Localizing Input Uncertainty Quantification for Large Language Models via Shapley Values

Seongjun Lee*, **Suwan Yoon***, **Changhee Lee**†
Dept. of Artificial Intelligence, Korea University
Seoul, Republic of Korea
{pyoung7307,suwanyoon,changheelee}@korea.ac.kr

Abstract

As large language models (LLMs) are increasingly integrated into high-stakes decision-making, the ability to reliably quantify uncertainty has become a critical requirement for safety and trust. However, current uncertainty quantification methods primarily operate at the output level, often failing to distinguish whether uncertainty arises from the model's lack of knowledge or from ambiguity in the user's input. While input-centric uncertainty quantification has recently emerged as a promising direction, it remains relatively underexplored and typically relies on coarse, input-level information. Consequently, users are provided with scalar uncertainty scores that offer little actionable guidance on *which parts of the input* should be clarified to improve reliability. To address this limitation, we propose **Sha**pley-based input uncertainty **Q**uantification (`ShaQ`), a framework for *span-level attribution* of input-induced uncertainty. Our approach models ambiguous spans in the input as players in a cooperative game and quantifies their contributions using Shapley values, defined via the weighted average of marginal reductions in conditional entropy obtained by clarifying each span coalition. Unlike existing input-level approaches, our formulation explicitly captures complex interactions among spans and provides a principled decomposition in which individual attributions sum exactly to the total input-induced uncertainty. We evaluate `ShaQ` on the AmbigQA and AmbiEnt benchmarks, where it achieves state-of-the-art performance in ambiguity detection. We further demonstrate its practical utility on a MediTOD benchmark, showing that `ShaQ` can precisely localize under-specified clinical utterances, facilitating more effective human-AI collaboration in high-stakes settings. Overall, our results show that `ShaQ` not only improves uncertainty estimation but also provides actionable insights for targeted input clarification. Codes are available here.

1 Introduction

Large language models (LLMs) have demonstrated strong zero-shot performance across a wide range of tasks and are increasingly deployed as decision-support tools in many real-world settings [1–4]. However, they often produce confident responses even when the input is ambiguous or under-specified [5, 6], making it critical to quantify the uncertainty in their outputs for trustworthy deployment. This uncertainty can be formalized within the classical uncertainty quantification (UQ) framework [7–12], which decomposes predictive uncertainty into *epistemic uncertainty* – stemming from incomplete knowledge of model parameters – and *aleatoric uncertainty* – arising from ambiguity or noise intrinsic to the input [13]. Building on this framework, a growing body of work has focused on estimating uncertainty in LLM outputs using signals such as token-level likelihoods, semantic consistency across sampled responses, or model parameter perturbations [14–19]. While these approaches have proven effective for detecting hallucinations and confabulations [20, 21], they

*Equal contribution. †Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

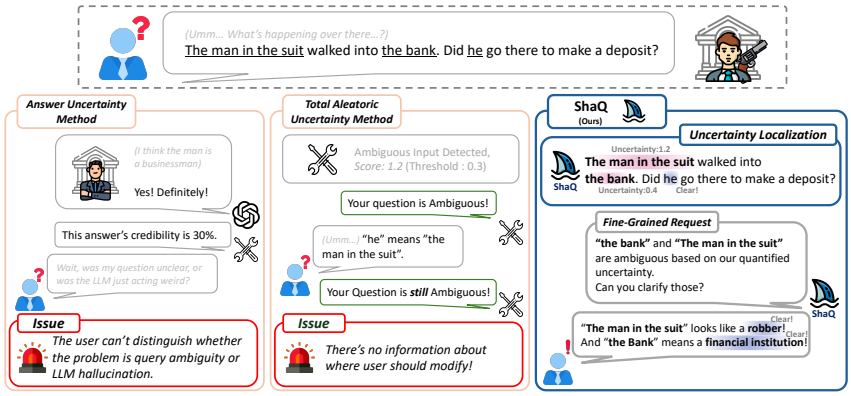


Figure 1: **Comparison of uncertainty methods to LLM-user interaction.** (**Left**) Output-level uncertainty methods provide confidence scores, but do not distinguish whether uncertainty arises from input ambiguity or model hallucination. (**Center**) Aggregate aleatoric uncertainty methods identify ambiguous inputs, but lack localization. (**Right**) ShaQ localizes uncertainty to individual spans, enabling targeted clarification of the specific sources of ambiguity.

remain fundamentally limited: a scalar uncertainty score indicates that an output is uncertain, but provides no actionable insight into *which part of the input* should be revised to resolve that ambiguity.

This limitation highlights a fundamental gap that reducing uncertainty ultimately requires understanding its origin within the input itself. However, input-centric uncertainty analysis remains largely underexplored [22]. The need for such analysis is especially pronounced in practical deployments, where LLMs are accessed via black-box APIs and modifying model parameters to reduce epistemic uncertainty is infeasible. In these settings, improving reliability must instead rely on resolving the *aleatoric uncertainty* inherent in the input. Yet, existing approaches to input-level uncertainty provide only an aggregate score for the entire input [23], offering little guidance on which parts of the input contribute to the uncertainty. As illustrated in Figure 1, the query *"The man in the suit walked into the bank. Did he go there to make a deposit?"* contains multiple ambiguous expressions: *the man* may refer to either a *robber* or a *customer*, while *the bank* may denote a *financial institution* or a *riverbank*. These ambiguous spans can interact in complex ways, further compounding uncertainty in the model's predictions. Without localization, users are often forced to clarify the entire input without knowing where ambiguity originates, leading to unnecessary effort and limited practical benefit.

To address this gap, we propose ***Sha****pley-based input uncertainty* ***Q****uantification* (ShaQ), a framework that localizes input-induced uncertainty to individual ambiguous spans within an input. ShaQ operates by resolving each span's ambiguity through *premises* – declarative statements that disambiguate a span's semantic role – and measuring the consequent reduction in predictive entropy. The fundamental challenge, as illustrated in our running example, is that ambiguous spans are often interdependent: the uncertainty resolved by one span is inherently conditioned on which other spans have already been clarified. This complex interaction structure motivates a Shapley formulation, where ambiguous spans act as *players* and an *information-theoretic* value function defines their coalitional utility in a cooperative game. To compute these values, we introduce a *bottom-up marginalization* algorithm that evaluates marginal contributions across all possible coalitions while provably guaranteeing non-negative uncertainty estimates. The resulting framework provides a principled and fine-grained decomposition of input-induced uncertainty, identifying *where* ambiguity arises and enabling users to focus their clarification efforts precisely where they are most needed.

Our contributions are summarized as follows:

- We propose ShaQ, a novel Shapley-based framework for localizing the input-induced uncertainty of LLM inputs at the span level.
- We theoretically show that ShaQ generalizes existing input-level uncertainty quantification methods [23], providing a principled decomposition into span-level contributions.
- We conduct experiments on AmbigQA [24] and AmbiEnt [25], where ShaQ achieves state-of-the-art ambiguity detection, and demonstrate its practical utility on MediTOD [26] in real-world medical dialogues.

2 Related Works

Uncertainty Quantification. Early work on uncertainty quantification (UQ) focused primarily on classification settings, estimating predictive uncertainty via posterior approximations, dropout, ensembles, or architectural modifications [8, 11, 12, 9, 10]. However, these approaches do not transfer readily to modern LLMs, which involve autoregressive generation, massive parameter scales, and black-box access. Consequently, recent LLM UQ research has pursued several alternative directions: exploiting token likelihoods or output entropy [14], measuring semantic consistency across multiple sampled outputs [17, 21, 18, 19], assessing prediction stability under input or model perturbations [16, 15], or eliciting verbalized confidence scores directly from the model [27]. Our framework departs from this perspective: rather than quantifying uncertainty in a generated answer, we localize and quantify the sources of ambiguity that cause an input to admit multiple valid answers.

Shapley Values for Explanation and Uncertainty. Shapley values originate from cooperative game theory as a principled solution for fairly distributing a coalition's payoff among players [28]. SHAP introduced this framework to machine learning by attributing a model's prediction to individual input features [29, 30]. Recent work extends this idea to uncertainty attribution by replacing the payoff with information-theoretic quantities such as predictive or conditional entropy [31], but remains confined to supervised settings with fixed feature spaces and tractable predictive distributions. How to define and estimate Shapley values for uncertainty quantification in LLMs – accounting for black-box access, semantic equivalence, and free-form generation – remains underexplored.

3 Problem Formulation

Notation. Let $\mathcal{D}$ denote an observed dataset $\mathcal{D}$ composed of inputs $X \in \mathcal{X}$ and corresponding outputs $Y \in \mathcal{Y}$, respectively. Following standard conventions, we use uppercase letters for random variables and lowercase letters for their realizations. For notational convenience, we omit the explicit conditioning on $\mathcal{D}$ and θ when it is clear from context.

3.1 Input Uncertainty Quantification in LLMs

A model's predictive uncertainty is conventionally decomposed into *epistemic* uncertainty, stemming from incomplete knowledge of model parameters, and *aleatoric* uncertainty, arising from ambiguity intrinsic to the input [13]. Reducing epistemic uncertainty requires access to model parameters, which is infeasible in black-box LLM deployments. To circumvent these limitations, recent work shifts the source of uncertainty from the parameter space to the input space [23]. Specifically, the model parameters are fixed to θ, and uncertainty is instead characterized through a latent variable C that represents plausible clarifications of an ambiguous or under-specified input x. The predictive uncertainty for a given input x is then quantified by the entropy $\mathcal{H}(Y \mid x)$, that can be decomposed as

$$\mathcal{H}(Y \mid x) = \underbrace{\mathcal{I}(Y; C \mid x)}_{\text{input-induced uncertainty}} + \underbrace{\mathbb{E}_{c \sim q(C|x)}\big[\mathcal{H}(Y \mid x, c)\big]}_{\text{remaining uncertainty}}, \tag{1}$$

where $q(C \mid x)$ denotes an approximate distribution over plausible clarifications given x. The mutual information term in (1) captures the uncertainty induced by input ambiguity, and following previous work [23], we treat it as an estimate of the *aleatoric uncertainty* of the input. The expected conditional entropy term quantifies the residual uncertainty after clarification, and under the assumption that input ambiguity is the primary source of aleatoric uncertainty, it serves as a proxy for the model's irreducible uncertainty. However, (1) treats aleatoric uncertainty as a monolithic property of the entire input, offering no means to attribute individual contributions of each input component.

3.2 Localizing Aleatoric Uncertainty

In this work, we aim to localize aleatoric uncertainty at a finer granularity by attributing it to specific components of the input. A central challenge is that, unlike structured data with predefined feature boundaries, natural language lacks a canonical decomposition into non-overlapping semantic units. Consequently, quantifying how different parts of the input contribute to overall uncertainty first requires identifying the units over which ambiguity arises.

Importantly, ambiguity is often context-dependent: a model's prediction can vary depending on how multiple phrases are jointly interpreted. This indicates that uncertainty is not solely a property of

individual terms, but also emerges from interactions among them. To capture these dynamics, we focus on contiguous phrases that form the minimal, semantically coherent units capable of supporting multiple plausible interpretations. We refer to these functional units as *ambiguous spans*:

Definition 3.1 (**Ambiguous Span**). An *ambiguous span* is a contiguous subsequence of the input sequence that constitutes a minimal semantically coherent unit admitting multiple plausible interpretations. We denote by $\mathcal{S} = \{s_1, \ldots, s_n\}$ a collection of non-overlapping ambiguous spans extracted from x.

Given an ambiguous span collection $\mathcal{S} = \{s_1, \ldots, s_n\}$, we associate each span s_k with a *clarification* random variable C_k, whose distribution $q(C_k \mid x)$ characterizes the set of plausible interpretations of s_k. We denote the joint clarification as $C = (C_1, \ldots, C_n)$ and the joint clarification excluding the k-th span as $C_{\setminus k}$. We use c, c_k, and $c_{\setminus k}$ to denote corresponding realizations, respectively.

A natural approach to quantifying the contribution of a span s_k is to measure how much information about Y is provided by its clarification C_k once all other ambiguities have been resolved. We formalize this intuition using a leave-one-out (LOO) attribution:

Definition 3.2 (**LOO Span Attribution**). The LOO attribution of span s_k is defined as the conditional mutual information

$$\phi_k^{\mathrm{LOO}} \triangleq \mathcal{I}(Y; C_k \mid x, C_{\setminus k}) = \mathcal{H}(Y \mid x, C_{\setminus k}) - \mathcal{H}(Y \mid x, C), \tag{2}$$

which quantifies the information about Y contributed by clarifying s_k after all remaining spans have been resolved.

Here, the second equality expresses the attribution as the difference between two conditional entropies. This form is particularly useful for estimation, as it allows us to approximate ϕ_k^{LOO} by sampling model outputs under partially clarified (i.e., conditioned on $C_{\setminus k}$) and fully clarified (i.e., conditioned on C) inputs. We provide the derivation in Appendix A.

Challenges. While Definition 3.2 provides a principled per-span attribution, it evaluates each span only in the setting where all other spans have already been resolved. When spans are semantically dependent, conditioning on $C_{\setminus k}$ can implicitly constrain the plausible interpretations of s_k. In our running example, clarifying *"the man in the suit"* as a bank robber effectively eliminates the riverbank interpretation of *"the bank"*, thereby reducing the uncertainty originally associated with that span. Such dependencies hinder accurate measurement of how much each span contributes to the total aleatoric uncertainty, and thus the sum of LOO attributions does not, in general, equal the total aleatoric uncertainty $\mathcal{I}(Y; C \mid x)$. To address this limitation, we introduce a *cooperative game formulation* based on Shapley values [28]. This approach accounts for the marginal contribution of each span across all possible subsets (coalitions) of clarifications, providing a fair allocation of uncertainty that satisfies the efficiency axiom [32].

4 The ShaQ Framework

4.1 Uncertainty Localization via Shapley Values

We formulate uncertainty localization as a cooperative game in which each ambiguous span $s_k \in \mathcal{S}$ acts as a player. The goal is to fairly distribute the total "payoff" – defined as the total aleatoric uncertainty $\mathcal{I}(Y; C \mid x)$ – across these spans. Let $\mathcal{N} = \{1, \ldots, n\}$ be the set of indices corresponding to the spans in $\mathcal{S}$. We specify the game by defining a value function over coalitions of spans $\mathcal{A} \subseteq \mathcal{N}$.

Definition 4.1 (**Value Function**). The value function $v : 2^N \to \mathbb{R}$ assigns to each coalition $\mathcal{A} \subseteq \mathcal{N}$ the mutual information between Y and the joint clarifications of the spans indexed by $\mathcal{A}$:

$$v(\mathcal{A}) = \mathcal{I}(Y; C_{\mathcal{A}} \mid x) = \mathcal{H}(Y \mid x) - \mathbb{E}_{c_{\mathcal{A}} \sim q(C_{\mathcal{A}} \mid x)}\big[\mathcal{H}(Y \mid x, c_{\mathcal{A}})\big], \tag{3}$$

where $C_{\mathcal{A}} = \big(C_j : j \in \mathcal{A}\big)$ denotes the joint clarifications associated with the spans in $\mathcal{A}$. By convention, we have $v(\emptyset) = 0$ and $v(\mathcal{N}) = \mathcal{I}(Y; C \mid x)$.

Based on the value function in (3), we define the *uncertainty attribution* for each span s_k using the Shapley value ϕ_k, which is defined as the weighted average of its marginal contributions $v(\mathcal{A} \cup \{k\}) - v(\mathcal{A})$ over all possible coalitions $\mathcal{A} \subseteq \mathcal{N} \setminus \{k\}$ [29]. The following theorem expresses these marginal contributions in terms of conditional mutual information:

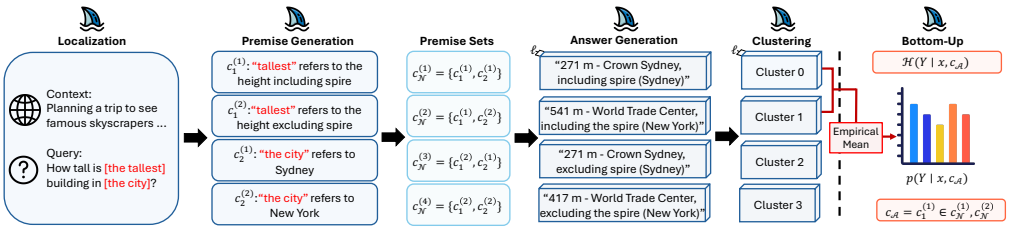


Figure 2: **An overview of the `ShaQ` pipeline.** Given an input query *"How tall is the tallest building in the city?"*, **(Step 1)** two ambiguous spans are localized, and then two premises are generated per span, providing four joint clarifications. For each clarification, **(Step 2)** the `Answerer` generates ℓ responses, which are then mapped into a shared semantic space by the `Clusterer`. **(Step 3)** Bottom-up marginalization is used to compute $\mathcal{H}(Y \mid x, c_{\mathcal{A}})$ for each premise (here, we set $c_{\mathcal{A}} = c_1^{(1)}$).

Theorem 4.1 (**Shapley Value of Ambiguous Span**). *For any coalition $\mathcal{A} \subseteq \mathcal{N} \setminus \{k\}$, the marginal contribution of span s_k satisfies*

$$v(\mathcal{A} \cup \{k\}) - v(\mathcal{A}) \;=\; \mathcal{I}(Y; C_k \mid x, C_{\mathcal{A}}). \tag{4}$$

Consequently, the Shapley value of span s_k can be written as

$$\phi_k \;=\; \sum_{\mathcal{A} \subseteq \mathcal{N} \setminus \{k\}} \frac{|\mathcal{A}|!(n - |\mathcal{A}| - 1)!}{n!} \mathcal{I}(Y; C_k \mid x, C_{\mathcal{A}}), \tag{5}$$

which characterizes ϕ_k as the expected information gain regarding Y from clarifying s_k, marginalized over all possible clarification contexts $C_{\mathcal{A}}$.

The derivation of (4) is provided in Appendix A. Notably, (5) reveals that the LOO attribution ϕ_k^{LOO} (Definition 3.2) is a specific boundary case of the Shapley value, where the context is restricted to the largest possible coalition $\mathcal{A} = \mathcal{N} \setminus \{k\}$. While LOO evaluates the contribution of a span only in the fully clarified setting, the Shapley formulation accounts for dependencies among spans by averaging their contributions across all clarification contexts. This leads to the following structural result:

Remark 4.1 (Efficiency)**.** The Shapley values in (5) partition the total input-induced uncertainty:

$$\sum_{k=1}^{n} \phi_k \;=\; v(\mathcal{N}) \;=\; \mathcal{I}(Y; C \mid x). \tag{6}$$

Remark 4.1 shows that our framework satisfies the efficiency axiom [28, 32], ensuring that the sum of *local* span-level attributions exactly recovers the *global* input-level aleatoric uncertainty. Together with Theorem 4.1, this establishes our approach as a *generalization* of prior input-level uncertainty estimation [23]. In particular, the global quantity $\mathcal{I}(Y; C \mid x)$ introduced in [23] is recovered via efficiency as $\sum_k \phi_k = v(\mathcal{N})$, while our Shapley formulation further provides a principled per-span decomposition that localizes the underlying sources of the total aleatoric uncertainty.

4.2 Practical Estimation of Uncertainty Attribution

Estimating the Shapley values defined in (5) reduces to computing the expected conditional entropy $\mathbb{E}_{c_{\mathcal{A}} \sim q(C_{\mathcal{A}} \mid x)}[\mathcal{H}(Y \mid x, c_{\mathcal{A}})]$ for all subsets $\mathcal{A} \subseteq \mathcal{N}$. Our framework employs four specialized LLMs, each assigned a distinct role to facilitate a modular estimation process as in Figure 2:

- `Localizer`: Extracts the set of ambiguous spans $\mathcal{S}$ within the input x.
- `Generator`: Samples logical premises (i.e., clarifications) for each identified ambiguous span.
- `Answerer`: Generates responses Y conditioned on x and joint clarifications for a coalition $\mathcal{A}$.
- `Clusterer`: Maps diverse responses into a unified semantic space for entropy calculation.

The estimation pipeline proceeds through the following three stages:

- **Stage 1: Span Localization and Premise Generation.** Given an input x, we identify a collection of ambiguous spans $\mathcal{S} = \{s_1, \ldots, s_n\}$ using the `Localizer`. This step reduces the combinatorial space of ambiguity by focusing on a small number of semantically coherent units that can have

multiple plausible interpretations. Then, for each identified span $s_k \in \mathcal{S}$, the `Generator` samples m conditioning premises $\{c_k^{(1)}, \ldots, c_k^{(m)}\}$, where each premise represents a distinct *logical clarification* of that span. We use the term *premise* to emphasize that these clarifications act as grounding statements: by specifying a plausible interpretation of an ambiguous span, a premise defines a concrete semantic context that can alter the meaning of the input.

- **Stage 2. Answer Sampling and Semantic Clustering.** For each $c_{\mathcal{N}}$, the `Answerer` generates ℓ responses conditioned on x and $c_{\mathcal{N}}$. This produces a global pool of answers across all possible joint clarifications. To maintain consistency across subset evaluations, we adopt a *global clustering* strategy, in which the `Clusterer` performs a single, comprehensive clustering over the entire answer pool. This defines a shared semantic output space $\mathcal{Y} = \{y_1, \ldots, y_K\}$, from which the empirical distribution $p(Y \mid x, c_{\mathcal{N}})$ is constructed. Using a common output space ensures that all entropy estimates are directly comparable across coalitions, avoiding the incomparability that would arise from independently constructed semantic clusters.
- **Stage 3. Hierarchical Conditional Entropy Computation.** Computing the Shapley value requires estimating $\mathbb{E}_{c_{\mathcal{A}} \sim q(C_{\mathcal{A}} \mid x)}[\mathcal{H}(Y \mid x, c_{\mathcal{A}})]$ for all coalitions $\mathcal{A} \subseteq \mathcal{N}$. A naive approach would estimate each coalition independently via separate LLM calls; however, such an approach generally fails to preserve the required marginalization consistency, $p(Y \mid x, c_{\mathcal{A}}) = \sum_{c_k} q(c_k \mid x)\, p(Y \mid x, c_{\mathcal{A}}, c_k)$ where $k \notin \mathcal{A}$. As a consequence, the empirical marginal contribution $\mathcal{I}(Y; C_k \mid x, C_{\mathcal{A}})$ can have negative values due to an artifact of inconsistent estimation, thereby undermining the reliability of the resulting uncertainty attributions. To resolve this issue, we introduce a *bottom-up marginalization* algorithm that treats unclarified spans in any coalition $\mathcal{A} \subset \mathcal{N}$ as marginalized variables. [2] The independence of our premise generation ensures that the premises of unclarified spans remain independent of those already specified, i.e., $q(c_{\setminus \mathcal{A}} \mid x, c_{\mathcal{A}}) = q(c_{\setminus \mathcal{A}} \mid x)$. This justifies deriving $p(Y \mid x, c_{\mathcal{A}})$ for any coalition by marginalizing over the remaining premises. Crucially, since the premises $c_{\setminus \mathcal{A}}$ are readily available from **Stage 1**, $p(Y \mid x, c_{\mathcal{A}})$ can be obtained via Monte Carlo approximation over the existing sampled premises, without additional sampling.

This bottom-up marginalization guarantees a fundamental structural property required for stable information-theoretic uncertainty quantification:

Property 4.1 (Hierarchical Monotonicity)**.** *For any subset $\mathcal{A} \subseteq \mathcal{N}$ and any span $k \notin \mathcal{A}$, the expected conditional entropy decreases monotonically as more spans are clarified:*

$$\mathbb{E}_{c_{\mathcal{A}}}[\mathcal{H}(Y \mid x, c_{\mathcal{A}})] \;\geq\; \mathbb{E}_{c_{\mathcal{A} \cup \{k\}}}[\mathcal{H}(Y \mid x, c_{\mathcal{A} \cup \{k\}})]. \tag{7}$$

Equivalently, every marginal contribution of the ambiguous span in (5) *is non-negative:*

$$\mathcal{I}(Y; C_k \mid x, C_{\mathcal{A}}) = \mathbb{E}_{c_{\mathcal{A}}}[\mathcal{H}(Y \mid x, c_{\mathcal{A}})] - \mathbb{E}_{c_{\mathcal{A} \cup \{k\}}}[\mathcal{H}(Y \mid x, c_{\mathcal{A} \cup \{k\}})] \geq 0. \tag{8}$$

The derivation is provided in Appendix A.

Algorithm 1 formalizes this procedure by computing $\mathbb{E}_{c_{\mathcal{A}}}[\mathcal{H}(Y \mid x, c_{\mathcal{A}})]$ for all coalitions through marginalization over unclarified spans. These quantities are then used to evaluate (5).

ALGORITHM 1: BOTTOM-UP MARGINALIZATION

Input: Bottom-level (finest) samples from $\{p(Y \mid x, c_{\mathcal{N}})\}$ for all joint clarifications.
For each subset $\mathcal{A} \subseteq \mathcal{N}$ and each premise combination $c_{\mathcal{A}}$**:**

1. Identify all bottom-level $c_{\mathcal{N}}$ consistent with $c_{\mathcal{A}}$, i.e., $c_{\mathcal{N}} = c_{\mathcal{A}} \cup c_{\setminus \mathcal{A}}$ for some $c_{\setminus \mathcal{A}}$.
2. Compute $p(Y \mid x, c_{\mathcal{A}}) = \sum_{c_{\setminus \mathcal{A}}} q(c_{\setminus \mathcal{A}} \mid x)\, p(Y \mid x, c_{\mathcal{A}} \cup c_{\setminus \mathcal{A}})$.
3. Compute $\mathcal{H}(Y \mid x, c_{\mathcal{A}}) = -\sum_y p(y \mid x, c_{\mathcal{A}}) \log p(y \mid x, c_{\mathcal{A}})$.

For each subset $\mathcal{A} \subseteq \mathcal{N}$**:**

4. Compute $\mathbb{E}_{c_{\mathcal{A}}}[\mathcal{H}(Y \mid x, c_{\mathcal{A}})] = \sum_{c_{\mathcal{A}}} q(c_{\mathcal{A}} \mid x)\, \mathcal{H}(Y \mid x, c_{\mathcal{A}})$.

Output: Cached values $\mathbb{E}_{c_{\mathcal{A}}}[\mathcal{H}(Y \mid x, c_{\mathcal{A}})]$ for Shapley computation via (5).

[2] This approach is structurally analogous to the interventional (marginal) Shapley value in feature attribution [29, 30, 33], where "absent" features are integrated out using their marginal distributions. See Appendix E for a detailed discussion.

Table 1: Uncertainty-based ambiguity detection.

Method	AmbigQA (GPT-4) F1 Score	AUROC	AUPRC	AmbiEnt (GPT-5.4-mini) F1 Score	AUROC	AUPRC
Other Uncertainty Methods						
Semantic Entropy	44.58	57.16	57.00	42.20	57.56	54.36
Sample Diversity	55.24	60.86	59.35	48.33	59.23	57.48
Sample Repetition	56.98	60.90	59.24	43.69	56.67	57.78
Self-Consistency	53.40	59.33	57.59	47.93	59.28	59.46
Aleatoric Methods						
Ask4Conf-D	56.36	52.44	53.97	65.43	50.82	52.54
Deep Ensembles	59.70	60.29	60.31	56.37	58.96	61.49
ICE	65.54	60.66	59.98	57.33	55.40	53.21
LOO	67.77	50.43	54.35	66.07	49.04	57.00
LOO (Max Span)	67.77	52.36	55.80	66.07	52.66	59.69
ShaQ **(Ours, Max Span)**	**70.38**	64.20	59.95	**74.60**	**77.80**	75.19
ShaQ **(Ours)**	**70.38**	**65.77**	**61.52**	73.84	77.68	**75.71**

Table 2: `Answerer & Clusterer` backbone with `GPT-5.4-mini` / `Gemini models`

Method	AmbigQA F1 Score	AUROC	AUPRC
`GPT-5.4-mini`			
Deep Ensembles	63.41	62.62	57.29
ICE	66.67	**64.61**	**63.11**
ShaQ **(Ours, Max Span)**	**69.53**	62.37	58.63
ShaQ **(Ours)**	**69.53**	63.60	62.08
`Gemini-3.1-flash-lite-preview`			
Deep Ensembles	44.30	55.92	54.26
ICE	63.21	63.41	**63.53**
ShaQ **(Ours, Max Span)**	**66.67**	62.97	58.95
ShaQ **(Ours)**	**66.67**	**64.22**	63.17
`Gemini-2.5-flash`			
Deep Ensembles	56.28	55.42	53.42
ICE	65.82	61.71	61.15
ShaQ **(Ours, Max Span)**	**68.75**	63.50	60.06
ShaQ **(Ours)**	**68.75**	**64.84**	**62.74**

Table 3: `Answerer & Clusterer` backbone with `Gemini models`

Method	AmbiEnt F1 Score	AUROC	AUPRC
`Gemini-3.1-flash-lite-preview`			
Deep Ensembles	27.52	46.74	47.76
ICE	50.70	50.71	49.45
Ask4Conf-D	49.12	45.51	49.57
LOO	**66.07**	53.23	55.63
LOO (Max Span)	65.47	52.94	56.20
ShaQ **(Ours, Max Span)**	65.59	**72.39**	**70.65**
ShaQ **(Ours)**	65.59	71.96	69.32
`Gemini-2.5-flash`			
Deep Ensembles	41.66	54.28	56.19
ICE	51.38	51.31	49.97
Ask4Conf-D	60.20	42.86	46.83
LOO	66.07	38.64	44.39
LOO (Max Span)	65.47	41.48	46.22
ShaQ **(Ours, Max Span)**	**71.42**	76.40	74.52
ShaQ **(Ours)**	**71.42**	**76.47**	**75.78**

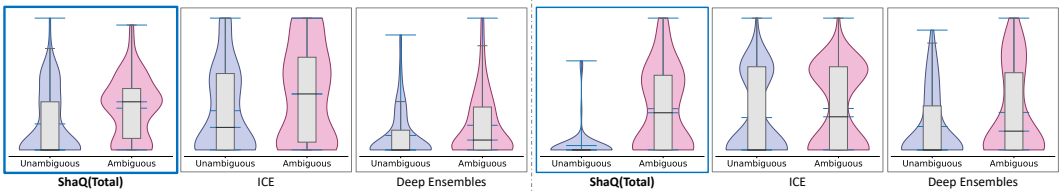


Figure 3: **Aleatoric uncertainty distributions on AmbigQA (Left) and AmbiEnt (Right).**

5 Experiments

5.1 Experiment setup

Baselines. For baseline methods that explicitly estimate aleatoric uncertainty, we use Input Clarification Ensemble (ICE) [23], Deep Ensembles [11], and Ask4Conf-D [23, 27]. We also evaluate answer-confidence-based estimators such as Semantic Entropy [17], Sample Diversity [34], Sample Repetition [35], and Self-Consistency [35], following prior work [23].

Implementation Details. For AmbigQA [24], we use `GPT-4` as the primary backbone for all baselines and modules of `ShaQ`, following the clarifier backbone used in ICE [23]. For AmbiEnt [25], we use `GPT-5.4-mini`, as its natural language inference (NLI) formulation requires stronger reasoning capabilities [36]. To examine whether the observed gains are robust to the choice of LLM, we additionally evaluate `ShaQ` and the aleatoric uncertainty baselines using `Gemini`-based models for ambiguity detection. Following prior work [23], *Deep Ensembles* are implemented by varying the in-context examples across ensemble members, since parameter-level ensembling is infeasible for frozen LLMs. We use both ICE and *Deep Ensembles* in their aleatoric uncertainty estimation modes and apply semantic clustering to both methods to ensure a fair comparison. Full configurations and prompts are provided in Appendix B.

5.2 Ambiguity Detection with Aleatoric Uncertainty

To assess whether the aleatoric uncertainty estimated by ShaQ aligns with true input ambiguity, we follow the ambiguity-detection protocol of prior work [23] and treat the aleatoric uncertainty produced by each method as signals for ambiguity detection. An effective aleatoric uncertainty measure should assign higher uncertainty to ambiguous questions and lower uncertainty to unambiguous ones. To this end, we evaluate two ambiguity benchmarks, AmbigQA [24] and AmbiEnt [25], measuring how well each method distinguishes ambiguous from unambiguous questions.

Since ShaQ produces span-level uncertainty attributions (ϕ_k), we use (6), i.e., $\sum_{k\in[n]} \phi_k$, as the total aleatoric uncertainty for ambiguity detection, and denote it by ShaQ. Additionally, we use the maximum span-level uncertainty attribution, $\max_{k\in[n]} \phi_k$, denoted by ShaQ (Max), as a direct variant. To examine the role of interaction modeling, we include LOO-based variants (2) as direct ablations, assessing whether evaluating a span's contribution under a fully-clarified context is sufficient in such settings: LOO (i.e., $\sum_{k=1}^{n} \phi_k^{\mathrm{LOO}}$) and LOO-Max (i.e., $\max_{k\in[n]} \phi_k^{\mathrm{LOO}}$).

Ambiguity Detection. We evaluate ambiguity detection performance using the best F1 score, AUROC, and AUPRC. As shown in Tables 1, 2 and 3, ShaQ and ShaQ (Max) consistently achieve strong performance across different backbones and evaluation metrics, demonstrating the robustness of our uncertainty localization framework. In particular, on AmbiEnt, where ambiguity often involves more complex and compositional phenomena, ShaQ and ShaQ (Max) outperform existing baselines by a substantial margin in AUROC and AUPRC. This indicates that ShaQ is especially effective at quantifying the degree of ambiguity, going *beyond simply detecting its presence*.

Furthermore, Figure 3 shows the distributions of aleatoric uncertainty for ambiguous and unambiguous inputs. Compared to the baselines, ShaQ exhibits the clearest separation between ambiguous and unambiguous inputs, assigning consistently higher uncertainty to ambiguous questions. This separation supports the effectiveness of ShaQ as a fine-grained ambiguity quantification method.

Overall, these results show that the uncertainty estimates produced by ShaQ closely align with human-annotated ambiguity, while the gains over the aggregate input-level baseline (i.e., ICE) highlight the benefit of explicitly localizing ambiguous spans.

Multi Span Ambiguity Detection. We further analyze performance on AmbiEnt inputs where our localization stage identifies two or more candidate ambiguous spans (i.e., $|\mathcal{S}| \geq 2$). This case provides a more rigorous evaluation of localization, as multiple sources of ambiguity may coexist and interact within a single input.

Table 4: Multi-span results on AmbiEnt

Method	**F1 Score**	**AUROC**	**AUPRC**
Ask4Conf-D	83.95	40.40	78.13
Deep Ensembles	58.62	45.95	75.25
ICE	59.64	53.40	77.54
LOO	85.36	49.11	79.59
LOO(Max Span)	85.36	51.64	80.26
ShaQ **(Ours, Max)**	**86.95**	**81.31**	**92.39**
ShaQ **(Ours)**	86.56	80.05	91.27

In Table 4, ShaQ and its variants achieve the best performance, outperforming all baselines and ablations, including LOO and LOO-Max. These results support the central motivation of ShaQ: in the presence of interacting ambiguity sources, aleatoric uncertainty is more accurately attributed through a dependency-aware decomposition, as opposed to a static, fully-clarified baseline. Appendix C.3 details further AmbigQA multi-span results and highlights ShaQ's inherent robustness to imperfect localization. Specifically, our formulation naturally neutralizes extraction errors by assigning negligible Shapley values to falsely identified spans.

5.3 Uncertainty-Guided Clarification

Clarification Simulation Setup. We evaluate whether each method provides *actionable uncertainty feedback* in an uncertainty-guided clarification setting. As illustrated in Figure 4, we *simulate a user-facing workflow* in which a method provides uncertainty feedback for an ambiguous input query, and an LLM user revises the original input x into a clarified input x'. We apply this procedure to ambiguous questions from the AmbigQA benchmark [24]. While baselines provide only input-level uncertainty feedback in the form of a single scalar score, ShaQ provides localized span-level uncertainty attributions, allowing us to test whether localization supports more targeted clarification while preserving the original intent. To avoid bias toward examples favored by a single estimator, we evaluate on a superset formed by the union of the top 20% examples ranked by ShaQ's maximum

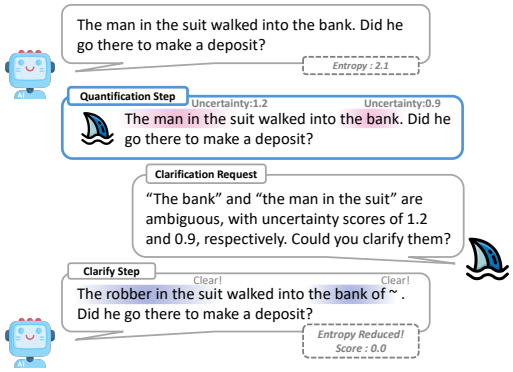


Figure 4: **Uncertainty-guided clarification.**

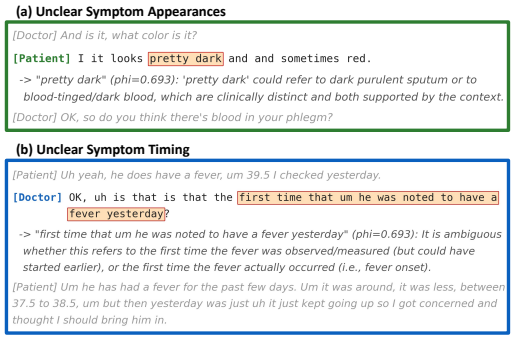


Figure 5: **Qualitative results on MediTOD.**

span-level aleatoric uncertainty and those ranked by the maximum aleatoric uncertainty among the baselines.

This captures both localized ambiguity and broadly high-uncertainty cases. We use the same backbone model as in Section 5.2, namely `GPT-4`. Experimental details are provided in Appendix B, and prompts are shown in Figures 18 and 17.

Evaluation Metrics. To assess the efficiency of `ShaQ`, we employ two complementary metrics that capture the trade-off between *user effort and information gain*: entropy reduction, $\Delta\mathcal{H} = \mathcal{H}(Y|x) - \mathcal{H}(Y|x')$ and word-level edit distance (*Edit*), computed as the Levenshtein distance over word tokens. These metrics should be interpreted jointly: high $\Delta\mathcal{H}$ indicates resolved model-side uncertainty, while low *Edit* indicates minimal modification. The desired outcome is high entropy reduction with low editing effort, showing that uncertainty information enables targeted, "surgical" refinement of the ambiguity sources.

Overall Results. Table 5 reports the uncertainty-guided clarification results on the high-uncertainty superset, where the *Clarified* column denotes the predictive entropy of the revised query, $\mathcal{H}(Y \mid x')$. We find that `ShaQ` achieves the highest entropy reduction while requiring the lowest word-level edits, indicating that span-level uncertainty provides actionable guidance for resolving ambiguity. We also observe that baseline-guided revisions often require larger modifications and may even increase predictive entropy (-0.0633), suggesting that input-level uncertainty alone can fail to identify which part of the query should be clarified. In Appendix C.6, we further analyze split evaluation sets under other selection criteria and observe the same trend.

Table 5: High Aleatoric Uncertainty Superset

Method	Clarified	$\Delta H(\uparrow)$	Edit ($\downarrow$)
Semantic Entropy	0.3423	0.0330	6.4545
ICE	0.4387	-0.0633	6.7878
Deep Ensembles	0.3831	-0.0076	7.4545
`ShaQ` **(Ours)**	**0.3365**	**0.0389**	**5.5151**

5.4 Analyses on Medical Dialogues

Qualitative Analysis. To demonstrate the applicability of `ShaQ` to complex human-to-human interactions, we evaluate it on MediTOD [26], a dataset of simulated doctor-patient dialogues. Figure 5 illustrates representative examples of clinically relevant ambiguities. In panel (a), `ShaQ` localizes the ambiguous expression *"pretty dark"*, which aligns with the physician's subsequent inquiry about blood in the phlegm. In panel (b), `ShaQ` identifies ambiguity in the reported timing of fever onset, which is subsequently clarified by the patient. Additional examples are provided in Appendix C.7.

LLM-Guided Quantitative Analysis. To quantitatively assess localization utility, we construct a balanced evaluation subset from MediTOD using `GPT-5.6 Luna Pro` as an annotator. Given up to eight preceding dialogue turns, the current patient utterance, and the immediately following turn, the annotator identifies cases where an unclear symptom appearance or quality is directly addressed by the physician's follow-up question. We retain 55 positive cases and randomly sample 55 negative cases, yielding 110 utterances. Following Section 5.2, we evaluate `ShaQ` and ICE using

Table 6: Quantitative results on MediTOD.

Method	**F1 Score**	**AUROC**	**AUPRC**
ICE	63.57	59.44	57.90
`ShaQ` **(Ours, Total)**	79.21	82.61	82.32
`ShaQ` **(Ours, Max)**	**80.00**	**83.12**	**82.87**
Follow-up Match		**72.09%**	

GPT-5.4-mini. Both methods receive the preceding dialogue history and current utterance as their initial inputs, without access to the actual follow-up turn.

As shown in Table 6, ShaQ outperforms ICE across all ambiguity detection metrics, achieving an AUROC of 82.61 compared to 59.44. To further assess localization utility, we consider gold-positive utterances for which ShaQ identifies at least one ambiguous span. We select the span with the highest Shapley attribution and employ GPT-5.6 Luna Pro as an LLM judge to assess its semantic alignment with the physician's actual follow-up question. The localized spans achieve a matching ratio of 72.09%, suggesting that ShaQ can identify ambiguity sources relevant to subsequent physician clarification in simulated clinical dialogues.

6 Conclusion

In this paper, we addressed a limitation of existing UQ methods in LLM: the inability to identify the specific input sources driving aleatoric uncertainty. To bridge this gap, we proposed ShaQ, a theoretically-grounded framework designed to localize input-induced uncertainty at the span level. By formulating ambiguity resolution as a cooperative game, ShaQ models the semantic dependencies between ambiguous spans. Empirically, ShaQ significantly improves ambiguity detection on benchmarks over monolithic baselines. Furthermore, it enables targeted input clarifications with minimal user effort and demonstrates potential for enhancing LLM reliability in high-stakes domains.

Acknowledgments and Disclosure of Funding

We thank the NeurIPS reviewers for their valuable comments and suggestions. This work is supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00358602) and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT), Artificial Intelligence Graduate School Program (No. RS-2019-II190079, Korea University), the Artificial Intelligence Star Fellowship Support Program to nurture the best talents (No. RS-2025-02304828), and the AI Research Hub Project (No. RS-2024-00457882). Also, this research was supported by the AI Seoul Tech Research Support Program of the Seoul Future Foundation.

References

[1] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] Siddartha Devic, Tejas Srinivasan, Jesse Thomason, Willie Neiswanger, and Vatsal Sharan. From calibration to collaboration: Llm uncertainty quantification should be more human-centered. *arXiv preprint arXiv:2506.07461*, 2025.

[4] Laura R Marusich, Jonathan Z Bakdash, Yan Zhou, and Murat Kantarcioglu. Using ai uncertainty quantification to improve human decision-making. *arXiv preprint arXiv:2309.10852*, 2023.

[5] Christine Herlihy, Jennifer Neville, Tobias Schnabel, and Adith Swaminathan. On overcoming miscalibrated conversational priors in llm-based chatbots. *arXiv preprint arXiv:2406.01633*, 2024.

[6] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Clam: Selective clarification for ambiguous questions with generative language models. *arXiv preprint arXiv:2212.07769*, 2022.

[7] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

[8] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[9] 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.

[10] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems*, 31, 2018.

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

[12] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. arxiv 2019. *arXiv preprint arXiv:1912.02757*, 2019.

[13] Yarin Gal et al. *Uncertainty in deep learning*, volume 1. University of Cambridge Cambridge, 2017.

[14] Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*, 2020.

[15] Tunyu Zhang, Haizhou Shi, Yibin Wang, Hengyi Wang, Xiaoxiao He, Zhuowei Li, Haoxian Chen, Ligong Han, Kai Xu, Huan Zhang, et al. Tokur: Token-level uncertainty estimation for large language model reasoning. *arXiv preprint arXiv:2505.11737*, 2025.

[16] Xiang Gao, Jiaxin Zhang, Lalla Mouatadid, and Kamalika Das. Spuq: Perturbation-based uncertainty quantification for large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2336–2346, 2024.

[17] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*, 2023.

[18] 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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[19] Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*, 2023.

[20] Potsawee Manakul, Adian Liusie, and Mark Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 9004–9017, 2023.

[21] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

[22] Xiaoou Liu, Tiejin Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. Uncertainty quantification and confidence calibration in large language models: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6107–6117, 2025.

[23] Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. Decomposing uncertainty for large language models through input clarification ensembling. *International Conference on Machine Learning*, 2024.

[24] 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)*, pages 5783–5797, 2020.

[25] Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. We're afraid language models aren't modeling ambiguity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 790–807, Singapore, December 2023. Association for Computational Linguistics.

[26] Vishal Vivek Saley, Goonjan Saha, Rocktim Jyoti Das, Dinesh Raghu, et al. Meditod: An english dialogue dataset for medical history taking with comprehensive annotations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16843–16877, 2024.

[27] Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D 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*, pages 5433–5442, 2023.

[28] Lloyd S Shapley et al. A value for n-person games. 1953.

[29] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[30] Hugh Chen, Ian C Covert, Scott M Lundberg, and Su-In Lee. Algorithms to estimate shapley value feature attributions. *Nature Machine Intelligence*, 5(6):590–601, 2023.

[31] David Watson, Joshua O'Hara, Niek Tax, Richard Mudd, and Ido Guy. Explaining predictive uncertainty with information theoretic shapley values. *Advances in Neural Information Processing Systems*, 36:7330–7350, 2023.

[32] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[33] Hugh Chen, Joseph D Janizek, Scott Lundberg, and Su-In Lee. True to the model or true to the data? *arXiv preprint arXiv:2006.16234*, 2020.

[34] Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. Prompting gpt-3 to be reliable. *arXiv preprint arXiv:2210.09150*, 2022.

[35] Jeremy Cole, Michael Zhang, Dan Gillick, Julian Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. Selectively answering ambiguous questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 530–543, 2023.

[36] Terufumi Morishita, Gaku Morio, Atsuki Yamaguchi, and Yasuhiro Sogawa. Enhancing reasoning capabilities of llms via principled synthetic logic corpus. *Advances in Neural Information Processing Systems*, 37:73572–73604, 2024.

[37] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626, 2023.

Appendix

Table of Contents

A Mathematical Derivations **14**

B Configurations and Prompts **15**

B.1 Dataset Description: AmbigQA 15

B.2 Dataset Description: AmbiEnt 16

B.3 Configurations 16

B.4 Prompts 17

B.5 Prompts and Configurations for Uncertainty-guided Clarification 17

C Additional Results **18**

C.1 Additional Results on Ambiguity Detection 18

C.2 Additional Ablation Results on Ambiguity Detection 20

C.3 Additional Results on Multi Span Ambiguity Detection 20

C.4 Stability Analysis 24

C.5 Distribution of Ambiguous Spans 24

C.6 Additional Results on Uncertainty-guided Clarification 25

C.7 Additional Results on Human-to-Human Medical Multi-turn Dialogue 26

D Computational Analysis **28**

E Limitations and Future Directions **29**

F Practical Considerations **29**

G Broader Impacts **30**

H Asset Licenses and Terms of Use **30**

A Mathematical Derivations

We restate Definition 3.2, Theorem 4.1 and Property 4.1 for convenience.

Definition A.1 (LOO Span Attribution). The LOO attribution of span s_k is defined as the conditional mutual information

$$\phi_k^{\mathrm{LOO}} \triangleq \mathcal{I}(Y; C_k \mid x, C_{\setminus k}) = \mathcal{H}(Y \mid x, C_{\setminus k}) - \mathcal{H}(Y \mid x, C), \tag{9}$$

which quantifies the information about Y contributed by clarifying s_k after all remaining spans have been resolved.

Proof. By definition of conditional mutual information,

$$\mathcal{I}(Y; C_k \mid x, C_{\setminus k}) = \mathcal{H}(Y \mid x, C_{\setminus k}) - \mathcal{H}(Y \mid x, C_k, C_{\setminus k}). \tag{10}$$

Since $(C_k, C_{\setminus k})$ jointly constitute C, the second term satisfies $\mathcal{H}(Y \mid x, C_k, C_{\setminus k}) = \mathcal{H}(Y \mid x, C)$, which gives

$$\mathcal{I}(Y; C_k \mid x, C_{\setminus k}) = \mathcal{H}(Y \mid x, C_{\setminus k}) - \mathcal{H}(Y \mid x, C). \tag{11}$$

□

Theorem A.1 (Shapley Value of Ambiguous Span). *For any coalition $\mathcal{A} \subseteq \mathcal{N} \setminus \{k\}$, the marginal contribution of span s_k satisfies*

$$v(\mathcal{A} \cup \{k\}) - v(\mathcal{A}) \;=\; \mathcal{I}(Y; C_k \mid x, C_{\mathcal{A}}). \tag{12}$$

Consequently, the Shapley value of span s_k can be written as

$$\phi_k \;=\; \sum_{\mathcal{A} \subseteq \mathcal{N} \setminus \{k\}} \frac{|\mathcal{A}|!(n - |\mathcal{A}| - 1)!}{n!}\, \mathcal{I}(Y; C_k \mid x, C_{\mathcal{A}}). \tag{13}$$

Proof. By Definition 4.1, the marginal contribution expands as

$$\begin{aligned} v(\mathcal{A} \cup \{k\}) - v(\mathcal{A}) &= \mathcal{I}(Y; C_{\mathcal{A}\cup\{k\}} \mid x) - \mathcal{I}(Y; C_{\mathcal{A}} \mid x) && (14)\\ &= \Big[\mathcal{H}(Y \mid x) - \mathbb{E}_{c_{\mathcal{A}\cup\{k\}}}\big[\mathcal{H}(Y \mid x, c_{\mathcal{A}\cup\{k\}})\big]\Big] && (15)\\ &\quad - \Big[\mathcal{H}(Y \mid x) - \mathbb{E}_{c_{\mathcal{A}}}\big[\mathcal{H}(Y \mid x, c_{\mathcal{A}})\big]\Big] && (16)\\ &= \mathbb{E}_{c_{\mathcal{A}}}\big[\mathcal{H}(Y \mid x, c_{\mathcal{A}})\big] - \mathbb{E}_{c_{\mathcal{A}\cup\{k\}}}\big[\mathcal{H}(Y \mid x, c_{\mathcal{A}\cup\{k\}})\big]. && (17) \end{aligned}$$

By the definition of conditional mutual information and conditional entropy,

$$\begin{aligned} \mathcal{I}(Y; C_k \mid x, C_{\mathcal{A}}) &= \mathcal{H}(Y \mid x, C_{\mathcal{A}}) - \mathcal{H}(Y \mid x, C_{\mathcal{A}}, C_k) && (18)\\ &= \mathbb{E}_{c_{\mathcal{A}}}\big[\mathcal{H}(Y \mid x, c_{\mathcal{A}})\big] - \mathbb{E}_{c_{\mathcal{A}\cup\{k\}}}\big[\mathcal{H}(Y \mid x, c_{\mathcal{A}\cup\{k\}})\big], && (19) \end{aligned}$$

where the second equality uses $(C_{\mathcal{A}}, C_k) = C_{\mathcal{A}\cup\{k\}}$. The two expressions are identical, which completes the proof. □

Property A.1 (Hierarchical Monotonicity). *For any subset $\mathcal{A} \subseteq \mathcal{N}$ and any span $k \notin \mathcal{A}$, the expected conditional entropy decreases monotonically as more spans are clarified:*

$$\mathbb{E}_{c_{\mathcal{A}}}[\mathcal{H}(Y \mid x, c_{\mathcal{A}})] \;\geq\; \mathbb{E}_{c_{\mathcal{A}\cup\{k\}}}[\mathcal{H}(Y \mid x, c_{\mathcal{A}\cup\{k\}})]. \tag{20}$$

Equivalently, every marginal contribution of the ambiguous span in Eq. (5) *is non-negative:*

$$\mathcal{I}(Y; C_k \mid x, C_{\mathcal{A}}) = \mathbb{E}_{c_{\mathcal{A}}}[\mathcal{H}(Y \mid x, c_{\mathcal{A}})] - \mathbb{E}_{c_{\mathcal{A}\cup\{k\}}}[\mathcal{H}(Y \mid x, c_{\mathcal{A}\cup\{k\}})] \geq 0. \tag{21}$$

Proof. By definition, the marginal contribution of span s_k to coalition $\mathcal{A}$ is given by the difference in expected conditional entropy:

$$\mathcal{I}(Y; C_k \mid x, C_{\mathcal{A}}) = \mathbb{E}_{c_{\mathcal{A}}}[\mathcal{H}(Y \mid x, c_{\mathcal{A}})] - \mathbb{E}_{c_{\mathcal{A}\cup\{k\}}}[\mathcal{H}(Y \mid x, c_{\mathcal{A}\cup\{k\}})]. \tag{22}$$

In our framework, the probability distribution for any coalition $\mathcal{A}$ is computed by marginalizing the completely clarified bottom-level states:

$$p(Y \mid x, c_{\mathcal{A}}) = \frac{1}{m^{|\mathcal{N}\setminus\mathcal{A}|}} \sum_{c_{\mathcal{N}\setminus\mathcal{A}}} p(Y \mid x, c_{\mathcal{A}} \cup c_{\mathcal{N}\setminus\mathcal{A}}). \tag{23}$$

To relate $p(Y \mid x, c_{\mathcal{A}})$ to $p(Y \mid x, c_{\mathcal{A}\cup\{k\}})$, we partition the set of unclarified spans $\mathcal{N} \setminus \mathcal{A}$ into the single target span $\{k\}$ and the rest of the unclarified spans, which we denote as $\mathcal{N} \setminus \mathcal{A}'$ where $\mathcal{A}' = \mathcal{A} \cup \{k\}$. Correspondingly, the combinations of premises can be decomposed as $c_{\mathcal{N}\setminus\mathcal{A}} = c_k \cup c_{\mathcal{N}\setminus\mathcal{A}'}$, and the total number of combinations factors as $m^{|\mathcal{N}\setminus\mathcal{A}|} = m \cdot m^{|\mathcal{N}\setminus\mathcal{A}'|}$. Substituting this decomposition into Equation (23) allows us to rewrite the single summation as a nested summation:

$$\begin{aligned} p(Y \mid x, c_{\mathcal{A}}) &= \frac{1}{m \cdot m^{|\mathcal{N}\setminus\mathcal{A}'|}} \sum_{c_k} \sum_{c_{\mathcal{N}\setminus\mathcal{A}'}} p(Y \mid x, c_{\mathcal{A}} \cup c_k \cup c_{\mathcal{N}\setminus\mathcal{A}'}) \\ &= \frac{1}{m} \sum_{c_k} \underbrace{\left(\frac{1}{m^{|\mathcal{N}\setminus\mathcal{A}'|}} \sum_{c_{\mathcal{N}\setminus\mathcal{A}'}} p(Y \mid x, c_{\mathcal{A}'} \cup c_{\mathcal{N}\setminus\mathcal{A}'}) \right)}_{\text{Definition of } p(Y|x,c_{\mathcal{A}'})}. \end{aligned} \tag{24}$$

Observe that the term inside the parentheses is exactly the bottom-up definition for the subset $\mathcal{A}' = \mathcal{A} \cup \{k\}$. Therefore, we can recursively express $p(Y \mid x, c_{\mathcal{A}})$ as:

$$p(Y \mid x, c_{\mathcal{A}}) = \frac{1}{m} \sum_{c_k} p(Y \mid x, c_{\mathcal{A}\cup\{k\}}). \tag{25}$$

Equation (25) mathematically establishes that $p(Y \mid x, c_{\mathcal{A}})$ is a strict convex combination of its immediate superset distributions $\{p(Y \mid x, c_{\mathcal{A}\cup\{k\}})\}_{c_k}$. Since the Shannon entropy $\mathcal{H}(\cdot)$ is a strictly concave function, applying Jensen's Inequality to this convex combination yields:

$$\mathcal{H}(Y \mid x, c_{\mathcal{A}}) \geq \frac{1}{m} \sum_{c_k} \mathcal{H}(Y \mid x, c_{\mathcal{A}\cup\{k\}}). \tag{26}$$

Now, we take the expectation over $c_{\mathcal{A}} \sim q(c_{\mathcal{A}} \mid x)$ on both sides. Since our premises are sampled uniformly with probability $\frac{1}{m^{|\mathcal{A}|}}$, this gives:

$$\begin{aligned} \mathbb{E}_{c_{\mathcal{A}}}[\mathcal{H}(Y \mid x, c_{\mathcal{A}})] &\geq \frac{1}{m^{|\mathcal{A}|}} \sum_{c_{\mathcal{A}}} \left(\frac{1}{m} \sum_{c_k} \mathcal{H}(Y \mid x, c_{\mathcal{A}\cup\{k\}}) \right) \\ &= \frac{1}{m^{|\mathcal{A}|+1}} \sum_{c_{\mathcal{A}} \cup c_k} \mathcal{H}(Y \mid x, c_{\mathcal{A}\cup\{k\}}) \\ &= \mathbb{E}_{c_{\mathcal{A}\cup\{k\}}}[\mathcal{H}(Y \mid x, c_{\mathcal{A}\cup\{k\}})]. \end{aligned} \tag{27}$$

Finally, substituting Equation (27) back into Equation (22), we conclude:

$$\mathcal{I}(Y; C_k \mid x, C_{\mathcal{A}}) \geq 0. \tag{28}$$

This confirms that calculating subset distributions purely via bottom-up marginalization mathematically guarantees non-negative marginal contributions across all possible coalitions. □

B Configurations and prompts

B.1 Dataset Description: AmbigQA

AmbigQA [24] is an open-domain question answering dataset designed to study ambiguity in naturally occurring information-seeking questions. It is constructed from NQ-Open and targets questions that may admit multiple plausible interpretations and therefore multiple valid answers. For each ambiguous question, AmbigQA provides a set of disambiguated question–answer pairs, where each rewritten question corresponds to one specific interpretation of the original ambiguous query. For unambiguous questions, the dataset provides a single answer annotation.

This annotation structure makes AmbigQA suitable for evaluating input ambiguity, since ambiguity is defined at the level of the user question rather than at the level of model-generated outputs. In our experiments, we use the presence of multiple plausible interpretations as the ambiguity label and evaluate whether uncertainty estimates assign higher scores to ambiguous questions than to unambiguous ones.

However, AmbigQA does not provide gold annotations for ambiguous spans or their corresponding premises. Therefore, it does not support direct supervised evaluation of span-level localization or premise generation. We instead use AmbigQA for input-level ambiguity detection, while span-level uncertainty attribution is evaluated through the resulting aggregate ambiguity score.

B.2 Dataset Description: AmbiEnt

AmbiEnt [25] is a natural language inference (NLI) benchmark designed to study ambiguity through its effect on entailment relations. Each example consists of a premise–hypothesis pair, where the task is to determine whether the premise entails, contradicts, or is neutral with respect to the hypothesis. Unlike standard NLI datasets that assume a single gold label for each example, AmbiEnt allows an example to be associated with multiple plausible NLI labels when different interpretations of the premise and/or hypothesis lead to different entailment relations.

The dataset contains linguist-annotated examples covering diverse forms of ambiguity, including lexical, syntactic, and pragmatic ambiguity. For ambiguous examples, AmbiEnt provides annotations that characterize the possible interpretations of the input and their corresponding entailment labels. This structure makes the dataset suitable for evaluating whether a model can recognize when an input admits multiple valid readings, rather than forcing all examples into a single deterministic NLI label.

In our setting, AmbiEnt is useful because ambiguity is defined over the input pair rather than over model-generated outputs. We therefore use AmbiEnt to evaluate whether aleatoric uncertainty estimates assign higher uncertainty to premise–hypothesis pairs with multiple plausible interpretations than to those with a single interpretation.

However, AmbiEnt is not designed as a supervised benchmark for span-level uncertainty attribution. Although the dataset provides ambiguity-sensitive NLI annotations, it does not directly provide gold uncertainty scores for individual spans. Therefore, we use AmbiEnt for input-level ambiguity detection, while the span-level attributions produced by our method are assessed through their aggregate uncertainty signal.

B.3 Configurations

Following the evaluation dataset sampling protocol of ICE [23], we conduct both ambiguity detection and uncertainty-guided clarification experiments on AmbigQA. For AmbiEnt, we construct a balanced evaluation set of 150 examples by matching the number of unambiguous and ambiguous instances. The ambiguous set includes premise-only, hypothesis-only, and premise–hypothesis ambiguities in equal proportions. For Self-Consistency, Sample Repetition, and Sample Diversity, we follow the same implementation protocol as [23] and use the configurations provided in [23]. For Self-Consistency, we use temperature 0.7 and sample 10 answers from the model, and use 1 minus the resulting confidence score as the ambiguity score. For Sample Diversity, we use temperature 0.5 and sample 10 answers, then compute the number of unique answers as the ambiguity score. For Sample Repetition, following the official implementation, we first generate an initial answer using greedy decoding and then resample 10 answers with temperature 0.5 to estimate how frequently the initial answer is reproduced; lower repetition corresponds to higher ambiguity.

For both `ShaQ` and ICE [23], we use the same sampling configuration to ensure a fair comparison. The number of generated premises or clarifications is set to 3. For each premise or clarification condition, the `Answerer` samples 5 answers. The `Answerer` temperature is set to 0.7, while the `Clarifier` in ICE and the Premise `Generator` in `ShaQ` use temperature 0.9. Following [23], we apply LLM-based semantic clustering to group semantically equivalent outputs for ICE and Deep Ensembles, using the prompts provided in [23]. Following [23], we also apply the same unknown-output handling procedure to `ShaQ`, ICE, and Deep Ensembles before computing entropy, ensuring that all entropy estimates are based on a consistent output-processing protocol.

B.4 Prompts

We list the AmbigQA prompts used in our experiments as follows. The prompts follow a unified format consisting of a role instruction, task description, explicit generation or decision rules, static in-context demonstrations, and instance-specific input fields. We use the same decomposition across tasks: the `Localizer` identifies ambiguous spans, the premise `Generator` produces clarification assumptions for each span, and the `Answerer` predicts the final task output under a given set of assumptions.

For AmbigQA, we use the `Answerer` prompt from ICE [23] and keep it identical across all methods to ensure a fair comparison. The `Answerer` also uses the same 6-shot static demonstrations from the official ICE code for all methods. For the ICE clarification model, we use the same 3-shot static demonstrations as those used for the premise `Generator` in `ShaQ`. For Ask4Conf-D and ICE, we adopt the corresponding prompts from ICE [23].

For `ShaQ`, AmbigQA provides gold final answers but does not include gold annotations for the intermediate outputs required by our decomposition, namely oracle spans for the `Localizer` and oracle premises for the premise `Generator`. Thus, direct comparison against dataset-provided gold labels is not applicable for these intermediate components. We therefore manually annotate a small set of instances to construct static in-context demonstrations for `ShaQ`, following predefined annotation guidelines consistent with our task decomposition. These manually annotated instances are used solely for prompting and are excluded from all evaluation data.

For AmbiEnt, we use the same prompt interface and decomposition as AmbigQA. Since the full NLI prompts differ mainly in task-specific rule blocks and static demonstrations, we do not duplicate the lengthy prompt text here. The changes are limited to: (i) replacing answer-changing QA ambiguity rules with label-changing NLI ambiguity rules, where ambiguity is defined as an interpretation difference that can change the relation between `entailment`, `neutral`, and `contradiction`; (ii) replacing the question input field with premise–hypothesis input fields; and (iii) replacing the AmbigQA in-context demonstrations with manually constructed AmbiEnt-NLI demonstrations in the same format. These NLI demonstrations cover unambiguous, premise-ambiguous, hypothesis-ambiguous, and both-side ambiguous cases, and are also excluded from the evaluation set.

This adaptation is intended only to express the same decomposition in the NLI task format, rather than to introduce method-specific prompting advantages. Within each dataset, shared modules such as the `Answerer` are kept fixed across all compared methods, and the AmbiEnt-NLI variants preserve the AmbigQA prompting interface while changing only the rules and demonstrations needed to define task-specific ambiguity.

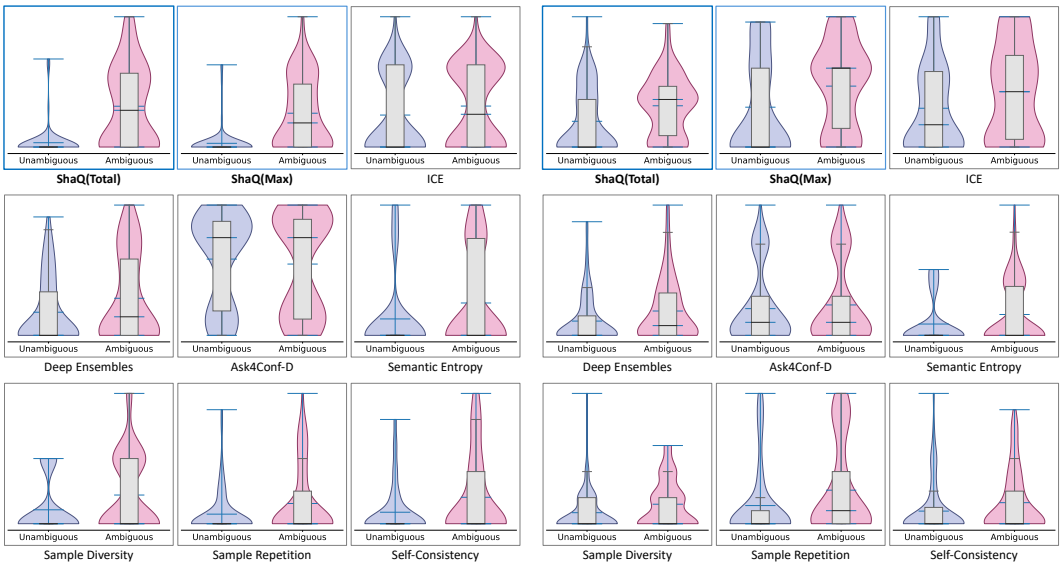
We list the prompts used in our experiments as follows:

- The `Localizer` prompt is shown in Figure 12.
- The Premise `Generator` prompt is shown in Figure 13.
- The `Answerer` prompt is shown in Figure 14.
- The `Clusterer` prompt is shown in Figure 15.
- The Ask4Conf-D prompt is shown in Figure 16.

B.5 Prompts and Configurations for Uncertainty-guided Clarification

We used same configurations for Uncertainty-guided Clarification as subsection B.3. We list the prompts and formats used in our Uncertainty-guided Clarification experiments as follows:

- Uncertainty-guided Question Clarification Prompt for Baseline Methods 17.
- Uncertainty-guided Question Clarification Prompt for `ShaQ` 18.
- Uncertainty Context Format for Uncertainty-guided Question Clarification 19.
- Uncertainty Context Format for Uncertainty-guided Question Clarification with `ShaQ` 20.



(a) AmbiEnt (`GPT-5.4-mini`) (b) AmbigQA (`GPT-4`)

Figure 6: Aleatoric uncertainty distributions on AmbiEnt (left) and AmbigQA (right).

C Additional Results

C.1 Additional Results on Ambiguity Detection

Uncertainty Distribution Analysis. Figures 6, 7, 8 visualize the aleatoric uncertainty distributions for ambiguous and unambiguous samples across different `Answerer` and `Clusterer` backbone models. Across these backbones, `ShaQ` exhibits a similar distributional pattern: ambiguous samples tend to receive higher uncertainty than unambiguous samples. This indicates that the uncertainty estimated by `ShaQ` remains consistently aligned with input ambiguity across different model backbones. Although the degree of separation varies across models, the overall trend remains stable. In particular, `ShaQ` generally produces more distinguishable distributions between ambiguous and unambiguous questions than the baseline methods, supporting the robustness of localized aleatoric uncertainty estimation.

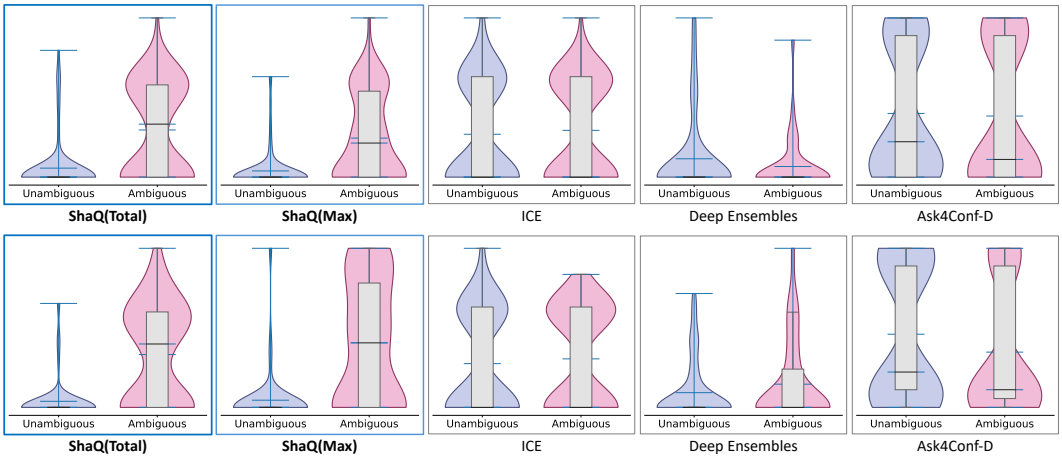


Figure 7: Aleatoric uncertainty distributions on AmbiEnt. Upper row: `Gemini-3.1-flash-lite-preview`, Lower row: `Gemini-2.5-flash`.

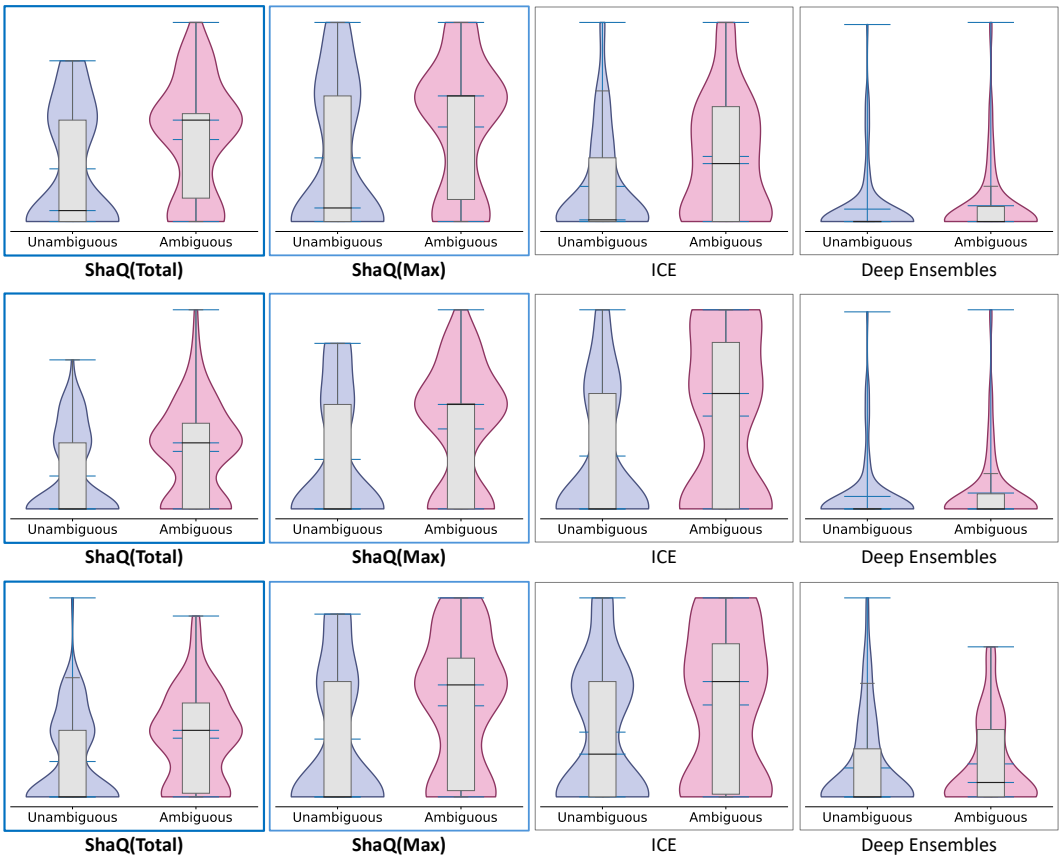


Figure 8: Aleatoric uncertainty distributions on AmbigQA. First row: `GPT-5.4-mini`, Mid row: `Gemini-3.1-flash-lite-preview`, Last row: `Gemini-2.5-flash`.

Table 7: Without semantic clustering and `Clusterer` result at AmbiEnt.

Method	**F1 Score**	**AUROC**	**AUPRC**
`GPT-5.4-mini`			
Direct SE	42.20	57.56	54.36
ICE	58.66	56.52	54.04
Deep Ensembles	56.00	56.71	55.41
LOO	66.07	49.12	57.59
LOO(Max Span)	66.07	56.92	63.24
`ShaQ` **(Ours, Max Span)**	**74.80**	**78.18**	75.72
`ShaQ` **(Ours)**	**74.80**	78.06	**76.30**
`Gemini-3.1-flash-lite-preview`			
LOO	**66.07**	53.23	55.63
LOO(Max Span)	65.47	52.94	56.20
`ShaQ` **(Ours, Max Span)**	65.59	**72.39**	**70.65**
`ShaQ` **(Ours)**	65.59	71.96	69.32
`Gemini-2.5-flash`			
LOO	66.07	38.64	44.39
LOO(Max Span)	65.47	41.48	46.22
`ShaQ` **(Ours, Max Span)**	**71.42**	76.40	74.52
`ShaQ` **(Ours)**	**71.42**	**76.47**	**75.78**

Table 8: Ambiguity detection results on multi-span examples ($|\mathcal{S}| \geq 2$) from AmbiEnt.

Method	**F1 Score**	**AUROC**	**AUPRC**
`GPT-5.4-mini`			
Semantic Entropy	39.13	57.07	78.65
Sample Diversity	38.29	53.66	78.40
Sample Repetition	45.83	60.98	82.47
Self-Consistency	45.83	60.85	82.12
Ask4Conf-D	83.95	40.40	78.13
Deep Ensembles	58.62	45.95	75.25
ICE	59.64	53.40	77.54
LOO	85.36	49.11	79.59
LOO(Max Span)	85.36	51.64	80.26
`ShaQ` **(Ours, Max Span)**	**86.95**	**81.31**	**92.39**
`ShaQ` **(Ours)**	86.56	80.05	91.27
`Gemini-3.1-flash-lite-preview`			
Ask4Conf-D	58.46	33.45	73.34
Deep Ensembles	50.00	40.02	73.21
ICE	15.00	49.49	76.92
LOO	**85.36**	63.38	84.65
LOO(Max Span)	**85.36**	60.85	84.42
`ShaQ` **(Ours, Max Span)**	84.05	**74.11**	**90.80**
`ShaQ` **(Ours)**	82.85	69.82	85.27
`Gemini-2.5-flash`			
Ask4Conf-D	78.94	42.29	76.05
Deep Ensembles	52.63	40.02	72.00
ICE	43.99	50.00	80.80
LOO	85.36	41.16	73.81
LOO(Max Span)	85.36	42.17	74.17
`ShaQ` **(Ours, Max Span)**	**87.87**	85.85	95.36
`ShaQ` **(Ours)**	86.95	**86.23**	**95.47**

C.2 Additional Ablation Results on Ambiguity Detection

As an ablation, we perform ambiguity detection on AmbiEnt, an NLI task whose answer space is limited to three labels: entailment, contradiction, and neutral. Since the possible outputs are already defined by these discrete labels, we omit semantic clustering and the `Clusterer`. The results at Table 7 show the same overall trend as Tables 1 and 3, suggesting that when semantic clustering is unnecessary, the computational cost of LLM-based estimation can be reduced.

C.3 Additional Results on Multi Span Ambiguity Detection

Table 9: Ambiguity detection results on multi-span examples ($|\mathcal{S}| \geq 2$) from AmbigQA. LOO: Leave-One-Out; DEEPENS.: Deep Ensembles; A4C: Ask4Conf-D; SE: Semantic Entropy; SDiv: Sample Diversity; SRep: Sample Repetition; SC: Self-Consistency.

	Aleatoric Methods							*Other Uncertainty Methods*			
Metric	ShaQ	ShaQ **Max**	LOO	LOO Max	ICE	DEEP ENS.	A4C-D	SE	SDiv	SRep	SC
Best F1 Score	**84.21**	83.87	82.53	82.53	75.00	72.00	68.08	60.46	76.59	76.00	75.00

Table 10: Ambiguity detection results on multi-span examples ($|\mathcal{S}| \geq 2$) from AmbigQA with `GPT-5.4-mini`, `Gemini-2.5-flash`, and `Gemini-3.1-flash-lite-preview`. DEEPENS.: Deep Ensembles.

Answerer & Localizer Backbone	ShaQ	ShaQ **Max**	ICE	DEEP ENS.
GPT-5.4-mini	**83.87**	82.53	73.07	75.00
Gemini-3.1-flash-lite-preview	84.21	**87.27**	76.92	61.90
Gemini-2.5-flash	79.31	79.31	**80.70**	68.08

Full Results of Multi-Span Ambiguity Detection (AmbiEnt). Table 8 reports the full results on AmbiEnt examples for which the `localizer` identifies two or more candidate ambiguous spans. In this setting, ShaQ and its max-span variant achieve the strongest overall performance across the evaluated backbone models, especially in AUROC and AUPRC. This indicates that the uncertainty scores produced by ShaQ provide a more reliable ranking of ambiguous versus unambiguous inputs than the compared baselines.

These results highlight the importance of interaction-aware span attribution in multi-span cases. When several candidate ambiguity sources appear in the same input, the uncertainty contribution of one span may depend on how other spans are interpreted. By averaging marginal contributions over span coalitions, ShaQ can capture these interactions more faithfully than methods that rely on aggregate input-level uncertainty or single-context leave-one-out perturbations. The strong performance of the max-span variant further suggests that the largest localized attribution often serves as an effective signal for detecting whether an input contains meaningful ambiguity.

Results of Multi-Span Ambiguity Detection (AmbigQA). Table 9 reports the best score of the full multi-span results from AmbigQA ($|S| \geq 2$). As a result, the ShaQ variants remain the strongest methods: ShaQ achieves the best F1 score, and ShaQ Max obtains the second-best score. In contrast, the newly included output-variation-based estimators, such as Sample Diversity, Sample Repetition, and Self-Consistency, do not outperform the ShaQ variants. This further supports that explicitly localizing input-induced ambiguity provides a stronger signal than relying only on aggregate input-level uncertainty or output variability. Table 10 reports the multi-span results from AmbigQA ($|S| \geq 2$) with recently released reasoning-capable models, including `GPT-5.4-mini`, `Gemini-2.5-flash`, and `Gemini-3.1-flash-lite-preview`. Across these backbones, ShaQ and its max-span variant achieve the best average performance among the compared aleatoric methods.



Figure 9: Qualitative Result of multi span ambiguous questions on AmbigQA (GPT-4). Highlighted span is colored with pink.

Qualitative Analysis on AmbigQA We further conduct a qualitative evaluation of ShaQ on three representative multi-span cases identified by the AmbigQA localizer:

- (1) cases in which all detected spans must be revised to resolve the ambiguity (Figure 9, bottom left)
- (2) cases in which revising a single span resolves the ambiguity associated with the remaining span(s) (Figure 9, top)
- (3) cases in which the localizer incorrectly identifies an unambiguous span as ambiguous (Figure 9, bottom right).

For case (1), ShaQ successfully highlights both spans requiring clarification, while LOO fails to identify them correctly. For case (2), ShaQ highlights the appropriate insertion point where adding information resolves the ambiguity of the other span; this behavior is consistent with the AmbigQA human annotation, which also resolves the ambiguity by introducing additional information. In contrast, LOO highlights all detected spans. For case (3), despite the localizer falsely marking a clear span as ambiguous, ShaQ avoids highlighting the unambiguous proper noun "ten commandments of computer ethics" and instead highlights only the genuinely ambiguous part, whereas LOO highlights the clear span as well. This indicates that, although the initial localization step relies on an LLM and may over-identify candidate spans, ShaQ can assign near-zero uncertainty contribution to spans whose clarification does not reduce predictive uncertainty.

These examples show that ShaQ can help filter out unnecessary or incorrectly localized spans by estimating their actual contribution to predictive uncertainty, even when the localizer over-identifies candidate spans or marks clear expressions as ambiguous. Overall, this suggests that ShaQ provides more precise and reliable fine-grained aleatoric uncertainty quantification than LOO.

Qualitative Analysis on AmbiEnt In AmbiEnt, ambiguity may arise independently from either the premise or the hypothesis. We therefore conduct a qualitative evaluation of ShaQ on two representative multi-span cases identified by the AmbiEnt localizer:

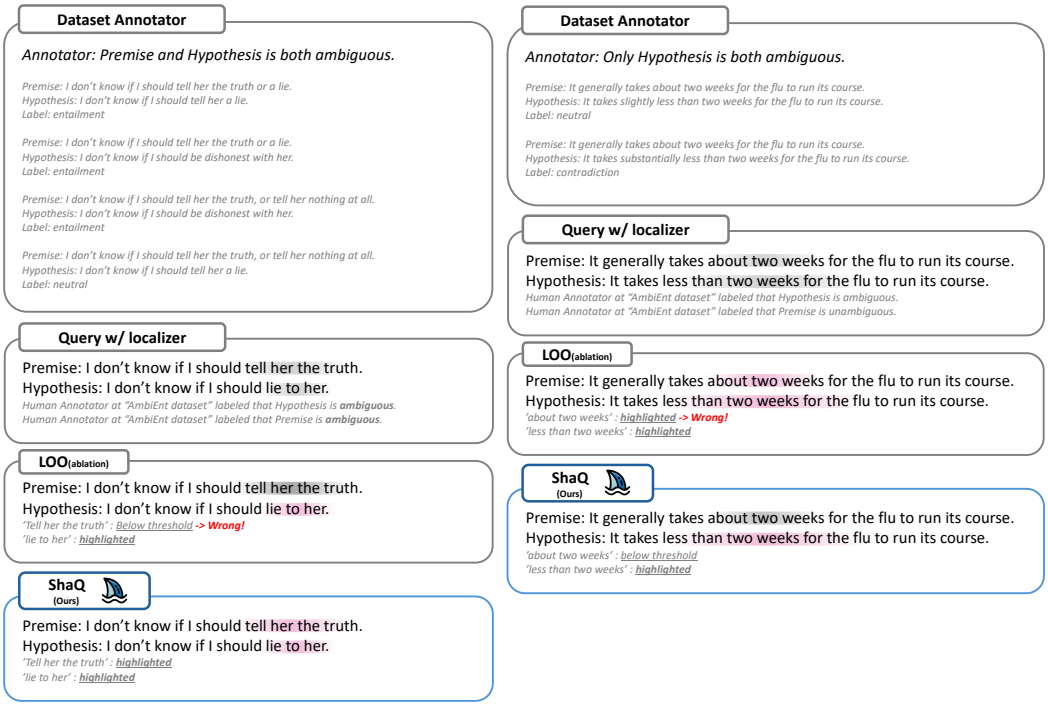


Figure 10: Qualitative Result of multi span ambiguous questions on AmbiEnt (`GPT-5.4-mini`). Highlighted span is colored with pink.

- (1) cases in which all detected spans must be revised to resolve the ambiguity (Figure 10, left);
- (2) cases in which the `localizer` incorrectly identifies an unambiguous span as ambiguous (Figure 10, right).

Although neither AmbiEnt nor AmbigQA provides span-level ambiguity annotations, AmbiEnt includes annotations indicating whether the premise and hypothesis are ambiguous, respectively. We use these annotations as a weak reference for evaluating span localization: if the annotator highlights a region in the premise or hypothesis, we regard the corresponding sentence component as ambiguous; otherwise, we treat it as unambiguous. This allows us to compare whether the spans highlighted by each method are consistent with the ambiguity labels assigned at the premise and hypothesis level.

For case (1), `ShaQ` successfully highlights both spans that require clarification, whereas LOO fails to identify them correctly. For case (2), despite the `localizer` falsely marking the unambiguous span "about two weeks" as a candidate ambiguous span, `ShaQ` avoids assigning a high uncertainty score to this clear span and instead highlights only the genuinely ambiguous part. In contrast, LOO also highlights "about two weeks", suggesting that it is more vulnerable to false-positive candidates produced by the localization stage. These results indicate that, although the initial localization step relies on an LLM and may over-identify candidate spans, `ShaQ` can assign near-zero uncertainty contribution to spans whose clarification does not reduce predictive uncertainty. This behavior demonstrates the robustness of `ShaQ` as a post-localization uncertainty attribution method, particularly in multi-span settings where ambiguous and unambiguous candidate spans may coexist.

Shapley-style interpretation of multi-span cases. The qualitative examples can be interpreted through the Shapley-style marginal contribution view underlying `ShaQ`. In our formulation, each localized span is treated as a player for attribution purposes, and its score is determined not by whether the span appears ambiguous in isolation, but by its average marginal reduction of predictive entropy across clarification contexts. This perspective helps explain several recurring multi-span patterns.

First, when the `localizer` includes an unambiguous span together with a genuinely ambiguous span, the unambiguous span behaves as a near-dummy player: clarifying it produces little or no reduction in predictive entropy under most contexts. This pattern appears in AmbigQA Case (3) and AmbiEnt Case (2), where the localizer over-identifies a clear span as a candidate ambiguity. For example,

in AmbiEnt Case (2), the span "about two weeks" is initially detected as a candidate ambiguous span, but clarifying it does not meaningfully reduce predictive uncertainty. ShaQ therefore assigns near-zero attribution to such spans, even if they were initially marked as candidates by the localizer.

Second, when multiple spans are complementary sources of ambiguity and the input becomes clear only after they are jointly clarified, the uncertainty reduction is shared across the corresponding spans. This pattern corresponds to AmbigQA Case (1) and AmbiEnt Case (1), where all detected spans must be revised to fully resolve the ambiguity. In this case, no single span fully accounts for the ambiguity in isolation, and the Shapley-style averaging distributes the joint uncertainty according to each span's marginal contribution across different clarification contexts.

Third, when multiple spans are partially redundant–that is, clarifying one span can substantially reduce the ambiguity associated with the others–the attribution need not be uniformly split. This pattern is illustrated by AmbigQA Case (2), where revising a single span or insertion point can resolve ambiguity associated with the remaining span(s). If the spans are symmetric in their effects on the answer distribution, the attribution is expected to be shared. However, if one span induces a richer set of plausible interpretations or yields larger entropy reductions across more contexts, ShaQ assigns a larger attribution to that span. Thus, concentrated attribution reflects an asymmetric contribution to the model's predictive uncertainty, rather than an artifact of the localizer.

C.4 Stability Analysis

We evaluate ShaQ and ICE on the same AmbiEnt examples using GPT-5.4-mini across five random seeds, keeping all other experimental settings fixed. For each example, we compute the variance of its uncertainty estimates across seeds and average these variances over the evaluation set. As shown in Table 11, ShaQ achieves a lower mean per-example variance than ICE (0.00260 vs. 0.00312), indicating comparatively more stable uncertainty estimates across random seeds in this setting.

Table 11: Stability of uncertainty estimates on AmbiEnt (lower is better).

Method	Mean variance ↓
ICE	0.00312
ShaQ	**0.00260**

C.5 Distribution of Ambiguous Spans

We examine the number of ambiguous spans localized by ShaQ in our main experiments on AmbigQA and AmbiEnt, using GPT-4 and GPT-5.4-mini, respectively, and excluding spans with negligible Shapley attribution. Table 12 reports the distribution by gold ambiguity label, with all percentages computed relative to the full evaluation set of each dataset. After filtering, 14% of AmbigQA inputs and 16% of AmbiEnt inputs retain at least two spans, motivating the consideration of potential interactions among spans. The dedicated evaluation on the AmbiEnt multi-span subset (Table 4 and Appendix Table 8) further shows that ShaQ achieves the best performance across F1, AUROC, and AUPRC.

Table 12: Distribution of localized ambiguous span counts (%).

Dataset	Gold	0	1	2	3	≥ 2
AmbigQA	Ambig.	10.50	30.00	10.50	0.50	11.00
	Unambig.	24.50	21.00	3.00	0.00	3.00
	Total	35.00	51.00	13.50	0.50	14.00
AmbiEnt	Ambig.	18.67	16.67	14.00	0.67	14.67
	Unambig.	47.33	1.33	1.33	0.00	1.33
	Total	66.00	18.00	15.33	0.67	16.00

Table 13: High ShaQ Max-Span Aleatoric Uncertainty Set (> 1).

Method	Clarified	$\Delta H(\uparrow)$	Edit ($\downarrow$)
Semantic Entropy	0.3429	0.1115	5.8571
ICE	0.5397	-0.0852	7.2857
Deep Ensembles	0.5245	-0.0700	7.3571
ShaQ **(Ours)**	**0.3296**	**0.1248**	**4.9285**

Table 14: High ShaQ Max-Span Aleatoric Uncertainty Set (top 20%).

Method	Clarified	$\Delta H(\uparrow)$	Edit ($\downarrow$)
Semantic Entropy	0.3165	0.0654	6.7142
ICE	0.4872	-0.1051	7.1428
Deep Ensembles	0.4426	-0.0605	7.1428
ShaQ **(Ours)**	**0.3159**	**0.0661**	**5.0476**

Table 15: High Baseline Max-Total Aleatoric Uncertainty Set (top 30).

Method	Clarified	$\Delta H(\uparrow)$	Edit ($\downarrow$)
Semantic Entropy	0.3374	-0.0174	6.1333
ICE	0.3684	-0.0277	6.6333
Deep Ensembles	0.3050	0.0149	7.5000
ShaQ **(Ours)**	**0.2848**	**0.0351**	**5.5333**

Table 16: High Baseline Max-Total Aleatoric Uncertainty Set (top 20%).

Method	Clarified	$\Delta H(\uparrow)$	Edit ($\downarrow$)
Semantic Entropy	0.3489	0.0066	6.1428
ICE	0.3477	0.0207	6.7142
Deep Ensembles	0.3239	0.0315	7.8095
ShaQ **(Ours)**	**0.3008**	**0.0547**	**5.8571**

Table 17: High Baseline Uncertainty Disagreement Set.

Method	Clarified	$\Delta H(\uparrow)$	Edit ($\downarrow$)
Semantic Entropy	0.1529	0.0199	5.5714
ICE	0.1676	0.0052	6.0952
Deep Ensembles	0.1758	-0.0029	6.7619
ShaQ **(Ours)**	**0.1355**	**0.0373**	**5.0476**

C.6 Additional Results on Uncertainty-guided Clarification

Breakdown by Selection Criterion. We provide a breakdown of uncertainty-guided clarification results under three complementary high-uncertainty settings. The first two settings correspond to the two selection criteria used to construct the evaluation superset in the Section 5.3: examples selected by high ShaQ max-span aleatoric uncertainty and examples selected by high baseline max-total aleatoric uncertainty. The third setting considers examples where baseline uncertainty estimators disagree with one another. Together, these settings allow us to analyze whether ShaQ remains effective across cases identified by its own localized uncertainty signal, cases identified by conventional uncertainty estimators, and cases where existing estimators provide unstable guidance.

High ShaQ Max-Span Aleatoric Uncertainty Set. This setting focuses on gold ambiguous examples for which ShaQ assigns high maximum span-level aleatoric uncertainty. We consider two variants: a threshold-based subset where the maximum span uncertainty is greater than 1, reported in Table 13, and the top-20% subset used in the main-text superset construction, reported in Table 14. These examples represent cases where ShaQ identifies a particularly strong localized source of ambiguity within the input.

As shown in Tables 13 and 14, ShaQ-guided clarification achieves the largest entropy reduction while requiring the fewest edits. This indicates that the examples strongly flagged by ShaQ are not merely high-scoring artifacts, but genuinely ambiguous cases for which localized clarification is especially useful. In this regime, baseline-guided rewrites often require larger modifications and may even fail to reduce entropy, suggesting that these methods do not reliably identify the specific part of the input that should be clarified. Thus, high max-span uncertainty under ShaQ captures fine-grained ambiguous samples that are difficult for aggregate uncertainty methods to handle.

High Baseline Max-Total Aleatoric Uncertainty Set. This setting focuses on gold ambiguous examples that existing uncertainty estimators strongly flag as uncertain. We consider both a top-30 subset, reported in Table 15, and the top-20% subset used in the main-text superset construction, reported in Table 16. These examples represent cases that conventional input-level or output-level uncertainty estimators regard as difficult.

As shown in Tables 15 and 16, ShaQ again achieves the strongest clarification performance, reducing entropy more effectively while requiring fewer edits. This shows that ShaQ is not only effective on samples selected by its own uncertainty score. Even when the evaluation examples are chosen according to baseline uncertainty signals, ShaQ can identify the localized source of ambiguity and provide more useful information for clarification. In other words, for examples that baselines can detect as uncertain, ShaQ can still provide better guidance on how the input should be revised.

High Baseline Disagreement Set. We further evaluate on a challenging subset of gold ambiguous examples where baseline uncertainty methods disagree, selected by the top 20% standard deviation of baseline uncertainty scores. This subset captures cases in which conventional input-level or output-level uncertainty estimators provide unstable or conflicting signals, making it difficult to determine which uncertainty estimate should guide clarification.

As shown in Table 17, ShaQ achieves the largest entropy reduction while requiring the smallest edit distance. This indicates that, even when existing uncertainty estimators disagree, ShaQ can still provide a more actionable ambiguity signal for clarification. In particular, localized span-level uncertainty helps identify the specific ambiguity that should be resolved, rather than relying on a single global uncertainty score that may be sensitive to the choice of estimator.

C.7 Additional Results on Human-to-Human Medical Multi-turn Dialogue

Dataset. We evaluate on MediTOD [26], a publicly available English dataset of staged doctor-patient interviews covering respiratory and musculoskeletal specialties, comprising 213 dialogues with 22,503 annotated utterances. Each dialogue follows the Objective Structured Clinical Examination (OSCE) format, in which medical professionals enact both doctor and patient roles to systematically elicit a patient history. Dialogues are highly conversational, averaging 96 utterances per dialogue, and are annotated with comprehensive intent and slot-attribute labels under the Comprehensive Medical Attribute Schema (CMAS).

Experimental Setup. A key distinction of this evaluation is that ShaQ is applied *online*: at each utterance, only the preceding dialogue history is provided as context, and future utterances are deliberately withheld from the prompt. This mirrors the real-world constraint that an interlocutor must assess potential misunderstanding in real time, without foreknowledge of how ambiguity will be resolved. Concretely, ShaQ localizes ambiguous spans within each utterance conditioned on the accumulated dialogue history, and the detected spans are logged at every turn. We use GPT-4.1 as the base model throughout. Notably, the vast majority of utterances in this setting are declarative statements rather than queries, which highlights a key strength of ShaQ: because its core mechanism operates by simulating clarification of an ambiguous span, it is agnostic to the surface form of the input and applies equally to questions, statements, and other utterance types. To accommodate this, we adopt a generalized version of ShaQ in which the Answerer, rather than generating an answer to a query, rewrites the current declarative utterance by embedding the meaning of a given premise directly into the statement – leaving all other components unchanged. Detailed prompts are provided in prompts 21 22 23 24.

Practical Significance. This setting exposes a practically important capability of ShaQ: rather than performing a one-shot analysis of a complete document, it can serve as a *turn-level ambiguity monitor* that alerts interlocutors to potentially problematic spans as a conversation unfolds. In the medical domain, where a misunderstood symptom descriptor or temporal qualifier can have serious downstream consequences for diagnosis and treatment, the ability to flag ambiguous spans at each turn – before the conversation proceeds – represents a concrete and high-value application beyond the LLM input setting for which ShaQ was originally designed.



Figure 11: **Extensive Analysis For MediTOD Dialogue Set.**

Extensive Medical Dialogue Case Study. Figure 11 presents ten representative cases spanning qualitatively distinct ambiguity types, demonstrating the breadth of clinically significant uncertainty that ShaQ is able to localize. We organize our observations as follows.

Diversity of ambiguity types. The detected spans exhibit a wide variety of linguistic and clinical ambiguity. *Lexical ambiguity* arises when a single expression admits multiple medically distinct readings: in (c), *"a bit feverish"* is ambiguous between a subjective thermal sensation and a measurable low-grade fever, and in (e), *"I haven't like right now"* is ambiguous between a temporary lapse in medication adherence and complete discontinuation of Claritin. *Referential ambiguity* occurs when the antecedent of a referring expression is underspecified: in (f), the demonstrative *"is that"* could resolve to loss of smell, loss of taste, or both, with the surrounding context supporting all three readings simultaneously. *Scope ambiguity* manifests in (b), where *"the other stuff"* is unclear as to whether it encompasses all co-occurring symptoms or only a subset, materially affecting the clinical interpretation of the patient's prior illness history, and in (j), where *"anybody in the home"* is ambiguous between household members and any recent visitor, altering the exposure risk assessment. *Syntactic ambiguity* appears in (g), where *"have you can have been consuming the drug"* is structurally malformed in a way that leaves the temporal scope–current, past, or lifetime use–and the referent–a specific drug or any drug–simultaneously underspecified.

Ambiguity in both speakers. Notably, ShaQ detects ambiguous spans in the utterances of *both* the patient and the doctor, underscoring that miscommunication risk is not unidirectional. Patient-side cases include (a), (b), (c), (d), and (e), where vague or hedged self-reports carry clinically consequential ambiguity. Doctor-side cases include (f), (g), (h), (i), and (j), where imprecise clinical phrasing risks eliciting misaligned patient responses. For instance, in (h), the physician's exclusion clause *"aside from when he was born"* is ambiguous as to whether it covers only the birth hospitalization or also a subsequent NICU admission for being underweight, a distinction that could alter the reported medical

Table 18: Computational cost and backbone ablation on AmbiEnt. Results in (b) are reported as Total / Max Span; Qwen denotes `Qwen3-VL-8B-Instruct`.

(a) End-to-end cost

Metric	ShaQ	ICE	Ratio
Time (s)	559.42	247.61	2.26×
API requests	2,635	1,125	2.34×
Total tokens	5,540,288	2,199,803	2.52×
API cost ($)	2.49	1.06	2.36×

(b) Backbone performance

Method	F1	AUROC	AUPRC
ICE	57.33	55.40	53.21
ShaQ **(Main)**	73.84 / 74.60	77.68 / 77.80	75.71 / 75.19
ShaQ **(Qwen L)**	75.15 / 74.85	73.14 / 71.08	69.47 / 65.71
ShaQ **(Qwen L+G)**	70.74 / 70.45	70.54 / 67.56	68.43 / 60.94

(c) L/G API cost

Backbone	Cost ($)
`GPT-5.4-mini`	0.42
`Qwen 8B`	0.16

history. In (i), the term *"Inorganic cause meaning something going on"* conflates a physical/medical etiology with a non-organic psychiatric one, representing a framing that could bias the patient's subsequent response.

Summary. Across all ten cases, the identified spans are clinically meaningful: misinterpretation of any one of them could lead to an incorrect diagnosis, an inappropriate treatment recommendation, or a missed follow-up question. These results collectively demonstrate that ShaQ generalizes well beyond the LLM question-answering setting for which it was designed, offering a principled and interpretable mechanism for turn-level ambiguity monitoring in real-world clinical dialogue.

D Computational Analysis

The primary computational bottleneck of ShaQ lies in the `Answerer` calls required for all joint clarifications. While concatenating multiple joint clarifications into a single LLM call may seem a natural shortcut, it incurs additional cross-attention computation among the concatenated premise sets, whose cost grows quadratically with the combined sequence length. Instead, since all joint clarification combinations share the same query x as a common prefix, each call effectively reduces to processing the premise suffix alone – a property that most modern LLM inference engines exploit via *prefix KV-caching* [37], caching the key – value representations of x and reusing them across calls. Consequently, ShaQ computes Shapley values by varying only the premise clarifications in the suffix across calls, leaving the shared prefix untouched. Furthermore, beyond inference optimizations, the overall computational cost of ShaQ can be significantly reduced by employing lightweight LLMs. As demonstrated in Tables 2 and 3, we evaluated ShaQ using smaller, cost-efficient models such as `GPT-5.4-mini`, `Gemini-3.1-flash-lite-preview`, and `Gemini-2.5-flash` for the `Answerer` and `Clusterer` backbones. Despite the reduced model capacity, ShaQ consistently outperforms baselines, demonstrating that our framework remains highly effective and computationally practical even in resource-constrained settings.

Compute resources and API budget. All LLM inference in our experiments was conducted through API-based inference via OpenRouter. We did not run any backbone models locally. Local compute was used only for experiment orchestration, prompt construction, response parsing, aggregation, and metric computation. The experiments were orchestrated from a CPU machine with 64 logical CPU cores and approximately 1 TiB RAM; no local GPU was used. Unless otherwise stated, runs used `max_workers=5` for example-level parallel API orchestration. For repeated-sampling calls, multiple requested samples were issued as repeated single-completion API calls, with concurrency subject to the same orchestration setting and OpenRouter rate limits.

We compare the end-to-end computational cost of ShaQ and ICE on 150 AmbiEnt examples using `GPT-5.4-mini`, a fixed random seed of 42, and five concurrent workers. As shown in Table 18(a), ShaQ requires 559.42 seconds and $2.49 in OpenRouter API cost, compared with 247.61 seconds and $1.06 for ICE. This corresponds to 2.26× the wall-clock time and 2.36× the API cost, or approximately 3.7 seconds and $0.017 per example for ShaQ.

Cost reduction with a smaller backbone. We further measure the combined API cost of the Localizer and Generator on the same 150 AmbiEnt examples, keeping the Answerer and Clusterer fixed to `GPT-5.4-mini`. As shown in Table 18(c), replacing `GPT-5.4-mini` with `Qwen3-VL-8B-Instruct` for both modules reduces their combined cost from $0.42 to $0.16, a reduction of approximately 62%. This reduction applies specifically to the Localizer/Generator stage, rather than the full pipeline. As shown in Table 18(b), although using a smaller backbone leads to some performance degradation,

ShaQ continues to outperform ICE across all three detection metrics when replacing either the Localizer alone or both the Localizer and Generator. In particular, the respective AUROC scores remain at 73.14 and 70.54, compared to 55.40 for ICE.

E Limitations and Future Directions

While ShaQ provides a principled framework for localizing input-induced uncertainty, several limitations remain. In our current formulation, premises for each ambiguous span are generated independently, and Shapley values are computed through bottom-up marginalization. This design follows the *Marginal* (or interventional) Shapley value approach commonly used in feature attribution, offering mathematical tractability, non-negative marginal information gains, and computational efficiency without requiring conditional premise sampling. However, this computational choice does not imply that the interpretations of different spans are semantically independent in natural language.

Another practical consideration concerns inputs without genuine ambiguity. Ideally, such inputs contain no input-induced uncertainty, i.e., $I(Y; C \mid x) = 0$. Under our theoretical formulation, the efficiency property and non-negativity of Shapley attributions then imply $\phi_k = 0$ for every candidate span. Thus, even if the Localizer identifies a spurious ambiguous span, its attribution should approach zero when clarifying it does not meaningfully reduce predictive entropy. In practice, however, imperfect localization, finite response sampling, and semantic clustering errors may introduce small nonzero estimates.

Computational overhead is another practical limitation, as the number of joint clarification combinations grows rapidly with the number of localized ambiguous spans. As shown in Table 18(a), ShaQ incurs $2.26\times$ the wall-clock time and $2.36\times$ the API cost of ICE on AmbiEnt. Although lightweight backbones and inference optimizations can mitigate this overhead, scaling to inputs with many ambiguous spans remains computationally challenging.

These limitations suggest several promising directions for future work. In complex contexts, the plausible interpretation of one span may be influenced by how another span is resolved, a relationship not explicitly modeled in our current formulation. One possible extension is to introduce a *Premise Generation Checker* that filters logically inconsistent premise combinations. More generally, modeling conditional dependencies among span interpretations could enable a form of *Conditional* Shapley estimation. Another promising direction is to extend ShaQ to explicitly model temporal dynamics in multi-turn dialogues, where contextual information and sources of ambiguity evolve over time. Investigating how to dynamically update span-level uncertainty attributions as new information becomes available, while maintaining attribution fidelity and computational efficiency, remains an interesting avenue for future work.

F Practical Considerations

Since ShaQ employs an LLM-based Localizer to identify candidate ambiguous spans, the quality of localization affects subsequent uncertainty attribution. In particular, missing a genuinely ambiguous span prevents its contribution from being recovered, whereas including a spurious span is generally less harmful, as clarifying such a span typically induces little entropy reduction and thus receives a near-zero Shapley attribution. This asymmetry suggests that the Localizer should favor broader candidate coverage over strict selectivity. Furthermore, incorporating relevant background knowledge into the localization process could help identify more compact and semantically coherent spans by better accounting for context-dependent ambiguity.

For practical interpretation, uncertainty attributions can be normalized according to the desired reference scale. Normalizing each attribution ϕ_k by the total predictive entropy $H(Y \mid x)$ indicates the fraction of predictive uncertainty attributable to the corresponding span, whereas normalization by the total input-induced uncertainty $I(Y; C \mid x) = \sum_k \phi_k$ represents its relative contribution among localized spans, with the normalized attributions summing to one. Alternatively, under the empirical distribution over K semantic clusters, the predictive entropy is upper-bounded by $\log K$. Thus, normalization by $\log K$ provides a reference scale relative to the maximum possible entropy over the observed semantic classes. These normalizations offer complementary interpretations of uncertainty attributions and assume nonzero denominators.

Finally, our implementation estimates semantic cluster probabilities using the empirical frequencies of sampled responses. While this approach is straightforward and model-agnostic, more fine-grained probability estimates could potentially improve the precision of uncertainty quantification. For tasks with predefined semantic classes, likelihood-based estimation or soft probabilistic assignments could provide more informative probability estimates than empirical frequencies alone. However, such approaches are challenging in open-ended generation, where the semantic output space is not known in advance and reliable class-level probabilities are difficult to obtain. In these settings, our frequency-based approach provides a practical alternative that naturally accommodates dynamically emerging semantic classes without requiring a predefined output space.

G Broader Impacts

This work addresses a fundamental limitation of existing uncertainty quantification methods for LLMs: the inability to identify which part of an input drives predictive uncertainty. By providing span-level uncertainty attribution, `ShaQ` enables more transparent and actionable human-AI interaction, which is especially valuable in real-world deployments where improving reliability must rely on resolving input ambiguity rather than modifying model parameters. The positive societal impacts of `ShaQ` are most pronounced in high-stakes domains. As demonstrated in our MediTOD evaluation, `ShaQ` can serve as a turn-level ambiguity monitor in clinical dialogues, flagging underspecified symptom descriptors or temporal qualifiers before a conversation proceeds. In such settings, targeted localization of ambiguous spans can help prevent misdiagnoses or inappropriate treatment recommendations arising from miscommunication between patients and clinicians. More broadly, `ShaQ` provides a principled mechanism for LLMs to communicate where users should focus their clarification efforts, reducing unnecessary user effort while improving the reliability of downstream decisions. Beyond the medical domain, `ShaQ` has potential applications wherever LLMs are deployed as decision-support tools and input quality directly affects output reliability – including legal document review, customer service, and educational tutoring systems. In these contexts, fine-grained uncertainty localization can enhance human oversight by drawing attention to the specific portions of an input that most warrant revision or verification. As with any method that automatically identifies ambiguous content in user inputs, `ShaQ` should be deployed with appropriate human oversight, particularly in sensitive domains where incorrect localization could mislead users. We recommend that practitioners treat `ShaQ`'s span-level attributions as decision-support signals rather than definitive judgments, and that the system be evaluated on domain-specific data before deployment in high-stakes settings.

H Asset Licenses and Terms of Use

We use only existing datasets, baselines, and proprietary model APIs for research evaluation. All datasets, baselines, and prior methods used in this work are credited through citations in the main paper and references. We use AmbigQA [24], AmbiEnt [25], and MediTOD [26] only for non-commercial academic research and evaluation, following their intended research use. We do not redistribute the original datasets; instead, we provide instructions and scripts for obtaining them from their official sources.

AmbigQA/AmbigNQ is based on NQ-Open and is distributed for research use; the HuggingFace mirror lists the dataset under CC BY-SA 3.0. AmbiEnt is released under CC BY 4.0 according to its official repository. MediTOD is an existing medical dialogue benchmark constructed from simulated doctor-patient dialogues; its source dialogues are described as simulated and publicly available under CC0 in the original dataset paper. We cite the original creators of all datasets and respect the corresponding license and terms-of-use requirements.

For baselines and prior methods, including ICE, Deep Ensembles, Ask4Conf-D, Semantic Entropy, Sample Diversity, Sample Repetition, and Self-Consistency, we cite the original papers and implement or adapt them only for comparative research evaluation.

For proprietary LLMs, we accessed GPT-based and Gemini-based models through OpenRouter, a third-party API gateway for LLM providers, rather than directly through individual provider APIs. Specifically, our experiments used `GPT-4`, `GPT-5.4-mini`, `Gemini-2.5-flash`, and `Gemini-3.1-flash-lite-preview`. These models were used only for inference in our experimental pipeline, including span localization, premise generation, answer generation, semantic clustering,

and baseline evaluation. Our use of these models is subject to the OpenRouter Terms of Service and, where applicable, the terms of service, usage policies, and data-handling policies of the underlying model providers.

We do not redistribute proprietary model weights, modify or fine-tune the underlying proprietary models, or claim ownership of the models themselves. Generated outputs are used solely for research evaluation and analysis. We will comply with any additional attribution, safety, usage, and publication requirements specified by OpenRouter and the corresponding model providers.

ShaQ Prompt: Localizer

You are an ambiguity detector for natural language questions.
Your task is to copy the input question exactly and wrap only the ambiguous spans with XML tags of the form `<ambig id="1">...</ambig>`.

Ambiguity Criteria
Mark a span as ambiguous if knowledgeable people would give DIFFERENT FACTUAL ANSWERS to the question because the span can be interpreted in multiple common ways.

MARK as ambiguous:

1. Entity overlap — the same name refers to two or more well-known entities that yield different answers. (e.g., "the flash" → 1990 TV series vs 2014 TV series)
2. Role/title holders — role-based questions where MORE THAN ONE person is commonly associated, especially with words like "current", "recent", or no time qualifier. (e.g., "Who is the administrator of the SBA?" — changed multiple times)
3. Missing specifier — a key dimension (sport, year, country, gender, edition) is absent and there are multiple well-known correct answers across that dimension. (e.g., "Who won the world cup?" — no sport, no year)
4. Temporal — the answer has changed over time AND the question does NOT fix a specific time period. Err toward marking temporal ambiguity when the role or record holder has changed within the past ∼20 years, even if one answer is "more famous".

Do NOT mark as ambiguous:

- Questions with a single universally dominant answer that virtually everyone would give. Example: "Who invented the theory of relativity?" → Einstein, clear.
- Questions where the subject has only one well-known referent despite theoretical alternatives. Example: "Who plays the doctor in Dexter Season 1?" → clear.
- Trivially under-specified questions where reasonable experts converge on the same answer.

Formatting Rules

- Do not rewrite, paraphrase, reorder, or clean up the sentence.
- The output sentence must match the original question EXACTLY after removing the XML tags.
- Preserve every character exactly, including typos, punctuation, and spacing.
- Never reuse an id. Use consecutive ids starting from 1.
- If the question is clear, return the original sentence unchanged as `tagged_sentence`.
- If ambiguity comes from missing context, insert an empty tag `<ambig id="k"> </ambig>` exactly at the position where the missing information should naturally appear.
- Ambiguous spans must be non-overlapping.
- Do not create nested or partially overlapping `<ambig>` tags.
- If the question is clear, return the original sentence unchanged as `tagged_sentence`.

Examples
`{Examples}`

Task
Input question: `{x}`

Figure 12: **Prompt used for the Localizer.**

ShaQ Prompt: Premise Generator

You are a precise premise generator for ambiguous questions.
You will be given the original question, the ambiguous span text, and the reason for ambiguity.
Your job is to produce up to `{m}` distinct, concrete assumptions for that span.

Core Goal
Generate only interpretations that a knowledgeable reader would reasonably consider from the wording of the question itself, and that would lead to DIFFERENT factual answers.

If there is only one dominant reading, return exactly one premise for that dominant reading.
If there are fewer than `{m}` strong interpretations, return fewer than `{m}`.
Do not add filler.

Rules

1. Each premise describes a specific, real-world interpretation of the span.
2. Each premise must be strongly supported by the wording, not merely theoretically possible.
3. DO NOT invent fictional or unverifiable entities. All premises must point to real, well-known facts, media, people, or events.
4. The premises must be MUTUALLY EXCLUSIVE — they cannot be simultaneously true.
5. Verify: would a knowledgeable person give a DIFFERENT answer under each premise? If not, do not include both.
6. Keep each premise concise, one short sentence.
7. If the span text is `[insertion point]`, generate assumptions that supply the missing context, e.g., a specific year, country, sport, or edition.
8. Prefer the MOST COMMON or WELL-KNOWN interpretations first.
9. Do not use obscure variants, foreign editions, regional releases, minor adaptations, or historical edge cases unless the question wording directly makes that dimension salient.
10. Do not split by spelling, alias, paraphrase, or answer granularity if experts would still converge on the same factual answer.
11. When in doubt, be conservative: return fewer premises rather than low-plausibility ones.

Examples
`{Examples}`

Task
Question: `{clean_sentence}`
Span id: `{span_id}`
Span text: `{span_text}`
Ambiguity reason: `{reason}`

The `"premises"` array must contain between 1 and `{m}` items.
Add another item only if it is a strong, common, answer-changing interpretation.

Figure 13: **Prompt used for the Premise Generator.**

Prompt: Answerer

You are a factual QA system that answers questions under specific assumptions.

Task
Answer the question using ONLY the specific interpretation described in the Assumptions.
Your answer must be the factual entity, name, number, or date that is correct
UNDER THAT SPECIFIC ASSUMPTION — not the most famous or default answer.

Rules

1. Your answer MUST be ≤ 5 words. Output ONLY the answer entity.
2. STRICTLY follow the assumption. If the assumption specifies a particular year, TV series, edition, or context, answer for THAT specific version — not the most well-known one.
3. DO NOT explain, hedge, or add qualifiers. No "According to..." or "I think...".
4. If the assumption points to an obscure but real entity, output that entity. DO NOT fall back to the most famous answer.
5. If Assumptions say "None.", answer the question with its most common interpretation.

Examples
`{Examples}`

Question: `{clean_sentence}`

Assumptions:
`{premises}`

Answer:

Figure 14: **Prompt used for the Answerer.**

Prompt: Clusterer

You are a semantic clustering assistant.
Your task is to group a list of short answers to a question into clusters based on whether they refer to the SAME real-world entity or fact.

Core Rule

Two answers belong to the SAME cluster if and only if they refer to the same real-world entity, fact, number, or date — regardless of surface wording.
Two answers belong to DIFFERENT clusters if they refer to different real-world entities, facts, numbers, or dates.

Detailed Rules

1. SAME cluster — surface variations of the same entity:

- "Paris" and "paris, france" → same cluster
- "1992" and "nineteen ninety-two" → same cluster
- "George Washington" and "Washington" → same cluster
- "The 2014 CW series" and "2014 Flash" → same cluster
- "22 episodes" and "22" → same cluster
- "Amy Adams" and "Amy Lou Adams" → same cluster

2. DIFFERENT clusters — genuinely distinct entities or facts:

- "Paris" and "London" → different clusters
- "1990" and "2014" → different clusters
- "22" and "23" → different clusters
- "France" and "Argentina" → different clusters

3. Refusals: "Unknown", "I don't know", "N/A", or any refusal to answer form their own single cluster, separate from all factual answers.

4. Partial overlap: If two answers share some words but refer to different entities, they go in DIFFERENT clusters.
- "New York Yankees" and "New York Mets" → different clusters

5. Assign cluster ids as consecutive integers starting from 0.

6. Every answer index must be assigned to exactly one cluster.

Examples
{`Examples`}
Task
Question: {`question`}
Answers: {answers_numbered_list}

Figure 15: **Prompt used for the Clusterer.**

Prompt: Ask4Conf-D

Read the following question:

Question: `{sentence}`

Provide the probability that this question is ambiguous due to factors such as ambiguous entities, ambiguous event references, or ambiguity over the answer type. Give ONLY the probability, with no other words or explanation.

For example:
Probability: <the probability between 0.0 and 1.0 that the question is ambiguous; 1.0 means the question is absolutely ambiguous. Do not include any extra commentary; output only the probability.>

Figure 16: **Prompt used for Ask4Conf-D.**

Uncertainty-guided Question Clarification Prompt for Baseline Methods

You rewrite ambiguous QA questions so that answer uncertainty is reduced while the original broad intent is preserved.
You are given only an original question and numeric uncertainty scores from one uncertainty method.

Original question:
`{{ORIGINAL_QUESTION}}`

Numeric uncertainty scores:
`{{UNCERTAINTY_CONTEXT}}`

Rules:
1. Return exactly one rewritten direct wh-question.
2. Preserve the original topic and answer type.
3. Do not answer the question.
4. Do not include a candidate answer in the question unless that text already appeared in the original question.
5. Do not ask a yes/no question and do not ask the model to list multiple alternatives.
6. If the original question is already clear, keep it nearly unchanged.

Figure 17: **Prompt used for the Uncertainty-guided Clarification.**

Uncertainty-guided Question Clarification Prompt for ShaQ

You rewrite ambiguous QA questions so that answer uncertainty is reduced while the original broad intent is preserved.

You are given only an original question, numeric uncertainty scores, and localized span-level uncertainty scores from one uncertainty method.

Definitions:
- query ambiguity: the total aleatoric uncertainty score assigned to the whole question. Larger values mean the question is more ambiguous.
- Fine Grained uncertainty scores: ambiguity scores assigned to specific parts of the question. Larger values indicate parts that should be prioritized for minimal clarification.
- Span: a contiguous text segment from the original question that was identified as a possible source of ambiguity.
- [insertion point]: a missing or implicit detail that is not written as a contiguous phrase in the original question, but should be inserted to clarify the question.

Original question:
{{ORIGINAL_QUESTION}}

Numeric uncertainty scores:
{{UNCERTAINTY_CONTEXT}}

Fine Grained uncertainty scores:
{{Fine_GRAINED_UNCERTAINTY_CONTEXT}}

Rules:
1. Return exactly one rewritten direct wh-question.
2. Preserve the original topic and answer type.
3. Prefer edits around spans with larger span uncertainty scores.
4. Do not answer the question.
5. Do not include a candidate answer in the question unless that text already appeared in the original question.
6. Do not ask a yes/no question and do not ask the model to list multiple alternatives.
7. If the original question is already clear, keep it nearly unchanged.

Figure 18: **Prompt used for the Uncertainty-guided Clarification with ShaQ.**

Uncertainty context

Numeric uncertainty scores:
query_ambiguity: {query_ambiguity} ,
query_ambiguity_stats:
"min": {min} ,
"max": {max} ,
"mean": {mean} ,
"median": {median} ,
"variance": {variance} ,
"threshold": {threshold} ,

Figure 19: **Uncertainty Context format used for the Uncertainty-guided Clarification.**

Uncertainty context

Fine Grained uncertainty scores:
span_ambiguity_stats:
{"min": {`min`},
"max": {`max`},
"mean": {`mean`},
"median": {`median`},
"variance": {`variance`},
"threshold": {`threshold`}}

Span {`Span 1`}: text='{`text`}'; ambiguity= {ϕ_1},
Span {`Span 2`}: text='{`text`}'; ambiguity= {ϕ_2},
Span {`Span 3`}: text='{`text`}'; ambiguity= {ϕ_3},
...

Figure 20: **Fine-grained Uncertainty Context format used for the Uncertainty-guided Clarification with `ShaQ`.**

ShaQ Prompt: Medical Dialogue Localizer

You are a conservative ambiguity detector for medical dialogues. Your task is to review the Current Utterance in light of the Medical Dialogue Context, and wrap only the truly ambiguous spans in the Current Utterance with XML tags of the form `<ambig id="1">...</ambig>`.

Ambiguity Standard
Mark a span as ambiguous ONLY if all of the following are true:

1. The span has at least two DISTINCT, mutually incompatible interpretations.
2. Both interpretations are COMMON and strongly supported by the wording plus context.
3. Choosing between them would materially change the clinical understanding, factual assumption, likely answer, or follow-up question.
4. No single interpretation is clearly dominant to a knowledgeable doctor or patient.

If a span is merely imprecise, approximate, broad, or missing detail, that alone is NOT enough.

MARK as ambiguous only when:

1. Ambiguous referents — a pronoun or vague noun could realistically refer to two different symptoms, body parts, time points, or events already active in context.
2. Competing medical readings — the wording naturally supports two different concrete clinical meanings, not just "more detail is needed."
3. Missing anchor with competing outcomes — omitted timing, target, or state creates two different plausible medical situations that knowledgeable readers would genuinely disagree between.
4. Clinically meaningful missing degree or duration — severity, frequency, amount, or duration is expressed so loosely that two different COMMON clinical readings remain live, and the context does not make one reading clearly dominant.

Do NOT mark as ambiguous:

- Clear statements with one dominant clinical interpretation, even if somewhat informal or approximate.
- Common symptom names with their ordinary meaning, such as "cough", "headache", "sore throat", or "fever", unless the context creates two concrete competing referents.
- Approximate or gradable expressions that still convey one main clinical bucket.
- Routine doctor questions that list possibilities or ask for clarification.
- Phrases that invite follow-up detail but do not create multiple strong current interpretations.
- Cases where one interpretation is overwhelmingly more natural than any alternative.
- Trivial conversational turns ("Yes", "Hello", "Thank you").

Span Selection

- Tag the SMALLEST span that creates the ambiguity.
- Do not tag an entire clause if only one word or short phrase is ambiguous.
- Do not tag text whose only issue is lack of precision, softness, or conversational hedging.

Formatting Rules

- Do not rewrite, paraphrase, reorder, or clean up the Current Utterance.
- The output `tagged_sentence` must match the original Current Utterance EXACTLY except for the added XML tags.
- Preserve every character exactly, INCLUDING the exact capitalization, punctuation, and leading/trailing spaces.
- Never reuse an id. Use consecutive ids starting from 1.
- Ambiguous spans must be non-overlapping.
- Do not create nested or partially overlapping `<ambig>` tags.
- If the utterance is clear, return the original Current Utterance unchanged as `tagged_sentence`.

Examples
`{Examples}`

Task
Medical Dialogue Context: `{context}`
Current Utterance: `{x}`

Figure 21: **Prompt for the Localizer**: identifies ambiguous spans in the current utterance given the dialogue history.

ShaQ Prompt: Medical Dialogue Premise Generator

You are a precise premise generator for ambiguous medical utterances.
You will be given the Medical Dialogue Context, the Current Utterance, the ambiguous span text, and the reason for ambiguity. Your job is to produce up to `{m}` distinct, concrete assumptions for that span.

Core Goal Generate only interpretations that a knowledgeable reader or doctor would reasonably consider from the wording of the utterance and the context, and that would lead to DIFFERENT clinical or factual meanings.
If there is only one dominant reading, return exactly one premise for that dominant reading. If there are fewer than `{m}` strong interpretations, return fewer than `{m}`. Do not add filler.

Rules

1. Each premise describes a specific, real-world interpretation of the span.
2. Each premise must be strongly supported by the context, not merely theoretically possible.
3. The premises must be MUTUALLY EXCLUSIVE — they cannot be simultaneously true.
4. Verify: would a knowledgeable person derive a DIFFERENT medical or semantic meaning under each premise? If not, do not include both.
5. Keep each premise concise (one short sentence).
6. Prefer the MOST COMMON or CLINICALLY RELEVANT interpretations first.
7. Do not use obscure variants, highly unlikely edge cases, or unrelated diagnoses unless the context directly makes that dimension salient.
8. When in doubt, be conservative: return fewer premises rather than low-plausibility ones.

Example
Context: Doctor: "When did the pain start getting worse?"
Current Utterance: "It hurt a little every time I exercised."
Span text: "exercised"
Ambiguity reason: "The type and intensity of the exercise is unspecified and could change the clinical understanding."
Output:

```
{
  "reasoning": "'exercised' could refer to heavy weightlifting or
    light cardio. These have different implications for muscle
    strain and pain.",
  "premises": [
    "\"exercised\" refers to strenuous weightlifting.",
    "\"exercised\" refers to light stretching or walking."
  ]
}
```

Task Medical Dialogue Context: `{context}`
Current Utterance: `{clean_sentence}`
Span id: `{span_id}`
Span text: `{span_text}`
Ambiguity reason: `{reason}`
The `premises` array must contain between 1 and `{m}` items. Add another item only if it is a strong, common, meaning-changing interpretation.
Respond strictly as JSON:

```
{
  "reasoning": "<brief reasoning>",
  "premises": [
    "<premise 1>"
  ]
}
```

Figure 22: **Prompt for the Premise Generator**: produces mutually exclusive clinical interpretations for each ambiguous span.

ShaQ Prompt: Medical Dialogue Answerer

You are a factual semantic specification assistant for medical dialogues.
You are given a Medical Dialogue Context, a Current Utterance, and specific Assumptions regarding an ambiguous span in the Current Utterance. Your task is to re-state or specify the factual meaning of the Current Utterance UNDER THAT SPECIFIC ASSUMPTION.

Rules

1. Your answer MUST be $\leq$ 20 words. Keep it short and factual.
2. STRICTLY follow the assumption. Rewrite the utterance to embed the specific meaning of the assumption into it.
3. DO NOT explain, hedge, or add qualifiers (e.g., skip "Under the assumption...", "This means..."). ONLY output the rewritten meaning.
4. If Assumptions say "None.", answer with the most straightforward interpretation.

Examples
Context: Doctor: "When did the pain start getting worse?"
Current Utterance: "It hurt a little every time I exercised."
Assumptions: "exercised" refers to strenuous weightlifting.
Answer: It hurt a little every time I did heavy weightlifting.

Context: Doctor: "When did the pain start getting worse?"
Current Utterance: "It hurt a little every time I exercised."
Assumptions: "exercised" refers to light stretching or walking.
Answer: It hurt a little every time I went for a light walk or stretched.

Task Medical Dialogue Context: `{context}`
Current Utterance: `{clean_sentence}`
Assumptions: `{premises}`
Answer:

Figure 23: **Prompt for the answerer**: rewrites the current utterance under a specific assumption about an ambiguous span.

ShaQ Prompt: Medical Dialogue Clusterer

You are a semantic clustering assistant for medical dialogues. Your task is to group a list of rewritten utterances (answers) into clusters based on whether they describe the SAME underlying clinical or real-world semantic meaning.

Core Rule Two answers belong to the SAME cluster if and only if they describe the same specific fact, medical state, entity, or semantic intent — regardless of surface wording. Two answers belong to DIFFERENT clusters if they describe different real-world states, types of exercise, types of pain, severities, or different factual outcomes.

Detailed Rules

1. Assign cluster ids as consecutive integers starting from 0.
2. Every answer index must be assigned to exactly one cluster.
3. Refusals ("Unknown", "N/A") go into their own independent cluster.

Task Medical Dialogue Context: `{context}`
Current Utterance: `{question}`
Answers: `{answers_numbered_list}`
Respond strictly in the following JSON format. Do not include any explanation or extra text.

```
{
  "clusters": {
    "0": <cluster_id>,
    "1": <cluster_id>
  }
}
```

Figure 24: **Prompt for the Clusterer**: groups rewritten utterances by shared clinical or semantic meaning.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main contributions: `ShaQ` localizes input-induced uncertainty at the span level using Shapley values, provides a principled decomposition of total input-induced uncertainty, and is evaluated on AmbigQA, AmbiEnt, and MediTOD. These claims are supported by the theoretical formulation in Sections 3, 4 and the empirical results in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations and future directions in Appendix E, including the independence assumption in premise generation and the potential need for conditional premise sampling when span interpretations are semantically coupled.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The theoretical results are stated in Sections 3, 4, including the Shapley formulation, efficiency property, and hierarchical monotonicity. Full derivations and proofs are provided in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides the experimental setup, datasets, baselines, model backbones, evaluation metrics, sampling configurations, and prompts in Section 5 and Appendix B, which together describe how to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper provides access to the code and describes the datasets, baselines, model backbones, prompts, and experimental configurations needed to reproduce the main results. The experiments use existing benchmarks, and the implementation details and prompts are provided in Appendix B.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setting is specified in Section 5 and Appendix B, including datasets, baselines, model backbones, sampling temperatures, number of sampled answers, number of generated premises or clarifications, and evaluation metrics.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: While the paper does not report explicit error bars, confidence intervals, or formal significance tests, it provides robustness evidence by evaluating the main results across multiple benchmarks, model backbones, and metrics. In particular, ShaQ consistently improves or remains competitive across AmbigQA and AmbiEnt, as well as GPT- and Gemini-based backbones, which provides appropriate evidence that the observed gains are not specific to a single experimental setting.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Justification: Appendix D reports the compute resources and API budget used for the experiments. Since all LLM inference was performed through OpenRouter API-based inference, we report the reproducibility-relevant API budget rather than local GPU hours. Appendix D includes approximate OpenRouter chat-completion request counts and observed wall-clock times for the main AmbigQA, AmbiEnt, Gemini backbone ablation, and clarification simulation experiments, and states that MediTOD was used as a qualitative case study. It also specifies that no models were run locally: local compute was used only for CPU-based orchestration, parsing, aggregation, and metric computation on a 64-logical-core machine with approximately 1 TiB RAM.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: We have reviewed the NeurIPS Code of Ethics and believe the research conforms to it. The work uses public benchmarks and simulated medical dialogues for uncertainty quantification, and does not involve collecting private user data or deploying a system to real users.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Section G discusses positive impacts such as improved transparency, targeted clarification, and reliability in high-stakes domains, as well as potential negative impacts from incorrect localization misleading users. It further recommends human oversight and domain-specific evaluation as safeguards before deployment.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a high-risk generative model, scraped dataset, or pre-trained model that would require controlled release safeguards. The method is an uncertainty localization framework evaluated on existing benchmarks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original creators of all datasets, baselines, and model APIs used in the paper, including AmbigQA, AmbiEnt, MediTOD, ICE, and other uncertainty baselines. Appendix H lists the relevant licenses and terms of use for these assets, states that the datasets are used only for research evaluation, and clarifies that we do not redistribute third-party datasets or proprietary model weights. For proprietary LLMs, we access GPT-based and Gemini-based models through OpenRouter and comply with the applicable OpenRouter terms of service and the terms and usage policies of the underlying model providers.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce a new dataset or pretrained model. If code is released, it will be documented with usage instructions and reproduction scripts in the supplemental material.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not conduct new crowdsourcing experiments or new research with human subjects. Experiments are conducted on existing benchmarks and simulated medical dialogue data.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve newly collected human-subject data or new interactions with study participants. It uses existing public benchmarks and simulated dialogue data.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs are central components of the proposed method: the Localizer, Generator, Answerer, and Clusterer modules are implemented with LLMs, and the paper describes their roles, model backbones, prompts, and configurations in Section 4 and Appendix B.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.