datasets. However, in the pure LLM setting as seen in App. A.14, length normalization appears to offer some advantages (see Table 8), suggesting that its effectiveness may be setting-dependent.

Metric	Method	VQAv2	AdVQA	OKVQA	MathVista	VQA-Rad
AUROC ↑	Without Length Normalized	**0.882**	**0.787**	0.755	0.822	**0.802**
	Length Normalized	0.875	0.779	**0.756**	**0.825**	0.792
CPC ↑	Without Length Normalized	0.946	**0.979**	**0.966**	**0.945**	0.908
	Length Normalized	**0.986**	0.978	0.946	0.936	**0.935**
ECE ↓	Without Length Normalized	**0.038**	0.042	0.036	0.071	**0.067**
	Length Normalized	0.062	**0.019**	**0.034**	**0.056**	0.068

Table 7: Comparison of UMPIRE with and without length normalization across various VQA datasets.

A.16. Single sampling method

We have also run experiments on basic uncertainty metrics that use only a single MLLM response, rather than a sampled set of MLLM responses. We ran the single-sample methods listed in Xiong et al. (2024b): Sequence Probability, Mean Token Entropy (Fomicheva et al., 2020), and Perplexity.

Table 9 shows these methods' results for the various MLLM datasets, along with UMPIRE, based on five response generations. Note that while the single-sampled methods may be cheaper to compute, they also produce significantly worse performance results compared to UMPIRE with $k = 5$. The appropriate metric to use would depend on the application

Metric	Method	CoQA	TriviaQA	NQ	SQuAD
AUROC ↑	Without Length Normalized	0.749	0.641	0.844	0.813
	Length Normalized	**0.799**	**0.720**	**0.853**	**0.836**
CPC ↑	Without Length Normalized	0.876	0.850	0.780	**0.888**
	Length Normalized	**0.937**	**0.923**	**0.892**	0.855
ECE ↓	Without Length Normalized	0.068	0.098	0.076	**0.117**
	Length Normalized	**0.061**	**0.054**	**0.022**	0.257

Table 8: Comparison of UMPIRE with and without length normalization across various text datasets.

requirements. For settings that require better uncertainty metric performance, UMPIRE would likely be a good choice especially since accelerated batched response generation (Kwon et al., 2023) is fast and typically not a computational resource bottleneck, while single-sample methods may be more suitable for very time-sensitive applications.

Dataset	Method	AUROC ↑	ECE ↓	CPC ↑
VQAv2	Single Prob	0.632	0.121	0.374
	Mean Token Entropy	0.628	0.129	0.046
	Perplexity	0.629	0.131	0.125
	Ours (k=5)	**0.873**	**0.067**	**0.923**
ADVQA	Single Prob	0.595	0.303	0.372
	Mean Token Entropy	0.590	0.302	0.170
	Perplexity	0.592	0.336	0.151
	Ours (k=5)	**0.774**	**0.055**	**0.959**
OKVQA	Single Prob	0.581	0.304	0.463
	Mean Token Entropy	0.580	0.303	0.039
	Perplexity	0.581	0.335	0.225
	Ours (k=5)	**0.740**	**0.097**	**0.944**
MathVista	SingleProb	0.628	0.539	0.322
	Mean Token Entropy	0.606	0.601	0.334
	Perplexity	0.616	0.643	0.224
	Ours (k=5)	**0.791**	**0.087**	**0.706**
VQA-RAD	Single Prob	0.540	0.525	0.140
	Mean Token Entropy	0.535	0.534	0.118
	Perplexity	0.537	0.550	0.168
	Ours (k=5)	**0.806**	**0.090**	**0.828**

Table 9: Comparison of the performance of single sampling methods and UMPIRE across various VQA datasets.