UNCERTAINTY QUANTIFICATION FOR MLLMS

Gregory Kang Ruey Lau[*†‡], **Hieu Dao**[*†], **Bryan Kian Hsiang Low**[†]
[†]Department of Computer Science, National University of Singapore, Singapore 117417
[‡]CNRS@CREATE, 1 Create Way, #08-01 Create Tower, Singapore 138602
{greglau,daohieu,lowkh}@comp.nus.edu.sg

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 and adversarial 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 & 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., 2023b; 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 e.g., 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. Such 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., 2023a; 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

[*]Equal contribution.

on either external verifiers (Liu et al., 2023b; Sun et al., 2023) or methods that involve relatively expensive computation (Zhang et al.; Khan & Fu, 2024) 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 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 (section 2.2) and analyzed challenges associated with existing approaches such as entropy-based methods (section 3.1), (2) proposed a novel MLLM uncertainty method and metric (section 3), and (4) empirical show how UMPIRE consistently outperforms all benchmarks with less computational time (section 4).

2 Problem formulation and desiderata

2.1 Problem formulation

We consider the setting where we have an open-source 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., $\mathcal{M}(I, q) \coloneqq p_{\mathcal{M}}(y|I, q) = p_{\mathcal{M}}(w_1|I, q)p_{\mathcal{M}}(w_2|I, q, w_1) \dots p_{\mathcal{M}}(w_n|I, q, w_{1:n-1})$.

We can apply the MLLM to multi-modal tasks $\mathcal{T}$ with task instances $t \in \mathcal{T}$, where $t \coloneqq (I_t, q_t)$ represents the input query containing both an image portion I_t and text portion q_t, and for clarity we explicitly denote $t^* \coloneqq (t; y_t^*)$ as task instances with known text ground truth output y_t^*. The MLLMs' response $\hat{y_t}$ to the task can then be sampled autoregressively from $\mathcal{M}(I_t, q_t)$, and its performance on the task instance can be evaluated by whether the response matches the ground truth, i.e. $a(\mathcal{M}, t^*) \coloneqq \mathbb{I}\{\hat{y_t} = y_t^*\}$, 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 task instances, i.e., $a(\mathcal{M}, \mathcal{T}) \coloneqq \mathbb{E}_{t \in \mathcal{T}} a(\mathcal{M}, t)$, 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 satisfy several key desiderata. First, the metric should be effective in approximating the uncertainty associated with each response. We assess this on two aspects:

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

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

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.

[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.

R2 Calibration. If provided a small dev set of labeled task instances $\mathcal{D}_v = \{t^*\}$, the metric u could be easily scaled to $\tilde{u} \in [0, 1]$ (e.g., using min-max scaling) such that it is well calibrated (Guo et al.), i.e.,

$$\mathbb{P}(a(\mathcal{M}, t^*) = 1 \mid \tilde{u}(\mathcal{M}, t) = p) \approx p, \quad \forall p \in [0, 1]. \tag{2}$$

Metrics satisfying this desiderata will reflect the extent of how uncertain a model is for a given task response, rather than just provide a binary classification given a threshold based on **R1**.

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

R3 Focus on semantics. The metric should depend on diversity in the semantic meaning of the responses, rather than 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 Multi-scale variations. The metric should be capable of quantifying and comparing across a wide range of semantic variation scales. Depending on the task and specific task instances, sampled MLLM responses could differ only in small nuances or convey very different meanings, and the metric would need to compare across them.

R5 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.

R6 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 requirements for other external pre-trained models or separately trained reward models as they incur additional costs and may not be feasible for some inference pipelines.

3 METHOD

3.1 CHALLENGES FACED BY EXISTING METHODS

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 of the existing methods will violate several of the desiderata in section 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., 2023b) or supervised training of classifiers (Li et al., 2024), (violating **R6**). A common approach is to rely on perturbing input queries and testing for the consistency of model responses as an indicator, with works proposing different perturbation approaches (Khan & Fu, 2024; Zhang et al.). However, such approaches tend to require a large number of perturbed samples to perform well. Even with a relaxation of the design desiderata by allowing them access to external models or more computation time, these methods underperform compared to our proposed method, UMPIRE (e.g., see empirical results in section 4), and may also not be well-calibrated (violating **R2**).

LLM uncertainty methods. While not originally developed for MLLMs, existing uncertainty quantification methods 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 section 4). However, these approaches still do not satisfy the desiderata in section 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 **R5**, resulting in poorer effectiveness.

Problems with entropy-based approaches. For both MLLM and LLM-specific existing works, a majority of these methods rely on computing some form of entropy measure. However, these approaches face several key challenges. First, entropy values are difficult to compare when they have different support sets (e.g., distributions defined on 2 discrete classes cannot be readily compared with those defined on 5 classes). This makes it hard to define entropy metrics that can be used to compare uncertainty across different task instances (violating **R4**) without assumptions that may not

hold in practice (e.g., assumptions that responses follow a Gaussian distribution in high dimensional semantic space and that such differential entropy can be reliably estimated(Chen et al., 2024)[2]). Furthermore, while there are works satisfying **R3** by computing some version of entropy in semantic space Farquhar et al. (2024); Nikitin et al. (2024); Zhang et al., these methods are typically sensitive to how the sampled responses are clustered (e.g., the number of clusters, model or algorithm for clustering) and may not consider the magnitude of the semantic differences among responses within clusters, hence violating **R4**. In addition, they typically involve external models to establish pairwise entailment relationships, which incurs significant computational costs and violates **R6**.

3.2 Overview of UMPIRE

To meet these design considerations, our proposed framework and metric draws inspiration from quantum physics and active learning research that have modeled systems of negative correlations and characterized sample diversity with determinantal point processes (DPP) (Kulesza, 2012).

Semantic volume. In our context, we posit that MLLMs, in the absence of strong anchor queries that they have a certain response to, will tend to generate a diverse set of responses when sampled. Hence, intuitively, given the input query of a task instance (I, q), the more diverse the set of responses that the MLLM produces when sampled, the more uncertain the MLLM is in its response to that task instance. We quantify this by computing the semantic volume enclosed by the response samples mapped on the model's embedding space. This explicitly takes into account the meaningful coverage and diversity of the model's responses (**R4**), while focusing on semantic dissimilarity rather than lexical variations (e.g., paraphrasing of the same response) which provides a better indication of model confidence for most tasks (**R3**), and avoids problems on semantic entropy-based techniques which we will elaborate on in section 3.3.

Implicit incoherence scores. In addition, during the generation process, the MLLM produces useful information on how coherent it assesses its individual responses are, conditional on the multimodal image and text queries (**R5**). While not a calibrated metric on its own, the aggregated logits of each MLLM response contains such information, and we use them to compute our proposed incoherence scores for each sampled MLLM response. These incoherence scores will be used to adjust the semantic volume term, similar to how sample quality metrics are used in DPP works, as we elaborate in section 3.4.

Computation framework. Putting everything together, our UMPIRE framework, which is surprisingly simple but effective, provides an uncertainty indicator for a given task with just a single batched inference forward pass without additional training (**R6**), via the following key steps:

1. We generate several responses from the MLLM via stochastic sampling. This process can be done efficiently via accelerated batch inference methods (Kwon et al., 2023a).
2. For each response we extract (a) the embedding of the response, represented by the last hidden layer vector of the last response token (before the `EOS` token), and (b) the associated incoherence score of that response, computed from product of the probabilities associated with the generation, from the model.
3. We can then compute our incoherence-adjusted semantic volume metric, which can be used as an indicator of how uncertain a model is when providing an answer to the task.

We summarize its implementation in algorithm 1, and provide elaboration in the following sections.

3.3 Semantic Volume

To compute the semantic volume for a set of MLLM responses, we first map each response to a normalized continuous vector $s \in \mathbb{R}^p$ in a p-dimensional semantic embedding space. While this

[2]Coincidentally, following this assumption leads to a metric that shares a similar form to one of the terms in our UMPIRE metric (unadjusted semantic volume). However, as explained in section 3.4, our framework naturally points to the need for the incoherence scores component, that completes the UMPIRE metric and allowing it to outperform these methods as shown in section 4.

mapping can be done with external embedding models Reimers & Gurevych (2019), the MLLM response generation process itself already computes such a representation, making it computationally efficient to extract. For a given response $\hat{y}_i$, we use the last hidden layer vector of the last response token (before the EOS token) as its embedding representation s_i, which captures the overall semantic meaning of both the preceding input query (I_t, q_t) and the response to it $\hat{y}_i$. Hence, given a sampled set of n MLLM responses $\mathcal{Y}_t^n = \{\hat{y}_i\}_{i=1}^n$ for a given task instance t, we can represent it as an $n \times p$ embedding matrix $R = [s_1, \ldots, s_n]$, where s_i is a $1 \times p$ row vector representing each MLLM response $\hat{y}_i$.

In this embedding space, we can define a kernel $k(s_i, s_j)$ that characterizes the similarity between any two response embeddings s_i and s_j. Given the set of sampled MLLM responses $\mathcal{Y}_t^n$, we can then compute the $n \times n$ Gram matrix $K_{\mathcal{Y}}$, where each element $K_{\mathcal{Y}}(i, j) = k(s_i, s_j)$ describes the similarity between two responses. A simple but effective choice for the kernel is the linear or cosine similarity kernel $K(s_i, s_j) = s_i \cdot s_j^T$, which is commonly used for semantic similarity computation using LLM embedding models (Reimers & Gurevych, 2019) where the embeddings are typically normalized and most semantic information is contained directionally.

Algorithm 1 UMPIRE algorithm

1: **Input:** MLLM model $\mathcal{M}$, task query $t = (I, q)$, number of response samples n, hyperparam α
2: **Output**: Uncertainty metric $\tilde{V}$
3: Sample a set of n model responses $\{\hat{y}\}$, where each response $\hat{y}_i$ consists of its embedding representation s_i (section 3.3) and incoherence score c_i (eq. (4)).
4: Compute coherence-adjusted semantic volume $\tilde{V}$ in eq. (6).
5: **return** $\tilde{V}$

This allows us to compute the semantic volume metric for the set of sampled responses by taking the logarithm (for numerical stability) of the associated Gram matrix determinant[3]:

$$\mathcal{V} = \log\det K = \log\det RR^T. \tag{3}$$

Intuitively, the larger the semantic volume enclosed by the set of responses, the larger the variation in semantic content that the MLLM response spans and hence the more uncertain it is in providing a single response to the query in the task instance.

3.4 Implicit Incoherence scores

However, the semantic volume alone defined in eq. (3) alone does not fully capture all available information regarding the model's uncertainty. A key consideration is also how coherent the MLLM considers each response to be with the task instance query. Through the stochastic response sampling process, the MLLM may generate responses with varying levels of coherence, and it is not optimal to consider all these responses equally when computing an uncertainty metric.

To quantify the coherence of the multi-modal task instance query and the text responses of the MLLM, we first compute the model-generated probability scores for each augmented MLLM response $\tilde{y}_i$, i.e., $p_i = \exp(\sum_j^N l_{i,j})$, where $l_{i,j}$ is the log probability of token j of the augmented response $\tilde{y}_i$ Note that these model-generated scores are not well-calibrated probabilities – they do not reflect the probabilities of whether each sequence is correct or will occur in texts, Hence, we do not use these values directly. As our goal is to compute an uncertainty indicator (the larger the score, the more uncertain), we define the incoherence score as

$$c_i = \exp \alpha(1 - p_i), \tag{4}$$

where α is a scaling hyperparameter that is fixed across instances of a given task, and as explained in section 3.5 could be heuristically set even *without* calibration in cases where there is no development set $\mathcal{D}_v$ and still yield good performance. The incoherence score intuitively captures how uncertain or the degree of incoherence of each response. For example, if the MLLM is very certain of the answer, there will only be one possible sample with $p_i = 1$, leading to $c_i = 1$ which is the smallest possible value. On the other hand, if the MLLM is very uncertain and has a large number of low probability responses, each of its sampled response will likely have a large value of c_i.

[3]We omit a constant factor of 2 for simplicity which does not affect the metric. For our setting, we also have $n < p$ as the semantic embedding space is usually high dimensional.

3.5 INCOHERENCE-ADJUSTED SEMANTIC VOLUME

Given the response-specific incoherence scores and aggregated response set-level semantic volume metric, a natural way to combine these into a single metric would be to scale the response embeddings by the incoherence scores – responses that are rated more incoherent by the MLLM based on eq. (4) are scaled to larger magnitude and will have more influence over the adjusted semantic volume score. Specifically, each response embedding s_i could be scaled by its incoherence score, i.e., $\tilde{s}_i = \exp \alpha(1 - p_i)s_i$, to generate an incoherence-adjusted embedding matrix,

$$\tilde{R} \coloneqq \operatorname{diag}\left[\exp \alpha(1-p)\right]R. \tag{5}$$

The corresponding Gram matrix and semantic volume score in eq. (3) will then be computed based on this adjusted embedding matrix. In the case of the linear kernel we can compute the incoherence-adjusted semantic volume, which surprisingly can be simplified into an easily interpretable form:

$$\tilde{\mathcal{V}} \coloneqq \log\det \tilde{R}\tilde{R}^T = \mathcal{V} + \tilde{\alpha}\mathbb{E}[1-p], \tag{6}$$

where the first term is the unadjusted semantic volume metric, and the expectation in the second term is computed empirically by Monte Carlo sampling of the MLLM responses (see appendix A.2 for derivation). The hyperparameter $\tilde{\alpha} = 2n\alpha$, where n is the number of sampled responses, can be interpreted as balancing the contribution between the two terms, and in practice could be roughly set such that the two terms have the same order of magnitude to avoid the need for fine-tuning while still producing good performance. For a given task $\mathcal{T}$ and a fixed number n of sampled responses, a task instance $t \in \mathcal{T}$ that results in very diverse sampled responses (high semantic volume), and a high expected incoherence score (high average $1 - p_i$ values for each response i) will result in a high metric score indicating high uncertainty.

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 general visual question-answering benchmark datasets such as VQAv2 (Goyal et al., 2017), OKVQA (Marino et al., 2019) and AdVQA (Li et al., 2021) that include various scenarios such as out-of-distribution and adversarial settings. 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 appendix A.5. To benchmark our UMPIRE framework, we considered not only methods from past works on MLLM uncertainty quantification, (1) Neighborhood Consistency (Khan & Fu, 2024), 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 & Gales, 2021), (3) Semantic Entropy (Kuhn et al., 2023), and EigenScore (Chen et al., 2024). More details on benchmarks are in appendix A.1, and additional ablation results are in appendix A.4.

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. Fig. 1 shows the AUROC evaluated over the test dataset for UMPIRE and the benchmarks on the VQAv2, OKVQA and AdVQA datasets. We see that UMPIRE consistently outperforms all other benchmarks, especially the multi-modal specific Neighborhood Consistency method, which faces significant difficulty in the OKVQA dataset that covers out-of-distribution scenarios and the AdVQA that covers adversarial scenarios.

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 our UMPIRE framework's better AUROC performance for **R1** translates to consistently higher True Positive Rates (TPR) given various FPR requirements.

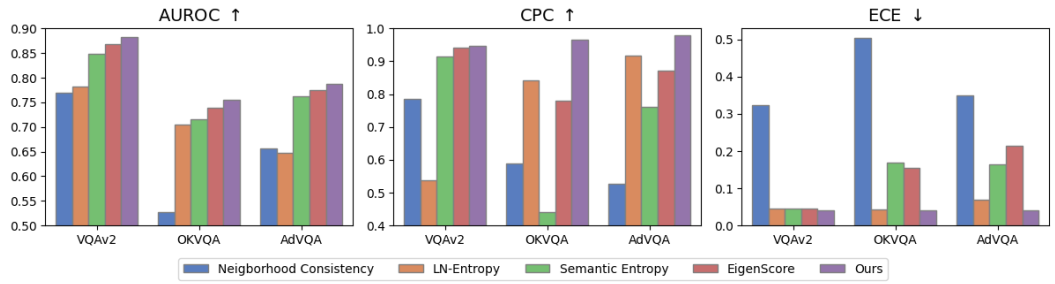

Figure 1: Performance comparison of different uncertainty quantification methods across VQA tasks. The metrics include AUROC (higher is better), CPC (higher is better), and ECE (lower is better). UMPIRE consistently surpasses existing approaches across all datasets.

	TPR at 0.1 FPR			TPR at 0.01 FPR		
	VQAv2	OKVQA	AdVQA	VQAv2	OKVQA	AdVQA
Neighborhood Consistency	0.362	0.095	0.189	0.049	0.008	0.019
LN-Entropy	0.282	0.244	0.168	0.057	0.030	0.066
Semantic Entropy	0.574	0.321	0.420	0.177	0.068	0.124
EigenScore	0.601	0.340	0.466	0.215	0.075	0.172
Ours	**0.629**	**0.368**	**0.477**	**0.230**	**0.091**	**0.185**

Table 1: UMPIRE outperforms all the benchmarks at different FPR levels across all datasets.

4.2 Uncertainty Calibration

Next, we assess whether UMPIRE and benchmarks satisfy **R2**. 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, (1) as-is by computing the calibration pearson correlation (CPC), and (2) after scaling with the help of a small dev set $\mathcal{D}_v$ by computing the expected calibration error (ECE).

Calibration Pearson Correlation. 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 fig. 1, UMPIRE consistently performs significantly better than benchmarks across all settings, achieving a CPC of 0.987, 0.949, and 0.983 on VQAv2, OKVQA, and AdVQA, respectively.

Expected Calibration Error. 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 **R2**. We can evaluate the expected calibration error (ECE)(Guo et al.) of all metrics by first using a development set of task instances (5% of the dataset) to perform min-max scaling before computing the ECE. As can be seen in fig. 1, UMPIRE achieves a very low ECE of around 0.04 on all datasets, and is significantly lower than benchmarks especially for the more challenging OKVQA (out-of-distribution) and AdVQA (adversarial) datasets.

4.3 Selective Answering

We also 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 (have low uncertainty), improving overall accuracy.

	VQAv2	OKVQA	AdVQA
Neighborhood Consistency	0.886	0.629	0.683
LN-Entropy	0.899	0.741	0.705
Semantic Entropy	0.902	0.734	0.742
EigenScore	0.913	0.753	0.753
Ours	**0.916**	**0.761**	**0.761**

Table 2: Comparison of Area under the rejection-accuracy curve (AURAC) across VQAv2, OKVQA, and AdVQA datasets for different uncertainty quantification methods. Our proposed method achieves the highest performance on all datasets

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

Table 3: Comparison of running times (in seconds) per query for different uncertainty quantification methods, averaged on 3000 samples of VQAv2 dataset. UMPIRE achieves the lowest running time, matching the most efficient baseline methods. *Note that Neighborhood Consistency costs extra time to train the VQG model on VQAv2 train split, which we did not include here. All experiments are conducted on a single L40 GPU.

To evaluate this capability, we follow past works Hüllermeier & Waegeman; 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.

As shown in table 2, our proposed method consistently achieves the highest AURAC for all datasets, VQAv2 (0.916), OKVQA (0.761), and AdVQA (0.761). These results demonstrate that our approach provides a more reliable uncertainty estimate, allowing for better decision-making in selective answering scenarios. By effectively identifying uncertain responses, our method enables the provider to optimize answer acceptance rates while maintaining high accuracy.

4.4 Computational efficiency

Finally, we assess the computational efficiency of the benchmarks, i.e., **R6**. 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 table 3 that UMPIRE takes almost 30% less time at 21.35s per query, compared to Semantic Entropy which takes 30.41s. This process can also be further sped up given recent advances in accelerated parallel LLM batch inference (Kwon et al., 2023b; Zhu et al., 2024; Gim et al., 2024).

5 Conclusion

In our work, 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 empirical show how UMPIRE consistently outperforms all benchmarks with less computational time.

ACKNOWLEDGMENTS

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-PhD/2023-01-039J) and is part of the programme DesCartes which is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

REFERENCES

Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. Hallucination of Multimodal Large Language Models: A Survey, April 2024.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. INSIDE: LLMs' Internal States Retain the Power of Hallucination Detection, October 2024.

Zijun Chen, Wenbo Hu, Guande He, Zhijie Deng, given-i=Zh family=ZHang, given=Zheng, and Richang Hong. Unveiling Uncertainty: A Deep Dive into Calibration and Performance of Multimodal Large Language Models. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 3095–3109. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.208/.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin 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.

In Gim, Guojun Chen, Seung-seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. Prompt Cache: Modular Attention Reuse for Low-Latency Inference, April 2024.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi 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.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. URL http://arxiv.org/abs/1706.04599.

Iryna Hartsock and Ghulam 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.

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

Chaoya Jiang, Haiyang Xu, Mengfan Dong, Jiaxing Chen, Wei Ye, Ming Yan, Qinghao Ye, Ji Zhang, Fei Huang, and Shikun Zhang. Hallucination Augmented Contrastive Learning for Multimodal Large Language Model, February 2024.

Zaid Khan and Yun Fu. Consistency and Uncertainty: Identifying Unreliable Responses From Black-Box Vision-Language Models for Selective Visual Question Answering, April 2024.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation, April 2023.

Alex 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.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023a.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023b.

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.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022. URL https://arxiv.org/abs/2201.12086.

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

Qing Li, Chenyang Lyu, Jiahui Geng, Derui Zhu, Maxim Panov, and Fakhri Karray. Reference-free Hallucination Detection for Large Vision-Language Models, August 2024.

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

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

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023c.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26296–26306, 2024.

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

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

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. Kernel Language Entropy: Fine-grained Uncertainty Quantification for LLMs from Semantic Similarities, May 2024.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov,

Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Nils Reimers and Iryna 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.

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

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming 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.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9568–9578, 2024.

Lei Wang, Jiabang He, Shenshen Li, Ning Liu, and Ee-Peng Lim. Mitigating fine-grained hallucination by fine-tuning large vision-language models with caption rewrites. In *International Conference on Multimedia Modeling*, pp. 32–45. Springer, 2024.

Qifan Yu, Juncheng Li, Longhui Wei, Liang Pang, Wentao Ye, Bosheng Qin, Siliang Tang, Qi Tian, and Yueting Zhuang. Hallucidoctor: Mitigating hallucinatory toxicity in visual instruction data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12944–12953, 2024.

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

Bohan Zhai, Shijia Yang, Chenfeng Xu, Sheng Shen, Kurt Keutzer, and Manling Li. Halle-switch: Controlling object hallucination in large vision language models. *arXiv e-prints*, pp. arXiv–2310, 2023.

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

Hanlin Zhu, Banghua Zhu, and Jiantao Jiao. Efficient Prompt Caching via Embedding Similarity, February 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. 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). 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 & Fu, 2024).** 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 try to 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 & Gales (2021).** This approach normalizes the joint log-probability of each sequence by dividing it by the sequence length and is proposed by Malinin & 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). Unlike us using the Gram matrix, they use the covariance matrix to show the correlation between samples. This leads to very small values of the covariance matrix, however, in their default settings, they use $1e-3$ jitter term, which is significantly large, compared to the values of the matrix. Therefore, we apply a smaller jitter term of $1e-8$ to improve their performance as well as a fairer comparison.

Other experimental settings In this work, we primarily use `LLaVA-v1.5-13B` as our MLLM, with further analysis on other models provided in appendix A.5. 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 ($T = 0.1$) and we use this answer $\hat{y}_t$ to evaluate the correctness of the model when answering this pair. 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 appendix A.4. 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 appendix A.3.

A.2 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. (6), an easily interpretable form.

$$\begin{aligned}
\tilde{V} &= \log\det(\tilde{R}\tilde{R}^T) && (7)\\
&= \log\det\left(\exp(\alpha\,\mathrm{diag}(1-p))R\right)\left(\exp(\alpha\,\mathrm{diag}(1-p))R\right)^T && (8)\\
&= \log\left[\det\left(\exp(\alpha\,\mathrm{diag}(1-p))\right)\det\left(RR^T\right)\det\left(\exp(\alpha\,\mathrm{diag}(1-p))^T\right)\right] && (9)\\
&= \log\det(RR^T) + 2\log\det\left(\exp(\alpha\,\mathrm{diag}(1-p))\right) && (10)\\
&= V + 2\log\prod_i \exp(\alpha(1-p_i)) && (11)\\
&= V + \tilde{\alpha}\mathbb{E}[1-p] && (12)
\end{aligned}$$

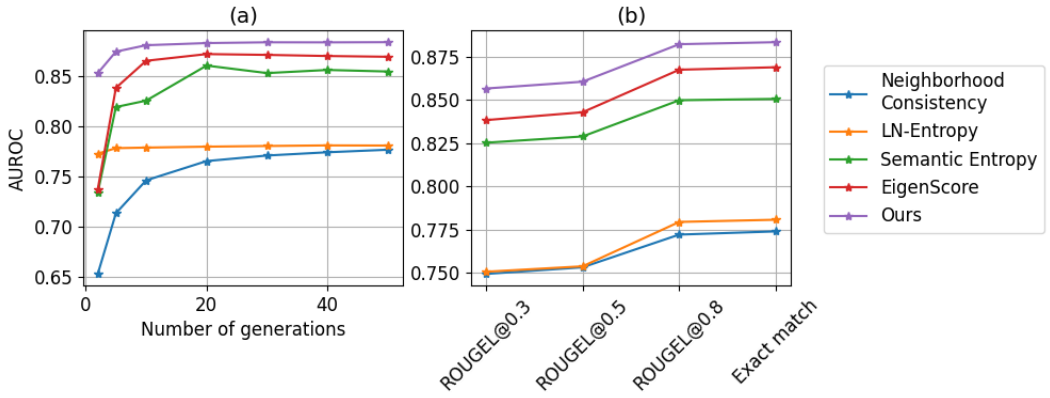

Figure 2: Ablation study on the (a) number of generations and (b) different evaluation functions $a(\mathcal{M}, t^*)$ for our method. (a) shows the AUROC performance as the number of generations increases, demonstrating the impact of additional generations. (b) compares AUROC across various correctness evaluation metrics $a(\mathcal{M}, t^*)$, including multiple levels of ROUGE-L and exact match. Our method consistently outperforms other baselines across different settings.

where eq. (8) follows from the definition of $\tilde{R}$ in eq. (5), eq. (9) uses the identity $\det(AB) = \det(A)\det(B)$, eq. (10) the identity $\log(AB) = \log(A) + \log(B)$, eq. (11) the definition of semantic volume in eq. (3), and eq. (12) noting that the sum is over a Monte-carlo sampling of model responses, with $\tilde{\alpha} = 2n\alpha$ redefined to absorb constants including n which is the number of sampled responses.

A.3 Number of generations analysis

To analyze the impact of the number of generations on evaluation performance, we conduct an ablation study by varying the number of generated outputs on the subset of the validation set of VQAv2. As shown in fig. 2 (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.4 Evaluation functions $a(\mathcal{M}, t^*)$ analysis

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.2(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.

A.5 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. 3, 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.

A.6 Prompts

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

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

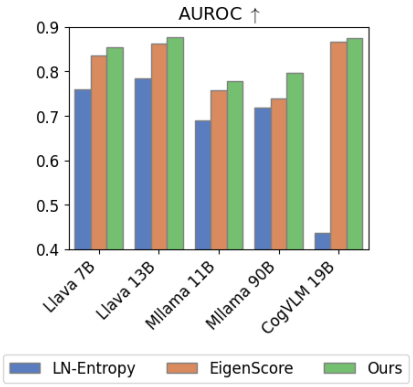

Figure 3: Ablation study across different models, evaluating AUROC performance for LN-Entropy, EigenScore, and UMPIRE. The results indicate that UMPIRE consistently achieves higher AUROC across various models, including `LLaVA-7B`, `LLaVA-13B`, `Mllama-11B`, `Mllama-90B`, and `CogVLM2-19B`. This highlights the robustness and effectiveness of our approach across different model architectures.