UNDERSTANDING THE SOURCES OF UNCERTAINTY FOR LARGE LANGUAGE AND MULTIMODAL MODELS

Ziran Yang[♣] **Shibo Hao**[♣] **Hao Sun**[♢] **Lai Jiang**[♣]
Qiyue Gao[♣] **Binglin Zhou**[♣] **Yian Ma**[♣] **Zhiting Hu**[♣]

[♣]UC San Diego, [♢]Cambridge University

ABSTRACT

Understanding and quantifying uncertainty in large model predictions is critical for their safe and trustworthy deployment. However, existing methods that estimate the overall prediction uncertainty often fail due to model overconfidence, where incorrect predictions are made with low uncertainty. Uncertainty decomposition provides a way to focus on some specific parts in total uncertainty, removing those unrelated components. Traditional uncertainty decomposition into epistemic (model-related) and aleatoric (data-related) components is insufficient for current model usage, as additional factors like prompt phrasing and context significantly influence the model's predictions. We introduce a unified uncertainty decomposition framework that systematically separates uncertainty contributions from various factors such as prompting, context, and preprocessing of multimodal inputs. By quantifying each component's uncertainty, our approach identifies which uncertainty terms are well-calibrated with the model's error rates, thereby enhancing error detection and model improvement. We validate our framework through applications in visual question answering and chain-of-thought reasoning, demonstrating that effective uncertainty calibrators can serve as metrics for error detection and improve model performance through self-training. Grounded in information theory and highly extensible, our framework provides a novel perspective on uncertainty quantification in large language and multimodal models, offering valuable insights for future research.

1 INTRODUCTION

Large language models (LLMs) and multimodal models, while very capable, are prone to producing inaccurate or unreliable outputs, such as hallucinations (Bender et al., 2021; Huang et al., 2023; Dziri et al., 2024) Understanding what these models do not know or are uncertain about is crucial for their safe and trustworthy deployment, and extensive work has been dedicated to quantifying the uncertainty of LLM outputs and use uncertainty for detecting hallucinations (Xiong et al., 2023; Kadavath et al., 2022; Farquhar et al., 2024; Kuhn et al., 2023; Hou et al., 2024). However, in real-world applications, existing methods to quantify the model's prediction uncertainty are not always reliable, e.g. the models may produces incorrect predictions very confidently, which we refer to as overconfidence (Xiong et al., 2023; Groot & Valdenegro-Toro, 2024; Yang et al., 2024). Previous work on uncertainty-based hallucination detection often overlooks or explicitly excludes this issue, e.g. Farquhar et al. (2024) stated that they do not consider the scenarios when "*LLMs continue to make errors due to incorrect training data*."

Uncertainty decomposition (Liu et al., 2019a; Der Kiureghian & Ditlevsen, 2009; Hüllermeier & Waegeman, 2021) provides a pathway to attributing the prediction uncertainty to different possible sources, which could potentially help us understand the uncertainty and find a more reliable estimation. Traditional methods (Hüllermeier & Waegeman, 2021; Schweighofer et al., 2023b) commonly decompose prediction uncertainty (or *total uncertainty*) into *epistemic uncertainty* (model's knowledge) and *aleatoric uncertainty* (inherent data randomness) components; Focusing on the decomposed epistemic component rather than the total uncertainty finds success in many applications (Charpentier et al., 2022; Osband et al., 2023; Hou et al., 2024; Ling et al., 2024), as it excludes the irreducible data-contributed part. However, this formulation falls short for foundation models,

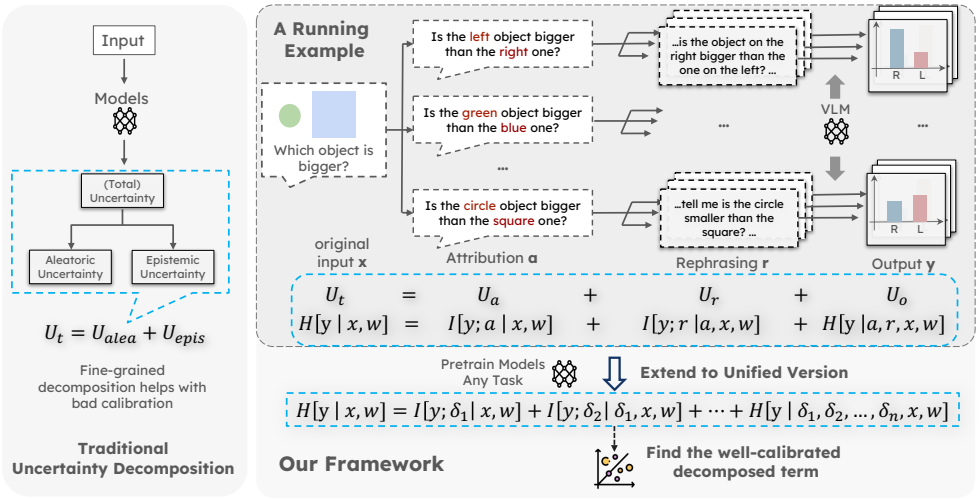


Figure 1: Outline of the paper. Traditional methods divide uncertainty into epistemic and aleatoric components (Sec. 2.1). Our framework starts with a running example: a vision language model is queried to compare the size of two objects in the image, where the attribution used to name the objects and the phrasing of the query alternate. We decompose total uncertainty (U_t) into those contributed by attribution (U_a), rephrasing (U_r), and remaining part (U_o). Furthermore, it can be extended to a unified version of any factors δ influencing model predictions (Sec. 2.2). Based on the decomposition identify those "effective calibrators" — uncertainty terms that strongly and positively correlates with error rates — which can improve model calibration and reliability(Sec. 2.3).

where uncertainty arises from many additional factors affecting model prediction, such as the phrasing of prompt, the choice of in-context examples (Wei et al., 2023; Dong et al., 2022; Wu et al., 2024), etc.

Go beyond traditional aleatoric-epistemic dualism, we introduce a systematic framework of **uncertainty decomposition** for large language and multimodal models (Sec. 2). It is flexible enough to measure the uncertainty contributed by various factors in extensible settings, as illustrated in Fig. 1. Our framework breaks down the total uncertainty into a sequence of mutual information between different factors, each of which measures the reduction in uncertainty the factor contributes to the model's output.

With the new framework, we are able to understand why the total uncertainty of models is not well correlated with error rates. Our empirical results show that some decomposed terms of uncertainty are significantly miscalibrated (Sec. 3), negatively impacting the total uncertainty. We further find out that some of the decomposed terms shows positive correlations. Thus we propose a simple calibration test method to distinguish decomposed uncertainty terms that are useful in terms of calibration to error rates (termed "effective **calibrators**") from those that are not. Experiments in two common scenarios, including visual question answering and chain-of-thoughts reasoning, validate that those identified effective calibrators can serve as better metrics for error detection, and could even help boost the performance of the model through self-training (Huang et al., 2022; Mukherjee & Awadallah, 2020; Yu et al., 2022). We believe that these findings provide a new perspective on uncertainty quantification in the foundation model and could provide valuable insights for future research.

2 METHODS

The objective of our paper is to go beyond the traditional perspective on uncertainty decomposition (Sec. 2.1) and quantify the uncertainty contributed by the choice of different variables, such as attribution a and rephrasing r in the running example (Fig. 1), to the model's output (Sec. 2.2). Furthermore, we aim to provide a diagnostic method for understanding the impact of different parts of uncertainty with respect to model performance (Sec. 2.3).

Notations. Without introducing any prior constrains on the task we focus on, we have the following notations: $\mathcal{D}$ is the training dataset, x is the input, w is the parametrized model, and y is the output defined on semantic space (Farquhar et al., 2024) which is basically we cluster all the answers that refer to the same ground truth option as one single y, despite its various expressions in natural language. $\mathbb{H}, I, D_{\mathrm{KL}}$ stand for entropy, mutual information, and KL-divergence respectively.

2.1 BACKGROUND

In traditional machine learning, uncertainty is typically decomposed into two types: aleatoric and epistemic uncertainty Hora (1996); Der Kiureghian & Ditlevsen (2009); Hüllermeier & Waegeman (2021). Aleatoric uncertainty represents the inherent randomness in the data, while epistemic uncertainty arises from the model's limited knowledge, often due to insufficient data and imperfect learning. This decomposition disentangles the uncertainty contributed by the imperfect training process from those inherent in data randomness, which was appropriate in classical ML settings or earlier deep learning contexts (Gal et al., 2016; Schweighofer et al., 2023b), where models' usage was relatively simple

We start with a training dataset $\mathcal{D}$ and an input x. Ideally, the distribution of the output y should be entirely determined by both $\mathcal{D}$ and x. Thus the prediction distribution is denoted as $p(y \mid x, \mathcal{D})$, and the total uncertainty in the prediction is measured by the entropy $\mathbb{H}[p(y \mid x, \mathcal{D})]$. In reality, the learning process introduces an intermediate variable, the model w, which is, from a Bayesian view, sampled from the distribution $p(w \mid \mathcal{D})$. As a result, the model's predictions are not directly conditioned on $\mathcal{D}$ and x, but rather on this intermediate variable w, i.e. $p(y \mid x, w)$ Using mutual information, the total uncertainty can be decomposed into aleatoric and epistemic components:

$$\underbrace{\mathbb{H}[y \mid x, \mathcal{D}]}_{\text{Total Uncertainty}} = \underbrace{\mathbb{H}[y \mid x, w]}_{\text{Aleatoric Uncertainty}} + \underbrace{I[y; w \mid x, \mathcal{D}]}_{\text{Epistemic Uncertainty}} \tag{1}$$

Here, the uncertainty contributed by the model weights w is measured by the mutual information between y and w given x and $\mathcal{D}$, called epistemic uncertainty. The aleatoric term $\mathbb{H}[y \mid x, w]$ represents the remaining uncertainty when w is known, often regarded as the inherent stochasticity of the data.

Recent works have adapted this traditional decomposition to inference-time LLM settings. These works are still from the perspective of aleatoric and epistemic uncertainty and propose different implementations to quantify uncertainty in specific settings, such as adding a clarification step to the input (Hou et al., 2024) or considering in-context examples (Ling et al., 2024). Later, we will show that these methods are special cases of our proposed framework. For example, in the clarification setting, an ambiguous question like x = "`In the image, is this object larger than another?`" can be clarified by a clarification c = "`The question refers to the red object.`"

$$\underbrace{\mathbb{H}[y \mid x, w]}_{\text{Total Uncertainty}} = \underbrace{\mathbb{H}[y \mid c, x, w]}_{\text{"Epistemic" Uncertainty}} + \underbrace{I[y; c \mid x, w]}_{\text{"Aleatoric" Uncertainty}} \tag{2}$$

The authors use this method to quantify aleatoric uncertainty (irreducible to clarification c) in ambiguous inputs, drawing a straightforward yet insightful analogy to traditional uncertainty decomposition.

Moving beyond the conventional *epistemic-aleatoric* dualistic perspective, which attributes uncertainty solely to two sources—one inherent in the data (aleatoric) and the other arising from model weights (epistemic)—our framework adopts a more nuanced approach. Instead of fitting uncertainty into the binary epi-alea categorization, we decompose it based on the specific variables that contribute to it. In the context of the clarification setting, this means partitioning uncertainty into the component introduced by the clarification c ($I[y; c \mid x, w]$) and the remaining uncertainty given the clarification ($\mathbb{H}[y \mid c, x, w]$). This perspective allows for a more flexible and interpretable understanding of uncertainty in large language models, highlighting how different factors contribute to the overall uncertainty beyond the traditional dual sources.

2.2 General Uncertainty Decomposition Framework

Going beyond the classical perspective of aleatoric and epistemic dualism, we introduce a general uncertainty decomposition framework that allows us to analyze the contributions of different factors to total uncertainty.

Running Example. We start with the running example (Fig.1) with two intermediate variables: attribution a and rephrasing r. We aim to quantify the uncertainty contributed by each of these variables. Inspired by previous work, we adapt a similar equation to first decompose the contribution from variable a, see Eq. 3. Intuitively, the uncertainty introduced by a variable should be quantified by how much it influences the model's predictions The term $\mathbb{H}[y \mid a, x, w]$ captures the remaining unpredictability of y even **when a is known**. Thus we decompose this term to find the uncertainty contributed by other variables without the influence of a. In the running example, taking another variable r into account, we can calculate the uncertainty contributed by r with $U_r = I[y; r \mid a, x, w]$, a part in $\mathbb{H}[y \mid a, x, w]$. Thus we derive the following equations for such a decomposition:

$$\underbrace{\mathbb{H}[y \mid x, w]}_{U_t\text{: total uncertainty}} = \mathbb{H}[y \mid a, x, w] + I[y; a \mid x, w] \tag{3}$$

$$= \underbrace{\mathbb{H}[y \mid r, a, x, w]}_{U_o\text{: observed uncertainty}} + \underbrace{I[y; r \mid a, x, w]}_{U_r\text{: contributed by rephrasing } r} + \underbrace{I[y; a \mid x, w]}_{U_a\text{: contributed by attribution } a} \tag{4}$$

Here, $U_a = I[y; a \mid x, w]$ quantifies the uncertainty contributed by choosing different attribution a, measuring how knowledge of a reduces uncertainty about y given x and w. Similarly, $U_r = I[y; r \mid a, x, w]$ captures the uncertainty due to varying prompts r, indicating the reduction in uncertainty about y when r is known in addition to a, x, and w. The term $U_o = \mathbb{H}[y \mid r, a, x, w]$ represents the residual uncertainty inherent in y when both a and r are known. It captures the inherent unpredictability in the model's outputs when both the prompt and context are known.

Unified Formulation. To generalize this approach, we propose a unified uncertainty decomposition framework that can handle multiple intermediate variables and different models. Let y be the model's prediction, x be the input, and $\delta_1, \delta_2, \ldots, \delta_n$ be intermediate variables that influence the model's predictions. We denote $\tilde{w}$ as the approximation of the true predictive distribution, which extends beyond a pre-selected off-the-shelf model w to also include more general scenarios such as ensembles i.e., $\tilde{w} = w \sim p(\cdot \mid \mathcal{D})$, or even the training data $\mathcal{D}$ itself (Schweighofer et al., 2024; 2023a). The total uncertainty can then be decomposed using the chain rule of mutual information:

$$\mathbb{H}[y \mid x, \tilde{w}] = \mathbb{H}[y \mid \delta_1, \ldots, \delta_n, x, \tilde{w}] + \sum_{i=1}^{n} I[y; \delta_i \mid \delta_{<i}, x, \tilde{w}] \tag{5}$$

where $\delta_{<i}$ denotes all variables before δ_i. This decomposition allows us to quantify the individual contributions of each intermediate variable to the total uncertainty. In the running example, $\delta_1 = a$, $\delta_2 = r$.

We can further elaborate on how the uncertainty contributed by δ_i is quantified:

$$I[y; \delta_i \mid \delta_{<i}, x, \tilde{w}] = \mathbb{E}_{\delta_{\leq i}}\left[D_{\mathrm{KL}}\left(p(y \mid \delta_{\leq i}, x, \tilde{w}) \parallel p(y \mid \delta_{<i}, x, \tilde{w})\right)\right] \tag{6}$$

where the marginal distribution: $p\left(y \mid \delta_{<i}, x, \tilde{w}\right) = \mathbb{E}_{\delta_i}\left[p\left(y \mid \delta_{\leq i}, x, \tilde{w}\right)\right]$. Mutual information here measures the expected reduction in uncertainty about y given knowledge of δ_i and it can be expressed as the expected KL divergence between the conditional and marginal distributions. It illustrates the physical meaning of uncertainty contributed by δ_i: The KL divergence quantifies the decrease in uncertainty (or increase of surprisal) when updating our belief from the marginal distribution to the conditional distribution, thereby capturing how additional information about δ_i influences the model's predictions

The **intermediate variables** δ_i represent factors that influence the model's predictions without changing the underlying distribution of the target variable y , such as the desired answer should be invariant to various prompting methods or context examples. Examples of intermediate variables include different network modules, prompt rephrasings, contextual information, and other elements inherent to the model's processing (Sec.2.4). To distinguish between input variables x and intermediate variables δ, consider their impact on the ground truth y: if a variable affects y, it is classified as part of x; otherwise, it should be treated as a δ.

Additionally, the **order of decomposition** among multiple intermediate variables can be determined based on the task setting and the dependencies among the variables. If a variable δ_j is generated conditionally based on δ_i, it is natural to decompose δ_i before δ_j. For further discussion please see Appendix B. Regardless of the order of decomposition, the decomposition and analysis principles remain consistent, allowing us to systematically quantify the uncertainty contributed by each variable. This flexibility enables our framework to adapt to a wide range of models and applications by selecting relevant intermediate variables based on the specific setting.

2.3 Understanding the Decomposed Uncertainty

In this section, we delve deeper into understanding uncertainty through our unified decomposition framework. Our motivation is to identify uncertainty from different sources to mitigate the overconfidence problem in uncertainty-based applications such as error detection and self-training, by dissecting the uncertainty contributed by different variables. The discussion is structured into two main parts. First, we introduce a method called the **calibration test**, designed to empirically assess the relationship between uncertainty components and model performance. Second, we provide intuitive insights into why some uncertainty terms may be negatively correlated with error rates.

Calibration Test. To empirically assess how each uncertainty component correlates with actual prediction errors, we introduce the calibration test. Uncertainty calibration refers to the alignment between predicted uncertainties and observed error rates, ensuring that uncertainty estimates reliably reflect the likelihood of errors (Gruber & Buettner, 2022). In our framework, each decomposed uncertainty term serves as a **calibrator** that predicts the probability of prediction errors based on its specific contribution to the total uncertainty.

In our running example, we evaluate the effectiveness of each calibrator by examining their correlation with actual prediction error rates. For each sample x, we calculate the error rate $\text{Error}(x)$ by averaging the prediction errors across various attributions a, rephrasings r, and sampled predictions. Using the decomposition from Eq. 4, we compute the uncertainty components for each sample: $U_a(x)$ for attribution uncertainty, $U_r(x)$ for rephrasing uncertainty, and $U_o(x)$ for the observed uncertainty. The total uncertainty $U_t(x)$ is the sum of these components.

We then calculate the Pearson correlation coefficient and the corresponding p-value between each calibrator and the error rate across all samples x. A strong positive correlation indicates that the uncertainty component effectively predicts a higher likelihood of errors, making it a reliable calibrator. Conversely, uncertainty terms with weak or negative correlations are less effective, as they do not consistently signal the probability of errors. This calibration test allows us to identify which aspects of uncertainty are most indicative of prediction performance, thereby informing strategies to enhance model reliability.

Effective calibrators provide reliable uncertainty estimates that are crucial for applications such as uncertainty-based error detection and self-training. By identifying which uncertainty components are strong predictors of errors, we can enhance the model's ability to detect when it is likely to make a mistake, allowing for corrective measures or human intervention. This not only improves model performance but also increases trust in the model's predictions in practical deployments.

Discussions. Based on our unified decomposition framework, especially the ensembling perspective outlined in Eq. 6 that each uncertainty part is quantified as KL divergence between individual prediction distributions and an aggregated average distribution, we can discuss why some uncertainty components are badly calibrated-negatively correlated with error rates. Ensemble-based methods are traditionally valued for enhancing model performance by reducing uncorrelated errors and increasing robustness through the diversity of ensemble members, thereby better capturing the $x \rightarrow y$ mapping (Rokach, 2010; Ganaie et al., 2022; Lakshminarayanan et al., 2017a). However, this benefit does not consistently extend to inference-time ensembling techniques, such as prompt augmentation or output bootstrapping (Jiang et al., 2023). In these cases, ensembling may fail to produce a more accurate mapping, meaning that higher uncertainty does not necessarily indicate a higher likelihood of error. This discrepancy arises primarily for two reasons. First, when the true distribution is approximated by model parameters rather than the data $\mathcal{D}$, ensembling can reinforce the model's inherent biases, conflicting with the actual data distribution and nullifying beneficial explorations from sampling stochasticity (Song et al., 2024). Second, joint training can lead to en-

sembling collapse, where ensemble members become overly similar, reducing their diversity and effectiveness (Jeffares et al., 2023; Liu et al., 2019b). This lack of diversity can result in spurious structures and ineffective ensembling during inference. Our unified decomposition framework quantifies uncertainty in these scenarios and provides diagnostic methods to identify sources of model uncertainty, thereby highlighting situations where ensembling might not enhance—and could even degrade—model performance.

Understanding these nuances is important for effectively using uncertainty estimates in applications. By analyzing the contributions of different uncertainty components, we can pinpoint which ones are reliable indicators of prediction errors and which are not. In Sec. 3, we revisit the downstream tasks like hallucination detection and self-training from the novel perspective of uncertainty decomposition.

2.4 Use Cases and Related Work

Table 1: Summary of uncertainty decompositions in different scenarios with examples. In the uncertainty-based exploration in RL, s stands for state, and a stands for action. For further discussion please refer to Appendix A.3.

Setting	δ	$\tilde{w}$	**Formula**	**Examples**
No Decomposition Uncertainty for Error Detection	-	w	$\mathbb{H}[y \mid x, w]$	(Farquhar et al., 2024; Kossen et al., 2024)
Traditional Decomposition	model w	$\mathcal{D}$	$\mathbb{H}[y \mid x, \mathcal{D}] = \mathbb{H}[y \mid x, w]$ $+I[y; w \mid x, \mathcal{D}]$	(Hüllermeier & Waegeman, 2021)
Uncertainty-Based Exploration in RL	observation network w	$\mathcal{D}$	$\mathbb{H}[a \mid s, \mathcal{D}] = \mathbb{H}[a \mid s, w]$ $+I[a; w \mid s, \mathcal{D}]$	(Osband et al., 2016; Burda et al., 2018)
In-Context Learning Input Clarification	context or clarification c	w	$\mathbb{H}[y \mid x, w] = \mathbb{H}[y \mid x, c, w]$ $+I[y; c \mid x, w]$	(Ling et al., 2024; Hou et al., 2024)
Prompt Rephrasing Augmentation	prompts q	w	$\mathbb{H}[y \mid x, w] = \mathbb{H}[y \mid x, w, q]$ $+I[y; q \mid x, w]$	(Jiang et al., 2023; Yadkori et al., 2024)
VLM Attributions Binding	attribution a, rephrasing r	w	$\mathbb{H}[y \mid x, w] = \mathbb{H}[y \mid x, a, r, w]$ $+I[y; r \mid x, a, w] + I[y; a \mid x, w]$	Sec. 3.1
LLM Math Reasoning	entity name c, prompting q	w	$\mathbb{H}[y \mid x, w] = \mathbb{H}[y \mid x, c, q, w]$ $+I[y; q \mid x, c, w] + I[y; c \mid x, w]$	Sec. 3.2

As we mentioned earlier, this framework is capable of decomposing various types of uncertainty, without imposing any prior assumptions on the intermediate variable δ. Table 1 provides examples of relevant use cases from the literature. For a detailed discussion on how our conclusions interpret and extend related work, please refer to Appendix A.3.

In traditional decomposition, previous works like Bootstrapped DQN (Lakshminarayanan et al., 2017b) and random network distillation Burda et al. (2018), in our view, can be summarized as follows: in these learning algorithms, training epistemic uncertainty serves as an effective calibrator. Thus, the effectiveness of learning is reflected in the uncertainty of w, i.e., the epistemic term, which supports the efficacy of ensemble learning (Dong et al., 2020; Osband et al., 2016; Ghasemipour et al., 2022) and pessimistic learning .

In uncertainty-based exploration in reinforcement learning, where some works measure uncertainty for exploration using $I[a; w_o \mid s]$ to quantify whether the current state has been encountered before, thereby eliminating aleatoric components inherent in the environment like the well-known "noisy TV" problem (Burda et al., 2018). In our view, these works are equivalent to a single statement that, under a well-defined observation model, exploration uncertainty serves as a effective calibrator. Therefore, this signal can encourage beneficial exploration, as high uncertainty genuinely corresponds to previously under-explored states (with many errors).

In recent work on LLMs, different parts have been decoupled according to the research context, each focusing on a single intermediate variable and drawing straightforward analogies with traditional uncertainty decomposition. For example, Hou et al. (2024) focused on input ambiguity, introducing an additional clarification step c. Ling et al. (2024) concentrated on sampling in-context examples, treating the context examples c as an intermediate variable. Additionally, Yadkori et al. (2024) implicitly used the prompt prefix or suffix as an intermediate variable, while Jiang et al. (2023) integrated various prompting design methods.

3 Applications and Experiments

In this section, we present two use cases that demonstrate the effectiveness of our uncertainty decomposition framework and the insights it provides. Specifically, we aim to address how our framework **distinguishes between effective and ineffective calibrators** by effectively differentiating various sources of uncertainty and revealing their relationships with prediction errors. Additionally, we investigate the behavior of different calibrators when applied to **downstream tasks**, examining whether identifying effective calibrators can enhance performance in tasks such as error detection and self-training, and exploring the implications of utilizing ineffective calibrators.

Experimental Setup In each setting, our experiments involve an inference-time dataset partitioned into two subsets: the development set and the test set. **Development Set**: A small subset with ground truth labels, used to identify effective and ineffective calibrators . **Test Set**: The whole inference-time data (without ground truth labels in real usage), where we apply the various calibrators identified from the development set to downstream self-supervised methods such as error detection and self-training. Here we validate our insights about the relationship between the view of the taxonomy of calibrators and these well-established methods with ground truth.

Due to the extensibility of our framework, these applications can be adapted to a wide range of scenarios, as suggested in the proposed directions outlined in Appendix A.3.

3.1 Application 1: VLM Attribution Binding Task

VLMs are expected to accurately identify all visible attributes of objects in an image and correctly associate these attributes with their respective objects. For instance, a model is expected to identify the color of an object, such as labeling an apple red and a cucumber green, rather than just percepting that there are an apple and a cucumber and red and green things when it comes to color. This issue relates to the well-known binding problem (Greff et al., 2020) or problem on compositionality (Han et al., 2024) (more background in Appendix A.3), and in this context, we refer to as **attribute binding**. However, they often struggle with this capability. Recent studies have highlighted challenges in this area, particularly in the comprehension of spatial relationships like position and occupancy (Rahmanzadehgervi et al., 2024; Kamath et al., 2023; Zeng et al., 2024). These works inspire us that by alternating the attribution a referred to in the input, the model performance is affected significantly. To address these challenges, we investigate how different manipulations of the input affect model performance, specifically focusing on altering referred attributions versus merely rephrasing the input. We approach this issue from the perspective of uncertainty, decoupling the uncertainty related to attribute binding (denoted as a) from that associated with prompt rephrasing (denoted as r).

Implementations We need to quantify the uncertainty on a and r in a controlled setting. Since there is no good benchmark available yet, we use the following synthetic data approach. For this task, we create a dataset consisting of images each depicting simple scenes of two objects with visible different occupancy, detailed in Appendix C.1. Along with the images, we pose all images

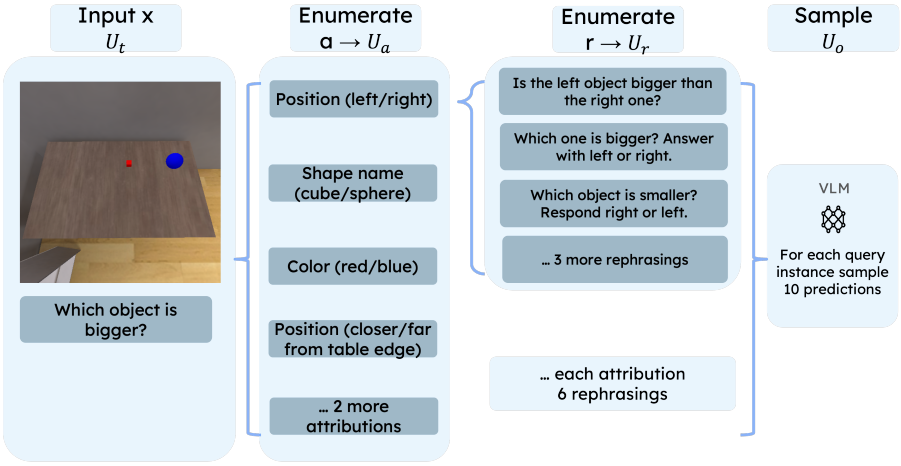


Figure 2: Decomstration of example implementation of variations on a, r in the VLM attribution binding task. We illustrate the pipeline that for a single query x, enumerates different a and r then sample predictions which then clustered into a distribution over y.

Table 2: Calibration test of different calibrators in the VLM attribution binding task. Using this table we can determine whether it is an effective or ineffective calibrator in our framework: here we interpret that U_a is an effective calibrator while the other three: U_r, U_o are negatively correlated and U_t is at chance level (or random).

Calibrator	`InternVL-2-4B` Corre. Coeff.	p-value	`Llava-1.6-7B` Corre. Coeff.	p-value	Description
U_o	0.0125	0.9016	0.0052	0.8341	random
U_r	-0.3729	0.0001	-0.3104	0.0023	negative correlated
U_a	0.5701	0.0011	0.5903	0.0009	positive correlated
U_t	0.0704	0.4867	0.0453	0.5120	random

with the same question x: "`Which object is bigger?`". Then we generate specific question instances incorporate two dimensions of variances: specific attributions a to reference the objects and the rephrasing r. We prompt `GPT-4o` using `Langfun` framework to generate both dimensions of the questions. For each sample, we generate 6 options for the attribution a and 6 options for rephrasing prompts r. Under each query instance, we sample 10 predictions from the tested model (`InternVL2-4B` and `Llava-1.6-7B`) (Chen et al., 2023; 2024) with temp $= 1.0$. Then we use sampled predictions to estimate the model prediction distribution $p(y \mid r, a, x, w)$, in which the y is defined on semantic space (Farquhar et al., 2024): which means, in this task, since y has only two options (two objects) while it can be expressed in various wording ways with different attributions like "`red object`", or the "`object on the right side`", we view those generations that referred to the same object as the same y. We use `GPT-4o` prompted with ground truth information to cluster the various predictions into two y options, judge their correctness, and calculate the average error rate for every x: Error(x). An illustration example is given in Fig. 2. Detailed implementation procedures and additional examples are provided in Appendix C.1.

Correlation Test Given the pre-selected w, we have the prediction distribution $p(y \mid a, r, x, w)$ for all a, r, x. Using these, we calculate $p(y \mid a, x, w) = \mathbb{E}_r[p(y \mid r, a, x, w)]$ and $p(y \mid x, w) = \mathbb{E}_a[p(y \mid a, x, w)]$ Then we quantify all uncertainty terms with $U_o(x) = \mathbb{E}_{a,r}\big[\mathbb{H}[p(y \mid r, a, x, w)]\big]$; $U_r(x) = \mathbb{E}_{a,r}\big[D_{\mathrm{KL}}[p(y \mid r, a, x, w) \parallel p(y \mid a, x, w)]\big]$; $U_a(x) = \mathbb{E}_a\big[D_{\mathrm{KL}}[p(y \mid a, x, w) \parallel p(y \mid x, w)]\big]$, and $U_t(x) = U_o(x) + U_r(x) + U_a(x)$. As introduced in Sec 2.3, we examine the correlation between different uncertainty terms and error rates Error(x) on the development set with ground truth labels. We list the statistics in Table 2.

We also present the scatter plot between uncertainty terms and error rate on the development set in Fig 9, which shows that for an ineffective calibrator, such as total uncertainty, there are samples in the low uncertainty region that exhibit both low and high error rates. The cluster of high error despite low uncertainty illustrates the model's overconfident behavior.

Revisit Downstream Tasks from the Perspective of Calibrators Here we demonstrate the empirical relationship between calibrators' effectiveness with downstream tasks like hallucination detection and self-training (uncertainty-based rejection sampling). It is shown that the calibration test along with uncertainty decomposition can identify those factors that will do well in these tasks, and reveal those that would perform poorly.

We start with evaluating the performance of different calibrators in **error detection**, as in previous work (Farquhar et al., 2024; Hou et al., 2024; Ling et al., 2024). We plot the ROC curves (Fig. 3) for different based on uncertainty and error rates and calculate the AUROC and AURAC metrics (Table 3) to validate the hypothesis that effective calibrators work well in uncertainty-based error detection while others work poorly.

Then we inspect to possibility of **self-training** with the help of an effective calibrator. This is quite similar to rejection sampling in LLM settings, however here we use the uncertainty metrics to select those maintained for fine-tuning. In this task, we use certain predictions of the model's own as pseudo-labels to finetune itself, just as rejection sampling which trains the model itself with its highest-scored generations. Specifically, we used the samples that fall into the lower 30% partition of each uncertainty term as training labels and fine-tuned the model. Here we show the results in Table 4.

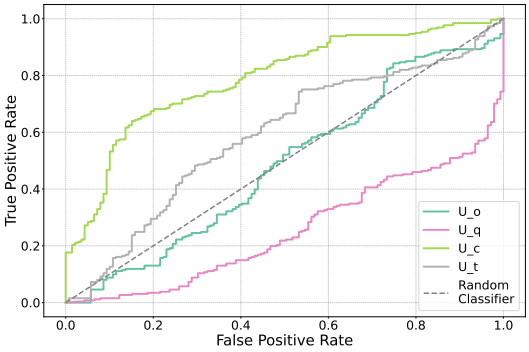


Figure 3: ROC curves in error detection task of `InternVL-2-4B` when using different uncertainty terms of VLM attribution binding experiments.

	`InternVL-2-4B`		`Llava-1.6-7B`	
	AUROC	AURAC	AUROC	AURAC
U_o	0.486	0.614	0.476	0.553
U_r	0.253	0.476	0.364	0.495
U_a	0.788	0.837	0.803	0.858
U_t	0.587	0.692	0.609	0.689

Table 3: AUROC and AURAC values in error detection task for different uncertainty parts across models.

Table 4: Accuracy under different Self-Training (uncertainty-based rejection sampling) conditions across various calibrators.

	Model	U_o	U_r	U_a	U_t
Self-Training	`InternVL-2-4B`	0.52 (-0.11)	0.41 (-0.23)	0.73 (+0.10)	0.60 (-0.03)
	`Llava-1.6-7B`	0.45 (-0.13)	0.38 (-0.20)	0.79 (+0.21)	0.62 (+0.04)

Analysis The intermediate variables a (attribution) and r (rephrasing) influence the model's prediction distributions without altering the ground truth, which is determined solely by the original input x—comprising the image content and the initial question. Consequently, different values of a and r diversify the generation trajectories, yet they ultimately converge to the same final predictions based on x. This separation allows us to isolate and examine the specific contributions of attribution binding and prompt phrasing to the model's uncertainty and accuracy.

The intuition behind why a turns out to be an effective calibrator is that given a VLM model's performance varies across different attributes; for example, it may achieve a relatively high accuracy

when questioned about color, but perform poorly in understanding spatial relationships. Without the need for expensive dense annotations, we could use a self-supervised approach where attributes with better performance help improve those with poorer performance. This could achieve a better level of binding, as different ways of asking about the same fact should theoretically be aligned with each other.

Our framework reveals that attribution a effectively calibrates the model's predictions, reducing uncertainty and improving accuracy. These results suggest that the ability to bind attributes to objects is crucial for reliable VLM performance and that improvements in attribution binding will have a more substantial impact than merely rephrasing prompts. It also hints that by properly using the binding structure, we can gain semi-supervised information gain for free, just like how self-training could improve the model's overall performance. In contrast, prompt r does not provide similar benefits and, along with both observed uncertainty and total uncertainty, serves as an ineffective calibrator.

3.2 Application 2: LLM Math Reasoning Task

In this section, we apply uncertainty decomposition to a math reasoning task using LLMs. The setting is that an LLM is queried with math problems, while the questions presented in the form of application problems often contain multiple entity names (such as human names), even though the underlying question remains unchanged. Additionally, the style of prompting, particularly in the form of chain-of-thought (CoT) prompts (Wei et al., 2023), can also vary with great influence on the model's output. We will examine these two variables: $\sigma_1 = c$ represents entity names, $\sigma_2 = q$ represents prompts design.

Implementation We choose the `SVAMP` benchmark (Patel et al., 2021) and model `Gemma-2-9B-it` (Team et al., 2024) for the task. The implementation is quite similar to that of the previous task. We utilize `GPT-4o` with `Langfun` framework to rephrase the entity names in a single math question, resulting in the same query in 6 different forms alternating the entity names, see Table 9. For example, we rephrase a question where "Paige raised 7 goldfish and 12 catfish in the pond, but stray cats loved eating them. Now she has 15 left" with alternative entity names, such as "Tom raised 7 rabbits and 12 hamsters in the yard, but wild foxes loved chasing them. Now he has 15 left," resulting in equivalent queries like "How many fishes disappeared?" and "How many pets vanished?" respectively. We then query them with 6 different designed CoT prompts templates respectively. For more details, see the Appendix C.2.

Using our framework we can naturally derive the decomposition:

$$\mathbb{H}[y \mid x, w] = \mathbb{H}[y \mid x, c, q, w] + I[y; q \mid x, c, w] + I[y; c \mid x, w]$$

and the corresponding quantifying equations as illustrated before in Eq. 6.

Results The experimental procedure is the same as in the previous Sec. 3.1. Here, we report the results of the calibration test and error detection. In this task, we performed a calibration test and evaluated the efficacy of different calibrators, focusing on the ability of the framework to distinguish between effective and ineffective calibrators for LLM-based math reasoning. The calibration results, presented in Table 5, indicate that U_q (uncertainty related to chain-of-thought prompt design) is an effective calibrator with a significant positive correlation, while U_o (observed uncertainty), U_c (entity-name-related uncertainty), and U_t (total uncertainty) show either weak or negative correlations with error rates.

For error detection, we plot ROC curves (Fig. 4) and report the AUROC and AURAC metrics for different calibrators, as shown in Table 3. The results confirm that U_q outperforms other uncertainty metrics in detecting errors, supporting the hypothesis that prompt design variability is more informative than entity names for error detection in math reasoning tasks. Self-training results align with this finding, as models fine-tuned with low U_q samples achieve higher accuracy, while other calibrators contribute little improvement or even negative effects.

Analysis The results show that prompt design q serves as an effective calibrator for LLM math reasoning, as its structure significantly influences reasoning accuracy. This aligns with the framework's

Table 5: Quantitative comparison of different calibrators in the uncertainty calibration task. This table can determine whether it is an effective or ineffective calibrator: here we can interpret that U_q is an effective calibrator, while the other three U_c, U_o, and U_t are not.

Calibrator	Corre. Coeff.	p-value	Description
U_o	-0.525	2.109e-08	negative correlated
U_q	0.460	1.495e-06	positive correlated
U_c	0.089	0.3767	random
U_t	-0.024	0.8103	random

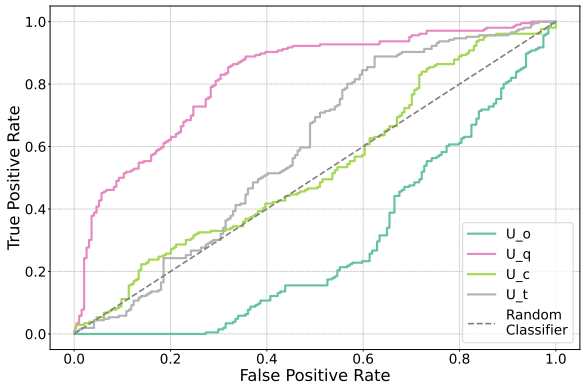


Figure 4: ROC curves in error detection task when using different uncertainty terms of LLM math reasoning experiments.

U_x	AUROC	AURAC
U_o	0.291	0.316
U_q	0.819	0.742
U_c	0.530	0.540
U_t	0.592	0.544

Table 6: AUROC and AURAC values in error detection task for different uncertainty parts.

goal of selecting task-specific calibrators: prompt variations guide error detection and enhance accuracy, while entity variations c add minimal value.

In essence, the framework reveals that, for reasoning tasks, refining prompt structure is more impactful for reducing uncertainty than altering surface attributes. This finding supports the broader application of uncertainty decomposition in selecting effective calibrators tailored to task characteristics, enhancing performance in reasoning-based applications.

The results here are actually supported by other works (Jiang et al., 2023), where techniques like prompt augmentation have been shown to effectively calibrate the model.

4 Conclusion

In conclusion, we have introduced a unified uncertainty decomposition framework that extends traditional concepts of aleatoric and epistemic uncertainty to encompass multiple intermediate variables inherent in LLMs and multi-modal language models. By systematically quantifying the uncertainty contributed by different components, we provide a principled method for diagnosing the sources of model uncertainty and their impact on performance. Our framework reveals that not all sources of uncertainty are equally informative; specifically, it distinguishes between effective calibrators, which correlate positively with error rates, and ineffective calibrators, which do not. This nuanced understanding better fits the practice reality where models exhibit high overconfidence and provide a pathway to better uncertainty estimation. Through two applications—the VLM attribution binding task and the LLM math reasoning task—we demonstrated the practical utility of our framework. In both cases, we identified specific variables that serve as effective calibrators and showed how they can enhance downstream tasks like error detection and self-training. By providing a scalable and flexible approach grounded in information theory, our framework opens new avenues for future research into the robustness and interpretability of large models, paving the way for safer and more trustworthy deployment of AI systems.

REFERENCES

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pp. 610–623, 2021.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation, 2018. URL https://arxiv.org/abs/1810.12894.

Bertrand Charpentier, Ransalu Senanayake, Mykel Kochenderfer, and Stephan Günnemann. Disentangling epistemic and aleatoric uncertainty in reinforcement learning. *arXiv preprint arXiv:2206.01558*, 2022.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024.

Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning, 2022.

Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. A survey on ensemble learning. *Frontiers of Computer Science*, 14:241–258, 2020.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, 36, 2024.

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. URL https://doi.org/10.1038/s41586-024-07421-0.

Yarin Gal et al. Uncertainty in deep learning. 2016.

Mudasir A Ganaie, Minghui Hu, Ashwani Kumar Malik, Muhammad Tanveer, and Ponnuthurai N Suganthan. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151, 2022.

Seyed Kamyar Seyed Ghasemipour, Shixiang Shane Gu, and Ofir Nachum. Why so pessimistic? estimating uncertainties for offline rl through ensembles, and why their independence matters, 2022. URL https://arxiv.org/abs/2205.13703.

Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks, 2020. URL https://arxiv.org/abs/2012.05208.

Tobias Groot and Matias Valdenegro-Toro. Overconfidence is key: Verbalized uncertainty evaluation in large language and vision-language models. *arXiv preprint arXiv:2405.02917*, 2024.

Sebastian Gruber and Florian Buettner. Better uncertainty calibration via proper scores for classification and beyond. *Advances in Neural Information Processing Systems*, 35:8618–8632, 2022.

Xu Han, Linghao Jin, Xiaofeng Liu, and Paul Pu Liang. Progressive compositionality in text-to-image generative models. *arXiv preprint arXiv:2410.16719*, 2024.

Stephen C Hora. Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. *Reliability Engineering & System Safety*, 54(2-3):217–223, 1996.

Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. Decomposing uncertainty for large language models through input clarification ensembling, 2024. URL https://arxiv.org/abs/2311.08718.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*, 2022.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions, 2023. URL https://arxiv.org/abs/2311.05232.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110(3):457–506, March 2021. ISSN 1573-0565. doi: 10.1007/s10994-021-05946-3. URL http://dx.doi.org/10.1007/s10994-021-05946-3.

Alan Jeffares, Tennison Liu, Jonathan Crabbé, and Mihaela van der Schaar. Joint training of deep ensembles fails due to learner collusion, 2023. URL https://arxiv.org/abs/2301.11323.

Mingjian Jiang, Yangjun Ruan, Sicong Huang, Saifei Liao, Silviu Pitis, Roger Baker Grosse, and Jimmy Ba. Calibrating language models via augmented prompt ensembles. In *International conference on machine learning*. PMLR, 2023.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know, 2022. URL https://arxiv.org/abs/2207.05221.

Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's "up" with vision-language models? investigating their struggle with spatial reasoning, 2023. URL https://arxiv.org/abs/2310.19785.

Amita Kamath, Cheng-Yu Hsieh, Kai-Wei Chang, and Ranjay Krishna. The hard positive truth about vision-language compositionality, 2024. URL https://arxiv.org/abs/2409.17958.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in llms, 2024. URL https://arxiv.org/abs/2406.15927.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation, 2023. URL https://arxiv.org/abs/2302.09664.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017a.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017b. URL https://arxiv.org/abs/1612.01474.

Paul Pu Liang, Yun Cheng, Xiang Fan, Chun Kai Ling, Suzanne Nie, Richard Chen, Zihao Deng, Nicholas Allen, Randy Auerbach, Faisal Mahmood, Ruslan Salakhutdinov, and Louis-Philippe Morency. Quantifying and modeling multimodal interactions: An information decomposition framework, 2023. URL https://arxiv.org/abs/2302.12247.

Chen Ling, Xujiang Zhao, Xuchao Zhang, Wei Cheng, Yanchi Liu, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Jie Ji, Guangji Bai, Liang Zhao, and Haifeng Chen. Uncertainty quantification for in-context learning of large language models, 2024. URL https://arxiv.org/abs/2402.10189.

Jeremiah Liu, John Paisley, Marianthi-Anna Kioumourtzoglou, and Brent Coull. Accurate uncertainty estimation and decomposition in ensemble learning. *Advances in neural information processing systems*, 32, 2019a.

Ling Liu, Wenqi Wei, Ka-Ho Chow, Margaret Loper, Emre Gursoy, Stacey Truex, and Yanzhao Wu. Deep neural network ensembles against deception: Ensemble diversity, accuracy and robustness. In *2019 IEEE 16th international conference on mobile ad hoc and sensor systems (MASS)*, pp. 274–282. IEEE, 2019b.

W. McGill. Multivariate information transmission. *Transactions of the IRE Professional Group on Information Theory*, 4(4):93–111, 1954. doi: 10.1109/TIT.1954.1057469.

Subhabrata Mukherjee and Ahmed Awadallah. Uncertainty-aware self-training for few-shot text classification. *Advances in Neural Information Processing Systems*, 33:21199–21212, 2020.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn, 2016. URL https://arxiv.org/abs/1602.04621.

Ian Osband, Zheng Wen, Seyed Mohammad Asghari, Vikranth Dwaracherla, Morteza Ibrahimi, Xiuyuan Lu, and Benjamin Van Roy. Epistemic neural networks. *Advances in Neural Information Processing Systems*, 36:2795–2823, 2023.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168.

Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind, 2024. URL https://arxiv.org/abs/2407.06581.

Lior Rokach. Ensemble-based classifiers. *Artificial intelligence review*, 33:1–39, 2010.

Kajetan Schweighofer, Lukas Aichberger, Mykyta Ielanskyi, and Sepp Hochreiter. Introducing an improved information-theoretic measure of predictive uncertainty, 2023a. URL https://arxiv.org/abs/2311.08309.

Kajetan Schweighofer, Lukas Aichberger, Mykyta Ielanskyi, Günter Klambauer, and Sepp Hochreiter. Quantification of uncertainty with adversarial models, 2023b. URL https://arxiv.org/abs/2307.03217.

Kajetan Schweighofer, Lukas Aichberger, Mykyta Ielanskyi, and Sepp Hochreiter. On information-theoretic measures of predictive uncertainty, 2024. URL https://arxiv.org/abs/2410.10786.

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. Trial and error: Exploration-based trajectory optimization for llm agents, 2024. URL https://arxiv.org/abs/2403.02502.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 2024.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Maria Mihaela Trusca, Wolf Nuyts, Jonathan Thomm, Robert Honig, Thomas Hofmann, Tinne Tuytelaars, and Marie-Francine Moens. Object-attribute binding in text-to-image generation: Evaluation and control, 2024. URL https://arxiv.org/abs/2404.13766.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Junda Wu, Zhehao Zhang, Yu Xia, Xintong Li, Zhaoyang Xia, Aaron Chang, Tong Yu, Sungchul Kim, Ryan A. Rossi, Ruiyi Zhang, Subrata Mitra, Dimitris N. Metaxas, Lina Yao, Jingbo Shang, and Julian McAuley. Visual prompting in multimodal large language models: A survey, 2024. URL https://arxiv.org/abs/2409.15310.

Yunlong Xiong, Jinhyuk Lee, Chandan Joshi, Jesse Finnie-Ansley, and Dragomir Radev. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*, 2023.

Yasin Abbasi Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvári. To believe or not to believe your llm, 2024. URL https://arxiv.org/abs/2406.02543.

Haoyan Yang, Yixuan Wang, Xingyin Xu, Hanyuan Zhang, and Yirong Bian. Can we trust llms? mitigate overconfidence bias in llms through knowledge transfer. *arXiv preprint arXiv:2405.16856*, 2024.

Yue Yu, Lingkai Kong, Jieyu Zhang, Rongzhi Zhang, and Chao Zhang. Actune: Uncertainty-based active self-training for active fine-tuning of pretrained language models. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 1422–1436, 2022.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it?, 2023. URL https://arxiv.org/abs/2210.01936.

Y. Zeng, Y. Huang, J. Zhang, Z. Jie, Z. Chai, and L. Wang. Investigating compositional challenges in vision-language models for visual grounding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14141–14151, Los Alamitos, CA, USA, jun 2024. IEEE Computer Society. doi: 10.1109/CVPR52733.2024.01341. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.01341.

A Background

A.1 Preliminary of Information Theory Used

Some basic rules:

$$\mathrm{CE}[P, Q] = \mathbb{H}[P] + D_{\mathrm{KL}}(P \parallel Q) \tag{7}$$

$$I(X; Y) = \mathbb{E}_Y \left[D_{\mathrm{KL}} \left(p_{X|Y} \parallel p_X \right) \right] \tag{8}$$

The uncertainty contributed to a specific variable δ is quantified by mutual information and by the equation $I(X; Y) = \mathbb{E}_Y \left[D_{\mathrm{KL}} \left(p_{X|Y} \parallel p_X \right) \right]$ are related to KL, which matches with the natural on KL-divergence. The KL calculates the averaged difference of surprisal, thus the decrease of uncertainty, note that $D_{\mathrm{KL}}(P \parallel Q) = \mathbb{E}_p[(-\log q) - (-\log p)]$ in which $-\log p$ is the surprisal.

A.2 Traditional UQ Decomposition

In classical formulations (Hüllermeier & Waegeman, 2021; Schweighofer et al., 2023b), the Bayesian framework offers a principled way to treat the uncertainty about the model weights through the posterior over hypothesis space $p(w \mid \mathcal{D}) \propto p(\mathcal{D} \mid w)p(w)$ for a given dataset $\mathcal{D}$. The Bayesian model average (BMA) predictive distribution is given by

$$p(y \mid x, \mathcal{D}) = \int_{\mathcal{W}} p(y \mid x, w) p(w \mid \mathcal{D}) \, dw \tag{9}$$

.

And the uncertainty of the BMA predictive distribution is commonly measured by the entropy $\mathbb{H}[p(y \mid x, \mathcal{D})]$. It refers to the total uncertainty, which can be decomposed into an aleatoric and an epistemic part. The BMA predictive entropy is equal to the posterior expectation of the cross-entropy between the predictive distribution of candidate models and the BMA, using Eq. 7.

Expected uncertainty when selecting a model w:

$$\mathrm{CE}[p(y \mid x, w), p(y \mid x, \mathcal{D})] = \mathbb{H}[p(y \mid x, w)] + D_{\mathrm{KL}}(p(y \mid x, w) \parallel p(y \mid x, \mathcal{D})) \tag{10}$$

Taking an expectation of w on Eq. 10 results in uncertainty formulation:

$$\begin{aligned} \underbrace{\mathbb{H}[p(y \mid x, \mathcal{D})]}_{\text{total uncertainty}} &= \mathbb{E}_{p(w|\mathcal{D})} \left[\mathrm{CE}[p(y \mid x, w), p(y \mid x, \mathcal{D})] \right] \\ &= \underbrace{\mathbb{E}_{p(w|\mathcal{D})} \left[\mathbb{H}(p(y \mid x, w)) \right]}_{\text{aleatoric uncertainty}} + \underbrace{\mathbb{E}_{p(w|\mathcal{D})} \left[D_{\mathrm{KL}}(p(y \mid x, w) \parallel p(y \mid x, \mathcal{D})) \right]}_{\text{epistemic uncertainty}} \end{aligned} \tag{11}$$

Or, writing in the form of mutual information and conditional entropy as:

$$\mathbb{H}[y \mid x, \mathcal{D}] = \mathbb{H}[y \mid x, w] + I[y; w \mid x, \mathcal{D}] \tag{12}$$

A.3 Application Background Discussion: VLM Binding (Compositionality) Problem

One major settings in our paper, which we refer to as VLM binding problem, has profound background. We provide a brief review here. We abstract this compositionality issue as the problem of binding objects with their multiple attributes.

Recent research has revealed that large-scale pretrained VLMs struggle with understanding compositionality in images (Zeng et al., 2024; Kamath et al., 2024). They exhibit limitations in integrating objects with their attributes and understanding spatial relationships (Rahmanzadehgervi et al., 2024; Kamath et al., 2023). We abstract this compositionality issue as the binding problem of objects and their multiple dimensions of attributions. When a model has binding issues between attributes and objects, it can lead to severe hallucinations, such as failing to distinguish between "`the grass is eating the horse`" and "`the horse is eating the grass`" (binding of the object and predicate), which appears absurd to humans. Even state-of-the-art VLMs can easily make

errors in determining which object is on the left and which is on the right (binding of position). Although the model can correctly identify two objects in an image, it often confuses them when referring to attributes like color or shape. Some works have analyzed this issue from the perspectives of flaws in pretraining data and model priors (Yuksekgonul et al., 2023; Trusca et al., 2024), but a systematic quantitative explanation is lacking.

Future works. We can also brainstorm some further applications within the uncertainty decomposition framework, listed in Table 7.

Name	σ_i and $\tilde{w}$	**Formula**
Traditional Training Decomposed into Two Terms	$x, w, \mathcal{D}$	$\mathbb{H}[\hat{y} \mid x, \mathcal{D}] = \mathbb{E}_{p(w\mid\mathcal{D})}[\mathbb{H}[\hat{y} \mid x, w]]$ $+I[\hat{y}; w \mid x, \mathcal{D}]$
Adding In-Context Learning to the Traditional Decomposition	$x, w, c, \mathcal{D}$	$\mathbb{H}[\hat{y} \mid x, c, \mathcal{D}] = \mathbb{H}[\hat{y} \mid x, w]$ $+I[\hat{y}; c \mid x, w] + I[\hat{y}; w \mid x, c, \mathcal{D}]$
Chain-of-Thought Inference Time Decomposition	$x, w, h, c, \mathcal{D}$	$\mathbb{H}[\hat{y} \mid x, c, \mathcal{D}] = \mathbb{H}[\hat{y} \mid x, h, w]$ $+I[\hat{y}; h \mid x, w, c] + I[\hat{y}; c \mid x, w] + I[\hat{y}; w \mid x, c, \mathcal{D}]$
Decomposition During Model Training Based on Model Architecture	$x, w, e, l, \mathcal{D}$	$\mathbb{H}[\hat{y} \mid x, \mathcal{D}] = \mathbb{H}[\hat{y} \mid x, e, l]$ $+I[\hat{y}; l \mid x, e, \mathcal{D}] + I[\hat{y}; e \mid x, \mathcal{D}]$
Model Merging	$x, w_1, w_2, \mathcal{D}$	$\mathbb{H}[\hat{y} \mid x, \mathcal{D}] = \mathbb{H}[\hat{y} \mid x, w_1, w_2]$ $+I[\hat{y}; w_1 \mid x, w_2, \mathcal{D}] + I[\hat{y}; w_2 \mid x, \mathcal{D}]$
Visual-Language Models with Multimodal Inputs	$x, v, w, \mathcal{D}$	$\mathbb{H}[\hat{y} \mid x, v, \mathcal{D}] = \mathbb{H}[\hat{y} \mid x, v, w]$ $+I[\hat{y}; v \mid x, w] + I[\hat{y}; w \mid x, v, \mathcal{D}]$
Incorporating Decoding Strategies in Language Generation	$x, w, h, c, \mathcal{D}, d$	$\mathbb{H}[\hat{y} \mid x, c, \mathcal{D}, d] = \mathbb{H}[\hat{y} \mid x, h, w, d]$ $+I[\hat{y}; d \mid x, h, w] + I[\hat{y}; h \mid x, w, c] + I[\hat{y}; w \mid x, c, \mathcal{D}]$
Adversarial Examples in Model Robustness	$x', w, \mathcal{D}$	$\mathbb{H}[\hat{y} \mid x', \mathcal{D}] = \mathbb{H}[\hat{y} \mid x', w]$ $+I[\hat{y}; x' \mid x, w] + I[\hat{y}; w \mid x', \mathcal{D}]$
Transfer Learning and Fine-Tuning	$x, w_{\text{pre}}, w_{\text{new}}, \mathcal{D}_{\text{pre}}, \mathcal{D}_{\text{new}}$	$\mathbb{H}[\hat{y} \mid x, \mathcal{D}_{\text{new}}, \mathcal{D}_{\text{pre}}] = \mathbb{H}[\hat{y} \mid x, w_{\text{pre}}, w_{\text{new}}]$ $+I[\hat{y}; w_{\text{new}} \mid x, w_{\text{pre}}, \mathcal{D}_{\text{new}}]$ $+I[\hat{y}; w_{\text{pre}} \mid x, \mathcal{D}_{\text{pre}}]$
Incorporating User Feedback in Reinforcement Learning	$x, w, u, \mathcal{D}$	$\mathbb{H}[\hat{y} \mid x, u, \mathcal{D}] = \mathbb{H}[\hat{y} \mid x, w]$ $+I[\hat{y}; u \mid x, w] + I[\hat{y}; w \mid x, u, \mathcal{D}]$

Table 7: Further applications of the uncertainty decompositions in different scenarios. Symbols such as x represent the input data, w denote the model parameters or weights, and $\mathcal{D}$ signify the dataset. Additional symbols like c (context), h (hidden states), e (encoder), l (later-layers behind the encoder), d (decoding strategies), v (visual inputs), and u (user feedback) are introduced to capture specific elements relevant to each application.

B Discussion On the Framework

In this section we discussion about details when applying our framework, using the setting in the running example presented in Sec. 2.2.

B.1 Intermediate Variables Designs

When first examining our uncertainty decomposition framework (e.g., Fig. 1), it may seem that intermediate variables, such as rephrasings (r) and attributions referred to (a), are arbitrarily added and inflate the total uncertainty. However, these variables are intrinsic to model operation and crucial for capturing prediction variability and uncertainty.

Ideally, the model's predictions should remain consistent regardless of variations in q and c. However, models often show sensitivity to these variations, causing output fluctuations. Including these intermediate variables in our decomposition explicitly captures the uncertainty resulting from this sensitivity.

These variables are not artificial constructs but integral to the model interaction. Every model query involves specific prompt rephrings r and context attributions a, naturally treated as random variables from underlying distributions rather than fixed inputs. This treatment reflects real-world usage and allows us to model the variability introduced by differing prompt formulations and contexts.

Our approach aligns with prompt optimization and ensembling techniques. Traditional prompt optimization seeks the best prompt instance for model performance, while we generalize by treating r as a distribution, quantifying uncertainty across prompt variations. Similarly, ensembling aggregates predictions across configurations to enhance robustness; our framework systematically accounts for uncertainty at this level by considering distributions over intermediate variables.

Explicitly modeling intermediate variables within uncertainty decomposition deepens our understanding of factors affecting predictions. It reveals model sensitivity to prompt or context shifts, essential for improving reliability. This method also highlights areas needing further training or refinement to enhance robustness.

B.2 Discussion of Decomposition Practice

About Multivariate Mutual Information. The extension of mutual information between 3 or more variables is an open question (Liang et al., 2023; McGill, 1954), so we not bother on introducing more complex decomposition which involves multivariate mutual information as in such scenarios the physical meaning of the decomposed terms remains unclear. However, we acknowledge this direction as worthy exploration.

About Decomposition Order. In applying the chain rule of conditional entropy for our uncertainty decomposition such as in Eq. 4, a natural question arises regarding the order in which we perform the decomposition: should we first condition on the context c or the prompt rephrasing q? The answer to this question is critical because the decomposition order affects the interpretation of the uncertainty components and must reflect the underlying conditional dependencies among the variables.

The appropriate decomposition order is determined by the conditional independence relationships among the variables. Conditional independence dictates how variables influence each other and, consequently, how uncertainty propagates through the model. When variables are conditionally independent given certain conditions, the order of decomposition should respect these relationships to ensure that each term accurately represents its contribution to the total uncertainty.

C Implementation Details and Examples

We present the details of experiments here.

C.1 VLM Attribution Binding Task

The Dataset Synthesis. In this dataset generation process, we use the `ManiSkill` simulator[1] (Tao et al., 2024), which is primarily designed for robot arm manipulation tasks. To adapt it to our needs, we remove the robot arm and transform it into a tabletop manipulation environment. This setup enables us to generate synthetic images that meet our controlled requirements for quantifying uncertainty in the variables a (attribution) and r (rephrasing) with minimal bias.

The dataset consists of simple scenes with two objects that differ in size and other attributes. To introduce controlled variation, we adjust several variables in each scene, including color, shape, camera position, background, and object size. Each variable has a set of options, such as nine color choices (rainbow colors plus black and white). Using nested for-loops, we exhaustively combine all options across these variables, resulting in over 1,000 unique data points.

About the Prompts We present the prompts used to generate intermediate variables a and r, along with the prompt used to evaluate the answers.

About Finetuning We use the official repo for InternVL[2] (Chen et al., 2024) for fine-tuning InternVL-2-4B. The hyperparameters are listed in Table 8.

[1] `https://github.com/haosulab/ManiSkill`
[2] `https://github.com/OpenGVLab/InternVL`

The question is asking about a fact in the image.
Please identify the fact it asks, and rephrase the question while keeping the fact the same but refer to different attribution in the image.
First reason about what are the attributions of the nouns in the question, and then rephrase the question.
For example, you can replace the nouns in the question using their color, shape, position, state, or alternate names, thus creating different questions.

Image: {{image}}

Question: {{question}}

Fact:

Rephrased questions:
1.
2.
3.
4.
5.

Figure 5: VLM Attribution prompt. It is for getting those variances of attribution a for any specific question sample x. The "fact" is just to improve Langfun's performance.

Your task is to determine if the model response is correct given the question and groundtruth response.
Ensure to interpret the model response in accordance to the the question.

If the question asks about the comparison of occupancy of two objects, if the groundtruth is A is bigger than B. Then the desired answer is either: "A is bigger than B" or "B is smaller than A". Both answers are correct.

If the question asks about a detail of an element that is not present in the image, A prediction of "yes", "no" or "nothing" should be considered incorrect because it inaccurately suggests that the element is presented in the image.
The correct prediction in such cases should acknowledge the absence of the element in question by stating the element is not present.
If prediction says that it can not assist or cannot provide an answer, then the prediction is incorrect.
If the question is about counting, then the prediction is correct only it matches the groundtruth counts exactly.

question={{question}},
model_response={{model_response}}
groundtruth_response={{groundtruth_response}},

Figure 6: VLM Attribution labeling prompt

C.2 LLM Reasoning Task

In this section, we detail the prompts utilized for the LLM reasoning tasks, as illustrated in Fig. 7 8. Additionally, Table 9 provides example bodies and their corresponding questions, offering a clear overview of the task setup and the types of problems addressed in our experiments.

About the Prompts

Table 8: Key Training Hyperparameters

Hyperparameter	Value
Total Batch Size	32
Number of Epochs	2
Learning Rate	6×10^{-6}
Weight Decay	0.05
Warmup Ratio	0.03
Learning Rate Scheduler	Cosine
Model Name or Path	`OpenGVLab/InternVL2-4B`
Image Size	448
Max Sequence Length	4096
Mixed Precision (bf16)	Yes
Gradient Checkpointing	Yes
Zero Optimization Stage	zero_stage1
Freeze (Vision) Backbone	True
Vision Select Layer	-1

Given the following math problem consisting of a Body and a Question, please rephrase it by replacing entity names which are irrelevant to the underlying mathematical reasoning. Keep all numbers and the core mathematical structure intact.

Original Body: {{body}}
Original Question: {{question}}

Please provide a rephrased version with different entity names but the same mathematical structure:

Rephrased Body:
Rephrased Question:

Figure 7: LLM Reasoning entity replacing prompt; The "body" and "question" are those corresponding keys in the SVAMP dataset.

C.3 More Analysis on the Calibration Test

In this subsection, we provide a comprehensive analysis of the calibration tests conducted across two distinct tasks, as illustrated in Fig. 9 and 10. These scatter plots demonstrate that when total uncertainty is employed, the model exhibits a pronounced overconfidence phenomenon, with numerous samples simultaneously showing high error rates and low uncertainty. Specifically, Fig 9 pertains to the VLM attribution binding task, while Fig 10 relates to the LLM math reasoning task. The consistent observation of high errors paired with low uncertainty across both tasks underscores the significant limitations of using total uncertainty alone. This finding highlights the critical importance and necessity of uncertainty decomposition, which enables a more nuanced and accurate calibration of model confidence, thereby enhancing the reliability and robustness of predictive performance.

Your task is to determine if the model response is correct given the question and groundtruth response. Ensure to interpret the model response in accordance to the the question.

If the question asks about the comparison of occupancy of two objects, if the groundtruth is A is bigger than B. Then the desired answer is either: "A is bigger than B" or "B is smaller than A". Both answers are correct.

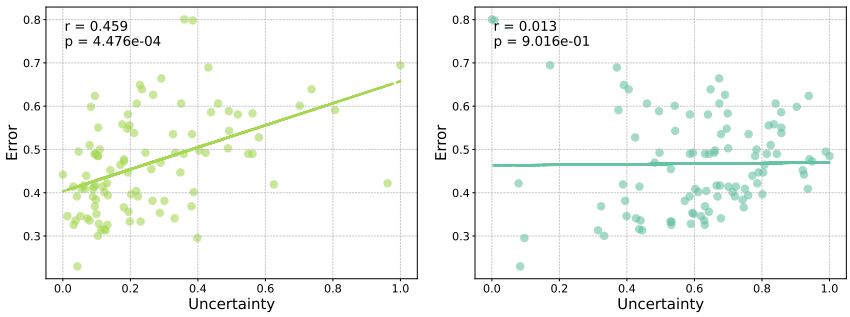
If the question asks about a detail of an element that is not present in the image, A prediction of "yes", "no" or "nothing" should be considered incorrect because it inaccurately suggests that the element is presented in the image.
The correct prediction in such cases should acknowledge the absence of the element in question by stating the element is not present.
If prediction says that it can not assist or cannot provide an answer, then the prediction is incorrect.
If the question is about counting, then the prediction is correct only it matches the groundtruth counts exactly.

question={{question}},
model_response={{model_response}}
groundtruth_response={{groundtruth_response}},

Figure 8: LLM Reasoning labeling prompt

Bodies	**Questions**
Paige raised 7 goldfish and 12 catfish in the pond but stray cats loved eating them. Now she has 15 left.	How many fishes disappeared?
Tom raised 7 rabbits and 12 hamsters in the yard but wild foxes loved chasing them. Now he has 15 left.	How many pets vanished?
Lisa raised 7 puppies and 12 kittens in the shelter but stray dogs loved bothering them. Now she has 15 left.	How many animals went missing?
Mark raised 7 ducks and 12 geese in the pond but hungry raccoons loved stealing them. Now he has 15 left.	How many birds were lost?
Emily raised 7 turtles and 12 frogs in the aquarium but curious cats loved disturbing them. Now she has 15 left.	How many creatures disappeared?
Jake raised 7 turtles and 12 slugs in the garden but wandering snails loved tasting them. Now he has 15 left.	How many critters went away?

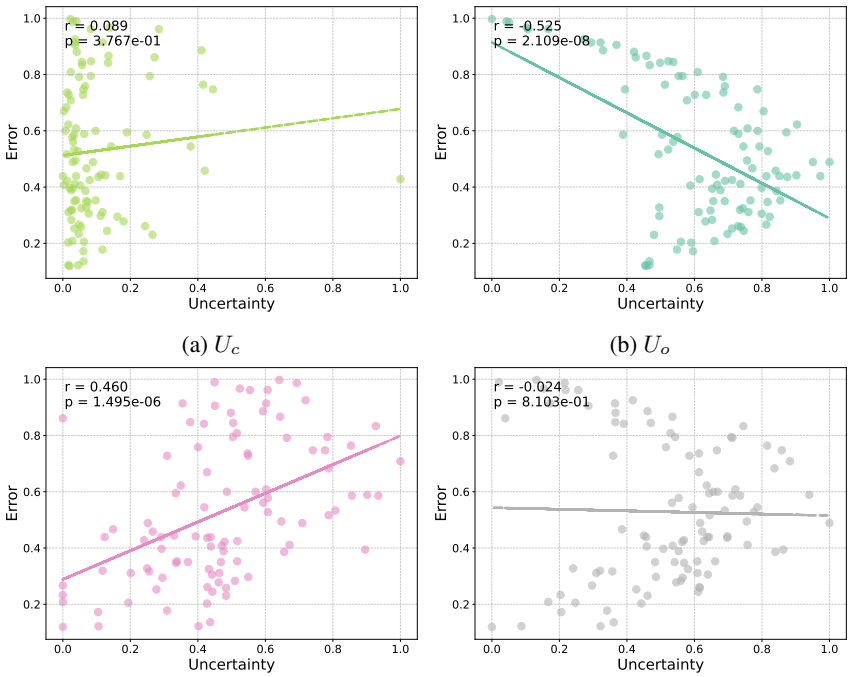
Table 9: Example Bodies and Corresponding Questions



(a) U_c (b) U_o

(c) U_q (d) U_t

Figure 9: Scatter plots between error rate and uncertainty parts U_c, U_o, U_q, and U_t in the VLM attribution binding task. We calculate the Pearson r and p-value to show the correlation relationships. U_c is a effective calibrator while others behave poorly.



(a) U_c (b) U_o

(c) U_q (d) U_t

Figure 10: Scatter plots between error rate and uncertainty terms U_c, U_o, U_q, and U_t in the LLM math reasoning task. We calculate the Pearson r and p-value to show the correlation relationships. U_q is a effective calibrator while others behave poorly.