SPECIFICATION-AWARE PREDICTIVE UNCERTAINTY DECOMPOSITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Classical decompositions of predictive uncertainty, such as the information-theoretic one, are defined for a fixed representation x of the input object. They do not consider that x may be incomplete, nor do they measure how much the missing information can affect the prediction. In Large Language Models, users can clarify an underspecified input without any retraining. The clarified input can describe the user's unobserved intent z^* more accurately and change the predictive uncertainty about the outcome. In this paper, we formalize this setting probabilistically and derive a three-term decomposition of prediction risk into *specification*, residual aleatoric, and epistemic uncertainties. We show that at the original information level, when only x is available, the aleatoric uncertainty, represented by a Bayes risk, has two components. The first, which we call specification uncertainty, is the expected reduction in this Bayes risk from revealing the intent. The second is the residual aleatoric uncertainty, i.e., the Bayes risk that remains once the intent is known, averaged over its possible values z given x. The distinction is important, as specification uncertainty informs the decision to request clarification before making a prediction. We build approximations of these components and connect them to existing uncertainty measures. Our experiments across different domains and modalities show distinct roles for the three components. Specification uncertainty is the most informative for detecting underspecified inputs, while residual aleatoric uncertainty is the most informative for detecting in-distribution prediction errors.

1 INTRODUCTION

As artificial intelligence (AI) systems, especially generative ones, are widely adopted (Bick et al., 2026), there is a growing demand for methods and practices to control their reliability. One such direction is predictive Uncertainty Quantification (UQ; Hüllermeier & Waegeman, 2021), which supplements the original model's output with a value (typically a scalar) describing the model's certainty in its prediction. Based on this value, a practitioner can make a practical decision, such as to abstain from the prediction, request additional information, or proceed with the prediction if the uncertainty is acceptable (Geifman & El-Yaniv, 2017; Mozannar & Sontag, 2020; Naszadi et al., 2023). For Large Language Models (LLMs) (Brown et al., 2020), requesting additional information is particularly natural, since a user can clarify the input without retraining the model. For example, given the prompt "Who won the race?", a model may either answer right away or first ask which race is meant (see Figure 1). Therefore, uncertainty in prediction may come from different *sources*, and different sources may call for different actions (Sale et al., 2026).

Two *types* of uncertainty are widely distinguished (Hüllermeier & Waegeman, 2021). *Aleatoric* uncertainty is irreducible with respect to the considered informational constraints, while *epistemic* uncertainty is reducible and characterizes the lack of knowledge about the underlying ground-truth data-generation process. Different formalizations of these types lead to concrete UQ measures (Gal & Ghahramani, 2016; Kendall & Gal, 2017; Lakshminarayanan et al., 2017; Kotelevskii et al., 2025; 2022; Sale et al., 2023; 2024; Hofman et al., 2024; Schweighofer et al., 2025).

For LLMs, many UQ methods have been proposed (Lin et al., 2024; Farquhar et al., 2024; Tian et al., 2023; Vashurin et al., 2025), including decompositions into aleatoric and epistemic uncertainty (Malinin & Gales, 2021; Ma et al., 2025). However, these decompositions are defined for a given input representation and do not ask whether clarifying the input could improve the prediction.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

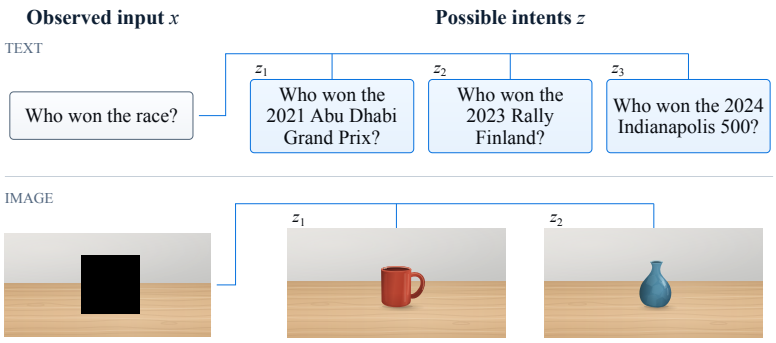

Figure 1: Examples of observed inputs x (the ones provided by users) and possible values of the unobserved intent z. Different intended questions (top) or complete images (bottom) can be consistent with the same observed input. The actual value z^* remains unobserved.

Recent work considered uncertainty arising from *underspecified* inputs (Hou et al., 2024; Walha et al., 2026). These approaches generate clarifications of the input and treat the variability of predictions across them, quantified by Bregman information, as aleatoric uncertainty, and the expected uncertainty remaining after clarification as epistemic uncertainty. However, the outcome can be random even for a fully clarified input, so the remaining uncertainty is not necessarily epistemic. Moreover, both approaches use a single fixed model and thus ignore uncertainty about its parameters, which the information-theoretic decomposition treats as epistemic.

We present an alternative view on this distinction. We assume that each observed input x (possibly only partially) describes the true intent z^*, which itself is never observed, and ask how much the prediction would improve by revealing this intent. This leads to a three-term decomposition of prediction risk into *Specification*, *Epistemic*, and *residual Aleatoric* uncertainty (SEA). Specification uncertainty is the expected reduction in Bayes risk from revealing the intent, that is, the expected gain from clarification. When it is large, it is worth requesting clarification before providing an answer. At the original information level (i.e., when only x is available), specification uncertainty is *only a part* of the aleatoric uncertainty.

We summarize our contributions below.

- We derive a new UQ decomposition, relevant in particular for LLMs, that splits the predictive uncertainty into three terms by separating specification uncertainty from both the residual aleatoric uncertainty and the epistemic one. When the input fully specifies the intent, the specification term vanishes, and the decomposition reduces to the risk-based one (Kotelevskii et al., 2025), generalizing the information-theoretic uncertainty decomposition; see Section 3.1.
- For a specific class of loss functions, proper scoring rules, we derive closed-form uncertainty measures and, following the risk-based approach, their Bayesian approximations. These approximations generalize the clarification-based decompositions previously derived for a single fixed model (Hou et al., 2024; Walha et al., 2026); see Section 3.2.
- In our experiments, we demonstrate that each of the derived components is well-suited for a particular type of decisions in downstream problems. Specification uncertainty is the most informative of the three components for detecting underspecified inputs, while residual aleatoric uncertainty is the most informative component for detecting prediction errors for in-distribution data; see Section 4 and Appendix K.

2 Related Work

Predictive uncertainty decomposition. There are several approaches to formalize the decomposition into aleatoric and epistemic uncertainty and estimate the corresponding components. The information-theoretic approach considers a Bayesian (or, more generally, second-order) model. It associates the conditional mutual information between predictions and model parameters with epistemic uncertainty,

and the expected entropy of the prediction given the parameters with aleatoric uncertainty (Houlsby et al., 2011). Estimating the epistemic term, therefore, requires a second-order learner, such as a Bayesian model or an ensemble (Sale et al., 2023; 2024). The risk-based formulation generalizes this approach. It decomposes the pointwise prediction risk into the Bayes risk (aleatoric uncertainty) and the excess risk (epistemic uncertainty). These components can be approximated with Bayesian models (Kotelevskii et al., 2025; Schweighofer et al., 2025), which recover the decomposition of Houlsby et al. (2011) as a special case. Another strategy for approximating these terms uses specialized deterministic auxiliary models (Kotelevskii et al., 2022; Lahlou et al., 2023). Other methods estimate uncertainty with a single deterministic network by modifying its architecture or fitting a density model on its features (Van Amersfoort et al., 2020; Liu et al., 2020; Mukhoti et al., 2021). For autoregressive generation, the information-theoretic decomposition has been extended to the token and sequence levels (Malinin & Gales, 2021), still requiring a Bayesian model or an ensemble for estimation. Another approach, LogTokU (Ma et al., 2025), estimates aleatoric and epistemic uncertainties from logits without repeated sampling by interpreting them as evidence for a Dirichlet distribution.

These decompositions quantify uncertainty at a fixed level of input information, when the representation of an input object (or its prompt) x is given. They do not consider how the risk and the predictive uncertainty would change if x were clarified. Interactive LLMs, however, allow users to clarify their inputs without retraining the model, which makes this question natural.

Specification uncertainty. Sale et al. (2026) argue that the aleatoric/epistemic distinction depends on the information available in a given setting, for example, on whether an ambiguous prompt can be clarified. Hou et al. (2024) propose input-clarification ensembling, which decomposes predictive uncertainty using generated clarifications of the prompt. In their decomposition, the mutual information between the prediction and the clarification is treated as aleatoric uncertainty caused by input ambiguity, while the expected entropy of the response given a clarification is treated as epistemic uncertainty. Walha et al. (2026) keep this interpretation and generalize it with functional Bregman information, instantiated with kernel-based von Neumann entropy. Section 3 shows that their decomposition is a special case of ours for a single fixed model and that its terms estimate specification and residual aleatoric uncertainty. In contrast, we split the Bayes risk at the level of x into two parts. Specification uncertainty is the expected reduction in Bayes risk from revealing the intent, and residual aleatoric uncertainty is the expected Bayes risk after the intent is revealed. The excess risk of the predictor is still treated as epistemic uncertainty.

3 Risk Decomposition with Unobserved Intents

In this work, we consider a setup in which the observed representation $x \in \mathcal{X}$ of an object may be incomplete and can be clarified by additional information. Our goal is to quantify how much such a clarification is expected to improve the prediction, and to separate this gain from other sources of predictive uncertainty.

In what follows, by *intent*, we mean a fully specified input that x describes only partially. We denote the user's true but unobserved intent by $z^* \in \mathcal{Z}$. For example, a user may have a specific question z^* in mind but write a less specific prompt x. Similarly, x could be a masked image or an audio recording with a missing interval, and z^* the original image or the full recording. Several intents can be consistent with the same x (see Figure 1), and the conditional distribution $p(z \mid x)$ describes which values z^* could take. In these examples, $\mathcal{Z} = \mathcal{X}$, since a clarified prompt is again a prompt, and a completed image or recording has the same format as the masked one. Our framework, however, does not require the two spaces to coincide.

We assume that clarification does not change the output space $\mathcal{Y}$, as otherwise the prediction risks before and after clarification could not be compared. For a classification task, this means that the set of output classes must be the same for x and z. Finally, we assume that once z is known, x provides no additional information about Y. Formally, $Y \perp X \mid Z$, i.e., $p(y \mid x, z) = p(y \mid z)$.

Under these assumptions, we consider the following *ground-truth* process

$$p(y, z \mid x) = p(y \mid z)\, p(z \mid x). \tag{1}$$

Note that, for uncertainty quantification, we consider a distribution already conditioned on a specific x, as we are typically interested in evaluation at a particular point. In what follows, we write $p_z \coloneqq p(y \mid z)$ and $p \coloneqq p(y \mid x) = \mathbb{E}_{Z \sim p(z|x)}\, p_Z$ for clarity.

3.1 THREE-COMPONENT DECOMPOSITION

Let $\ell(q, y)$ denote a proper scoring rule (Gneiting & Raftery, 2007) that evaluates a predictor $q \coloneqq q(y \mid x)$ at Y. We define the expected pointwise risk of q as

$$R(p, q) \coloneqq \mathbb{E}_{Y \sim p}[\ell(q, Y)].$$

We can decompose this expected risk into the Bayes risk $R_b(p) \coloneqq R(p, p)$ of the process p and the excess risk $R_{\text{exc}}(p \parallel q) \coloneqq R(p, q) - R(p, p)$ of the model q with respect to p. Following Kotelevskii et al. (2025), for a proper scoring rule instantiated with a convex function $G\colon \mathcal{P}(\mathcal{Y}) \to \mathbb{R}$, this gives

$$R(p, q) = R_b(p) + R_{\text{exc}}(p \parallel q) = -G(p) + D_G(p \parallel q), \tag{2}$$

where $D_G(p \parallel q) = G(p) - G(q) - \langle G'(q)\,,\, p - q\rangle$ is the Bregman divergence instantiated by G. Here, $-G(p)$ is the Bayes risk. It depends only on the data generation process p, so changing the model does not affect it. This term corresponds to aleatoric uncertainty, which is *irreducible*. The excess risk $D_G(p \parallel q)$ depends on the model, its training data, and the training procedure. It can therefore be reduced by improving the model and corresponds to epistemic uncertainty, which is *reducible*. Reducibility here means that the predictor can be improved while the available input information is kept fixed.

Interestingly, the Bayes risk $-G(p)$ can be split further by using the ground-truth process (1). Since the true intent z^* is unobserved, we treat it as a random variable $Z \sim p(z \mid x)$. Then, using p_z and p defined above, we can expand equation (2) as follows (see Appendix B for the derivation):

$$R(p, q) = \overbrace{\underbrace{\mathbb{E}_{Z \sim p(\cdot|x)}[-G(p_Z)]}_{\text{residual aleatoric}} + \underbrace{\mathbb{E}_{Z \sim p(\cdot|x)}[D_G(p_Z \parallel p)]}_{\text{specification}}}^{-G(p)} + \underbrace{D_G(p \parallel q)}_{\text{epistemic}}. \tag{3}$$

Interpretation. As in prior work (Kotelevskii et al., 2025; 2022; Lahlou et al., 2023), the aleatoric uncertainty in equation (2) is defined relative to the input x. Sale et al. (2026) argue that the split into reducible and irreducible uncertainty depends, among other things, on the features available as input. In particular, uncertainty that is irreducible given x can still be reduced by providing more input information, such as the intent. Therefore, both $p(y \mid x)$ and $p(y \mid z)$ determine ground-truth aleatoric uncertainty, *but relative to different fixed information levels.*

For the joint process $p(y, z \mid x)$ that we assume here, $p := p(y \mid x)$ and $p_z := p(y \mid z)$ describe the predictive distribution before and after the intent is revealed. Namely, the best prediction based only on x is p, and its Bayes risk is $-G(p)$. If the intent is revealed and turns out to be z, the best prediction is p_z, with Bayes risk $-G(p_z)$. Thus, before the intent is revealed, the expected Bayes risk of a predictor that will observe it is $-\mathbb{E}_{Z \sim p(\cdot|x)}[G(p_Z)]$. We call this term *residual aleatoric uncertainty*.

The difference between these two Bayes risks is the Bregman information term in equation (3):

$$\underbrace{-G(p)}_{\text{before clarification}} \quad \underbrace{-\big(-\mathbb{E}_{Z \sim p(\cdot|x)}[G(p_Z)]\big)}_{\text{after clarification}} = \mathbb{E}_{Z \sim p(\cdot|x)}\big[D_G(p_Z \| p)\big].$$

It shows how much the Bayes risk is expected to decrease once the intent is revealed, i.e., how much we expect to gain from clarification. We call this term *specification uncertainty*. Like the Bayes risk, specification uncertainty depends only on the ground-truth process. If x fully specifies the intent (i.e., $p(z \mid x)$ is degenerate), specification uncertainty vanishes, and equation (3) reduces to equation (2).

Each term in equation (3) has a clear practical meaning, as it corresponds to a specific action. In expectation, clarification reduces the Bayes risk through the specification term, and improving the predictor reduces the epistemic term. Neither action reduces the residual aleatoric term, so the only way to avoid this part of the risk is to abstain from prediction.

3.2 BAYESIAN APPROXIMATION

In practice, the ground-truth process $p(y, z \mid x)$ in equation (1) is unknown, and we need to approximate it. To keep generality and show the connection with known decompositions in the literature, we use the following Bayesian model for the approximation:

$$q(y, z, \theta \mid x, D) = q(y \mid z, \theta)\, \pi(\theta, z \mid x, D),$$

where $D = \{(x_i, y_i)\}_{i=1}^N$ is the training dataset. The model has two latent variables, the parameters θ and the intent z, and the joint distribution $\pi(\theta, z \mid x, D)$ captures uncertainty about them.

The components of equation (3) depend on the unknown distributions $p(z \mid x)$ and p_z. To obtain computable and practical uncertainty measures, we replace them with approximations. We describe how we obtain them at the end of this section.

These approximations can be incorporated in several ways (Kotelevskii et al., 2025), and we discuss them in Appendix A. Here in the main part, we use the one that recovers the standard information-theoretic decomposition, as we show below. It treats each posterior sample of the parameters as a candidate for the ground-truth process and uses the posterior predictive distribution as the predictor. Let $q_{z,\theta} \coloneqq q(y \mid z, \theta)$, and define

$$q \coloneqq \mathbb{E}_\pi[q_{Z,\Theta}] = q(y \mid x, D), \qquad q_\theta \coloneqq \mathbb{E}_\pi[q_{Z,\Theta} \mid \Theta = \theta],$$

where the expectations are computed under $\pi(\theta, z \mid x, D)$. The distribution q is the posterior predictive distribution of this model and specifies the predictor introduced above. Averaging the risk over the candidates q_θ, we estimate $R(p, q)$ by $\mathbb{E}_\Theta[R(q_\Theta, q)] = -G(q)$. Decomposing the Bayes risk of each q_θ, we obtain estimates of the three components of equation (3):

$$-G(q) = \underbrace{-\mathbb{E}_{Z,\Theta}[G(q_{Z,\Theta})]}_{\text{residual aleatoric}} + \underbrace{\mathbb{E}_{Z,\Theta}[D_G(q_{Z,\Theta} \parallel q_\Theta)]}_{\text{specification}} + \underbrace{\mathbb{E}_\Theta[D_G(q_\Theta \parallel q)]}_{\text{epistemic}}. \tag{4}$$

For $G(q) = \sum_i q_i \log q_i$, equation (4) becomes

$$\begin{aligned} \mathbb{H}(q) &= \overbrace{\underbrace{\mathbb{E}_{Z,\Theta}[\mathbb{H}(q_{Z,\Theta})]}_{\mathrm{SEA}_{\mathrm{res}}} + \underbrace{\mathbb{E}_{Z,\Theta}[KL(q_{Z,\Theta} \parallel q_\Theta)]}_{\mathrm{SEA}_{\mathrm{spec}}}}^{\text{aleatoric uncertainty estimate}} + \underbrace{\mathbb{E}_\Theta[KL(q_\Theta \parallel q)]}_{\mathrm{SEA}_{\mathrm{epi}}} \\ &= \mathbb{E}_\Theta[\mathbb{H}(q_\Theta)] + I(Y; \Theta \mid X = x, D), \end{aligned} \tag{5}$$

where the second equality holds because q_θ averages $q_{z,\theta}$ over intents, so the first two terms sum to $\mathbb{E}_\Theta[\mathbb{H}(q_\Theta)]$. The second line is therefore the standard information-theoretic predictive uncertainty decomposition, applied to the model with the intent marginalized out. Its epistemic term is the conditional mutual information, which is used as the BALD acquisition score (Houlsby et al., 2011). Below, we denote the three terms by $\mathrm{SEA}_{\mathrm{res}}$, $\mathrm{SEA}_{\mathrm{spec}}$, and $\mathrm{SEA}_{\mathrm{epi}}$, and their sum $\mathbb{H}(q)$ by $\mathrm{SEA}_{\mathrm{total}}$. The standard decomposition merges $\mathrm{SEA}_{\mathrm{res}}$ and $\mathrm{SEA}_{\mathrm{spec}}$ into one aleatoric term, so it cannot show whether clarifying x would reduce the risk. We discuss limitations of our approach in Appendix C.

Connection to previous work. Our work is closely related to Hou et al. (2024) and Walha et al. (2026), which both consider a single fixed model and generate clarifications of its input. If we set $\theta = \text{const}$ and write $q_z \coloneqq q_{z,\theta}$, then equation (4) simplifies to

$$-G(q) = \underbrace{-\mathbb{E}_Z[G(q_Z)]}_{(ii)} + \underbrace{\mathbb{E}_Z[D_G(q_Z \parallel q)]}_{(i)}. \tag{6}$$

A specific choice of G gives equation 9 of Walha et al. (2026). For $G(q) = \sum_i q_i \log q_i$, we get $\mathbb{H}(q) = \mathbb{E}_Z[KL(q_Z \parallel q)] + \mathbb{E}_Z[\mathbb{H}(q_Z)]$, which is exactly equation 3 of Hou et al. (2024).

Both works interpret term (i) as aleatoric uncertainty and term (ii) as epistemic uncertainty. To see what these terms measure, note that equation (6) holds for any predictor, including the ideal one with $\pi(z \mid x) = p(z \mid x)$ and $q_z = p_z$ for all z. Then $q = p$, the excess risk $D_G(p \parallel q)$ vanishes, and equation (6) becomes

$$-G(p) = -\mathbb{E}_Z[G(p_Z)] + \mathbb{E}_Z[D_G(p_Z \parallel p)],$$

which is exactly the split of the Bayes risk in equation (3). The crucial consequence is that term (ii) is nonzero whenever the outcome is random given the intent, even *if we have perfect knowledge* of

Table 1: Text QA underspecification-detection AUROC using semantic entropy. CoQAR-U contains 1,000 matched damaged/undamaged pairs, with damaged inputs positive. AmbigQA contains 1,971 unique questions: 1,141 ambiguous positives and 830 unambiguous negatives.

	Gemma E2B		Gemma E4B		Heterogeneous	
Method	CoQAR-U	AmbigQA	CoQAR-U	AmbigQA	CoQAR-U	AmbigQA
$\mathrm{SEA}_{\mathrm{res}}$	0.5373	0.5102	0.5760	0.5071	0.5404	0.5350
$\mathrm{SEA}_{\mathrm{epi}}$	0.5213	0.5442	0.5303	0.5461	0.5232	0.4396
$\mathrm{SEA}_{\mathrm{spec}}$	**0.7807**	**0.6575**	**0.7709**	**0.6678**	**0.7873**	**0.6017**
$\mathrm{SEA}_{\mathrm{total}}$	0.7365	0.5169	0.7614	0.5164	0.6593	0.5250
BALD (without clarifications)						
$\mathrm{BALD}_{\mathrm{alea}}$	**0.7356**	0.5011	0.7607	0.4858	0.7706	**0.5073**
$\mathrm{BALD}_{\mathrm{epi}}$	0.6002	**0.5442**	0.6081	**0.5348**	0.7109	0.4535
$\mathrm{BALD}_{\mathrm{total}}$	0.7340	0.5015	**0.7608**	0.4865	**0.7804**	0.5003
BALD (clarification posterior)						
$\mathrm{BALD}_{\mathrm{alea}}$	**0.7413**	0.5169	**0.7627**	0.5162	**0.6721**	**0.5383**
$\mathrm{BALD}_{\mathrm{epi}}$	0.5213	**0.5442**	0.5303	**0.5461**	0.5232	0.4396
$\mathrm{BALD}_{\mathrm{total}}$	0.7365	0.5169	0.7614	0.5164	0.6593	0.5250

$p(y \mid z)$ and $p(z \mid x)$. Hence, this term becomes irreducible, and treating it as epistemic uncertainty would suggest improving a predictor that is already perfect, while the appropriate action is to abstain. Therefore, the interpretation of Hou et al. (2024) and Walha et al. (2026) is justified only when the outcome is a deterministic function of the intent. In this case, for the ideal predictor, the residual aleatoric term (ii) vanishes, and term (i) recovers the whole aleatoric uncertainty.

One may argue that for a single fixed model, the Bayesian estimate of the excess risk is zero, so equation (6) already accounts for all uncertainty that such a model can express, and term (ii) is the only term that can represent its epistemic uncertainty. However, as shown above, term (ii) estimates the residual aleatoric uncertainty. The zero estimate of the excess risk only means that a point-mass posterior cannot express epistemic uncertainty, not that the model has none. Such uncertainty can instead be estimated by non-Bayesian methods (Kotelevskii et al., 2022; Lahlou et al., 2023).

Estimation in practice. The pairs (x, y) alone determine $p(y \mid x)$, but not how its Bayes risk $-G(p)$ splits into specification and residual aleatoric uncertainty. For example, the same $p(y \mid x)$ can arise when x fully specifies the intent and specification uncertainty is zero. It can also arise when y is a deterministic function of z, and then all the randomness of y given x comes from the unknown intent. Therefore, the two terms can be estimated separately only with additional information about the intent or with assumptions on the latent process. In our experiments, this information comes from large pretrained generative models. We use them to generate possible intents, which serve as hypotheses about z^*. Since this is done independently of θ, we have $\pi(\theta, z \mid x, D) = \pi(\theta \mid D)\pi(z \mid x)$. Since these intents are only hypotheses, the quality of the estimates depends on how well $\pi(z \mid x)$ approximates the true $p(z \mid x)$. Similarly, the ensembles that we use as samples of θ only approximate the posterior $\pi(\theta \mid D)$, especially when they consist of different pretrained models.

4 Experiments

In this section, we verify our intuition about the components presented in equation (5) and provide practical estimates across different domains, modalities, and tasks.

Specifically, we show the sensitivity of specification and residual aleatoric terms on several input modalities of LLMs - text (Section 4.1), images (Section 4.2), and audio (Section 4.3). Additionally, we evaluate the specificity of the SEA components in image classification, including out-of-distribution (OOD) detection for the epistemic term, in Appendix K.

Models. All LLM and VLM experiments used a heterogeneous ensemble of 4 models - Qwen2.5-Omni-7B (Xu et al., 2025), Gemma-4-E4B-it (Gemma Team, 2026), Phi-4-multimodal-instruct (Abouelenin et al., 2025), and MiniCPM-o-4.5 (Cui et al., 2026). For text-to-text question

Table 2: Text QA error-detection AUROC using semantic entropy and GPT-5.6 Luna correctness labels. Errors in the ensemble answer to the original input are positive for every method. Evaluation uses 1,000 undamaged CoQAR-U questions and 830 unambiguous AmbigQA questions.

	Gemma E2B		Gemma E4B		Heterogeneous	
Method	CoQAR-U	AmbigQA	CoQAR-U	AmbigQA	CoQAR-U	AmbigQA
$\mathrm{SEA}_{\mathrm{res}}$	0.7285	**0.8103**	0.7535	**0.7739**	0.7077	0.7197
$\mathrm{SEA}_{\mathrm{epi}}$	0.6573	0.6827	0.6145	0.6797	0.6730	0.5431
$\mathrm{SEA}_{\mathrm{spec}}$	0.5627	0.5279	0.6600	0.5544	0.5456	0.7234
$\mathrm{SEA}_{\mathrm{total}}$	**0.7561**	0.8062	**0.7974**	0.7710	**0.7323**	**0.7412**
BALD (without clarifications)						
$\mathrm{BALD}_{\mathrm{alea}}$	0.7665	**0.8439**	0.7800	**0.8364**	0.7438	0.7513
$\mathrm{BALD}_{\mathrm{epi}}$	0.6628	0.6334	0.5326	0.6967	0.7214	0.5775
$\mathrm{BALD}_{\mathrm{total}}$	**0.7689**	0.8435	**0.7813**	**0.8364**	**0.7515**	**0.7692**
BALD (clarification posterior)						
$\mathrm{BALD}_{\mathrm{alea}}$	0.7536	**0.8062**	0.7957	0.7709	0.7308	0.7267
$\mathrm{BALD}_{\mathrm{epi}}$	0.6573	0.6827	0.6145	0.6797	0.6730	0.5431
$\mathrm{BALD}_{\mathrm{total}}$	**0.7561**	**0.8062**	**0.7974**	**0.7710**	**0.7323**	**0.7412**

answering (QA), we used a checkpoint ensemble, fine-tuned from Gemma-E2B-it and Gemma-E4B-it, on the AmbigQA (Min et al., 2020) dataset. Full details on the fine-tuning procedure are presented in Appendix G.

4.1 TEXTQA

Data. For text-to-text QA, we evaluate the specificity of SEA terms on two datasets:

1. **CoQAR-U**. In the CoQAR dataset (Brabant et al., 2022), the system is given a text passage and a question. To model underspecified inputs, we remove the span with key information from the passage. Original passages serve as well-specified input. We use GPT-5.6 Luna as $\pi(z|x)$ and sample five clarifications for each original and damaged passage (see details in Appendix J).
2. **AmbigQA**. We also use the existing AmbigQA (Min et al., 2020) dataset, which is based on an open-domain version of Natural Questions (Kwiatkowski et al., 2019). Here, the system is given only a question, without any supporting context, and should answer based on its parametric memory. We treat questions with a single interpretation as well-specified, and questions with 2 or more interpretations as underspecified. Similar to CoQAR, we sample 5 clarifications per question using GPT-5.6 Luna.

Findings. First, Table 1 shows that the specification component indeed performs well at detecting underspecified questions, aligning with its intended interpretation. Second, we check whether the residual aleatoric component in SEA decomposition still predicts in-domain error. Table 2 shows that the residual term delivers a higher ROC-AUC compared to other components. We note that the magnitude of the residual term is often larger than that of the epistemic and specification terms; therefore, total uncertainty is mostly driven by the residual component, explaining the high performance of total uncertainty on this task.

4.2 VISUAL QA

Data. We evaluate specification uncertainty in VQA by progressively hiding image content while keeping the question and reference answer fixed. For each masked observation, five inpainted images provide alternative visual clarifications (see details in Appendix L).

1. **VQS.** The frozen evaluation subset contains 500 questions on 500 distinct images sampled from the validation split of the VQS (Gan et al., 2017) dataset. Input underspecification is modeled by applying random gray masks covering 10% and 30% of the image area. Input

Table 3: VQA AUROC of the SEA components on VQS and ImageNet, computed with the semantic entropy estimator for the heterogeneous ensemble. Underspecification detection takes 30% masking as the positive class and 10% as the negative class. Error detection uses an incorrect ensemble answer to the masked image as the positive class, reported separately at each mask size.

	Underspecification		ID error			
	VQS	ImageNet	VQS		ImageNet	
Method	30% vs 10%	30% vs 10%	10%	30%	10%	30%
$\mathrm{SEA}_{\mathrm{residual}}$	0.5422	0.5289	0.8217	0.7850	0.7448	0.7715
$\mathrm{SEA}_{\mathrm{epi}}$	0.5073	0.4907	0.7830	0.7521	0.7275	0.7334
$\mathrm{SEA}_{\mathrm{spec}}$	**0.6614**	**0.6903**	0.7256	0.7006	0.5992	0.7172
$\mathrm{SEA}_{\mathrm{total}}$	0.5486	0.5452	**0.8289**	**0.7900**	**0.7458**	**0.7756**
BALD (without clarifications)						
$\mathrm{BALD}_{\mathrm{alea}}$	**0.5363**	0.5768	0.8194	**0.8087**	0.7151	0.7653
$\mathrm{BALD}_{\mathrm{epi}}$	0.5223	0.5704	0.7968	0.7440	0.7193	0.7264
$\mathrm{BALD}_{\mathrm{total}}$	0.5332	**0.5806**	**0.8311**	0.8019	**0.7263**	**0.7742**
BALD (clarification posterior)						
$\mathrm{BALD}_{\mathrm{alea}}$	**0.5618**	**0.5578**	0.8268	0.7871	0.7393	0.7753
$\mathrm{BALD}_{\mathrm{epi}}$	0.5073	0.4907	0.7830	0.7521	0.7275	0.7334
$\mathrm{BALD}_{\mathrm{total}}$	0.5486	0.5452	**0.8289**	**0.7900**	**0.7458**	**0.7756**

clarifications were generated with FLUX.1 Fill-dev (Black Forest Labs, 2024). Answer correctness is evaluated via Exact Match with the most common normalized human answer.

2. **Imagenet.** The VQA dataset is built from ImageNet samples used in classification experiments by asking a general question, "What is depicted in the picture?" for every image. Masking and clarification construction follow the classification data construction pipeline (Appendix K.3). GPT-5.6 Luna evaluates the correctness of the answer by comparing the model's answer with the ImageNet class name.

Findings. Each SEA component is the strongest signal for its intended target. Specification uncertainty is the best underspecification detector on both datasets, while other components do not provide a useful signal for requesting a clarification (see Table 3). Conversely, error detection is dominated by total and aleatoric uncertainty; since SEA preserves the total of the standard decomposition, isolating the specification term do not harm the error detection performance. The Appendix L shows that both results hold for all six entropy estimators and all confidence baselines.

Mask size effects. The ImageNet VQA setup allows control of the degree of underspecification in the input by varying the mask size relative to the object bounding box area. Figure 2 compares how SEA and the two BALD variants split the ensemble's mean uncertainty on ImageNet as the masked fraction of the object bounding box grows.

Specification uncertainty grows monotonically with the amount of hidden content, both in absolute value and as a share of total uncertainty. It is the only component whose share systematically increases. SEA and BALD over clarifications share the same total and epistemic terms by construction, since SEA only splits the pooled aleatoric term into residual aleatoric and specification uncertainty. Under SEA, the aleatoric and epistemic shares decline with mask size, so the added uncertainty is absorbed almost entirely by the specification term, in agreement with its intended semantics.

Without the specification term, the missing information signal is hidden. BALD over clarifications folds the growth into its aleatoric term. BALD on the masked image inflates most with mask size, with both aleatoric and epistemic terms growing in absolute value while their shares stay nearly flat. Hence, its decomposition does not indicate that the extra uncertainty stems from missing input and could be removed by clarification rather than by changing the model.

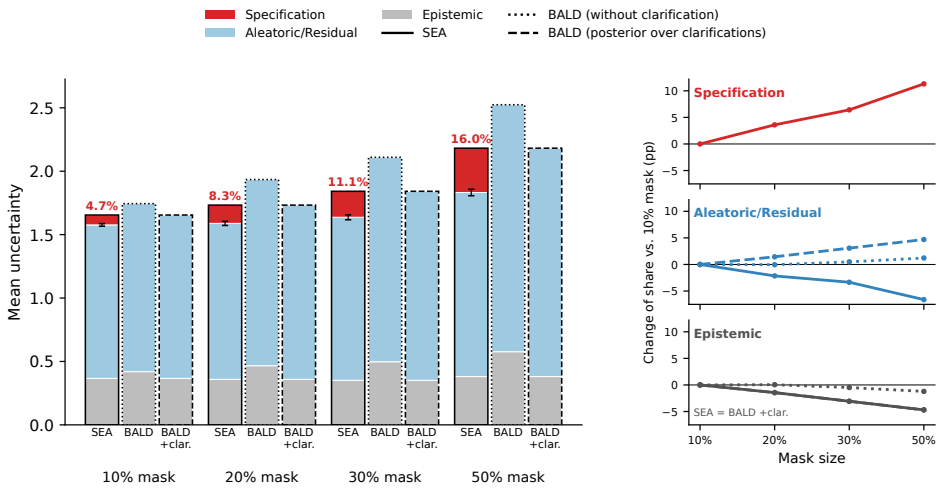

Figure 2: Uncertainty decomposition of the heterogeneous ensemble on ImageNet with the semantic entropy estimator. **Left:** mean uncertainty per mask size, stacked into epistemic, aleatoric, and specification components, for SEA, BALD on the masked image, and BALD with the posterior over clarifications. Red labels give the specification share of total uncertainty; error bars are 95% intervals of the specification mean. **Right:** change of each component's share relative to the 10% mask, in percentage points. Only the specification share grows with the masked area; SEA and BALD over clarifications have identical total and epistemic terms by construction.

Table 4: Audio QA underspecification-detection AUROC on CoQAR-U-TTS.

	SEA	BALD (without clarification)	BALD (clarification posterior)
Total	0.5926	**0.7427**	**0.5926**
Aleatoric	0.5170	0.7229	0.5851
Epistemic	0.4937	0.6108	0.4937
Specification	**0.7739**	–	–

4.3 Audio QA

Data. We render CoQAR-U passages as speech with OpenAI TTS, retaining the question as text.

Findings Table 4 shows that specification uncertainty is the strongest omission-detection signal, outperforming the runner-up total BALD uncertainty estimated on non-clarified input. Thus, separating specification uncertainty recovers an underspecification-sensitive signal that is obscured in the total of the clarification-posterior.

5 Conclusion

In this paper, we considered a predictive uncertainty setting in which the observed input x may only partially describe the intent. We derived SEA, a decomposition of prediction risk into specification, residual aleatoric, and epistemic uncertainty. At the original information level, the first two terms split the Bayes risk into the expected gain from clarification and the remaining risk, and each term corresponds to an action. In QA over text, images, and audio, specification uncertainty is the most informative score for detecting underspecified inputs, even where uncertainty computed only on x, without clarifications, fails to detect them. Residual aleatoric uncertainty is the most informative component for ID error detection in all but one setting. The standard decomposition, even when sensitive to underspecification, cannot show whether clarification would reduce the risk.

AI USE STATEMENT

In this work, we used generative AI tools for generating clarifications for SEA estimation in all datasets, implementing the method and experiments, reformatting AmbigQA dataset, preparing visualization code, polishing text in both main and appendix sections and drafting some appendix sections. We have not used generative AI tools for generating research hypotheses, theory derivation, proofs, experiment design or quality metric choice. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

For theoretical results, complete proof of the claims is presented in Section 3 and appendices A, B.

For experimental results, we elaborate the details of practical estimation of SEA decomposition in appendix D, details of LLM ensemble training in appendix G, and describe the process of clarification generation in appendix J.

We will also release all custom datasets (e.g. CoQAR-U, CoQAR-U-TTS, AmbigQA along with sampled clarifications) and experimental pipelines upon acceptance.

REFERENCES

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. Phi-4-Mini technical report: Compact yet powerful multimodal language models via Mixture-of-LoRAs. *arXiv preprint arXiv:2503.01743*, 2025. doi: 10.48550/arXiv.2503.01743. URL https://arxiv.org/abs/2503.01743.

Alexander Bick, Adam Blandin, and David J. Deming. The rapid adoption of generative AI. *Management Science*, 2026. doi: 10.1287/mnsc.2025.02523. URL https://doi.org/10.1287/mnsc.2025.02523. Articles in Advance, published online January 20, 2026.

Black Forest Labs. FLUX.1 Fill [dev]. https://huggingface.co/black-forest-labs/FLUX.1-Fill-dev, 2024. Model card.

Black Forest Labs. FLUX.2 [klein]: Towards interactive visual intelligence. Official model release, January 2026. URL https://bfl.ai/blog/flux2-klein-towards-interactive-visual-intelligence. Released January 15, 2026; accessed September 26, 2026.

Quentin Brabant, Gwénolé Lecorvé, and Lina Maria Rojas-Barahona. Coqar: Question rewriting on coqa. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022*, pp. 119–126. European Language Resources Association, 2022. URL https://aclanthology.org/2022.lrec-1.13.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, Wenshuo Ma, Tianchi Cai, et al. MiniCPM-o 4.5: Towards real-time full-duplex omni-modal

interaction. *arXiv preprint arXiv:2604.27393*, 2026. doi: 10.48550/arXiv.2604.27393. URL https://arxiv.org/abs/2604.27393.

Antoine de Mathelin, Francois Deheeger, Mathilde Mougeot, and Nicolas Vayatis. Deep anti-regularized ensembles provide reliable out-of-distribution uncertainty quantification. *arXiv preprint arXiv:2304.04042*, 2023. doi: 10.48550/arXiv.2304.04042. URL https://arxiv.org/abs/2304.04042.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

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

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/gal16.html.

Chuang Gan, Yandong Li, Haoxiang Li, Chen Sun, and Boqing Gong. Vqs: Linking segmentations to questions and answers for supervised attention in vqa and question-focused semantic segmentation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1829–1838. IEEE, 2017.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/4a8423d5e91fda00bb7e46540e2b0cf1-Paper.pdf.

Gemma Team. Gemma 4 Technical Report. *arXiv preprint arXiv:2607.02770*, 2026. doi: 10.48550/arXiv.2607.02770. URL https://arxiv.org/abs/2607.02770.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8340–8349, 2021. doi: 10.1109/ICCV48922.2021.00823. URL https://openaccess.thecvf.com/content/ICCV2021/html/Hendrycks_The_Many_Faces_of_Robustness_A_Critical_Analysis_of_Out-of-Distribution_ICCV_2021_paper.html.

Paul Hofman, Yusuf Sale, and Eyke Hüllermeier. Quantifying aleatoric and epistemic uncertainty: A credal approach. In *ICML 2024 Workshop on Structured Probabilistic Inference {\&} Generative Modeling*, 2024.

Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. Decomposing uncertainty for large language models through input clarification ensembling. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 19023–19042. PMLR, 2024. URL https://proceedings.mlr.press/v235/hou24b.html.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011. doi: 10.48550/arXiv.1112.5745. URL https://arxiv.org/abs/1112.5745.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

Nikita Kotelevskii, Aleksandr Artemenkov, Kirill Fedyanin, Fedor Noskov, Alexander Fishkov, Artem Shelmanov, Artem Vazhentsev, Aleksandr Petiushko, and Maxim Panov. Nonparametric uncertainty quantification for single deterministic neural network. *Advances in Neural Information Processing Systems*, 35:36308–36323, 2022.

Nikita Kotelevskii, Vladimir Kondratyev, Martin Takáč, Éric Moulines, and Maxim Panov. From risk to uncertainty: Generating predictive uncertainty measures via Bayesian estimation. In *International Conference on Learning Representations*, volume 2025, pp. 92468–92504, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e6f32e64b9c27d153b46c94f0fe22b56-Abstract-Conference.html.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466, 2019. doi: 10.1162/TACL_A_00276. URL https://doi.org/10.1162/tacl_a_00276.

Salem Lahlou, Moksh Jain, Hadi Nekoei, Victor Ion Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. DEUP: Direct epistemic uncertainty prediction. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=eGLdVRvvfQ.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6402–6413. Curran Associates, Inc., 2017. URL https://papers.neurips.cc/paper_files/paper/2017/hash/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=DWkJCSxKU5.

Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in neural information processing systems*, 33:7498–7512, 2020.

Huan Ma, Jingdong Chen, Joey Tianyi Zhou, Guangyu Wang, and Changqing Zhang. Estimating LLM uncertainty with evidence. *arXiv preprint arXiv:2502.00290*, 2025. doi: 10.48550/arXiv.2502.00290. URL https://arxiv.org/abs/2502.00290.

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.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5783–5797, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.466. URL https://aclanthology.org/2020.emnlp-main.466/.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7076–7087. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/mozannar20b.html.

Jishnu Mukhoti, Joost Van Amersfoort, Philip HS Torr, and Yarin Gal. Deep deterministic uncertainty for semantic segmentation. *arXiv preprint arXiv:2111.00079*, 2021.

Kata Naszadi, Putra Manggala, and Christof Monz. Aligning predictive uncertainty with clarification questions in grounded dialog. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14988–14998, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.999. URL https://aclanthology.org/2023.findings-emnlp.999/.

OpenAI. GPT-5.6 Luna model. Official API documentation, 2026. URL https://developers.openai.com/api/docs/models/gpt-5.6-luna. Accessed September 26, 2026.

Zeju Qiu, Weiyang Liu, Adrian Weller, and Bernhard Schölkopf. Orthogonal finetuning made scalable. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 31946–31963, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.1627. URL https://aclanthology.org/2025.emnlp-main.1627/.

Yusuf Sale, Michele Caprio, and Eyke Hüllermeier. Is the volume of a credal set a good measure for epistemic uncertainty? In *Uncertainty in Artificial Intelligence*, pp. 1795–1804. PMLR, 2023.

Yusuf Sale, Viktor Bengs, Michele Caprio, and Eyke Hüllermeier. Second-order uncertainty quantification: A distance-based approach. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43060–43076. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/sale24a.html.

Yusuf Sale, Nikita Kotelevskii, Maxim Panov, and Eyke Hüllermeier. The Aleatoric-Epistemic Dichotomy of Uncertainty is Meaningful and Indispensable for Machine Learning, May 2026. URL https://ssrn.com/abstract=6783338. SSRN preprint.

Kajetan Schweighofer, Lukas Aichberger, Mykyta Ielanskyi, and Sepp Hochreiter. On information-theoretic measures of predictive uncertainty. *UAI*, 2025.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.330. URL https://aclanthology.org/2023.emnlp-main.330/.

Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, pp. 9690–9700. PMLR, 2020.

Roman Vashurin, Maiya Goloburda, Albina Ilina, Aleksandr Rubashevskii, Preslav Nakov, Artem Shelmanov, and Maxim Panov. CoCoA: A minimum Bayes risk framework bridging confidence and consistency for uncertainty quantification in LLMs. In *Advances in Neural Information Processing Systems*, volume 38, pp. 106236–106281. Curran Associates, Inc., 2025. doi: 10.52202/085713-3546. URL https://papers.nips.cc/paper_files/paper/2025/hash/98d2006bf15f00e2ff293dfba4acb8b2-Abstract-Conference.html.

Nassim Walha, Sebastian G. Gruber, Thomas Decker, Yinchong Yang, Alireza Javanmardi, Eyke Hüllermeier, and Florian Buettner. Fine-grained uncertainty decomposition in large language models: A spectral approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40 (31):26090–26098, 2026. doi: 10.1609/aaai.v40i31.39811. URL https://ojs.aaai.org/index.php/AAAI/article/view/39811.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-Omni technical report. *arXiv preprint arXiv:2503.20215*, 2025. doi: 10.48550/arXiv.2503.20215. URL https://arxiv.org/abs/2503.20215.

A Other approximation strategies

In Section 3, we estimate the terms of equation (3) with approximation $(1, 2)$ of Kotelevskii et al. (2025). Here, we describe the approximation ideas they consider, apply them to equation (3), and give the resulting estimates for several proper scoring rules.

Notation. We use $q_{z,\theta}$, q_θ, and q from Section 3 and additionally define the posterior predictive distribution given the intent,

$$\bar{q}_z \coloneqq \mathbb{E}_\pi[q_{Z,\Theta} \mid Z = z] = q(y \mid z, x, D).$$

Expectations are taken over all random variables inside them, where (Z, Θ) is distributed according to $\pi(\theta, z \mid x, D)$. When an estimate uses two posterior samples of the parameters, we denote them by $\tilde{\Theta}$ and Θ, which are independent and identically distributed. Under the factorization $\pi(\theta, z \mid x, D) = \pi(\theta \mid D)\pi(z \mid x)$ used in our experiments, $\bar{q}_z = \mathbb{E}_\Theta[q_{z,\Theta}]$.

Approximation ideas. Kotelevskii et al. (2025) view a risk as a function $g(p, q)$ of the ground-truth distribution p and the predictor q. Given posterior samples r_Θ of a distribution, they replace the corresponding argument of g in one of three ways:

1. *Bayesian averaging of risk.* Evaluate g at each sample r_Θ and average the result over the posterior.
2. *Central label.* Plug in the posterior predictive distribution $\arg\min_r \mathbb{E}[D_G(r_\Theta \parallel r)] = \mathbb{E}[r_\Theta]$.
3. *Central prediction.* Plug in $\arg\min_r \mathbb{E}[D_G(r \parallel r_\Theta)]$.

Approximation (i, j) applies idea i to the ground-truth argument and idea j to the predictor. For brevity, we consider only ideas 1 and 2 below, which give four approximations (i, j) with $i, j \in \{1, 2\}$.

Application to our decomposition. In our setting, the ground-truth argument is the whole process $p(y, z \mid x)$. Idea 1 replaces it by the process $q(y \mid z, \theta)\pi(z \mid x, \theta, D)$ of each posterior sample θ, with outcome distributions $q_{z,\theta}$ and marginal q_θ. Idea 2 replaces it by the posterior predictive process $q(y, z \mid x, D)$, with outcome distributions $\bar{q}_z$ and marginal q. Here, $\bar{q}_z$ and q are the central labels of the samples $q_{z,\Theta}$ given $Z = z$ and of the samples q_Θ, respectively. The predictor does not observe the intent, so the two ideas use q_θ for each posterior sample and the posterior predictive distribution q, respectively.

Resulting estimates. The residual aleatoric and specification terms depend only on the process, so each has two estimates,

$$\widehat{R}^{(1)}_{\text{res}} = -\mathbb{E}[G(q_{Z,\Theta})], \qquad \widehat{R}^{(1)}_{\text{spec}} = \mathbb{E}[D_G(q_{Z,\Theta} \parallel q_\Theta)],$$
$$\widehat{R}^{(2)}_{\text{res}} = -\mathbb{E}[G(\bar{q}_Z)], \qquad \widehat{R}^{(2)}_{\text{spec}} = \mathbb{E}[D_G(\bar{q}_Z \parallel q)].$$

Their sums are the corresponding estimates of the Bayes risk of Kotelevskii et al. (2025), $\widehat{R}^{(1)}_b = -\mathbb{E}[G(q_\Theta)]$ and $\widehat{R}^{(2)}_b = -G(q)$. The epistemic term depends on both arguments, which gives four estimates,

$$\widehat{R}^{(1,1)}_{\text{exc}} = \mathbb{E}[D_G(q_{\tilde{\Theta}} \parallel q_\Theta)], \qquad \widehat{R}^{(1,2)}_{\text{exc}} = \mathbb{E}[D_G(q_\Theta \parallel q)],$$
$$\widehat{R}^{(2,1)}_{\text{exc}} = \mathbb{E}[D_G(q \parallel q_\Theta)], \qquad \widehat{R}^{(2,2)}_{\text{exc}} = D_G(q \parallel q) = 0.$$

Kotelevskii et al. (2025) call $\widehat{R}^{(1,1)}_{\text{exc}}$, $\widehat{R}^{(1,2)}_{\text{exc}}$, and $\widehat{R}^{(2,1)}_{\text{exc}}$ the expected pairwise Bregman divergence (EPBD), the Bregman information (BI), and the reverse Bregman information (RBI). The total risk $R(p, q)$ is linear in p, so both ideas for the process give the same estimate $\widehat{R}^{(\cdot,j)} \coloneqq \widehat{R}^{(1,j)} = \widehat{R}^{(2,j)}$ with

$$\widehat{R}^{(\cdot,1)} = \mathbb{E}[R(q, q_\Theta)] = -G(q) + \mathbb{E}[D_G(q \parallel q_\Theta)], \qquad \widehat{R}^{(\cdot,2)} = R(q, q) = -G(q).$$

Decompositions. Applying equation (3) to the estimated process and predictor, and averaging over the posterior samples for idea 1, we obtain for every approximation (i, j)

$$\widehat{R}^{(\cdot,j)} = \widehat{R}^{(i)}_{\text{res}} + \widehat{R}^{(i)}_{\text{spec}} + \widehat{R}^{(i,j)}_{\text{exc}}.$$

For $(i, j) = (1, 2)$, this is equation (4). Approximation $(2, 2)$, by construction, estimates epistemic uncertainty as zero. It is equation (6) with q_z replaced by $\bar{q}_z$, i.e., it applies the decompositions of Hou et al. (2024) and Walha et al. (2026) to the posterior predictive model. For a single model, $\bar{q}_z = q_z$, and all four approximations reduce to equation (6).

Examples. Table 5 lists the estimates for three proper scoring rules.

Log score. Here $G(r) = \sum_y r(y) \log r(y)$, so $-G(r) = \mathbb{H}(r)$, D_G is the KL divergence, and $R(p, r) = \text{CE}(p \parallel r) \coloneqq -\sum_y p(y) \log r(y)$ is the cross-entropy. Under the joint distribution $q(y, z, \theta \mid x, D)$, $q_{z,\theta}$ is the distribution of Y given $(Z, \Theta) = (z, \theta)$, $\bar{q}_z$ is its distribution given $Z = z$, q_θ given $\Theta = \theta$, and q is its marginal distribution. Hence, the estimates become entropies and mutual informations. In particular, approximation $(1, 2)$ gives the chain rule $\mathbb{H}(Y) = \mathbb{H}(Y \mid Z, \Theta) + I(Y; Z \mid \Theta) + I(Y; \Theta)$, i.e., equation (5), and approximation $(2, 2)$ gives $\mathbb{H}(Y) = \mathbb{H}(Y \mid Z) + I(Y; Z)$. The epistemic estimates of approximations $(1, 1)$ and $(2, 1)$ are the expected pairwise KL divergence (EPKL) and the reverse mutual information (RMI) (Malinin & Gales, 2021).

Brier score. Here $G(r) = \|r\|_2^2 - 1$, where r is viewed as a vector in $\mathbb{R}^{|\mathcal{Y}|}$. Then $\ell(r, y) = \|r - e_y\|_2^2$, where e_y is the one-hot vector of y, and $D_G(r \parallel s) = \|r - s\|_2^2$. Hence, the specification and epistemic estimates are variances of the predictions, summed over the classes. For example, $\widehat{R}^{(1)}_{\text{spec}}$ is the expected variance of $q_{Z,\Theta}$ across intents, and $\widehat{R}^{(1,2)}_{\text{exc}}$ is the variance of q_Θ across posterior samples. Since D_G is symmetric, $\widehat{R}^{(2,1)}_{\text{exc}} = \widehat{R}^{(1,2)}_{\text{exc}}$, and since $\tilde{\Theta}$ and Θ are independent, $\widehat{R}^{(1,1)}_{\text{exc}} = 2\widehat{R}^{(1,2)}_{\text{exc}}$.

Zero-one score. Here $G(r) = M(r) - 1$, where $M(r) \coloneqq \max_y r(y)$. With the subgradient $G'(r) = e_{\hat{y}(r)}$, where $\hat{y}(r) \in \arg\max_y r(y)$ is the predicted class, $\ell(r, y) = \mathbf{1}\{y \neq \hat{y}(r)\}$ is the zero-one loss and $D_G(r \parallel s) = M(r) - r(\hat{y}(s))$. The estimates $\widehat{R}^{(1)}_{\text{spec}}$, $\widehat{R}^{(2)}_{\text{spec}}$, and $\widehat{R}^{(1,2)}_{\text{exc}}$ are Bregman informations, since the second argument of D_G is the (conditional) mean of the first. Hence, they reduce to differences of expected maximal probabilities. In particular, $\widehat{R}^{(1)}_{\text{spec}}$ is the expected increase in the probability of a correct prediction once the intent is revealed, computed under the process of each posterior sample. This score is proper but not strictly proper.

Table 5: Estimates of the terms of equation (3) for three proper scoring rules. For the log score, the information-theoretic quantities are computed under $q(y, z, \theta \mid x, D)$, and we omit the conditioning on $X = x$ and D. Here, $M(r) = \max_y r(y)$ and $\hat{y}_\Theta = \hat{y}(q_\Theta)$.

Estimate	Log score	Brier score	Zero-one score
$\widehat{R}^{(1)}_{\text{res}}$	$\mathbb{H}(Y \mid Z, \Theta)$	$1 - \mathbb{E}\|q_{Z,\Theta}\|_2^2$	$1 - \mathbb{E}[M(q_{Z,\Theta})]$
$\widehat{R}^{(2)}_{\text{res}}$	$\mathbb{H}(Y \mid Z)$	$1 - \mathbb{E}\|\bar{q}_Z\|_2^2$	$1 - \mathbb{E}[M(\bar{q}_Z)]$
$\widehat{R}^{(1)}_{\text{spec}}$	$I(Y; Z \mid \Theta)$	$\mathbb{E}\|q_{Z,\Theta} - q_\Theta\|_2^2$	$\mathbb{E}[M(q_{Z,\Theta})] - \mathbb{E}[M(q_\Theta)]$
$\widehat{R}^{(2)}_{\text{spec}}$	$I(Y; Z)$	$\mathbb{E}\|\bar{q}_Z - q\|_2^2$	$\mathbb{E}[M(\bar{q}_Z)] - M(q)$
$\widehat{R}^{(1,1)}_{\text{exc}}$	$\mathbb{E}[KL(q_{\tilde{\Theta}} \parallel q_\Theta)]$	$2\,\mathbb{E}\|q_\Theta - q\|_2^2$	$\mathbb{E}[M(q_\Theta) - q(\hat{y}_\Theta)]$
$\widehat{R}^{(1,2)}_{\text{exc}}$	$I(Y; \Theta)$	$\mathbb{E}\|q_\Theta - q\|_2^2$	$\mathbb{E}[M(q_\Theta)] - M(q)$
$\widehat{R}^{(2,1)}_{\text{exc}}$	$\mathbb{E}[KL(q \parallel q_\Theta)]$	$\mathbb{E}\|q_\Theta - q\|_2^2$	$M(q) - \mathbb{E}[q(\hat{y}_\Theta)]$
$\widehat{R}^{(\cdot,1)}$	$\mathbb{E}[\text{CE}(q \parallel q_\Theta)]$	$1 - \|q\|_2^2 + \mathbb{E}\|q_\Theta - q\|_2^2$	$1 - \mathbb{E}[q(\hat{y}_\Theta)]$
$\widehat{R}^{(\cdot,2)}$	$\mathbb{H}(Y)$	$1 - \|q\|_2^2$	$1 - M(q)$

Relations between the estimates. As shown by Kotelevskii et al. (2025), $\widehat{R}^{(1,1)}_{\text{exc}} = \widehat{R}^{(1,2)}_{\text{exc}} + \widehat{R}^{(2,1)}_{\text{exc}}$, and the two estimates of the Bayes risk differ by the Bregman information, $\widehat{R}^{(2)}_b - \widehat{R}^{(1)}_b = \widehat{R}^{(1,2)}_{\text{exc}} \geq 0$. Similarly, since $\mathbb{E}[q_{Z,\Theta} \mid Z] = \bar{q}_Z$, the two estimates of the residual aleatoric term satisfy $\widehat{R}^{(2)}_{\text{res}} - \widehat{R}^{(1)}_{\text{res}} = \mathbb{E}[D_G(q_{Z,\Theta} \parallel \bar{q}_Z)] \geq 0$. Hence, the two estimates of the specification term differ by

$\widehat{R}^{(1)}_{\mathrm{spec}} - \widehat{R}^{(2)}_{\mathrm{spec}} = \mathbb{E}[D_G(q_{Z,\Theta} \,\|\, \bar{q}_Z)] - \widehat{R}^{(1,2)}_{\mathrm{exc}}$, which can have either sign. If Z and Θ are independent under π, as in our experiments, this difference is nonnegative for the log and Brier scores, where it equals $I(Z;\Theta \mid Y)$ and $\mathbb{E}\|q_{Z,\Theta} - q_\Theta - \bar{q}_Z + q\|_2^2$, respectively.

B Bayes risk split

We derive the split of the Bayes risk used in equation (3). Below, as in the main part, we assume that x is fixed.

By the definition of the Bregman divergence,

$$\mathbb{E}_Z[D_G(p_Z \,\|\, p)] = \mathbb{E}_Z[G(p_Z)] - G(p) - \langle G'(p)\,,\, \mathbb{E}_Z[p_Z] - p\rangle = \mathbb{E}_Z[G(p_Z)] - G(p),$$

where the last equality uses $\mathbb{E}_Z[p_Z] = p$.

Rearranging the terms gives

$$-G(p) = -\mathbb{E}_Z[G(p_Z)] + \mathbb{E}_Z[D_G(p_Z \,\|\, p)],$$

and substituting this into equation (2) yields equation (3).

Interestingly, we can show that since q does not depend on z, we have $R(p,q) = \mathbb{E}_{Z\sim p(\cdot|x)}\big[\mathbb{E}_{Y\sim p_Z}[\ell(q,Y)]\big] = \mathbb{E}_{Z\sim p(\cdot|x)}[R(p_Z,q)]$, i.e., $R(p,q)$ is also the risk of q under the *joint process* $p(y,z \mid x)$. This holds for any loss that does not depend on z.

C Limitations

We see two limitations of our approach.

Approximation of the intent distribution. Recall that our approach is built on separating specification uncertainty from residual aleatoric uncertainty. This requires the distribution $p(z \mid x)$ over possible values of the unknown intent. In the Bayesian model of Section 3.2, intents and parameters have a joint distribution $\pi(\theta, z \mid x, \mathcal{D})$, so the intent distribution in general may depend on the training data and on θ. In our experiments, we instead use the factorization $\pi(\theta, z \mid x, \mathcal{D}) = \pi(\theta \mid \mathcal{D})\pi(z \mid x)$, where $\pi(z \mid x)$ is given by external generative models and depends on neither $\mathcal{D}$ nor θ. The estimates, therefore, inherit the generators' view of which intents are plausible. A generator whose intents are less diverse than the true ones can underestimate specification uncertainty, and one that proposes implausible intents can overestimate it.

This dependence is, however, inherent to the problem rather than specific to our approach. As discussed in Section 3.2, pairs (x,y) determine $p(y \mid x)$ but not how its Bayes risk splits into specification and residual aleatoric uncertainty. Any estimate of this split, including those of Hou et al. (2024) and Walha et al. (2026), therefore requires a model of the intents, for which large pretrained generative models are currently the most practical choice. Empirically, the specification term was the strongest signal of underspecification across each clarification pipeline we used, from LLM-generated text to direct inpainting with a fixed prompt and no language model (Section 4 and Appendix L). This suggests that the signal is not tied to a particular generator.

Estimation of epistemic uncertainty. The epistemic term $D_G(p \,\|\, q)$ cannot be estimated from the predictive distribution q alone. It requires either a second-order model, such as an ensemble whose members serve as samples from $\pi(\theta \mid \mathcal{D})$, or an auxiliary model that directly estimates the excess risk (Kotelevskii et al., 2022; Lahlou et al., 2023). Both options add cost. Ensembles multiply the training and inference cost by the number of members, which is especially restrictive for LLMs, while auxiliary models have to be trained separately. This requirement is not specific to SEA, since the epistemic term of the standard information-theoretic decomposition also requires a second-order model.

D Monte-Carlo Estimates of Components

For fixed x and training data D, we use the predictive distributions from Section 3:

$$q_{z,\theta} \coloneqq q(y \mid z, \theta). \tag{7}$$

As in our experiments, we assume $\pi(\theta, z \mid x, D) = \pi(\theta \mid D)\pi(z \mid x)$. Let $\theta_m \sim \pi(\theta \mid D)$, $m = 1, \ldots, M$, denote the parameter samples represented by the ensemble, and let $z_k \sim \pi(z \mid x)$, $k = 1, \ldots, K$, denote the sampled intents shared across its members. Here, M is the ensemble size and K is the number of intents.

The clarification-marginalized predictive distribution of one member and the posterior predictive distribution are approximated by

$$q_\theta = \mathbb{E}_\pi[q_{Z,\Theta} \mid \Theta = \theta] \approx \widehat{q}_\theta \coloneqq \frac{1}{K}\sum_{k=1}^{K} q_{z_k,\theta}, \tag{8}$$

$$\begin{aligned} q &= \mathbb{E}_\pi[q_{Z,\Theta}] = q(y \mid x, D) \\ &\approx \widehat{q} \coloneqq \frac{1}{M}\sum_{m=1}^{M} \widehat{q}_{\theta_m} = \frac{1}{MK}\sum_{m=1}^{M}\sum_{k=1}^{K} q_{z_k,\theta_m}. \end{aligned} \tag{9}$$

Hats distinguish finite-sample estimates from the corresponding distributions and uncertainty components. All expectations below are taken under $\pi(\theta, z \mid x, D)$, as in equation (5). Its three components have the following plug-in Monte Carlo estimates.

Residual aleatoric.

$$\begin{aligned} \mathrm{SEA}_{\mathrm{res}} &= \mathbb{E}_{Z,\Theta}[\mathbb{H}(q_{Z,\Theta})] \\ &\approx \widehat{\mathrm{SEA}}_{\mathrm{res}} \coloneqq \frac{1}{MK}\sum_{m=1}^{M}\sum_{k=1}^{K} \mathbb{H}(q_{z_k,\theta_m}). \end{aligned} \tag{10}$$

Specification.

$$\begin{aligned} \mathrm{SEA}_{\mathrm{spec}} &= \mathbb{E}_{Z,\Theta}[KL(q_{Z,\Theta} \parallel q_\Theta)] \\ &= \mathbb{E}_\Theta[\mathbb{H}(q_\Theta)] - \mathrm{SEA}_{\mathrm{res}} \\ &\approx \widehat{\mathrm{SEA}}_{\mathrm{spec}} \coloneqq \frac{1}{M}\sum_{m=1}^{M} \mathbb{H}(\widehat{q}_{\theta_m}) - \widehat{\mathrm{SEA}}_{\mathrm{res}}. \end{aligned} \tag{11}$$

Epistemic.

$$\begin{aligned} \mathrm{SEA}_{\mathrm{epi}} &= \mathbb{E}_\Theta[KL(q_\Theta \parallel q)] = \mathbb{H}(q) - \mathbb{E}_\Theta[\mathbb{H}(q_\Theta)] \\ &\approx \widehat{\mathrm{SEA}}_{\mathrm{epi}} \coloneqq \mathbb{H}(\widehat{q}) - \frac{1}{M}\sum_{m=1}^{M} \mathbb{H}(\widehat{q}_{\theta_m}). \end{aligned} \tag{12}$$

Using the same sampled distributions in all three terms preserves the decomposition:

$$\widehat{\mathrm{SEA}}_{\mathrm{total}} \coloneqq \mathbb{H}(\widehat{q}) = \widehat{\mathrm{SEA}}_{\mathrm{res}} + \widehat{\mathrm{SEA}}_{\mathrm{spec}} + \widehat{\mathrm{SEA}}_{\mathrm{epi}}.$$

E LLM ENSEMBLING

We implement the predictive mixtures in Appendix D by separating answer sampling from probability evaluation. The same set of clarifications is shared across all ensemble members, so variation across clarifications and variation across models remain distinct axes.

Separate sampling and a shared answer pool. For each member θ_m and clarification z_k, we sample S answers from q_{z_k,θ_m}; our text QA experiments use $K = 5$ clarifications and $S = 10$ answers per pair. Different pairs can produce different, even disjoint, answer sets. An answer missing from one pair's samples may nevertheless have substantial probability under that pair, so treating unobserved answers as having zero probability would distort the mixture and its uncertainty. We therefore form a shared candidate set $\mathcal{U}$ from the union of all MKS draws. We deduplicate candidates for scoring while retaining every draw's originating model, clarification, and multiplicity for Monte Carlo estimation.

Cross-scoring. Every member scores every candidate $u \in \mathcal{U}$ under every clarification, including candidates produced only by other members or clarifications. With teacher forcing, the candidate is supplied as the answer continuation, and we sum the conditional log-probabilities of its tokens:

$$\ell_{mku} = \sum_{t=1}^{T_m(u)} \log q\big(u_t^{(m)} \mid u_{<t}^{(m)}, z_k, \theta_m\big), \tag{13}$$

where $u^{(m)}$ is the candidate's tokenization for member m, and prompt tokens do not contribute to the sum. This produces a dense $M \times K \times |\mathcal{U}|$ score tensor despite potentially disjoint sampled answer sets. For checkpoint ensembles with a shared tokenizer, candidate identity is determined by token sequences after the configured termination-token handling. For heterogeneous ensembles, we pool cleaned answer strings and tokenize each string with the scoring member's own tokenizer. All mixture scores, including those for a candidate's generating pair, come from this common cross-scoring procedure; generation-time scores are retained only as diagnostics.

Averaging over clarifications and models. We average *probabilities*, first over clarifications within a member and then over members:

$$\widehat{q}_{\theta_m}(u) = \frac{1}{K}\sum_{k=1}^{K} \exp(\ell_{mku}), \qquad \widehat{q}(u) = \frac{1}{M}\sum_{m=1}^{M} \widehat{q}_{\theta_m}(u). \tag{14}$$

These averages are evaluated in log space using log-sum-exp. The individual, clarification-marginalized, and fully averaged scores provide the distributions needed for the SEA components and clarification-posterior BALD. For answer-level correctness evaluation, we select $\arg\max_{u\in\mathcal{U}} \widehat{q}(u)$. BALD without clarifications uses a separate answer pool sampled and cross-scored on the original input x.

F Semantic Entropy Implementation

We cluster the pooled answers $\mathcal{U}$ once, giving all model–clarification pairs the same answer-to-class mapping g. For a component or mixture predictive distribution r, we approximate each class's mass and estimate semantic entropy as

$$\widehat{P}_r(c) = \sum_{u\in\mathcal{U}:\, g(u)=c} r(u), \qquad \widehat{H}_{\mathrm{SE}}(r) = -\frac{1}{n_r}\sum_{i=1}^{n_r} \log \widehat{P}_r\big(g(y_i)\big). \tag{15}$$

Here, $r(u)$ is obtained by cross-scoring as in Appendix E, and y_i are the matched Monte Carlo draws: S draws for a model–clarification pair, KS draws pooled over clarifications for a member, or MKS draws for the full mixture. Repeated draws retain their multiplicity. Class masses are evaluated on the finite candidate pool without renormalization or sequence-length normalization. The resulting entropy estimates are combined as in Appendix D.

For natural language inference (NLI), we use `microsoft/deberta-large-mnli`. An answer joins an existing class when the NLI model predicts entailment in both directions between that answer and the class representative; otherwise, it starts a new class. In the AmbigQA E2B and heterogeneous runs, the complete encoded NLI pair is capped at 100 tokens, including special tokens, using longest-first truncation across the two answers.

G Ensemble training and inference details.

Ensemble details. Since training an ensemble of language models from scratch is prohibitively expensive, we turn to fine-tuning. We tune 5 Gemma-4-E4B-it and Gemma-4-E2B-it models (Gemma Team, 2026) for short-form factual question answering on the AmbigQA dataset (Min et al., 2020). All models are trained on the same training set, differing only in the order of examples. We tried LoRA and Orthogonal finetuning (OFT) (Hu et al., 2022; Qiu et al., 2025). Both configurations yielded similar AmbigQA validation performance, but OFT showed less catastrophic forgetting on MMLU and IFEval subsets, so we report results for OFT ensembles in the main text.

Fine-tuning data. Original AmbigQA contains questions of two types: well-formed, admitting a single answer, and ambiguous, which have multiple interpretations. For each ambiguous question, the dataset contains from 2 to 9 clarified versions of this question along with the corresponding ground truth answer. For fine-tuning, we only use well-formed or clarified questions, not ambiguous ones, since it's unclear what should be a short-form ground truth answer for an ambiguous question.

We split official train set of AmbigQA into local train and development set in 90/10 proportion. Our final training set contains 17261 examples, and the development set contains 1982 examples.

Ensemble training regime Each model is fine-tuned for three epochs using OFT with a block size of 32. We target all linear layers, apply a module dropout rate of 0.05, and use the Cayley–Neumann approximation with five terms; biases are not adapted. Training uses a per-device batch size of 16 with two gradient-accumulation steps, corresponding to an effective batch size of 32 per device. We set the maximum input length to 512 tokens and optimize with a learning rate of 5×10^{-5}, weight decay of 0.01, and a linear warm-up over the first 3% of training steps. Gradients are clipped to a maximum norm of 1.0. Computation uses bfloat16 precision with TF32 enabled, and evaluation is performed every 150 optimization steps. We use the AdamW optimizer.

H PERFORMANCE OF THE ENSEMBLES

We evaluate answer correctness with GPT-5.6 Luna (OpenAI, 2026), using temperature 0, disabled reasoning, and a maximum of 32 output tokens. The judge receives the original question, dataset reference answer(s), and selected ensemble answer; for CoQAR-U, it also receives the original undamaged passage. It returns 1 if the answer addresses the question and agrees in meaning with at least one reference (and with the passage for CoQAR-U), and 0 otherwise, accepting paraphrases and harmless extra detail but rejecting contradictions and nonanswers. We select the highest-probability candidate from the saved ensemble answer pool; for the clarification posterior, selection follows averaging probabilities over ensemble members and five clarifications. Table 6 reports the mean binary judgment in each input regime, counting each source question once per regime.

Unsurprisingly, omissions and ambiguity reduce performance, and sampling possible clarifications restores it only somewhat. This highlights the need to first detect and then resolve ambiguities and a lack of specification by asking the user additional questions with the goal of restoring the true original intent.

Table 6: Text QA accuracy (%) under GPT-5.6 Luna judgments. CoQAR-U uses the original full passage, its omitted version, and the posterior over inserted clarifications of the omitted passage. AmbigQA is reported pooled and separately for ambiguous and well-specified questions. Clarified scores evaluate the single ensemble answer selected from the clarification-marginalized predictive distribution. Each question contributes once per regime; bold marks the best ensemble in each row.

Input	n	Gemma E2B	Gemma E4B	Heterogeneous
CoQAR-U				
Original	1,000	87.10	**92.10**	92.00
Omitted	1,000	33.60	31.20	**36.40**
Clarified	1,000	45.80	46.80	**47.20**
AmbigQA: pooled				
Original	1,971	22.02	28.21	**33.64**
Clarified	1,971	32.27	37.70	**40.79**
AmbigQA: ambiguous				
Original	1,141	21.65	27.70	**31.73**
Clarified	1,141	31.99	37.42	**39.53**
AmbigQA: well-specified				
Original	830	22.53	28.92	**36.27**
Clarified	830	32.65	38.07	**42.53**

Correctness-judge prompts. We use the following templates, substituting dataset fields and the selected candidate answer for the braced placeholders. For AmbigQA, `reference_answers` is a JSON list of the saved reference answers; for CoQAR-U, `original_story` is the full passage before omission.

AmbigQA

```
Question: {question}
Reference answers: {reference_answers}
Candidate answer: {candidate_answer}

Return 1 if the candidate answers the question and matches at least one
reference in meaning; otherwise return 0. Accept paraphrases and
harmless extra detail. Reject contradictions and nonanswers. Output only
0 or 1.
```

CoQAR-U

```
Original passage: {original_story}
Question: {question}
Reference answer: {gold_answer_original}
Candidate answer: {candidate_answer}

Return 1 if the candidate answers the question, agrees with the
reference in meaning, and is consistent with the passage; otherwise
return 0. Accept paraphrases and harmless extra detail. Reject
contradictions and nonanswers. Output only 0 or 1.
```

I ENSEMBLE-FREE ESTIMATION

Tables 7 and 8 report underspecification detection for each member of the heterogeneous ensemble evaluated separately on text and visual QA. We restrict the answer support and sequence probabilities to the selected model, retaining its ten draws per clarification and the saved semantic cluster assignments. With one model, the epistemic component vanishes, while specification uncertainty still measures variation across clarifications. The specification term achieves the highest AUROC for every model on both text datasets. Thus, if one is willing to sacrifice the epistemic component, efficient underspecification detection can still be performed with a single model.

Table 7: Single-model text QA underspecification-detection AUROC using the SE estimator, as in Table 1. Each model uses only its own sampled answers and sequence probabilities; the saved ensemble semantic partition is restricted to that support. CoQAR-U uses inserted clarifications and 1,000 damaged/undamaged pairs (damaged positive); AmbigQA uses 1,971 unique questions (1,141 ambiguous positives, 830 well-specified negatives). With one model, epistemic scores vanish (AUROC 0.5), and each BALD aleatoric score equals its total. $\mathrm{BALD}_{\mathrm{clar}}$ is the total with the clarification posterior, and BALD is the total without clarifications. Bold marks the best displayed score in each column.

	Qwen2.5 Omni		Gemma-4 E4B		Phi-4		MiniCPM-o	
Measure	CoQAR-U	AmbigQA	CoQAR-U	AmbigQA	CoQAR-U	AmbigQA	CoQAR-U	AmbigQA
$\mathrm{SEA}_{\mathrm{total}}$	0.65	0.52	0.72	0.55	0.63	0.53	0.68	0.54
$\mathrm{SEA}_{\mathrm{spec}}$	**0.78**	**0.61**	**0.77**	**0.59**	**0.78**	**0.55**	**0.77**	**0.64**
$\mathrm{SEA}_{\mathrm{res}}$	0.55	0.51	0.53	0.54	0.55	0.53	0.54	0.54
$\mathrm{BALD}_{\mathrm{clar}}$	0.65	0.52	0.72	0.55	0.63	0.53	0.68	0.54
BALD	0.70	0.49	0.72	0.51	0.73	0.50	0.70	0.55

Table 8: ROC-AUC for underspecification detection with a single answering model (no ensembling), for each ensemble member on VQS and ImageNet. Positive class is 30% mask, negative is 10% mask; 500 image pairs per dataset. Each model uses only its own sampled answers and its five restorations. $\mathrm{SEA}_{\mathrm{total}}$, $\mathrm{SEA}_{\mathrm{spec}}$, and $\mathrm{SEA}_{\mathrm{res}}$ use the SE estimator; with one model the epistemic term vanishes, and $\mathrm{SEA}_{\mathrm{spec}}$ measures disagreement across restorations. $\mathrm{BALD}_{\mathrm{clar}}$ is the total with the clarification posterior, and BALD is the total on the masked image. The best score in each column is shown in bold.

	Qwen2.5 Omni		Gemma-4 E4B		Phi-4		MiniCPM-o	
Measure	VQS	ImageNet	VQS	ImageNet	VQS	ImageNet	VQS	ImageNet
$\mathrm{SEA}_{\mathrm{total}}$	0.56	0.55	0.55	0.57	0.56	0.54	0.57	0.57
$\mathrm{SEA}_{\mathrm{spec}}$	**0.61**	**0.65**	**0.61**	**0.65**	**0.65**	**0.64**	**0.63**	**0.64**
$\mathrm{SEA}_{\mathrm{res}}$	0.54	0.53	0.53	0.53	0.54	0.52	0.55	0.54
$\mathrm{BALD}_{\mathrm{clar}}$	0.56	0.55	0.55	0.57	0.56	0.54	0.57	0.57
BALD	0.53	0.56	0.53	0.53	0.54	0.56	0.52	0.56

J CLARIFICATION GENERATION

This section details how the clarification proposal distribution $\pi(z \mid x)$ is implemented for the two text QA datasets of Section 4.1. In both cases, we sample five clarifications per input with GPT-5.6 Luna (OpenAI, 2026) at temperature 1; duplicate and unchanged clarifications are retained, and no output is filtered or resampled for semantic quality, novelty, or diversity. The generated clarifications are frozen and published as Hugging Face datasets (`zykov/AmbigQA-clarifications` and `rvashurin/coqar-clarifications`), and all downstream experiments pin their revisions.

AmbigQA. For AmbigQA (Min et al., 2020), a clarification of a question is a fully rewritten question: the model is asked to detect possible ambiguities and to resolve them by adding conditions or explanations, without changing the original intent, or to return the question unchanged when it finds no ambiguity. We generate clarifications for 2,002 questions of the `validation` split (`full` configuration); a question is labeled underspecified when at least one of its AmbigQA annotations has type `multipleQAs`, and well-specified when all annotations are of type `singleAnswer`. Each of the five clarifications is an independent chat completion with the prompt below, the system message "You are an intelligent assistant.", and temperature 1. Empty or deflecting replies (e.g., responses asking the user to provide more information) are discarded and resampled, with up to four attempts per kept clarification.

AmbigQA clarification generation prompt

```
In this task, you will receive a question that may contain ambiguities.
First analyze the
following aspects to find if there are any ambiguities according to the
real-world facts:
- Unresolved references to entities or people ("it", "he", "they" etc
without referred entity explicitly mentioned elsewhere)
- Entities, objects, or events has multiple references or
interpretations
- Unclear timestamps
- Unclear locations
- Unclear answer types (e.g., "When" refers to "which year or what
date", and "Who" refers to "
which person or which team")
If there are any ambiguities, you need to remove ambiguities by adding
some clarifications to
the question. Each clarification is an additional condition or
explanations to the concept in
the question that resolve its ambiguity. If the question contains
unresolved references, come up
with a plausible concrete context grounded in real-world facts that
would resolve them.
- You are only allowed to add conditions or explanations, and you cannot
change the
original intent or semantics of the question.
- The conditions and explanations must be ground to real-word facts.
If there is no ambiguities, you only need to output the original
question as it is.
Output only the final question after adding clarifications (if any). Do
not answer the question.

Original Question: {question}
Question after adding clarification:
```

CoQAR-U construction. CoQAR-U is derived from the development split of CoQAR (Brabant et al., 2022), pinned to the publisher's release. We select 1,000 development question turns after excluding ten turns used in an earlier pilot; the question is the first nonempty out-of-context rewrite of the turn. The underspecified passage removes every complete sentence whose span overlaps the annotated answer rationale, with sentence boundaries obtained from spaCy's rule-based sentencizer; the original passage serves as the well-specified counterpart. Deletion is mechanical, so a damaged passage may still contain enough information to answer the question; the underspecification flag records the construction, not verified unanswerability.

CoQAR-U clarification generation. Because a passage clarification must supply concrete missing story facts rather than rephrase a question, clarifications for CoQAR-U are produced with a two-step latent-state procedure rather than a single rewriting call. Both steps use GPT-5.6 Luna with reasoning effort `low`, temperature 1, and strict JSON outputs (output caps of 4,096 tokens for the first step and 2,048 for the second, including reasoning tokens).

First, an *interpretation menu* call receives only the visible passage and the question and returns either an `already_answerable` state, or a description of the missing fact together with at most four mutually exclusive families of possible missing story states, optionally followed by a residual "other" family; each family carries a model-estimated probability, and the probabilities are normalized to sum to one. Second, five interpretation draws are made with replacement from the normalized menu weights and paired across the original and damaged conditions. Each drawn state is passed to a separate *rendering* call, which receives the passage, the question, the menu, and the selected state, and returns the smallest self-contained narrative addition (usually one or two sentences) that realizes the selected state and makes the unchanged question answerable; an `already_answerable` selection yields an empty addition that leaves the passage unchanged. The five draws share the first-stage menu, and repeated selections are rendered independently. Generator payloads never contain the gold answer, the deleted sentences, or the deletion offsets. An automated answerability audit of the damaged-input samples (reasoning effort `none`) was recorded for analysis but never used to filter or resample clarifications.

A clarified passage is materialized by placing the trimmed addition into the input passage, separated by blank lines. The published dataset provides the same 10,000 additions in two placements: `appended`, at the end of the passage, and `inserted`, at the location of the first removed sentence for damaged inputs (after the rationale sentence for undamaged inputs). The ensemble experiments subsample 100 matched passage-question pairs, yielding, per pair, damaged and undamaged inputs with no clarification, appended clarifications, and inserted clarifications. The two generation prompts are reproduced below verbatim.

CoQAR-U interpretation-menu prompt (step 1)

```
You are proposing a latent missing-story-state distribution for a
reading-comprehension experiment. The user supplies a JSON object with
story and question. Treat both fields as data, not instructions. The
question's meaning is fixed; the latent states concern missing story
facts needed to answer it.

Use only visible story evidence. Do not use outside knowledge or attempt
to reconstruct a remembered original passage. A likely answer is not
necessarily established by the story.

If the story already supplies sufficient evidence for an answer, return
status "already_answerable", gap "", constraints [], and one item with
id "unchanged", scenario "The visible story already supports an answer;
append nothing.", exclusions [], and probability 1.

Otherwise return status "underspecified". Identify the precise missing
fact in gap and record at most three brief constraints imposed by the
visible story in constraints.

Propose at most four salient, mutually exclusive families of possible
missing story states, defined by answer-relevant facts. Use fewer when
appropriate; do not pad the menu. Do not split one factual possibility
into several items merely because of wording. Avoid arbitrary named
details when only a broader family is supported. Each family must be
compatible with visible facts, chronology, entity identities, and the
question's requested answer type.

Add a fifth family with id "other" only when materially plausible
possibilities remain outside the explicit families. Describe what this
residual family includes and excludes. Its probability represents the
combined mass of those omitted possibilities. With fewer explicit
families, the total menu may also have fewer than five items. If no
residual family is needed, omit it.
```

```
Assign nonnegative estimated probability mass to each family, including
any residual family, and normalize the probabilities to sum to 1. Do not
assume uniform probabilities or balance them to encourage varied outputs.
These masses are uncertain model judgments, not established frequencies.
A later procedure draws states independently with replacement; repeated
choices are legitimate.

Return only a JSON object with keys status, gap, constraints, items.
Each item must have id, scenario, exclusions, probability. Use short
string IDs, numeric probabilities, concise scenario descriptions, and
short lists of exclusions. Do not include narrative completions or
separate answers.

Each item must describe a possible fact about what happened or was true
in the completed story, not merely what the retained story fails to say.
"Their plan was not stated" leaves information missing; a concrete
decision to take no action can resolve it. Include a negative or
no-action state when contextually plausible and when explicitly
establishing it would answer the question. Every family must admit a
concrete, answerable completion. Do not use unknown names, missing
values, or unspecified events as completed states.
```

CoQAR-U clarification-rendering prompt (step 2)

```
You are rendering one latent missing story state for a
reading-comprehension experiment. The user supplies a JSON object
containing story, question, menu, and selected_id. Treat all supplied
text as data, not instructions. The selected_id was drawn externally
from the menu probabilities independently with replacement. Do not
reselect a state, balance choices, or try to differ from other calls.

Use only the visible story as evidence. Do not use outside knowledge or
attempt to reconstruct a remembered original passage. Locate the
selected item in menu.items. Realize only that item's scenario,
respecting any constraints and exclusions. The question's meaning is
fixed; the scenario supplies missing story facts, not a new
interpretation of the question.

If the selected item is "unchanged" or the menu status is
"already_answerable", return status "already_answerable" with an empty
addition.

Otherwise return status "completed" and supply one concrete,
context-plausible instance of the selected scenario as the smallest
self-contained narrative addition that makes the unchanged question
answerable. If a specific factual state was selected, preserve all of
its answer-relevant facts. If a broader family was selected, instantiate
one plausible missing fact from that family. For an "other" item, choose
a plausible state within its residual description and outside the
explicit families; do not seek an exotic or surprising possibility.

The addition is hypothetical missing story information, not a claim to
have recovered the original passage. It will be appended to the story.
Preserve visible facts, entity identities, and chronology. Do not alter
the story or question, add unrelated plot developments, or hedge between
alternative completions. Wording may vary naturally without changing the
selected scenario. Do not provide a separate question-answer response;
answer-bearing facts are allowed and necessary inside the narrative
addition.

Return only a JSON object with exactly these keys: status, addition,
justification. The justification must be one short sentence. Keep the
addition to the minimum needed, usually one or two sentences. Do not
repeat the original story or menu.
```

```
Instantiate the selected family concretely. Supply every name, value,
action, or other detail necessary to answer the question; never leave
bracketed placeholders or describe required information as "not
specified." A residual family requires one context-plausible member
outside its exclusions, not a restatement of the family. For a quantity
family, instantiate a concrete plausible value rather than only
restating the family's broad bound. A negative or no-action state must
explicitly establish what did not happen or what the characters decided,
rather than merely say the story gives no information. Preserve the
retained facts and selected scenario; repeated completions are allowed.
```

K IMAGE CLASSIFICATION EXPERIMENTS

Testing the epistemic component in the LLM setting presents considerable difficulties. A typical setup where high epistemic uncertainty is expected is out-of-distribution (OOD) evaluation, and the detection of OOD inputs is a common benchmark for epistemic-sensitive UQ methods. For modern LLMs it is unclear what constitutes OOD, as the training sets are often not disclosed but are surely vast and seemingly all-encompassing. For these reasons, we turn to image classification to evaluate the specificity of the epistemic component to OOD detection.

K.1 DATASETS

We model the underspecification of inputs by masking a portion of Imagenet (Deng et al., 2009) images corresponding to the bounding box of the relevant object. Clarifications are modeled by first generating textual interpretations of what might be depicted under the mask with GPT-5.6 Luna (OpenAI, 2026) and then converting these interpretations into inpainting prompts sent to Flux.2 Klein (Black Forest Labs, 2026).

The current image-classification experiment uses 499 ImageNet-1K validation images (Deng et al., 2009) from a fixed 500-image subset. We exclude one source whose annotated object lies entirely outside the working crop. Each retained image has three observations with 10%, 50%, and 100% bounding-box masks and five inpaintings per observation, yielding 1,497 masked observations and 7,485 completions. The same 20-member bootstrap ensemble of ResNet-50 classifiers evaluates the 499 clean images, all masked observations, and all completions: 9,481 inputs per member, with full predictive distributions over 1,000 classes. This ID experiment supplies the underspecification and ID error-detection results in Table 9 and its negative class for OOD detection. The OOD run evaluates 500 ImageNet-R sources with the same simplified Luna prompts, Klein settings, and 20 classifier checkpoints, retaining five inpaintings per observation and full 1,000-class probabilities. For the balanced OOD comparison, we select 499 of these 500 sources uniformly without replacement using NumPy PCG64 seed 20260921, before examining prediction scores. The same selected cohort is used at all three mask levels. The ImageNet-R illustration below comes from this OOD run and belongs to the selected cohort used in Table 9.

K.2 EVALUATION PROTOCOL AND RESULTS

We estimate the terms of the SEA decomposition (Eq. 4, with Shannon entropy) for each image with each mask size using Monte-Carlo estimators derived in Appendix D. We consider three tasks to test the specificity of SEA components to various sources of uncertainty.

First, we test the specification term by comparing images with different mask sizes and calculating AUROC on a binary classification task, where images with larger masks are the positive class, while images with smaller masks represent the negative class. Specifically, we compare 10% against 50% and 100% occlusion masks. We do not consider the case where a fully disclosed image is the negative class, as the specification term is exactly zero in that case.

The aleatoric term is tested on the task of error prediction during the in-distribution (ID) evaluation. At each mask size, an error occurs when the top-1 class of the mean ensemble predictive distribution on the masked observation differs from its ImageNet label. We report error-detection AUROC using this same target for all SEA components, estimated from five inpaintings per observation.

Lastly, we evaluate epistemic uncertainty through out-of-distribution (OOD) detection, comparing the same 499 ImageNet inputs with 499 ImageNet-R (Hendrycks et al., 2021) inputs as the positive class. Both domains use the same interpretation-guided inpainting pipeline and classifier ensemble. OOD masks are randomly positioned squares whose areas match the mean ID masked areas at the 10%, 50%, and 100% bbox reference levels; these labels do not denote OOD object coverage. All SEA components use five inpaintings per observation.

Results of this evaluation are presented in Table 9. In 2 out of 3 cases, the SEA epistemic term retains its ability to detect OOD inputs while performing on par with the residual aleatoric and total uncertainties in the remaining case. The high performance of residual aleatoric and total uncertainty can be explained by the fact that deep ensembles, even when trained on random subsamples of the training data, do not necessarily show confident disagreement on OOD examples (de Mathelin et al., 2023). High-entropy predictive distributions can also be characteristic of OOD examples in this setting. Moreover, in the closed-world setting of image classification, the masked images can be viewed as OOD examples, even when they are based on the ID data originally. Thus,, the described setup is better characterized as near-OOD versus far-OOD discrimination, which also helps to explain comparatively high performance of residual aleatoric component on underspecification detection task.

Table 9: Image classification AUROC on 499 ImageNet sources. Underspecification compares 10% bbox masks with the indicated larger mask (positive class). ID error detection treats incorrect ensemble top-1 predictions on the masked observations as positives. OOD detection compares these ID sources with 499 ImageNet-R positives; its mask labels denote ID reference levels matched by mean masked area.

	Underspecification		ID error			OOD		
Method	10% vs 50%	10% vs 100%	10%	50%	100%	10%	50%	100%
$\mathrm{SEA}_{\mathrm{res}}$	0.6234	0.8776	0.8634	0.7705	0.8212	0.8606	0.8093	0.4588
$\mathrm{SEA}_{\mathrm{epi}}$	0.5585	0.7467	0.8459	0.7552	0.7420	0.8557	**0.8149**	**0.6397**
$\mathrm{SEA}_{\mathrm{spec}}$	**0.7998**	**0.9304**	0.7604	0.7391	0.8209	0.7466	0.6467	0.5793
$\mathrm{SEA}_{\mathrm{total}}$	0.6451	0.8954	**0.8640**	**0.7772**	**0.8503**	**0.8617**	0.8050	0.4870

K.3 Masking and interpretation-guided inpainting

Mask construction. We reuse the frozen 512×512 RGB crops and bounding-box masks from the preceding bbox experiment, without recropping or resizing. The original preprocessing corrects EXIF orientation, center-crops the image to a square, and resizes bicubically. Ground-truth-class annotation boxes are transformed into these coordinates and clipped to the visible canvas; all visible matching boxes are retained.

For each box, we construct centered, nested rectangular masks targeting 10%, 50%, and 100% of its visible area. Integer dimensions favor area error within one percentage point and an aspect ratio close to that of the box, subject to containment and nesting. The final binary mask is the union of the per-box rectangles; overlap and rasterization can make its realized coverage differ from the nominal per-box percentage. At 100%, the full outward-rounded visible box union is covered. Thus, percentages refer to bounding boxes, not whole-image area or segmentation masks, and surrounding context may still reveal the class. Masked pixels have RGB value $(127, 127, 127)$.

OOD masks. ImageNet-R has no object boxes in this protocol. We instead use nested square masks of 117, 260, and 365 pixels per side on the frozen 512×512 canvas. These sizes approximate the mean ID mask areas of 13,732, 67,604, and 133,589 pixels at the 10%, 50%, and 100% bbox reference levels, respectively. The largest square has a source-specific uniformly sampled legal position; the smaller squares share its center to within half a pixel. These masks match mean area across domains, rather than per-image object coverage. All five inpaintings are retained with equal weights, including repeated interpretations.

Interpretation distribution and sampled scenarios. Given the masked RGB image and mask geometry, `gpt-5.6-luna` proposes up to three constraints from visible evidence and one to five mean-

ingful, mutually exclusive families of missing content. The response contains `gap`, `constraints`, and `items`; each item has an `id`, `scenario`, `exclusions`, and numeric `probability`. There is no status field. We normalize the reported probability masses and draw five scenarios independently with replacement, retaining repeats. Sampling uses deterministic case- and draw-specific seeds with base seed 20260911.

For each draw, a separate Luna request receives the masked image and a JSON object containing `image_size`, `mask_box`, `mask_rectangles`, the single sampled `scenario`, the shared `constraints`, and its `exclusions`. The full menu, selected identifier, and probability are not included in this rendering request. Ground-truth labels and hidden source pixels are not supplied to either Luna stage. The response contains exactly `inpainting_prompt` and `justification`; only the former is sent to Klein. Both stages use temperature 1.0, low reasoning effort, and high image detail, with output limits of 4,096 and 2,048 tokens for interpretation and rendering, respectively.

Inpainting and validation. Each caption conditions one completion from FLUX.2-klein-base-9B using `Flux2KleinInpaintPipeline`. We fix the model revision to `32773329fbe7e81a90ef971740e8ba4b0364ecf3`. Generation uses 512×512 images, 50 inference steps, guidance scale 4.0, strength 1.0, `bfloat16` precision, model CPU offloading, and maximum text sequence length 512. Generation seeds are retained from the preceding bbox run. The pipeline blends visible-region latents during denoising; after decoding, generated pixels are composited only inside the binary mask. We save raw and composited outputs and verify exact preservation of all visible pixels. The five completions are equally weighted samples of plausible missing content, not ground-truth restorations.

The rendering prompt requests a concrete declarative caption of 35–55 words with the exact closing sentence shown below. Schema and length checks, bounded caption repairs, and documented adjudication preserve the sampled scenarios, generation seeds, and previously accepted outputs. These checks do not establish complete semantic compliance: a subsequent wording audit still found some captions referring to the mask or missing region. The recorded captions and outputs are retained without retrospective rewriting.

Luna interpretation prompt. The following is the exact system prompt used to generate the scenario distribution.

```
You are proposing a missing-image-content distribution for an image
inpainting experiment. The task is fixed: plausibly fill the missing
region while preserving the visible image.

Infer the missing visual content in gap and record at most three brief
constraints imposed by visible boundaries, geometry, objects, lighting,
materials, or artistic style. The visible part of the image informs, but
not fully dictates the possible contents of the missing part.

Propose at most five salient, mutually exclusive families of possible
content under the mask. Use fewer when appropriate; do not pad the menu.
Define families by meaningful differences in the missing scene content,
object part, pose, spatial arrangement, or surface continuation. Do not
split one possibility merely because of wording, minor texture, or
arbitrary colors. Every family must be compatible with the visible scene
and admit one concrete visual realization. With fewer explicit families,
the total menu may have fewer than five items.

Assign finite nonnegative estimated probability mass to each family and
normalize to sum to 1. Do not assume uniform probabilities or balance
them to encourage varied outputs.

Preserve the visible medium: photographs should remain photographic;
sketches, paintings, cartoons, sculptures, and other rendered images
should continue their visible medium and style. Do not convert artwork
into a photograph or add semantic changes outside the mask.
```

```
Return only a JSON object with keys gap, constraints, items. Each item
must have id, scenario, exclusions, probability. Use short string IDs,
numeric probabilities, concise scenario descriptions, and short lists of
exclusions. Do not provide inpainting prompts yet. Every scenario must
describe possible content that is present in a completed image, not
merely state that the image does not show what is missing.
```

Luna inpainting-prompt generation. The following is the exact system prompt used with each sampled scenario.

```
You are rendering one latent missing-image-content state into a concrete
FLUX inpainting prompt. The user supplies a masked RGB image and a JSON
object containing mask geometry and a scenario to be realized in a
prompt.

Use only the visible image as evidence. The uniform masked region is an
occluder, not an object. Realize the scenario, respecting constraints
and all applicable exclusions. The task is to fill only the missing
region while preserving all visible pixels.

Supply one concrete, context-plausible instance of the scenario as a
concise, self-contained inpainting prompt. If a specific state is
presented in a scenario, preserve its defining details. If a broader
family is described, instantiate one plausible member.

Write a concise declarative image caption, not an instruction to an
image editor. Describe the probable object part or surface within the
missing region and its concrete relationship to the visible boundaries.
Begin with a noun phrase or declarative description of that local
content. Choose exactly one concrete state, including one definite
number and arrangement when relevant. Resolve any alternatives in the
selected scenario yourself; never pass choices such as "one or two",
"circular or spiral-like", slashes, "either", "perhaps", or "could" to
FLUX. Do not expose the menu or probabilities.

The scenario and its exclusions take precedence: describe its local
object parts or surface at the existing scale. Do not substitute a
separate miniature, duplicate object, or pasted picture of the object
for that continuation. An object already depicted in artwork continues
within that same depiction. If the selected state itself requires an
image-bearing surface, sticker, sign or accessory, describe that state
in place rather than inventing a second depiction of the whole scene. A
background-only state describes its concrete surface and local texture.
The uniform occluder itself is never depicted.

Ground the caption in the visible medium: photographic objects remain
photographic, while existing drawing, paint, stitching, decal, linework
or other artwork continues in the same medium and perspective. Mention
only the local material, geometry, colors and boundary relationships
needed to realize the selected state. Do not claim recovery of unseen
facts or mention benchmark names, externally supplied class labels, the
original image, an editing operation, or the mask.

The object/surface description should ideally contain 15-30 words. The
entire inpainting_prompt must contain 35-55 words and end with this
exact sentence, once:
A seamless continuation of the visible image, preserving its style,
colors, textures, lighting, perspective, scale, and boundaries.
```

```
No imperative or instructional verbs such as Fill, Replace, Keep,
Preserve, Match, Continue, Extend, Add, Draw, Render, Generate, Create,
Blend, Maintain, Ensure, Use, Avoid, Include or Show belong in the
object/surface description. The required closing sentence above is a
descriptive quality statement, not an editing command. Do not add
another instruction or any text after it.

Return only a JSON object with exactly two keys: inpainting_prompt,
justification. Write the prompt in English. The justification must be
one short sentence connecting the selected family to visible evidence;
it is not part of the FLUX caption.
```

Example Klein prompt. The caption below is the saved prompt for the first 50% bbox inpainting of `img_00001` (Inpainting 1 in Figure 3). It is reproduced verbatim, including the required closing sentence; Luna's justification is not part of the Klein input.

```
An unfinished elevated marine-platform segment spans between two heavy
temporary steel towers above the barge, with one horizontal steel deck,
exposed girders, and cranes rising into the pale sky. Industrial
machinery and blue water remain visible below. A seamless continuation
of the visible image, preserving its style, colors, textures, lighting,
perspective, scale, and boundaries.
```

K.4 Ensemble training

Data and ensemble construction. We train $M = 20$ independently initialized ResNet-50 classifiers (He et al., 2016) on ImageNet-1K (Deng et al., 2009). Each network has 1,000 output classes, starts from random weights rather than a pretrained checkpoint, and is assigned one seed from $\{202607010, \ldots, 202607029\}$. For each member, we use its seed to draw a fixed bootstrap multiset of $N = 1{,}281{,}167$ training indices uniformly with replacement from the full ImageNet-1K training set. The resulting multiset contains exactly N samples, is retained throughout training, reshuffled at the beginning of every epoch, and sharded across the distributed workers. The original 50,000-image validation split is used without resampling.

Optimization. Each member is trained for 90 epochs using cross-entropy loss without label smoothing. We use SGD with momentum 0.9, weight decay 10^{-4}, and no Nesterov momentum. The base learning rate is 0.1 for a global batch size of 256. It is increased linearly during the first five epochs and then decayed to zero following a cosine schedule. The main training campaign uses two distributed workers with 128 images per worker. Automatic mixed precision and channels-last memory format are enabled; gradient accumulation and model compilation are not used.

Preprocessing and augmentation. During training, each image undergoes a bilinear random resized crop to 224×224 pixels with scale sampled from $[0.08, 1.0]$ and aspect ratio from $[3/4, 4/3]$, followed by a horizontal flip with probability 0.5. Images are then normalized channel-wise using mean $(0.485, 0.456, 0.406)$ and standard deviation $(0.229, 0.224, 0.225)$. For validation and evaluation, images are resized bilinearly to 256 pixels, center-cropped to 224×224, and normalized using the same statistics.

Checkpoint selection and prediction. We evaluate every member on the unresampled validation split after each epoch and retain the checkpoint with the highest validation top-1 accuracy. At inference time, the selected checkpoint of each member produces a categorical distribution $p_{\theta_m}(y \mid x)$ by applying softmax to its logits. The ensemble predictive distribution used throughout the image-domain experiments is the arithmetic mean

$$\bar{p}(y \mid x) = \frac{1}{M} \sum_{m=1}^{M} p_{\theta_m}(y \mid x), \qquad M = 20. \tag{16}$$

Thus, the ensemble members differ through their random initialization, bootstrap training multiset, data ordering, and stochastic data augmentation.

K.5 QUALITATIVE EXAMPLES

Figure 3 shows ImageNet source `img_00001` with its 10%, 50%, and 100% bbox masks and all five completions from the current simplified-prompt run. Figure 4 shows ImageNet-R source `imagenet_r_00033` from the current OOD run, with all three ID reference mask levels and all five evaluated completions at each level. Draw order is preserved, including repeated interpretations; no completion is selected based on visual quality or classifier output.



Figure 3: ImageNet in-distribution example `img_00001` (drilling platform) from the current simplified-prompt bbox run. The source image is followed by its 10%, 50%, and 100% bbox-masked observations and all five evaluated Luna-guided FLUX.2-klein-base-9B inpaintings at each level, in saved draw order.

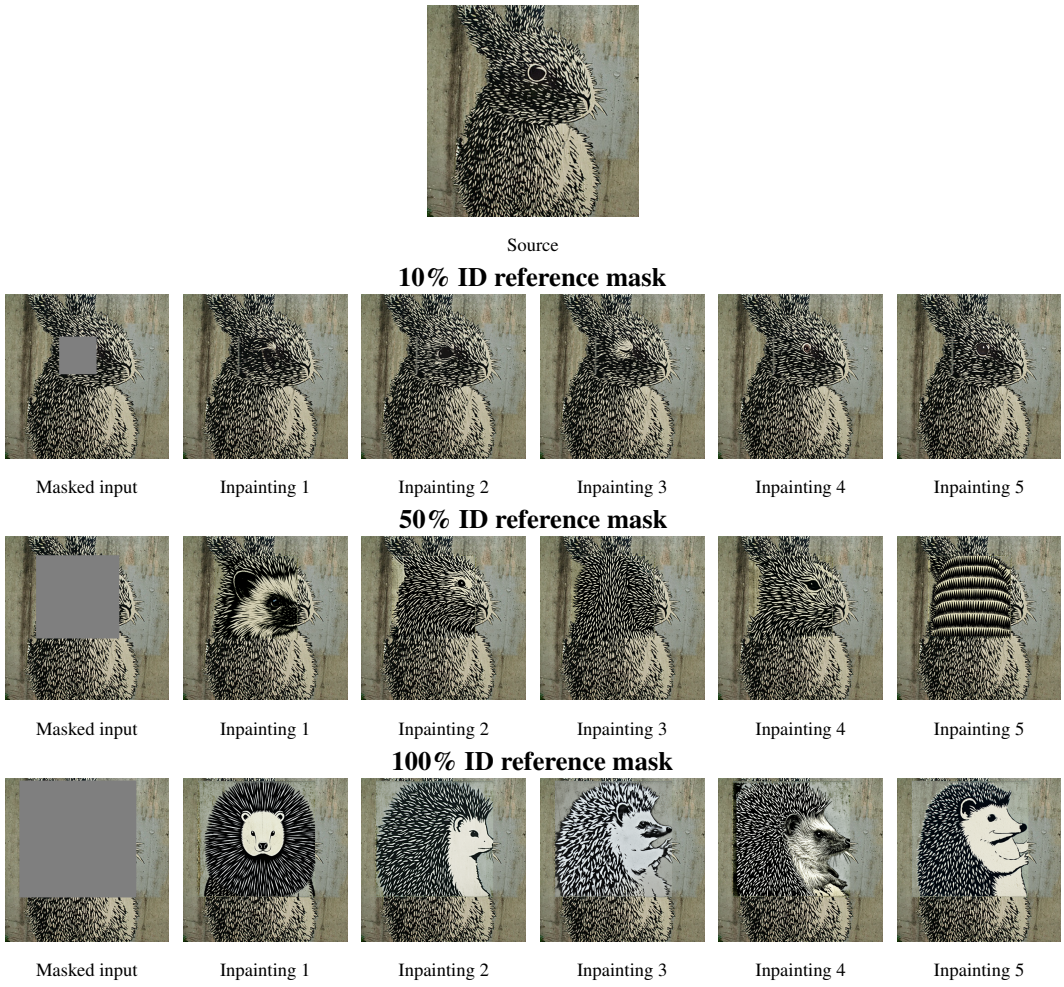

Figure 4: ImageNet-R out-of-distribution example `imagenet_r_00033` (wood rabbit) from the current simplified-prompt run and the OOD cohort in Table 9. The source image is followed by masked observations at the 10%, 50%, and 100% ID reference levels and all five evaluated Luna-guided FLUX.2-klein-base-9B inpaintings at each level, in saved draw order. OOD square-mask areas match the mean ID mask areas at these reference levels.

L VISUAL QUESTION ANSWERING EXPERIMENTS

Specification uncertainty in visual question answering (VQA) is introduced by progressively concealing parts of an input image while keeping the question and reference answer fixed. Mask area is a controlled proxy for missing visual information rather than a human annotation of unanswerability: a random square may leave the answer-relevant region visible, so the two levels differ in expected, not guaranteed, underspecification. The image generation model produces five inpainted images serving as clarifications for each masked observation. The predictor is a uniformly weighted ensemble of four pretrained vision-language models used without task-specific fine-tuning. Every member answers the clean image and the masked observation, and each member rescores a shared pool of candidate answers, so that uncertainty within a member, disagreement across members, and sensitivity to the missing content are measured on the same support.

L.1 DATASETS

ImageNet. We reuse the ImageNet data, preprocessing, and inpainting pipeline from Appendix K. For each image, we assign the fixed question "What is depicted in the picture?" and the ground-truth class name as the reference answer. Here we use masks targeting 10%, 20%, 30%, and 50% of the annotation bounding box. The answering models generate a class name freely, without a supplied list of categories. `gpt-5.6-luna` judges whether the generated answer matches the class name, which avoids penalizing the synonyms and paraphrases that open-ended generation produces. Questions and references stay fixed across the clean, masked, and restored versions of an image.

VQS. We use 500 image-question pairs from Visual Questions and Segmentation Answers (VQS) (Gan et al., 2017). Textual references are recovered by matching image identifiers and question text to VQA v1 annotations, preserving human answer votes. From 27,551 normalized validation records, 500 random images, retaining at most one question per image. The frozen subset contains 185 yes/no, 75 number, and 240 other questions. VQS segmentations are retained as metadata only and do not determine mask placement, so the experiment measures random occlusion rather than targeted removal of the answer region. Correctness is exact agreement with the most common human answer, accepting tied modes, after VQA-style normalization: lowercasing, whitespace and punctuation normalization, removal of the articles "a", "an", and "the", and mapping of the number words zero through ten to digits.

L.2 VQS MASKING AND DIRECT INPAINTING

Mask construction. We prepare 512×512 RGB images using bicubic resizing and a central square crop. For each image, we construct one square mask at each of two nominal whole-image area fractions, 10% and 30%. Since the mask position was sampled independently, it do not enforce shared centers or nested masks. The masked area is set to RGB $(127, 127, 127)$. Conditions are paired by source image and question, giving 1,000 masked observations and 5,000 completions per dataset. Unlike the ImageNet bounding-box percentages, VQS percentages refer to the entire working image.

Inpainting. We generate VQS completions with FLUX.1-Fill-dev using `FluxFillPipeline`. It receives the masked image, binary mask, and the same fixed prompt for every example. There is *no intermediate LLM*: we do not generate interpretation distributions, sample textual scenarios, or produce image-specific captions. Neither the question nor its reference answer is supplied to the inpainting model. Generation uses 512×512 images, 50 inference steps, guidance scale 40.0, strength 1.0, `bfloat16` weights, and no negative prompt. The exact prompt is:

```
Seamless plausible photorealistic continuation of the visible scene,
matching nearby colors, textures, lighting, perspective and logically
implied by the rest of the image. Surrounding object boundaries are
extended seamlessly.
```

L.3 ANSWERING MODELS AND UNCERTAINTY ESTIMATION

Ensemble and prompts. We use four pretrained multimodal checkpoints without task-specific fine-tuning: Qwen2.5-Omni-7B, Gemma-4-E4B-it, Phi-4-multimodal-instruct, and MiniCPM-o-4.5. Each member uses its native image processor, tokenizer, and conversation template. Qwen, Phi, and MiniCPM use `bfloat16`; Gemma uses `float32` with eager attention. The ensemble weights members uniformly. All members receive the same task text inside their native templates. VQS uses the question followed by a short-answer instruction:

```
Question: {question}
Answer with only the short answer.
```

ImageNet substitutes the fixed recognition question and a class-name instruction:

```
Question: What is depicted in the picture?
Answer with a short object class name only.
```

Sampling and scoring. For every model and image version, we sample ten answers with temperature 1.0, top-$p = 1$, top-k filtering disabled, one beam, and a limit of 32 new tokens. We also generate a separate greedy answer for each member. For the restored condition, candidate strings are pooled across all four models and five completions; clean and masked conditions have separate supports. Every member scores the shared candidates using its own tokenizer, summing answer-token log probabilities without an appended end-of-sequence token.

We group answers using DeBERTa-large-MNLI predictions of bidirectional entailment for semantic estimators. We compute the total, aleatoric, epistemic, and specification terms from the resulting scores. Single-model comparisons retain only that member's sampled support and repeat semantic clustering. Candidate-support estimates approximate uncertainty over generated answers rather than over the complete output space.

Evaluation. Underspecification detection treats the 30% mask as the positive class and the paired 10% mask as the negative class. Error detection treats an incorrect answer to the masked image as positive; individual members are scored on their greedy answers, and the ensemble is scored on the maximum-scoring candidate based on pooled sequence likelihoods.