# Decomposed Prompting Does Not Fix Knowledge Gaps, But Helps Models Say *"I Don't Know"*

**Dhruv Madhwal**[1,*] **Lyuxin David Zhang**[2,*] **Dan Roth**[2,3]
**Tomer Wolfson**[2,†] **Vivek Gupta**[1,†]
[1]Arizona State University [2]University of Pennsylvania [3]Oracle AI
{dmadhwal, vgupt140}@asu.edu, {davidzlx, danroth, wolfsont}@upenn.edu

## Abstract

Large language models often struggle to recognize their knowledge limits in closed-book question answering, leading to confident hallucinations. While decomposed prompting is typically used to improve accuracy, we investigate its impact on reliability. We evaluate three task-equivalent prompting regimes: Direct, Assistive, and Incremental, across different model scales and multi-hop QA benchmarks. We find that although accuracy gains from decomposition diminish in frontier models, disagreements between prompting regimes remain highly indicative of potential errors. Because factual knowledge is typically stable while hallucinations are stochastic, cross-regime agreement provides a precise signal of internal uncertainty. We leverage this signal to implement a training-free abstention policy that requires no retrieval or fine-tuning. Our results show that disagreement-based abstention outperforms standard uncertainty baselines as an error detector, improving both F1 and AUROC across settings. This demonstrates that decomposition-based prompting can serve as a practical diagnostic probe for model reliability in closed-book QA. All code, data and prompts are available at: https://github.com/dhruvmadhwal/disagreement-based-abstention.

## 1 Introduction

Large language models (LLMs) are increasingly deployed in settings where they must answer factual questions without access to external retrieval or verification, such as privacy-restricted systems, on-device applications, and time-critical decision-making pipelines (Wang et al., 2025; Urlana et al., 2025; Qu et al., 2025). In these closed-book environments, models must rely entirely on their internal knowledge and reasoning capabilities. This makes closed-book factual question answering particularly challenging: when a model lacks sufficient knowledge it cannot defer to evidence and instead risks producing confident but incorrect answers (Simhi et al., 2025). As a result, a reliable closed-book QA model should not merely have high factual accuracy but should also be able to recognize uncertainty and appropriately refrain from answering when the requisite knowledge is unknown.

A widely adopted way to improve closed-book QA is to change how models are prompted to reason. *Direct* reasoning asks the model to generate an answer in one step without an explicit intermediate structure, relying on the model's internal reasoning. Question decomposition (Min et al., 2019; Wolfson et al., 2020; Khot et al., 2023) explicitly structures the solution process before producing an answer. Two decomposition variants are *Assistive*, where a full set of intermediate steps is generated and then used to answer, and *Incremental*, where the model is guided through a sequence of subquestions and answers them one step at a time. Crucially, these decompositions change only the reasoning path and do not introduce new facts or supervision, which makes them a controlled lens for studying accuracy and reliability in closed-book QA.

Our study spans across six multi-hop QA benchmarks and nine LLMs at varying scales. Its results reveal a striking, scale-dependent shift in how question decomposition functions. For smaller and mid-scale models, decomposition acts primarily as a scaffold that improves accuracy by providing structure. However, for frontier LLMs, these accuracy gains diminish, while the diagnostic value of the prompt increases. We find that for large-scale models, a disagreement between direct and decomposed outputs is a highly precise signal of an underlying error. In effect, as models scale, decomposition evolves from a tool for improving performance into a powerful training-free auditor for identifying fragile beliefs.

[*]Equal contribution.
[†]Equal advising.



*Findings of the Association for Computational Linguistics: ACL 2026*, pages 36688–36710

Based on these insights, we introduce Disagreement-Based Abstention (DBA), a simple and effective reliability method. DBA compares a model's Direct answer to an answer obtained through a task-equivalent decomposition and triggers an *"I don't know"* response whenever the two final answers are conflicting. Unlike other LLM uncertainty methods, DBA requires no additional training or external retrieval and relies on the model's exhibited behavior rather than self-reported confidence scores. Across nine LLMs, this approach outperforms standard uncertainty baselines, improving both F1 and AUROC and flagging confident-but-incorrect answers that other calibration methods often miss.

In summary, our contributions are: (a) a comprehensive empirical analysis of how factual accuracy and cross-prompt consistency jointly vary across LLM scales, evaluation tasks, and prompting methods in closed-book multi-hop QA, comparing direct prompting with Assistive and Incremental decomposed prompting; and (b) *Disagreement-Based Abstention* (DBA), a simple training-free abstention method that treats cross-prompt disagreement between direct and decomposed prompting as a reliability signal for deciding when closed-book answers should be trusted.

## 2 Decomposition-based Prompting

We evaluate closed-book multi-hop QA across three prompting regimes that share a fixed gold decomposition for each question: **(a) Direct:** where the model answers the query directly without intermediate structure or explicit decomposition; **(b) Assistive:** the model receives the full sequence of sub-questions as its fixed context and generates all intermediate answers in a single call; and **(c) Incremental:** where the decomposition is executed through separate model calls, with sub-questions being answered one at a time and follow-up questions are derived using the answers from previous steps. In the incremental approach, each LLM call is focused solely on the current sub-question, without being distracted by the previous computation history.

For each question, we measure: (1) correctness relative to the ground truth and (2) consistency, defined as semantic agreement between the Direct answer and the final answer produced by each decomposed prompting regime.

**Why these prompting regimes?** Direct provides the baseline for closed-book multi-hop QA, and is a conventional single-step answer prompt in which question decomposition is done implicitly by the LLM. The two decomposed prompting regimes, Assistive and Incremental, capture the most common decomposition strategies used in the literature. Assistive mirrors single-prompt, full-decomposition approaches that present an explicit plan or chain of intermediate steps up front (Press et al., 2023; Radhakrishnan et al., 2023; Wu et al., 2024), while Incremental mirrors step-wise or least-to-most execution that decomposes the problem into a sequence of sub-questions answered one at a time (Zhou et al., 2023; Khot et al., 2023). By having a fixed gold-standard question decomposition, we ensure that the semantics of the Assistive and Incremental approaches are identical to that of the Direct question. This controlled comparison enables us to evaluate how the execution style, whether single-call (Direct), single-call with explicit plan (Assistive), or iterative QA (Incremental), affects both accuracy and answer consistency.

**Verified Reference Decompositions** A key requirement of our study is a fixed, gold-standard decomposition for each question. Using the syntax outlined in Wolfson et al. (2026), we represent the question decomposition using a domain-specific language (DSL) that specifies variables, answer types, and sub-questions (Figure 1). This representation gives us a precise, model-agnostic plan that can be reused across our prompting regimes.

We generate DSL decompositions using a two-stage pipeline with a strong LLM for decomposition. For benchmarks that provide step-by-step annotations, we prompt the LLM to translate the original steps into DSL. For benchmarks that lack labeled decompositions, we prompt `Gemini-2.5-Flash` to synthesize a DSL program from the original question. We then manually verify every DSL decomposition to ensure it is semantically and syntactically accurate, thereby eliminating any planning errors as a confounding factor.

The structured decomposition has three roles in our experimental design. First, decomposition steps can refer to the answers of previous steps using placeholders, thereby outlining the solution plan without the need to explicitly generate all intermediate answers in advance. Second, it promotes deterministic execution by defining expected answer types (for example, integers or lists), which

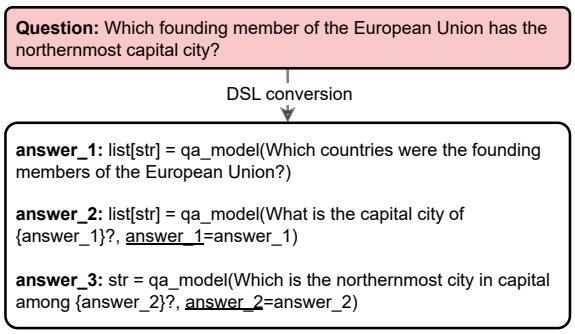



Figure 1: DSL decomposition for a multi-hop question.

keeps intermediate outputs in a consistent format for subsequent steps. Third, it guarantees a semantic equivalence between the decomposition and the original query, such that if a model answers all sub-questions correctly, its final answer must be identical to the answer of the original query. As a result, Direct, Assistive and Incremental differ only in how they execute the same plan. Cross-regime answer disagreements can be interpreted as errors in factual knowledge or execution rather than an erroneous plan (Sinha et al., 2025).

## 3 Experiments

**Models.** To study how reasoning consistency behaves across different models at varying scales, we evaluate nine instruction-tuned LLMs spanning three distinct parameter sizes alongside a suite of state-of-the-art frontier systems. At the smaller scale (≈8B parameters), we employ `Mistral-7B-Instruct-v0.3` (Jiang et al., 2023), `Llama-3.1-8B-Instruct` (Grattafiori et al., 2024), and `Qwen3-8B` (Yang et al., 2025a). Our medium-scale representative is `Qwen3-32B` (Yang et al., 2025a), while for the large-scale tier (≈70B parameters), we use `Llama-3.3-70B-Instruct` (Grattafiori et al., 2024) and `Qwen2.5-72B-Instruct` (Yang et al., 2025b). Furthermore, we include three closed-source frontier models: `GPT-5.1` (OpenAI, 2025), `Gemini-2.5-Pro`, and `Gemini-2.5-Flash` (Comanici et al., 2025). All models are evaluated in their released instruction-tuned form using fixed decoding settings to reduce output variance. For all models except `GPT-5.1`, we use greedy decoding ($T = 0$). For `GPT-5.1`, temperature is not exposed, so we use the API default with the reasoning effort set to 'medium'.

**Datasets.** We evaluate on six multi-hop QA datasets: *Bamboogle* (Press et al., 2023), *FRAMES* (El Asri et al., 2017), *MuSiQue* (Trivedi et al., 2022), *CRAG* (Yang et al., 2024), *HotpotQA* (Yang et al., 2018), and *Mintaka* (Sen et al., 2022). We filter questions to avoid ambiguity and to ensure their multi-hop complexity, temporal-independence, and semantic clarity—resulting in 1,433 verified instances. Our full filtering statistics and benchmark splits are reported in Appendix B.

**Prompting Regimes.** Direct, Assistive, and Incremental all share the same closed-book multi-hop QA instruction and decoding settings. Direct is prompted using only instructions, while Assistive adds the gold DSL program, a small set of dataset-specific few-shot examples, and the model is instructed to answer all sub-questions in one response. Incremental executes the DSL line by line: at step $k$, we fill the template for sub-question $Q_k$ with answers from previous steps (e.g., "In what city was [answer_1] born?" → "In what city was Albert Einstein born?") and issue it as a new query. Each hop is then posed as an isolated factual lookup, without access to the original top-level question or to any of the past or future steps.

**Consistency Protocol.** We measure factual consistency by anchoring to the Direct response, which represents the model's zero-shot answer based on its parametric knowledge. We compute two pairwise comparisons: Direct vs. Assistive and Direct vs. Incremental. An *internal disagreement* is recorded whenever the answer to the decomposed approach is not semantically equivalent to the Direct answer. This definition is independent of the actual correctness of either answer.

**Evaluation Measures.** We evaluate each prompting regime along two axes: accuracy, defined as agreement with the gold answer, and consistency, defined as semantic agreement across regimes. To address the brittleness of lexical measures such as EM or ROUGE, we employ an LLM-as-judge protocol, using `Gemini-2.5-Flash`, to assess semantic equivalence. The judge follows a fixed rubric described in Appendix C. Namely, the rubric: (i) normalizes surface variation, including units, aliases, and abbreviations; (ii) enforces numeric tolerance and exact date matching; and (iii) penalizes explicit contradictions. **Accuracy** is computed by comparing a regime's final answer to the gold reference. **Consistency** is computed by comparing outputs from two regimes, such as Direct and Incremental, to determine whether they express the same underlying claim, independent of factual correctness.

### 3.1 Results and Analysis

We analyze how LLM scale and prompting regime affect factual correctness and cross-regime consistency. We identify three consistent patterns across benchmarks and LLMs. First, both accuracy and consistency increase with model scale, though non-trivial inconsistency persists even in frontier models. Second, decomposed prompting substantially improves accuracy for non-frontier models, but yields diminishing or negative returns for frontier LLMs. Third, agreement between prompting regimes is highly indicative of accuracy, highlighting consistency as a signal for answer confidence.

**Scale, Accuracy and Consistency.** Both accuracy and cross-regime consistency increase with model scale. Larger models are more likely to produce correct answers while maintaining agreement across the Direct, Assistive, and Incremental regimes, as shown in Figure 2. The figure illustrates a clear shift towards higher accuracy and consistency, as the LLM size increases.

However, significant inconsistency persists even among frontier models. Our most accurate LLMs still exhibit measurable cross-regime disagreement, for example, consistency on the MuSiQue benchmark reaches only 59.7%. A comprehensive breakdown of accuracy and consistency across all models and datasets is provided in Appendices A.1-A.2. This result reveals a fundamental failure of logical invariance: in a knowledge-grounded setting, semantically-equivalent queries should yield identical outputs regardless of prompting regime. However, while all prompting regimes answer either the original question or a semantically equivalent decomposition, the lack of LLM consistency highlights failures in their execution or underlying parametric knowledge (Sinha et al., 2025).

**Decomposition Gains Plateau in Frontier LLMs.** Our results show a sharp scaling breakpoint in the efficacy of decomposed prompting (Table 1). For non-frontier models ($\leq$ 70B), decomposition serves as a vital reasoning scaffold, leading to substantial accuracy gains. A striking example is Qwen-72B, where question decomposition leads to a $+26.8\%$ improvement on BAMBOOGLE. These persistent, double-digit improvements ($\Delta A, \Delta I$) suggest that even high-capacity non-frontier models still benefit from explicit decomposition structure to solve complex multi-hop tasks.

However, this advantage vanishes for frontier LLMs. For Gemini-2.5 and GPT-5.1, the gains from decomposition plateau, falling to near-parity or even slipping into negative returns. We hypothesize that this transition reflects a "ceiling effect" where frontier models have internalized the necessary reasoning chains. In such models, explicit scaffolding, in the form of decomposed prompting, may no longer enhance, and could occasionally degrade performance.

**Consistency as an Answer Reliability Signal.** Decomposed prompting delivers large accuracy gains for non-frontier models but offers little gain for frontier LLMs (Table 1). This raises a natural question: if decomposition no longer helps frontier models answer questions more accurately, what is it useful for? Our analysis suggests that its primary utility shifts from *intervention* to *inspection*: cross-prompt consistency becomes a direct signal of how consistent the model's beliefs are.

**Accuracy-Consistency Correlation.** LLM accuracy and cross-regime consistency are closely tied across models and datasets, with Pearson correlation coefficients reaching up to $r = 0.98$, as shown in Figure 2. This tight coupling motivates a quantitative measure of how strongly consistency helps identify correct answers. We define the **Reliability Multiplier** $RM = \frac{\text{correct} \wedge \text{consistent}}{\text{correct} \wedge \text{inconsistent}}$, where correctness is anchored to the Direct answer. In practice, an $RM$ value of $k$ means that correct Direct answers occur $k$ times as often among consistent cases as among inconsistent cases.

**Reliability across LLM Scales.** The Reliability Multiplier ($RM$) varies sharply with model scale, reflecting a transition from noisy, stochastic behavior to more stable agreement across prompting regimes (Table 2). Smaller models ($\approx$8B) often exhibit near one or inverted $RM$ values, consistent with a floor effect in which correct answers are too sparse for $RM$ to provide a stable signal. When correctness is sparse, $RM$ is statistically fragile. This pattern is reflected in the near-one or inverted RM values for smaller models in Table 2. As model capacity increases into the 30B to 70B range, the multiplier grows steadily, typically reaching values between $2\times$ and $10\times$. At frontier scale, the effect can become much larger, with values exceeding $50\times$ for Gemini Pro in our measurements. These patterns indicate that, as scale increases, correct Direct answers increasingly concentrate in the subset of cases where regimes agree. As a result,

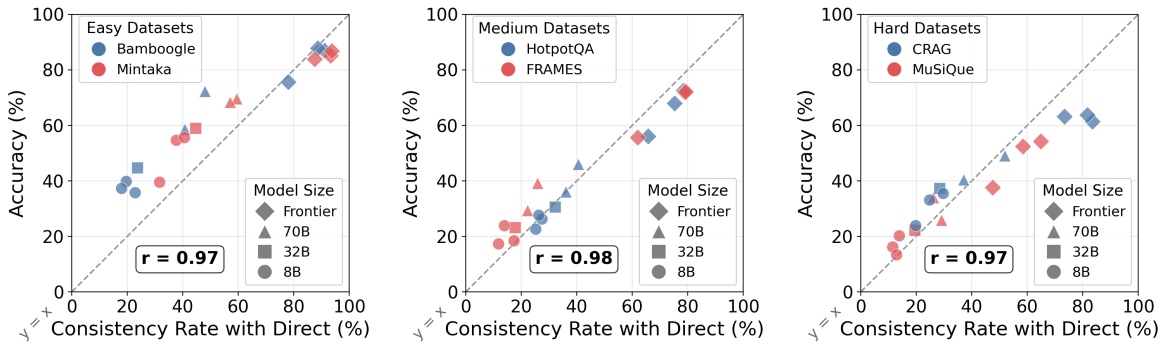



Figure 2: Accuracy vs. consistency rate across 9 models and 6 datasets, grouped by difficulty. Each point represents a (model, dataset) pair. Marker shape encodes LLM size (Frontier, 70B, 32B, 8B) while colors encode different evaluation datasets.

| Model | Bamboogle | | | Mintaka | | | HotpotQA | | | CRAG | | | FRAMES | | | MuSiQue | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Dir | ΔA | ΔI | Dir | ΔA | ΔI | Dir | ΔA | ΔI | Dir | ΔA | ΔI | Dir | ΔA | ΔI | Dir | ΔA | ΔI |
| Mistral 7B | 15.6 | +20.2 | +21.0 | 40.9 | +14.8 | +9.7 | 24.1 | +3.6 | +0.4 | 25.2 | +8.0 | −0.6 | 11.9 | +6.5 | +1.7 | 12.8 | +7.5 | +0.7 |
| Qwen 8B | 12.2 | +25.2 | +20.6 | 24.6 | +15.0 | +9.3 | 21.7 | +0.9 | −5.3 | 19.6 | +4.3 | +1.8 | 11.6 | +12.3 | +13.4 | 11.7 | +1.8 | +3.6 |
| Llama 8B | 21.3 | +18.5 | +26.7 | 33.3 | +21.4 | +24.7 | 25.3 | +1.0 | +2.2 | 27.6 | +8.0 | +14.1 | 8.9 | +8.5 | +9.9 | 9.2 | +7.0 | +7.2 |
| Qwen 32B | 18.7 | +26.0 | +25.2 | 38.6 | +20.5 | +14.8 | 31.4 | −0.7 | −3.7 | 27.0 | +10.4 | +8.6 | 14.3 | +8.9 | +7.2 | 16.0 | +6.2 | +3.9 |
| Qwen 72B | 31.7 | +26.8 | +29.3 | 53.7 | +14.6 | +18.1 | 34.9 | +1.2 | +1.2 | 31.9 | +8.6 | +9.2 | 17.4 | +11.9 | +11.6 | 23.0 | +2.8 | +3.2 |
| Llama 70B | 45.5 | +26.8 | +22.8 | 61.6 | +8.0 | +12.7 | 41.3 | +4.6 | +1.5 | 41.7 | +7.4 | +10.4 | 26.6 | +12.6 | +14.0 | 24.5 | +9.6 | +6.7 |
| Gemini Flash | 83.7 | −8.1 | −3.2 | 83.2 | +0.7 | −0.4 | 59.7 | −3.7 | −6.4 | 64.4 | −1.2 | −2.5 | 61.4 | −5.8 | −5.8 | 37.2 | +0.4 | −1.4 |
| Gemini Pro | 85.4 | +1.6 | +1.5 | 85.6 | −0.3 | −2.3 | 68.1 | −0.1 | −4.3 | 64.4 | −3.1 | −4.9 | 71.7 | +0.3 | −6.8 | 47.5 | +5.0 | −2.0 |
| GPT 5.1 | 84.6 | +3.2 | +2.4 | 86.2 | +0.7 | −4.0 | 73.7 | −1.2 | −3.0 | 65.0 | −1.2 | −3.1 | 72.4 | −0.3 | −6.5 | 53.9 | +0.4 | −7.8 |

Table 1: Direct accuracy (Dir, %) and accuracy change from Direct to Assistive (ΔA) and Incremental (ΔI). Non-frontier models (≤70B) benefit substantially from decomposition; frontier models show diminishing or negative returns.

cross-regime agreement becomes a progressively stronger accuracy indicator.

**Utility Shift.** These results highlight question decomposition as a dual-purpose tool, whose primary role shifts with LLM scale. For weaker models, decomposition functions primarily as a generative aid that improves accuracy. For frontier models, where these gains saturate, decomposition serves as a diagnostic auditor: cross-regime agreement becomes a reliable signal for accuracy. While this signal is strongest for frontier LLMs, it remains useful across the full range of model sizes, with reliability multipliers greater than one in most settings.

## 4 Selective Abstention Framework

Our analysis shows that while Assistive and Incremental decompositions may not fundamentally expand a model's internal knowledge, *agreement* across different decomposed prompting regimes is a strong indicator of factual correctness. We exploit this cross-regime consistency to build a multi-prompt-based abstention policy. When different prompting regimes yield inconsistent final answers, we treat the instance as unreliable and output an "I don't know" response. Unlike other uncertainty prompting methods that rely on potentially unfaithful self-reported calibration, our approach uses answer inconsistency across decomposed prompting regimes as a direct signal of factual error.

### 4.1 LLM Abstention on Factual Questions

**Disagreement-based Abstention (DBA).** Given a question, we prompt an LLM twice, once using the Direct regime and once using a task-equivalent decomposed regime. If the two final answers are semantically distinct, the model abstains. DBA instantiates this procedure using either an Assistive or an Incremental decomposition, referred to as DBA-A and DBA-I, respectively. Semantic equivalence is determined using the same LLM-as-judge protocol described in Section 3. The overall DBA pipeline is shown in Figure 3.

**Evaluation Metrics** We evaluate LLM abstention policies as *error detectors* where the positive class is **incorrect** Direct predictions. We treat the model's Direct response as the primary claim-of-interest. Namely, correctness is defined by whether this Direct claim matches the ground-truth answer, and the Assistive/Incremental executions serve only as diagnostic signals for deciding whether to accept the Direct answer, or to abstain. This is in line with prior work (our abstention baselines below) which also focus on the model's direct answer.

We report four measures: **(a) Precision**, the fraction of rejected (abstained) instances whose *Direct* answer is incorrect; **(b) Recall**, the fraction of all incorrect *Direct* answers that are rejected; **(c) F1**,

| Model | Bamboogle | | Mintaka | | HotpotQA | | CRAG | | FRAMES | | MuSiQue | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | I | A | I | A | I | A | I | A | I | A | I |
| Mistral 7B | 1.3x | 1.6x | 2.0x | 1.4x | 1.3x | 0.7x | 0.8x | 0.5x | 1.1x | 0.5x | 1.1x | 0.8x |
| Qwen 8B | 4.0x | 6.5x | 2.2x | 2.1x | 2.0x | 1.3x | 1.1x | 0.9x | 1.3x | 1.0x | 0.6x | 0.7x |
| Llama 8B | 2.2x | 3.3x | 3.5x | 3.8x | 1.8x | 1.9x | 1.5x | 1.6x | 1.0x | 0.7x | 1.6x | 0.7x |
| Qwen 32B | 3.6x | 2.8x | 5.5x | 4.1x | 2.2x | 2.2x | 2.1x | 2.1x | 1.3x | 1.0x | 2.0x | 1.4x |
| Qwen 72B | 8.8x | 12.0x | 6.2x | 7.0x | 2.0x | 2.0x | 4.8x | 4.2x | 1.4x | 2.2x | 2.2x | 1.2x |
| Llama 70B | 10.2x | 10.2x | 4.7x | 10.4x | 3.3x | 2.5x | 2.6x | 3.7x | 2.0x | 1.8x | 1.7x | 1.5x |
| Gemini Flash | 5.4x | 8.4x | 14.6x | 13.6x | 4.9x | 2.6x | 6.1x | 5.2x | 4.7x | 3.7x | 3.6x | 2.6x |
| Gemini Pro | 34.0x | 16.5x | 50.2x | 27.4x | 13.3x | 5.4x | 10.8x | 10.8x | 10.1x | 5.0x | 5.7x | 2.8x |
| GPT 5.1 | 20.2x | 12.2x | 42.0x | 27.7x | 9.4x | 6.0x | 12.4x | 10.9x | 9.5x | 5.3x | 5.3x | 3.2x |

Table 2: **The Reliability Multiplier.** ratio of (consistent, correct) to (inconsistent, correct) Direct answers for each dataset and decomposed prompting regime (A = Assistive, I = Incremental).

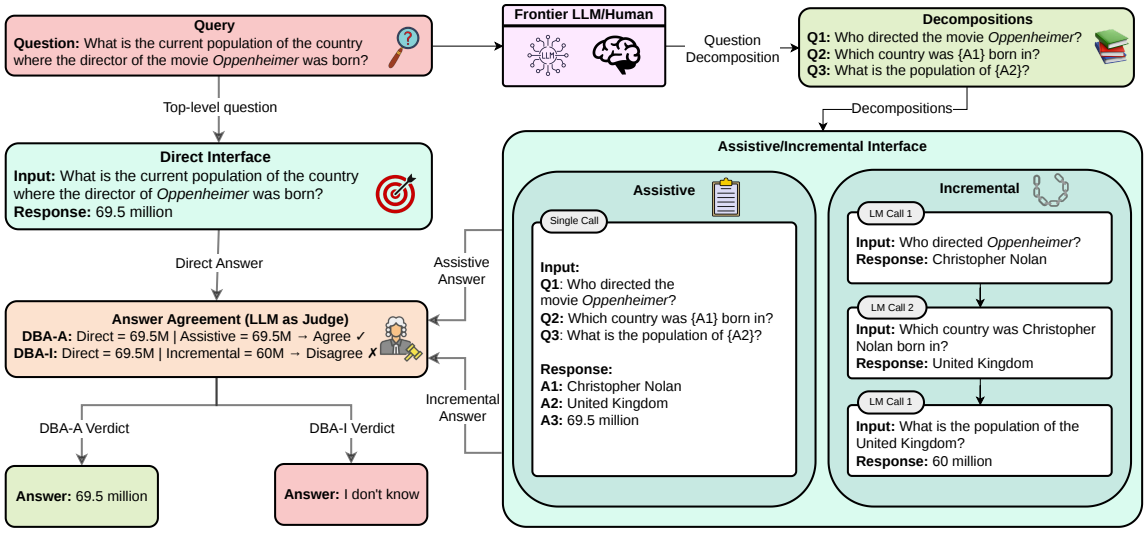



Figure 3: **Disagreement-Based Abstention (DBA) Framework.** Our method compares a Direct answer against Assistive/Incremental reasoning paths, if the semantic claims disagree, the model abstains (IDK).

the harmonic mean of precision and recall; and **(d) AUROC**, which measures the method's ability to separate incorrect from correct *Direct* answers.

While precision, recall, and F1 are sensitive to the base error rate, which varies widely across model scales (e.g., ∼10% to ∼90% on *Bamboogle*), AUROC is less sensitive. In our binary setting, AUROC corresponds to the probability that an incorrect *Direct* answer is rejected more strongly than a correct one, with a naive always-accept or always-reject strategy yielding AUROC $= 0.50$.

**Baselines.** We compare DBA against three uncertainty-detection baselines. The first two (AYS, IC-IDK) are drawn from prior work (Cohen et al., 2023). Our third baseline is self-consistency, a popular prompting approach (Wang et al., 2023).

- **AYS (Are You Sure?)** first produces a Direct answer, then issues a binary follow-up query asking the model to verify its answer (*"Are you sure regarding the correctness of <answer>?"*). The Direct answer is accepted if the model responds affirmatively and otherwise rejected.
- **IC-IDK (In-Context IDK)** appends an instruction to the Direct prompt that allows the model to respond with an explicit "I don't know" when uncertain (Cohen et al., 2023). For each model and dataset, we prepend $K = 15$ demonstrations constructed from a held-out pool disjoint from the evaluation set. To ensure these demonstrations reflect each model's specific knowledge gaps, we select $D = 4$ instances where that same model's Direct answer was verified incorrect against the ground truth as IDK demonstrations, and 11 instances where it answered correctly as standard answer demonstrations.
- **Self-consistency** samples multiple responses from the Direct regime at higher temperatures and predicts by majority vote (Wang et al., 2023). In our experiments, a heuristic majority vote self-consistency achieves high precision but critically low recall, which results in poor F1 performance. We therefore report its detailed analysis in Appendix D.2 and focus our main comparison on the remaining baselines.

**Ensembles with DBA.** We define ensemble variants that combine AYS with DBA. ENSEMBLE-A abstains if either AYS or DBA-A abstains. ENSEMBLE-I is defined analogously using DBA-I.

| Model | Method | Bamboogle | | CRAG | | FRAMES | | HotpotQA | | Mintaka | | MuSiQue | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | F1 | AUC | F1 | AUC | F1 | AUC | F1 | AUC | F1 | AUC | F1 | AUC |
| GPT 5.1 | AYS | 0.47 | 0.66 | 0.66 | 0.74 | 0.51 | 0.67 | 0.51 | 0.67 | 0.42 | 0.66 | 0.38 | 0.61 |
| | IC-IDK | 0.43 | 0.64 | 0.60 | 0.70 | 0.24 | 0.56 | 0.58 | 0.71 | 0.31 | 0.60 | 0.27 | 0.57 |
| | DBA-A | **0.73** | **0.81** | 0.53 | 0.67 | 0.58 | 0.71 | 0.70 | 0.79 | 0.43 | 0.65 | 0.66 | 0.72 |
| | DBA-I | 0.56 | 0.72 | 0.49 | 0.65 | **0.66** | **0.77** | 0.64 | 0.75 | 0.48 | 0.68 | 0.64 | 0.68 |
| | Ensemble-A | 0.67 | 0.79 | **0.75** | **0.81** | 0.60 | 0.73 | **0.73** | **0.81** | 0.46 | 0.70 | **0.69** | **0.73** |
| | Ensemble-I | 0.59 | 0.74 | 0.74 | 0.80 | 0.64 | **0.77** | 0.68 | 0.78 | **0.49** | **0.73** | 0.68 | 0.71 |
| Llama 3.3 70B | AYS | 0.51 | 0.63 | 0.53 | 0.64 | 0.72 | 0.68 | 0.53 | 0.62 | 0.49 | 0.64 | 0.70 | 0.69 |
| | IC-IDK | 0.48 | 0.60 | 0.75 | 0.67 | 0.13 | 0.53 | 0.74 | 0.71 | 0.62 | 0.68 | 0.22 | 0.55 |
| | DBA-A | **0.92** | **0.91** | 0.71 | 0.71 | 0.89 | **0.78** | **0.84** | **0.81** | 0.77 | 0.81 | 0.86 | **0.74** |
| | DBA-I | 0.84 | 0.84 | 0.74 | **0.76** | 0.89 | 0.77 | 0.81 | 0.78 | **0.79** | **0.83** | 0.84 | 0.71 |
| | Ensemble-A | 0.89 | 0.87 | 0.74 | 0.70 | **0.90** | 0.75 | **0.84** | 0.78 | 0.77 | 0.82 | **0.88** | 0.72 |
| | Ensemble-I | 0.85 | 0.84 | **0.77** | 0.74 | **0.90** | 0.74 | 0.81 | 0.75 | 0.78 | 0.82 | **0.88** | 0.70 |
| Qwen 8B | AYS | 0.73 | 0.66 | 0.75 | 0.68 | 0.88 | 0.71 | 0.76 | 0.71 | 0.70 | 0.68 | 0.79 | 0.64 |
| | IC-IDK | 0.59 | 0.62 | 0.76 | 0.61 | 0.44 | 0.57 | 0.62 | 0.65 | 0.44 | 0.53 | 0.54 | 0.62 |
| | DBA-A | 0.93 | **0.82** | 0.89 | **0.73** | 0.93 | **0.74** | 0.87 | **0.76** | 0.87 | **0.79** | 0.91 | **0.65** |
| | DBA-I | 0.80 | 0.80 | 0.89 | **0.73** | 0.94 | 0.73 | 0.84 | 0.70 | 0.81 | 0.74 | 0.89 | **0.65** |
| | Ensemble-A | **0.94** | 0.80 | **0.91** | 0.67 | 0.94 | 0.67 | **0.91** | 0.75 | **0.88** | 0.75 | **0.93** | 0.62 |
| | Ensemble-I | 0.90 | 0.76 | **0.91** | 0.70 | **0.95** | 0.69 | 0.89 | 0.70 | 0.85 | 0.69 | 0.91 | 0.60 |

Table 3: Per-model error-detection performance (F1 / AUROC). Prompting baselines (AYS, IC-IDK), DBA, and ensembles (AYS ∪ DBA). Within each model block and for each dataset/metric, **bold** indicates the best overall value (ties included) and underlined indicates the best non-ensemble value among AYS/IC-IDK/DBA-A/DBA-I (ties included). If a non-ensemble method is also best overall, it is both bold and underlined. AUROC is reported as AUC.

### 4.2 Results and Analysis

Table 3 presents error detection results for three representative models spanning distinct parameter scales: GPT-5.1 (Frontier), Llama-3.3-70B (Large), and Qwen3-8B (Small). We observe similar trends across the remaining models reported in Table 9. The results in both tables clearly demonstrate the efficacy of DBA, showing its superior performance across nine distinct LLMs and six QA benchmarks.

DBA outperforms the AYS baseline (in terms of F1) in 17 of the 18 evaluated model-dataset pairs. For the GPT-5.1 frontier LLM, DBA-A yields substantial gains on datasets where base consistency is moderate-to-high. We observe F1 improvements ranging from $+18$ to $+28$ points across *Bamboogle*, *HotpotQA*, and *MuSiQue*. For Qwen3-8B, DBA delivers strong error detection performance across datasets, with F1 typically at or above $0.80$. For Llama-3.3-70B, DBA exceeds $0.80$ F1 on most datasets. In all cases, AUROC remains above random, ranging from the mid-$0.60s$ to low-$0.90s$. This demonstrates that the success of DBA does not merely stem from being more abstention-prone, showing that it effectively separates incorrect answers from correct ones.

**DBA Improves Performance through Recall.** Results in Table 8 and Table 9 show that the main driver of DBA's F1 gains is a substantial increase in the recall of incorrect Direct answers. Standard uncertainty baselines such as AYS often exhibit overconfidence when the model hallucinates, which leads to low recall and missed errors. DBA mitigates this failure mode by treating cross-regime disagreement as a hard veto. For open-source models and more challenging datasets, this strategy typically increases recall by +30 to +50 points compared to AYS. For example, on Bamboogle, recall for Llama-3.3-70B improves from 37 to 90.

We also observe clear boundary conditions for frontier models such as GPT-5.1. When base accuracy is high, as on *Mintaka*, errors are rare and disagreement signals are correspondingly sparse, which allows AYS to remain competitive. Conversely, on complex datasets with lower accuracy, such as *CRAG*, frontier models can produce stable but incorrect predictions across regimes. In these cases, DBA recall decreases because consistent errors do not generate disagreement signals that the method can exploit.

**Complementary Failure Modes and Method Ensembling.** To address cases where errors are consistent, we leverage an ensembling approach. This strategy combines the strengths of both paradigms: AYS detects stable-but-uncertain errors, while DBA flags confident-but-fragile hallucinations. The complementary nature of both approaches is most visible on GPT-5.1 when evaluated on *CRAG*: while AYS outperforms DBA-A individually (66 vs. 53 F1), their union achieves the highest overall performance (75 F1). This trend generalizes to other datasets like *HotpotQA*, where ensembling boosts F1 to 73.

### 4.3 Efficiency and Deployment Overhead

We evaluate the computational requirements of each method using call volume and token multipliers, averaged across all models and datasets (Table 4). IC-IDK requires a single model call,

| Method | LM Calls | Total Tokens | Total× |
|---|---|---|---|
| Direct | 1 | 833 | 1.00 |
| AYS | 2 | 1,054 | 1.26 |
| IC-IDK | 1 | 963 | 1.16 |
| DBA-A | 4 | 4,702 | 5.64 |
| DBA-I | 6.4 | 4,912 | 5.89 |
| Self-Consistency | 8 | 5,420 | 6.51 |

Table 4: Comparison of computational cost across abstention methods. We report the number of model calls and total token usage, along with relative cost (×) normalized to Direct prompting.

while AYS adds one verification turn for a total of two calls. IC-IDK's cost is also influenced by the length of in-context demonstrations, which vary across datasets.

Among the multi-call approaches, DBA-A and Self-Consistency (SC) allocate compute differently. DBA-A requires 4 calls and $5.64\times$ the total tokens of Direct prompting, whereas SC requires 8 calls and $6.51\times$. Notably, DBA-A generates $38\%$ fewer output tokens than SC, reflecting its focus on structural verification rather than producing multiple full-length candidate answers. Detailed token usage statistics, including a breakdown of input versus output overhead, are in Appendix A.6.

While DBA-I issues one call per decomposition step, each hop carries only a small context, so its total token cost is only marginally higher than DBA-A despite the larger number of calls, and downstream performance is comparable. Overall, moving from lower-overhead methods such as AYS to DBA-A increases total token usage from $1.26\times$ to $5.64\times$ relative to Direct prompting, highlighting the additional cost of multi-call reliability mechanisms. However, this resource trade-off yields the significant gains in abstention accuracy and reliability metrics reported in Section 4.2.

## 5 Discussion

Our results reveal a persistent discrepancy between decomposed prompting regimes and direct question answering. If an LLM's reasoning were indeed consistent over stable internal knowledge, then Direct and Assistive answers would be expected to be equivalent. Instead, even frontier LLMs exhibit only moderate cross-prompt consistency, with agreement rates around 70%. This pattern suggests that LLM outputs strongly depend on the question decomposition rather than reflecting a consistent reasoning process over semantically equivalent plans. We present eight representative inconsistency cases for frontier LLMs in Table 22, drawn from a manual error analysis of 100 random examples taken from all of our six evaluation datasets.

**Potential for Answer Correction** The analysis in Table 21 reveals a boundary condition where the efficacy of DBA is limited. Among the frontier-model disagreement cases summarized in Table 21, 67% fall into the *Both* category, where both the Direct and Assistive regimes are wrong. This indicates that, for frontier models, DBA is highly effective at detecting "reasoning shortcuts" (Jiang and Bansal, 2019), but it cannot correct errors that stem from a lack of parametric knowledge.

**Decomposition Hurts (Shortcut Hypothesis)** Cases in which decomposition reduces performance suggest that correct Direct answers may arise from reasoning shortcuts, as the LLM fails to reach an accurate result via a decomposed (step-by-step) execution. Example 4 in Table 22 shows a case where the LLM produces the correct final answer under Direct prompting but fails to recover the required intermediate facts when executing the gold decomposition. Due to this being a fixed gold-standard decomposition, the plan is bound to be accurate. Hence, the inconsistency with the Direct answer, highlights the LLM's lack of the relevant intermediate knowledge required in the execution of the gold-standard plan.

**Decomposition Helps (Scaffolding Hypothesis)** Conversely, when decomposition helps performance the LLM appears to possess the knowledge of the requisite atomic facts but lacks the executive function to organize them zero-shot. In Example 2, the Direct model hallucinates the date, while the Assistive prompt successfully scaffolds the retrieval of the correct intermediate entity and final answer.

## 6 Related Work

**Decomposition and Planning.** Question decomposition has been formalized as a standalone question-understanding problem through QDMR and the BREAK benchmark (Wolfson et al., 2020). More recent work has shown that smaller language models can be leveraged to generate and rank high-quality synthetic decompositions, and that fine-tuning on this data yields decomposition performance that is comparable to larger models under distribution shift (Han and Gardent, 2025). Traditional decomposition strategies function as gener-

ative interventions, modularizing execution to enhance performance on complex multi-hop queries (Khot et al., 2023; Wolfson et al., 2022; Zhou et al., 2023; Press et al., 2023; Radhakrishnan et al., 2023; Wu et al., 2024). Recent work has further leveraged these structures to improve reasoning faithfulness by enforcing structural constraints (Radhakrishnan et al., 2023). In contrast, we employ question decomposition as an experimental control: we fix a gold-standard decomposition and use it across multiple task-equivalent but structurally distinct prompting regimes, thereby decoupling planning from execution and using cross-prompt consistency as a lens on model reliability.

**Uncertainty and Abstention.** Our work relates to uncertainty estimation and selective prediction for deciding when a model should abstain. One approach elicits explicit self-evaluation signals, prompting models to estimate their own accuracy or internal knowledge (Kadavath et al., 2022). Other methods train calibrated "I don't know" behavior through self-detection and reflection on diverse question variants (Zhao et al., 2024). Additionally, verification pipelines like Chain-of-Verification utilize structured drafting and fact-checking stages to revise potential errors (Dhuliawala et al., 2024). These strategies generally treat uncertainty as a property of a single execution path. In contrast, we infer reliability behaviorally from the agreement across task-equivalent execution regimes, requiring no specialized supervision, auxiliary classifiers, or confidence heads.

**Consistency as a Reliability Signal.** Consistency across generations is a widely used indicator of correctness. Self-consistency decoding leverages majority agreement among sampled reasoning paths as a confidence heuristic (Wang et al., 2023), which can be further refined by weighting semantic rationale similarity (Knappe et al., 2024). Disagreement also helps flag hallucinations. Frameworks like SelfCheckGPT and SAC3 measure conflict among alternative continuations or perturbations (Manakul et al., 2023; Zhang et al., 2023), while multi-agent systems treat inconsistencies during cross-examination as evidence of falsehood (Cohen et al., 2023). Beyond sampling, consistency under prompt variation serves as a core reliability axis that correlates with accuracy (Nalbandyan et al., 2025; Novikova et al., 2025), and can be actively optimized (Raj et al., 2025). Unlike these methods based on sampling or paraphrasing, we evaluate consistency across structurally distinct yet task-equivalent reasoning interfaces. We demonstrate that disagreement between direct and decomposed regimes provides a simple, prompt-based signal that effectively tracks factual correctness.

## 7 Conclusion

In this work, we investigate the role of decomposed prompting through the lens of LLM reliability. By auditing semantically-equivalent reasoning paths across different model scales, we identify a scale-dependent shift in the efficacy of question decomposition. For weaker models, decomposition often functions as a scaffold that improves accuracy, while for frontier models these accuracy gains plateau. At the same time, LLM consistency between different prompting regimes serves as a strong signal for LLM performance.

We show that LLM inconsistency between Direct answering and decomposed prompting provides a reliable indicator of factual errors across different LLMs and outperforms self-reported confidence. Leveraging this insight, we introduce Disagreement-Based Abstention (DBA), a training-free approach which substantially improves error detection compared to standard uncertainty methods. Ultimately, our findings suggest that as LLMs internalize more reasoning capability, the marginal benefit of decomposed prompting shifts from expanding *what can be answered* by the LLM to auditing its confidence of *what is known*.

## Limitations

We highlight three main limitations of our analysis and proposed DBA method. First, DBA can only detect errors that manifest as disagreement across prompting regimes. When a model reproduces the same incorrect answer under Direct, Assistive, and Incremental prompting, DBA treats that answer as stable and provides no corrective signal.

Second, DBA relies on access to high-quality decompositions expressed in our DSL. In the main experiments, we use decompositions that are manually verified or produced by a strong teacher model, so our reported setup still assumes access to reliable plans. This makes the method less self-contained, especially for weaker models, which often struggle to generate good multi-hop decompositions on their own. However, this limitation is less severe for stronger models; as shown in Appendix D.4, a 70B-scale open model can generate usable decom-

positions in most manually audited cases.

Third, DBA incurs additional computational cost and latency, since it requires at least one decomposed execution in addition to the Direct call. Incremental prompting further increases this cost by introducing one model invocation per hop. In contrast, baselines such as AYS and IC-IDK operate within a single prompting regime and are therefore cheaper to deploy, although they provide weaker error-detection performance.

## Ethics Statement

Our study evaluates existing, publicly available QA benchmarks and therefore inherits any biases or limitations in those datasets. We do not collect new data, solicit human subjects, or intentionally include personally identifying information beyond what may already be present in the original benchmarks. Our method is training-free and intended for research on hallucination detection; it does not guarantee correctness and may abstain unevenly across topics or domains. Because it requires additional model calls, it increases inference cost/latency, which should be considered in any deployment.

## Acknowledgements

This research has been supported in part by the ONR Contract N00014-23-1-2364, and conducted as a collaborative effort between *Arizona State University* and the *University of Pennsylvania*. We gratefully acknowledge the *Complex Data Analysis and Reasoning Lab* at the *School of Computing and Augmented Intelligence*, *Arizona State University*, and the *Cognitive Computation Group*, *University of Pennsylvania*, for providing computational resources and institutional support. We also thank the anonymous reviewers for their constructive feedback and valuable suggestions.

## References

Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. 2023. LM vs LM: Detecting factual errors via cross examination. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12621–12640, Singapore. Association for Computational Linguistics.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3563–3578, Bangkok, Thailand. Association for Computational Linguistics.

Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 207–219, Saarbrücken, Germany. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Kelvin Han and Claire Gardent. 2025. Generating complex question decompositions in the face of distribution shifts. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1189–1211, Albuquerque, New Mexico. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Yichen Jiang and Mohit Bansal. 2019. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop QA. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2726–2736, Florence, Italy. Association for Computational Linguistics.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and

17 others. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed prompting: A modular approach for solving complex tasks. In *The Eleventh International Conference on Learning Representations*.

Tim Knappe, Ryan Luo Li, Ayush Chauhan, Kaylee Chhua, Kevin Zhu, and Sean O'Brien. 2024. Enhancing language model reasoning via weighted reasoning in self-consistency. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. 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, Singapore. Association for Computational Linguistics.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6097–6109, Florence, Italy. Association for Computational Linguistics.

Grigor Nalbandyan, Rima Shahbazyan, and Evelina Bakhturina. 2025. SCORE: Systematic COnsistency and robustness evaluation for large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 470–484, Albuquerque, New Mexico. Association for Computational Linguistics.

Jekaterina Novikova, Carol Myrick Anderson, Borhane Blili-Hamelin, and Subhabrata Majumdar. 2025. Consistency in language models: Current landscape, challenges, and future directions. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*.

OpenAI. 2025. Gpt-5.1 instant and gpt-5.1 thinking system card addendum. Accessed 2026-01-03.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

Guanqiao Qu, Qiyuan Chen, Wei Wei, Zheng Lin, Xianhao Chen, and Kaibin Huang. 2025. Mobile edge intelligence for large language models: A contemporary survey. *IEEE Communications Surveys & Tutorials*.

Ansh Radhakrishnan, Karina Nguyen, Anna Chen, Carol Chen, Carson Denison, Danny Hernandez, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Sam McCandlish, Sheer El Showk, Tamera Lanham, Tim Maxwell, Venkatesa Chandrasekaran, and 5 others. 2023. Question decomposition improves the faithfulness of model-generated reasoning. *Preprint*, arXiv:2307.11768.

Harsh Raj, Vipul Gupta, Domenic Rosati, and Subhabrata Majumdar. 2025. Improving consistency in large language models through chain of guidance. *Transactions on Machine Learning Research*.

Priyanka Sen, Alham Fikri Aji, and Amir Saffari. 2022. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1604–1619, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Adi Simhi, Itay Itzhak, Fazl Barez, Gabriel Stanovsky, and Yonatan Belinkov. 2025. Trust me, I'm wrong: LLMs hallucinate with certainty despite knowing the answer. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 14665–14688, Suzhou, China. Association for Computational Linguistics.

Akshit Sinha, Arvindh Arun, Shashwat Goel, Steffen Staab, and Jonas Geiping. 2025. The illusion of diminishing returns: Measuring long horizon execution in llms. *Preprint*, arXiv:2509.09677.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Ashok Urlana, Charaka Vinayak Kumar, Ajeet Kumar Singh, Bala Mallikarjunarao Garlapati, Srinivasa Rao Chalamala, and Rahul Mishra. 2025. Llms with industrial lens: Deciphering the challenges and prospects – a survey. *Preprint*, arXiv:2402.14558.

Rui Wang, Zhiyong Gao, Liuyang Zhang, Shuaibing Yue, and Ziyi Gao. 2025. Empowering large language models to edge intelligence: A survey of edge efficient llms and techniques. *Computer Science Review*, 57:100755.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Tomer Wolfson, Daniel Deutch, and Jonathan Berant. 2022. Weakly supervised text-to-SQL parsing through question decomposition. In *Findings of the Association for Computational Linguistics: NAACL*

*2022*, pages 2528–2542, Seattle, United States. Association for Computational Linguistics.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198.

Tomer Wolfson, Harsh Trivedi, Mor Geva, Yoav Goldberg, Dan Roth, Tushar Khot, Ashish Sabharwal, and Reut Tsarfaty. 2026. MoNaCo: More natural and complex questions for reasoning across dozens of documents. *Transactions of the Association for Computational Linguistics*, 14:23–46.

Jian Wu, Linyi Yang, Yuliang Ji, Wenhao Huang, Börje F. Karlsson, and Manabu Okumura. 2024. Gendec: A robust generative question-decomposition method for multi-hop reasoning. *Preprint*, arXiv:2402.11166.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Qwen: An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025b. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Gui, Ziran Jiang, Ziyu JIANG, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Ethan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, and 8 others. 2024. CRAG - comprehensive RAG benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley Malin, and Sricharan Kumar. 2023. SAC$^3$: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15445–15458, Singapore. Association for Computational Linguistics.

Yukun Zhao, Lingyong Yan, Weiwei Sun, Guoliang Xing, Chong Meng, Shuaiqiang Wang, Zhicong Cheng, Zhaochun Ren, and Dawei Yin. 2024. Knowing what LLMs DO NOT know: A simple yet effective self-detection method. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7051–7063, Mexico City, Mexico. Association for Computational Linguistics.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

# A Complete Results Tables

This appendix provides the complete accuracy and consistency tables across all models, datasets, and reasoning interfaces.

## A.1 Full Accuracy Results

Table 5 reports accuracy (%) for all nine models across six datasets and three reasoning interfaces (Direct, Assistive, and Incremental).

## A.2 Full Consistency Results

Table 6 reports the consistency rate (%) between the Direct interface and each structured interface (Assistive and Incremental), measuring how often the two interfaces produce semantically equivalent answers.

## A.3 Cross-Regime Consistency by Dataset

Table 7 summarizes cross-regime consistency rates, averaged across all models, for each dataset.

## A.4 Complete Baseline Results

Table 8 reports complete precision, recall, and F1 scores for all error detection methods across the three primary models (GPT-5.1, Llama-3.3-70B, and Qwen3-8B) discussed in Section 4.

Table 9 reports complete precision, recall, and F1 scores for all remaining models across all datasets.

## A.5 Incremental Consistency Analysis

Figure 4 shows the same accuracy–consistency relationship as Figure 2 in the main paper, but using Incremental consistency (Direct–Incremental agreement) and Incremental accuracy. We observe similarly strong positive correlations across all difficulty levels, confirming that the relationship between consistency and correctness is robust to the choice of reasoning interface.

## A.6 Token Cost and Call Overhead

Table 10 reports the full call and token overhead averaged across all evaluated methods, models, and datasets.

# B Data Filtering and Quality Control

Table 11 summarizes the dataset splits used in our experiments, along with dataset sizes before and after filtering.

## B.1 Filtering Criteria

We manually verified all datasets and their gold decompositions and applied the following filters to ensure high-quality, unambiguous evaluation instances.

**Bamboogle.** We used 123 out of 125 of all questions in the Bamboogle dataset, as all questions are multi-hop by design.

**CRAG.** The CRAG dataset contains a mix of single-hop and multi-hop questions, as well as questions whose answers change over time. We retained only questions that are both multi-hop and temporally stable (static or slow-changing), as defined in the original dataset, yielding 163 questions.

**Other Datasets.** For HotpotQA, FRAMES, Mintaka, and MuSiQue, we began with 300 randomly sampled questions from the standard evaluation split and applied three successive filters:

1. **Temporal Stability:** We excluded questions containing explicit time-dependent markers (e.g., “currently”, “now”, “as of today”) to ensure answers remain valid regardless of evaluation date.

2. **Semantic Clarity:** We removed ambiguous, underspecified, or logically flawed questions identified during manual verification of the decompositions.

3. **DSL Validity:** We excluded questions whose gold decompositions were malformed, truncated, or did not logically entail the final answer.

# C Evaluation and Generation Prompts

This appendix documents the prompt templates and configurations used for answer generation, decomposition, and evaluation across all experiments.

## C.1 LLM-as-Judge Configuration

We evaluate answer correctness using an LLM-as-judge protocol configured as follows:

- **Judge model:** Gemini-2.5-Flash (`google/gemini-2.5-flash`)
- **Decoding:** Greedy decoding (temperature $=$ 0)
- **Few-shot examples:** Four examples per dataset

| Model | Bamboogle | | | Mintaka | | | HotpotQA | | | CRAG | | | FRAMES | | | MuSiQue | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | D | A | I | D | A | I | D | A | I | D | A | I | D | A | I | D | A | I |
| Mistral 7B | 15.6 | 35.8 | 36.6 | 40.9 | 55.7 | 50.7 | 24.1 | 27.7 | 24.4 | 25.1 | 33.1 | 24.5 | 11.9 | 18.5 | 13.7 | 12.8 | 20.3 | 13.5 |
| Qwen 8B | 12.2 | 37.4 | 32.8 | 24.6 | 39.6 | 33.9 | 21.7 | 22.6 | 16.4 | 19.6 | 23.9 | 21.5 | 11.6 | 23.9 | 25.0 | 11.7 | 13.5 | 15.2 |
| Llama 8B | 21.3 | 39.8 | 48.0 | 33.3 | 54.7 | 58.0 | 25.3 | 26.3 | 27.5 | 27.6 | 35.6 | 41.7 | 8.9 | 17.4 | 18.8 | 9.2 | 16.2 | 16.4 |
| Qwen 32B | 18.7 | 44.7 | 43.9 | 38.6 | 59.1 | 53.4 | 31.4 | 30.7 | 27.7 | 27.0 | 37.4 | 35.6 | 14.3 | 23.2 | 21.5 | 16.0 | 22.2 | 19.9 |
| Qwen 72B | 31.7 | 58.5 | 61.0 | 53.7 | 68.3 | 71.8 | 34.9 | 36.1 | 36.1 | 31.9 | 40.5 | 41.1 | 17.4 | 29.4 | 29.0 | 23.1 | 25.9 | 26.2 |
| Llama 70B | 45.5 | 72.4 | 68.3 | 61.6 | 69.6 | 74.3 | 41.3 | 46.0 | 42.8 | 41.7 | 49.1 | 52.1 | 26.6 | 39.2 | 40.6 | 24.5 | 34.0 | 31.2 |
| Gemini Flash | 83.7 | 75.6 | 80.5 | 83.2 | 83.9 | 82.8 | 59.7 | 56.0 | 53.3 | 64.4 | 63.2 | 62.0 | 61.4 | 55.7 | 55.6 | 37.2 | 37.6 | 35.8 |
| Gemini Pro | 85.4 | 87.0 | 86.9 | 85.6 | 85.2 | 83.2 | 68.1 | 68.0 | 63.9 | 64.4 | 61.4 | 59.5 | 71.7 | 72.0 | 64.8 | 47.5 | 52.5 | 45.5 |
| GPT 5.1 | 84.5 | 87.8 | 87.0 | 86.2 | 86.9 | 82.2 | 73.7 | 72.5 | 70.7 | 65.0 | 63.8 | 62.0 | 72.3 | 72.1 | 65.9 | 53.9 | 54.3 | 46.1 |

Table 5: Accuracy (%) by model, dataset, and reasoning approach. D=Direct, A=Assistive, I=Incremental.

| Model | Bamboogle | | Mintaka | | HotpotQA | | CRAG | | FRAMES | | MuSiQue | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | I | A | I | A | I | A | I | A | I | A | I |
| Mistral 7B | 22.8 | 20.3 | 40.6 | 36.2 | 26.3 | 20.8 | 24.7 | 21.6 | 17.4 | 13.0 | 13.8 | 13.1 |
| Qwen 8B | 17.9 | 14.6 | 31.5 | 28.5 | 25.2 | 19.7 | 19.8 | 19.1 | 13.9 | 10.0 | 12.8 | 16.7 |
| Llama 8B | 19.5 | 26.0 | 37.6 | 38.3 | 27.4 | 26.3 | 29.6 | 27.8 | 11.7 | 10.6 | 11.4 | 12.1 |
| Qwen 32B | 23.6 | 22.8 | 44.6 | 43.0 | 32.1 | 32.9 | 28.4 | 28.4 | 17.7 | 12.7 | 19.4 | 21.4 |
| Qwen 72B | 40.6 | 39.8 | 57.0 | 58.7 | 36.1 | 34.7 | 37.0 | 37.0 | 22.2 | 20.9 | 29.1 | 23.8 |
| Llama 70B | 48.0 | 50.4 | 59.4 | 66.1 | 40.5 | 39.4 | 51.9 | 56.2 | 25.8 | 24.0 | 26.2 | 28.4 |
| Gemini Flash | 78.0 | 83.7 | 87.6 | 88.9 | 65.7 | 57.7 | 73.5 | 72.8 | 61.9 | 58.4 | 47.5 | 43.3 |
| Gemini Pro | 91.1 | 89.4 | 93.3 | 93.0 | 75.2 | 70.4 | 83.4 | 83.4 | 78.8 | 71.6 | 58.5 | 55.2 |
| GPT 5.1 | 88.6 | 89.4 | 93.6 | 92.3 | 78.5 | 76.3 | 81.6 | 82.8 | 79.3 | 69.7 | 64.9 | 59.7 |

Table 6: Consistency rate (%) with direct answers by model, dataset, and reasoning approach. A=Assistive, I=Incremental.

Table 7: Cross-regime consistency (equivalence rate with Direct), averaged across all models per dataset. Higher indicates more stable answers across reasoning approaches.

| Dataset | Assistive | Incremental |
|---|---|---|
| Bamboogle | 0.439 | **0.452** |
| CRAG | 0.556 | **0.572** |
| FRAMES | **0.334** | 0.295 |
| HotpotQA | 0.496 | **0.508** |
| Mintaka | **0.573** | 0.562 |
| MuSiQue | **0.290** | 0.273 |
| *Average* | **0.421** | 0.418 |

### C.2 Direct Answering Prompt

The Direct interface uses the following zero-shot prompt for open-ended question answering (Table 12).

### C.3 Correctness Evaluation Prompt

The LLM-as-judge prompt used to evaluate answer correctness is shown in Table 13.

### C.4 Consistency Evaluation Prompt

The LLM-as-judge prompt used to evaluate pairwise answer equivalence (consistency) is shown in Table 14.

### C.5 Assistive Execution Prompt

The Assistive interface receives a DSL decomposition and executes it step-by-step using the prompt shown in Table 15.

### C.6 Incremental Execution Prompt

The Incremental interface processes each subquestion independently using the prompt shown in Table 16.

### C.7 Baseline Abstention Prompts

The instructions appended to standard prompts for the abstention baselines are shown in Table 17.

## D Extended Baseline Analysis

### D.1 Per-Model Baseline Comparison

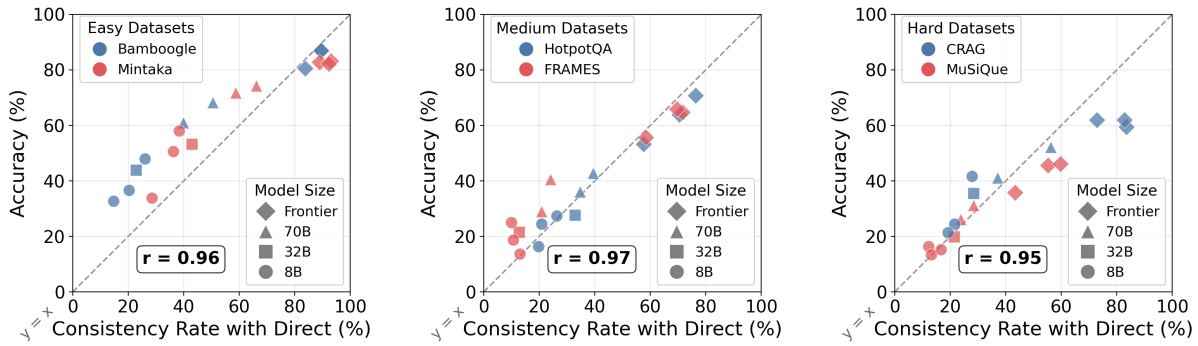
Tables 8 and 9 provide the complete per-model baseline results, including precision, recall, and F1 scores for all error detection methods across all models and datasets.

### D.2 Self-Consistency vs. DBA

We compare DBA against self-consistency (Wang et al., 2023), a standard method for reliability

| | | Bamboogle | | | | CRAG | | | | FRAMES | | | | HotpotQA | | | | Mintaka | | | | MuSiQue | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **Method** | **P** | **R** | **F1** | **AUC** | **P** | **R** | **F1** | **AUC** | **P** | **R** | **F1** | **AUC** | **P** | **R** | **F1** | **AUC** | **P** | **R** | **F1** | **AUC** | **P** | **R** | **F1** | **AUC** |
| GPT 5.1 | AYS | 0.64 | 0.37 | 0.47 | 0.67 | 0.75 | 0.59 | 0.66 | 0.74 | 0.64 | 0.42 | 0.51 | 0.67 | **0.87** | 0.36 | 0.51 | 0.67 | 0.42 | 0.41 | 0.42 | 0.66 | **0.91** | 0.24 | 0.38 | 0.61 |
| | IC-IDK | 0.67 | 0.32 | 0.43 | 0.64 | 0.71 | 0.52 | 0.60 | 0.70 | **0.73** | 0.14 | 0.24 | 0.56 | 0.74 | 0.47 | 0.58 | 0.71 | 0.37 | 0.27 | 0.31 | 0.60 | **0.91** | 0.16 | 0.27 | 0.57 |
| | DBA-A | **0.86** | **0.63** | **0.73** | **0.81** | **0.77** | 0.40 | 0.53 | 0.67 | 0.66 | 0.52 | 0.58 | 0.71 | 0.78 | 0.64 | 0.70 | 0.79 | **0.68** | 0.32 | 0.43 | 0.65 | 0.75 | 0.59 | 0.66 | 0.72 |
| | DBA-I | 0.69 | 0.47 | 0.56 | 0.72 | 0.75 | 0.37 | 0.49 | 0.65 | 0.62 | 0.70 | **0.66** | **0.77** | 0.68 | 0.61 | 0.64 | 0.75 | 0.64 | 0.39 | 0.48 | 0.68 | 0.68 | 0.60 | 0.64 | 0.68 |
| | Ensemble-A | 0.71 | **0.63** | 0.67 | 0.79 | 0.73 | **0.77** | **0.75** | **0.81** | 0.59 | 0.60 | 0.60 | 0.73 | 0.77 | **0.69** | **0.73** | **0.81** | 0.42 | 0.51 | 0.46 | 0.70 | 0.75 | 0.65 | **0.69** | **0.73** |
| | Ensemble-I | 0.67 | 0.53 | 0.59 | 0.74 | 0.73 | 0.75 | 0.74 | 0.80 | 0.56 | **0.74** | 0.64 | **0.77** | 0.67 | 0.68 | 0.68 | 0.78 | 0.43 | **0.59** | **0.49** | **0.73** | 0.69 | **0.66** | 0.68 | 0.71 |
| Llama 70B | AYS | 0.81 | 0.37 | 0.51 | 0.63 | 0.82 | 0.39 | 0.53 | 0.64 | 0.87 | 0.61 | 0.72 | 0.68 | 0.79 | 0.40 | 0.53 | 0.62 | 0.69 | 0.38 | 0.49 | 0.64 | 0.90 | 0.57 | 0.70 | 0.69 |
| | IC-IDK | 0.73 | 0.36 | 0.48 | 0.60 | 0.70 | **0.81** | 0.75 | 0.67 | **0.94** | 0.07 | 0.13 | 0.53 | 0.78 | 0.71 | 0.74 | 0.71 | 0.59 | 0.65 | 0.62 | 0.68 | **0.96** | 0.12 | 0.22 | 0.55 |
| | DBA-A | **0.94** | 0.90 | **0.92** | **0.91** | 0.79 | 0.65 | 0.71 | 0.71 | 0.88 | 0.89 | 0.89 | **0.78** | **0.85** | 0.84 | **0.84** | **0.81** | **0.83** | 0.72 | 0.77 | 0.81 | 0.88 | 0.85 | 0.86 | **0.74** |
| | DBA-I | 0.89 | 0.81 | 0.84 | 0.84 | **0.87** | 0.65 | 0.74 | **0.76** | 0.87 | 0.91 | 0.89 | 0.77 | 0.83 | 0.80 | 0.81 | 0.78 | 0.82 | 0.77 | **0.79** | **0.83** | 0.86 | 0.82 | 0.84 | 0.71 |
| | Ensemble-A | 0.87 | **0.91** | 0.89 | 0.87 | 0.76 | 0.72 | 0.74 | 0.70 | 0.85 | **0.95** | **0.90** | 0.75 | 0.79 | **0.90** | **0.84** | 0.78 | 0.73 | 0.82 | 0.77 | 0.82 | 0.85 | **0.92** | **0.88** | 0.72 |
| | Ensemble-I | 0.85 | 0.85 | 0.85 | 0.84 | 0.80 | 0.74 | **0.77** | 0.74 | 0.85 | **0.95** | **0.90** | 0.74 | 0.78 | 0.85 | 0.81 | 0.75 | 0.73 | **0.84** | 0.78 | 0.82 | 0.85 | 0.91 | **0.88** | 0.70 |
| Qwen3 8B | AYS | 0.94 | 0.59 | 0.73 | 0.66 | **0.90** | 0.64 | 0.75 | 0.68 | **0.94** | 0.83 | 0.88 | 0.71 | **0.91** | 0.65 | 0.76 | 0.71 | 0.90 | 0.57 | 0.70 | 0.68 | 0.93 | 0.68 | 0.79 | 0.64 |
| | IC-IDK | 0.94 | 0.43 | 0.59 | 0.62 | 0.86 | 0.68 | 0.76 | 0.61 | **0.94** | 0.29 | 0.44 | 0.57 | **0.91** | 0.47 | 0.62 | 0.65 | 0.80 | 0.30 | 0.44 | 0.53 | **0.95** | 0.38 | 0.54 | 0.61 |
| | DBA-A | 0.96 | 0.90 | 0.93 | **0.82** | 0.89 | 0.89 | 0.89 | **0.73** | **0.94** | 0.92 | 0.93 | **0.74** | **0.91** | 0.83 | 0.87 | **0.76** | **0.92** | 0.82 | 0.87 | **0.79** | 0.92 | 0.91 | 0.91 | **0.65** |
| | DBA-I | **0.99** | 0.67 | 0.80 | 0.80 | 0.89 | 0.89 | 0.89 | **0.73** | **0.94** | 0.95 | 0.94 | 0.73 | 0.88 | 0.80 | 0.84 | 0.70 | 0.90 | 0.74 | 0.81 | 0.74 | 0.92 | 0.87 | 0.89 | **0.65** |
| | Ensemble-A | 0.95 | **0.94** | **0.94** | 0.80 | 0.87 | **0.95** | **0.91** | 0.67 | 0.92 | 0.96 | 0.94 | 0.67 | 0.89 | **0.93** | **0.91** | 0.75 | 0.88 | **0.89** | **0.88** | 0.75 | 0.91 | **0.95** | **0.93** | 0.62 |
| | Ensemble-I | 0.95 | 0.85 | 0.90 | 0.76 | 0.88 | **0.95** | **0.91** | 0.70 | 0.92 | **0.99** | **0.95** | 0.69 | 0.87 | **0.93** | 0.89 | 0.70 | 0.85 | 0.84 | 0.85 | 0.69 | 0.91 | 0.90 | 0.91 | 0.60 |

Table 8: Complete error-detection results (Precision, Recall, F1, AUROC; AUROC is reported as AUC in the table) for the three main models across all datasets. Within each model block and dataset/metric pair, **bold** marks the best overall value across all methods (ties included), while underlined marks the best non-ensemble value among AYS, IC-IDK, DBA-A, and DBA-I (ties included). When a non-ensemble method is also best overall, it is both **bold** and underlined.




(a) Easy (Bamboogle, Mintaka) (b) Medium (HotpotQA, FRAMES) (c) Hard (CRAG, MuSiQue)

Figure 4: Incremental accuracy vs. Incremental consistency rate (Direct-Incremental agreement) across 9 models and 6 datasets. Similar to the Assistive case (Figure 2), we observe strong positive correlations, confirming that the relationship between consistency and correctness generalizes across reasoning interfaces.

estimation. We report two complementary self-consistency analyses: the original majority-vote protocol and a continuous-score evaluation based on agreement across sampled answers.

### D.2.1 Majority-vote analysis

We compare DBA against self-consistency (Wang et al., 2023), a standard method for reliability estimation. While DBA measures agreement across *different prompting regimes* (Direct vs. Decomposed), self-consistency measures *within-prompt* agreement across stochastic samples.

For each question, we generate seven candidate answers using the Direct interface: six sampled generations at temperature $T = 0.7$ and one deterministic generation at $T = 0$.[1] We apply a majority vote threshold: if a single semantic claim appears in at least four of the seven candidates, the model returns that answer; otherwise, it abstains (flags the instance as an error).

We ran self-consistency in 4 datasets and present the results in Table 19.

### D.2.2 Continuous-score AUROC evaluation

We also re-evaluate self-consistency by treating the number of semantically equivalent answers (out of 7) as a continuous score and computing AUROC. For each question, we identify the dominant

[1]For GPT-5.1, we sampled 7 responses via medium reasoning effort as temperature control was not exposed.

| | | Bamboogle | | | | CRAG | | | | FRAMES | | | | HotpotQA | | | | Mintaka | | | | MuSiQue | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **Method** | **P** | **R** | **F1** | **AUC** | **P** | **R** | **F1** | **AUC** | **P** | **R** | **F1** | **AUC** | **P** | **R** | **F1** | **AUC** | **P** | **R** | **F1** | **AUC** | **P** | **R** | **F1** | **AUC** |
| Mistral 7B | AYS | **0.95** | 0.38 | 0.54 | 0.63 | **0.92** | 0.20 | 0.33 | 0.58 | 0.91 | 0.41 | 0.57 | 0.56 | 0.90 | 0.46 | 0.60 | 0.64 | 0.81 | 0.30 | 0.43 | 0.60 | 0.92 | 0.42 | 0.57 | 0.58 |
| | IC-IDK | 0.92 | 0.22 | 0.36 | 0.56 | 0.90 | 0.51 | 0.65 | 0.67 | **0.96** | 0.37 | 0.53 | 0.63 | **0.95** | 0.10 | 0.18 | 0.54 | **0.88** | 0.21 | 0.34 | 0.58 | 0.92 | 0.58 | 0.71 | 0.61 |
| | DBA-A | 0.94 | 0.43 | 0.59 | 0.64 | 0.88 | 0.81 | 0.84 | 0.75 | 0.93 | 0.87 | 0.90 | **0.69** | 0.89 | 0.60 | 0.72 | 0.69 | 0.85 | 0.38 | 0.52 | 0.64 | **0.93** | 0.92 | **0.93** | **0.72** |
| | DBA-I | 0.91 | 0.40 | 0.56 | 0.60 | 0.87 | **0.85** | 0.86 | 0.73 | 0.91 | 0.90 | 0.90 | 0.62 | 0.85 | 0.55 | 0.67 | 0.62 | 0.74 | 0.36 | 0.48 | 0.59 | 0.92 | 0.92 | 0.92 | 0.68 |
| | Ensemble-A | 0.93 | **0.62** | **0.75** | **0.68** | 0.89 | 0.84 | **0.87** | **0.76** | 0.92 | 0.89 | 0.91 | 0.66 | 0.88 | **0.78** | **0.83** | **0.71** | 0.82 | **0.57** | **0.67** | **0.69** | 0.92 | **0.94** | **0.93** | 0.67 |
| | Ensemble-I | 0.92 | 0.57 | 0.70 | 0.65 | 0.86 | **0.85** | 0.86 | 0.71 | 0.91 | **0.93** | **0.92** | 0.61 | 0.85 | 0.72 | 0.78 | 0.65 | 0.76 | 0.55 | 0.64 | 0.65 | 0.90 | **0.94** | 0.92 | 0.61 |
| Llama 3.1 8B | AYS | 0.92 | 0.50 | 0.65 | 0.68 | 0.86 | 0.54 | 0.66 | 0.66 | **0.97** | 0.40 | 0.57 | 0.64 | 0.89 | 0.52 | 0.66 | 0.66 | 0.86 | 0.55 | 0.67 | 0.68 | 0.95 | 0.45 | 0.61 | 0.61 |
| | IC-IDK | 0.83 | 0.05 | 0.10 | 0.51 | **0.91** | 0.25 | 0.40 | 0.59 | **0.97** | 0.12 | 0.21 | 0.54 | **1.00** | 0.02 | 0.04 | 0.51 | 0.82 | 0.07 | 0.13 | 0.52 | **1.00** | 0.10 | 0.19 | 0.55 |
| | DBA-A | **0.93** | 0.83 | 0.88 | **0.80** | 0.88 | 0.84 | 0.86 | 0.76 | 0.95 | 0.92 | 0.94 | **0.71** | 0.91 | 0.77 | 0.83 | **0.77** | **0.91** | 0.76 | 0.83 | **0.80** | 0.96 | 0.93 | **0.95** | **0.77** |
| | DBA-I | **0.93** | 0.69 | 0.79 | 0.75 | 0.89 | 0.87 | **0.88** | **0.79** | 0.94 | 0.93 | 0.93 | 0.67 | 0.88 | 0.78 | 0.83 | 0.73 | **0.91** | 0.73 | 0.81 | 0.79 | 0.94 | 0.91 | 0.92 | 0.67 |
| | Ensemble-A | 0.89 | **0.89** | **0.89** | 0.76 | 0.84 | 0.88 | 0.86 | 0.71 | 0.95 | 0.94 | 0.94 | 0.70 | 0.87 | 0.85 | 0.86 | 0.74 | 0.85 | **0.85** | **0.85** | 0.77 | 0.95 | **0.96** | **0.95** | 0.67 |
| | Ensemble-I | 0.91 | 0.84 | 0.88 | 0.77 | 0.84 | **0.91** | 0.87 | 0.71 | 0.94 | **0.96** | **0.95** | 0.65 | 0.86 | **0.89** | **0.88** | 0.73 | 0.85 | **0.85** | **0.85** | 0.77 | 0.93 | 0.93 | 0.93 | 0.64 |
| Qwen3 32B | AYS | 0.90 | 0.56 | 0.69 | 0.65 | **0.90** | 0.61 | 0.72 | 0.71 | **0.93** | 0.74 | 0.82 | 0.71 | **0.94** | 0.48 | 0.63 | 0.70 | 0.84 | 0.52 | 0.65 | 0.68 | **0.94** | 0.64 | 0.76 | 0.71 |
| | IC-IDK | 0.94 | 0.15 | 0.26 | 0.55 | 0.86 | 0.32 | 0.46 | 0.58 | 0.92 | 0.10 | 0.17 | 0.52 | 0.92 | 0.06 | 0.12 | 0.52 | **1.00** | 0.10 | 0.17 | 0.55 | **0.94** | 0.20 | 0.33 | 0.57 |
| | DBA-A | **0.95** | 0.89 | **0.92** | **0.84** | 0.89 | 0.87 | **0.88** | 0.79 | **0.93** | 0.89 | 0.91 | **0.73** | 0.86 | 0.83 | 0.85 | **0.77** | 0.88 | 0.83 | 0.86 | **0.83** | 0.93 | 0.90 | **0.92** | **0.78** |
| | DBA-I | 0.94 | 0.89 | 0.91 | 0.81 | **0.90** | 0.86 | **0.88** | **0.80** | 0.92 | 0.94 | 0.93 | 0.72 | 0.85 | 0.84 | 0.85 | 0.76 | 0.87 | 0.83 | 0.85 | 0.82 | 0.91 | 0.86 | 0.88 | 0.72 |
| | Ensemble-A | 0.90 | 0.90 | 0.90 | 0.73 | 0.85 | 0.89 | 0.87 | 0.74 | 0.91 | 0.96 | 0.93 | 0.68 | 0.85 | 0.89 | 0.87 | **0.77** | 0.84 | **0.92** | **0.88** | 0.82 | 0.92 | **0.93** | **0.92** | 0.74 |
| | Ensemble-I | 0.90 | **0.91** | 0.91 | 0.74 | 0.86 | **0.91** | **0.88** | 0.74 | 0.91 | **0.97** | **0.94** | 0.69 | 0.84 | **0.91** | **0.88** | 0.76 | 0.82 | 0.91 | 0.86 | 0.78 | 0.91 | 0.90 | 0.90 | 0.70 |
| Qwen 2.5 72B | AYS | 0.87 | 0.63 | 0.73 | 0.71 | **0.91** | 0.72 | 0.80 | 0.78 | 0.91 | 0.74 | 0.81 | 0.69 | 0.84 | 0.69 | 0.76 | 0.72 | 0.70 | 0.66 | 0.68 | 0.70 | 0.82 | 0.74 | 0.78 | 0.60 |
| | IC-IDK | 0.79 | 0.87 | 0.82 | 0.68 | 0.84 | 0.47 | 0.60 | 0.64 | **0.97** | 0.25 | 0.40 | 0.61 | **0.90** | 0.41 | 0.57 | 0.66 | 0.74 | 0.26 | 0.39 | 0.59 | **0.92** | 0.42 | 0.58 | 0.65 |
| | DBA-A | 0.94 | 0.82 | 0.88 | **0.86** | **0.91** | 0.83 | **0.87** | **0.83** | 0.91 | 0.85 | 0.88 | 0.72 | 0.84 | 0.84 | 0.84 | **0.77** | 0.86 | 0.77 | 0.81 | 0.83 | 0.90 | 0.83 | 0.86 | **0.76** |
| | DBA-I | **0.97** | 0.73 | 0.83 | 0.84 | 0.90 | 0.81 | 0.85 | 0.81 | 0.93 | 0.89 | 0.91 | **0.79** | 0.84 | 0.84 | 0.84 | **0.77** | **0.87** | 0.78 | **0.82** | **0.84** | 0.86 | 0.86 | 0.86 | 0.71 |
| | Ensemble-A | 0.87 | **0.90** | **0.89** | 0.81 | 0.87 | 0.87 | **0.87** | 0.80 | 0.90 | 0.94 | 0.92 | 0.70 | 0.80 | **0.91** | **0.85** | 0.74 | 0.74 | **0.90** | 0.81 | 0.80 | 0.83 | 0.93 | **0.88** | 0.66 |
| | Ensemble-I | 0.90 | 0.85 | 0.87 | 0.82 | 0.86 | **0.88** | **0.87** | 0.78 | 0.90 | **0.96** | **0.93** | 0.72 | 0.80 | 0.90 | **0.85** | 0.74 | 0.72 | 0.89 | 0.79 | 0.78 | 0.81 | **0.95** | **0.88** | 0.61 |
| Gemini Flash | AYS | 0.67 | 0.40 | 0.50 | 0.68 | **0.85** | 0.67 | 0.75 | 0.80 | 0.68 | 0.50 | 0.57 | 0.68 | 0.73 | 0.29 | 0.42 | 0.61 | 0.56 | 0.43 | 0.49 | 0.68 | 0.86 | 0.27 | 0.41 | 0.60 |
| | IC-IDK | **1.00** | 0.05 | 0.10 | 0.53 | 0.64 | 0.28 | 0.39 | 0.59 | 0.70 | 0.12 | 0.21 | 0.55 | **0.87** | 0.23 | 0.37 | 0.60 | 0.43 | 0.12 | 0.19 | 0.54 | **0.87** | 0.27 | 0.41 | 0.60 |
| | DBA-A | 0.43 | 0.60 | 0.50 | 0.72 | 0.77 | 0.57 | 0.65 | 0.74 | **0.72** | 0.71 | 0.71 | 0.77 | 0.74 | 0.64 | 0.69 | 0.74 | **0.63** | 0.44 | 0.52 | 0.69 | 0.84 | 0.71 | 0.77 | **0.74** |
| | DBA-I | 0.57 | 0.60 | 0.58 | 0.76 | 0.73 | 0.57 | 0.64 | 0.73 | 0.69 | 0.74 | 0.71 | 0.76 | 0.66 | 0.70 | 0.68 | 0.73 | 0.58 | 0.35 | 0.44 | 0.65 | 0.82 | 0.74 | 0.78 | 0.73 |
| | Ensemble-A | 0.45 | **0.75** | 0.57 | **0.79** | 0.75 | **0.82** | **0.78** | **0.83** | 0.67 | **0.86** | **0.75** | **0.79** | 0.70 | 0.69 | **0.70** | **0.75** | 0.52 | **0.62** | **0.56** | **0.75** | 0.83 | 0.75 | **0.79** | **0.74** |
| | Ensemble-I | 0.52 | 0.70 | **0.60** | **0.79** | 0.72 | **0.82** | 0.77 | 0.82 | 0.66 | 0.83 | 0.73 | 0.78 | 0.62 | **0.73** | 0.67 | 0.71 | 0.48 | 0.58 | 0.53 | 0.73 | 0.81 | **0.78** | **0.79** | 0.73 |
| Gemini Pro | AYS | 0.78 | 0.39 | 0.52 | 0.68 | **0.80** | 0.56 | 0.66 | 0.74 | 0.55 | 0.22 | 0.31 | 0.57 | 0.74 | 0.22 | 0.34 | 0.59 | 0.44 | 0.32 | 0.37 | 0.62 | 0.74 | 0.21 | 0.33 | 0.56 |
| | IC-IDK | **1.00** | 0.06 | 0.10 | 0.53 | 0.77 | 0.30 | 0.43 | 0.62 | 0.67 | 0.05 | 0.09 | 0.52 | **0.86** | 0.20 | 0.32 | 0.59 | **0.71** | 0.27 | 0.39 | 0.62 | **0.93** | 0.18 | 0.29 | 0.58 |
| | DBA-A | 0.57 | 0.44 | 0.50 | 0.69 | 0.70 | 0.33 | 0.45 | 0.63 | **0.69** | 0.52 | 0.59 | 0.71 | 0.81 | 0.66 | **0.73** | **0.79** | 0.52 | 0.30 | 0.38 | 0.62 | 0.83 | 0.66 | 0.73 | **0.75** |
| | DBA-I | 0.67 | 0.44 | 0.53 | 0.70 | 0.65 | 0.30 | 0.41 | 0.61 | 0.58 | 0.58 | 0.58 | 0.71 | 0.72 | 0.59 | 0.65 | 0.74 | 0.65 | 0.32 | 0.43 | 0.64 | 0.72 | 0.61 | 0.66 | 0.68 |
| | Ensemble-A | 0.63 | **0.67** | **0.65** | **0.80** | 0.76 | **0.70** | **0.73** | **0.79** | 0.62 | 0.58 | **0.60** | **0.72** | 0.77 | **0.67** | 0.72 | 0.78 | 0.42 | 0.47 | 0.44 | 0.67 | 0.79 | **0.70** | **0.74** | **0.75** |
| | Ensemble-I | 0.67 | 0.56 | 0.61 | 0.75 | 0.75 | 0.68 | 0.71 | 0.77 | 0.55 | **0.65** | **0.60** | **0.72** | 0.72 | 0.62 | 0.66 | 0.74 | 0.47 | **0.49** | **0.48** | **0.69** | 0.72 | 0.67 | 0.69 | 0.69 |

Table 9: Complete error-detection results (Precision, Recall, F1, AUROC; AUROC is reported as AUC in the table) for all remaining models. Within each model block and dataset/metric pair, **bold** marks the best overall value across all methods (ties included), while underlined marks the best non-ensemble value among AYS, IC-IDK, DBA-A, and DBA-I (ties included). When a non-ensemble method is also best overall, it is both **bold** and underlined.

| **Method** | **LLM Calls** | **Input Tokens** | **Output Tokens** | **Total Tokens** | **Input×** | **Output×** | **Total×** |
|---|---|---|---|---|---|---|---|
| Direct | 1 | 634 | 199 | 833 | 1.00 | 1.00 | 1.00 |
| AYS | 2 | 706 | 348 | 1,054 | 1.11 | 1.74 | 1.26 |
| IC-IDK | 1 | 753 | 210 | 963 | 1.19 | 1.06 | 1.16 |
| DBA-A | 4 | 4,197 | 505 | 4,702 | 6.61 | 2.53 | 5.64 |
| DBA-I | 6.4 | 4,374 | 538 | 4,912 | 6.89 | 2.70 | 5.89 |
| Self-Consistency | 8 | 4,610 | 810 | 5,420 | 7.27 | 4.07 | 6.51 |

Table 10: Detailed comparison of computational cost and token distribution across different prompting and abstention methods. All token counts and multipliers are normalized against the Direct prompting baseline.

semantic claim among the seven generations and use its support count as the abstention score. We then sweep thresholds from 1 to 7, abstaining when the support count falls below the threshold, and report Precision, Recall, and F1 at the F1-optimal threshold.

We evaluate this protocol on three datasets spanning easy, medium, and hard settings (Bamboogle, FRAMES, MuSiQue) and across four representative model scales (Qwen3-8B, Qwen3-32B, Llama-3.3-70B, Gemini-2.5-Pro) as defined in Section 3. Table 18 reports metrics averaged across the three datasets.

Across all settings, the F1-optimal threshold is consistently high (6 or 7), suggesting that self-consistency is most reliable under near-unanimous

| Dataset | Split | Init. | Final | Rmvd. | % Lost |
|---|---|---|---|---|---|
| Bamboogle | Full | 125 | 123 | 2 | 1.6% |
| CRAG† | train | 163 | 163 | 0 | 0% |
| FRAMES | test | 300 | 293 | 7 | 2.3% |
| HotpotQA | val | 300 | 274 | 26 | 8.7% |
| Mintaka | val | 300 | 298 | 2 | 0.7% |
| MuSiQue | val | 300 | 282 | 18 | 6.0% |
| **Total** | – | **1488** | **1433** | **55** | **3.7%** |

Table 11: Dataset splits and sizes before and after filtering. †CRAG uses the multi-hop subset from the original dataset; single-hop questions were excluded by design. Other datasets use the standard evaluation split and were filtered for temporal stability, semantic clarity, and DSL validity.

agreement. While self-consistency achieves competitive AUROC, its optimal-threshold F1 is generally lower than DBA-A. Additionally, it requires seven sampled generations, semantic aggregation with an LLM judge, and explicit threshold tuning.

### D.3 Comparative Advantage of DBA

Table 19 demonstrates that DBA, which varies the decomposed prompting regime, is a far more effective probe for error detection than varying the *decoding parameters* (Self-Consistency).

1. **Recall Disparity:** DBA-A and DBA-I consistently achieve Recall scores 4–5× higher than Self-Consistency. For Qwen3-8B on *MuSiQue*, Self-Consistency Recall is 0.46, whereas DBA-A Recall is 0.91.

2. **Comparable Precision:** Despite the massive gain in Recall, DBA maintains Precision levels comparable to Self-Consistency (e.g., for Mistral 7B on *FRAMES*, Self-Consistency Precision is 1.00 vs. DBA-A Precision of 0.87).

3. **F1 Score Dominance:** Consequently, DBA dominates on F1 score. For Gemini Pro on Mintaka, Self-Consistency achieves an F1 of 0.16, while DBA-A achieves 0.38.

### D.4 Feasibility of Model Generated Decomposition

Our main experiments use manually verified reference decompositions so that comparisons across Direct, Assistive, and Incremental prompting are not confounded by decomposition errors. A natural practical question, however, is whether such decompositions must always be provided by humans or a frontier teacher model at deployment time.

To assess this, we conduct an auxiliary feasibility analysis using Qwen2.5-72B-Instruct as an automatic decomposition generator. We compare its generated decompositions against the gold DSL references in two ways. First, we perform an automatic comparison that measures how well the generated decomposition matches the reference sequence of reasoning steps. Second, because automatic matching may under-credit valid paraphrases or minor structural variations, we also perform a manual audit. This auxiliary analysis is also consistent with prior work showing that high-quality decompositions can be obtained automatically, either by using strong teacher models with manual verification or by using compact-model pipelines to generate and rank synthetic decompositions (Wolfson et al., 2020; Han and Gardent, 2025).

For the manual audit, we inspect 120 generated decompositions (20 per dataset) and judge whether each decomposition remains consistent with the original question and preserves the intended multi-hop reasoning structure. Under this criterion, 103 of 120 decompositions are valid, for an overall validity rate of 85.8%. This indicates that strong open models can generate usable decompositions in most cases.

For completeness, we also report an automatic structural comparison against the gold DSL templates. Averaged across the six datasets, Qwen2.5-72B-Instruct achieves 0.841 precision, 0.869 recall, and 0.854 F1 under hop-level alignment, with an average hop ratio of 1.081. These results are broadly consistent with the manual audit, although we view the manual audit as the more interpretable measure of practical usability.

Overall, these results suggest that the decomposition requirement of DBA is less restrictive for frontier and 70B-class models than the main experimental setup alone might imply. At the same time, we do not replace the manually verified decompositions in the main experiments, since the goal of the main study is a controlled comparison under fixed, high-quality plans, and automatic decomposition remains less dependable for weaker models.

The automatic structural comparison reported above uses an LLM-as-judge on top of `google/gemini-2.5-flash`; the exact prompt fed to the judge is shown in Table 20. For each question, the judge sees both the gold DSL and the model-generated DSL and returns a strict JSON object with two fields: `equivalent_final` (0/1) for end-to-end equivalence, and `matches` (integer)

**System prompt:**
You are a precise question answering assistant. Always answer the question; if unsure, provide your best single answer. Do not expose hidden reasoning or use `<think>` tags.

GENERAL GUIDELINES:

- Always return your best single answer even if uncertain.
- Keep prose minimal—prefer the shortest phrasing that still answers the question.

REFERENCE DATE:
Treat all time-relative terms (e.g., "now", "currently") and age calculations as referring to {reference_date} unless the question specifies another date.

NORMALIZATION:

- Dates: if day known → YYYY-MM-DD; if only month+year → YYYY-MM; if only year → YYYY.
- Units: prefer symbols (km, m, ft, kg, $mi^2$, $km^2$). Keep one consistent unit.
- Multi-item strings: separate with semicolons exactly like "A; B; C".
- Names: prefer canonical full names (e.g., "Thomas Edison" not "Edison").

OUTPUT:
`answer:`
`<one line with only the normalized final answer>`

Table 12: Direct answering prompt used for open-ended question answering in zero-shot setting.

**System prompt:**
You are an impartial string evaluator.

Given the original question, the gold (ground-truth) answer, and the prediction, decide whether the prediction communicates the same essential fact(s).

[question]: {question}
[gold]: {gold_answer}
[prediction]: {prediction}

CRITICAL TOLERANCE RULES:

1) **Normalization** — Ignore case, leading/trailing spaces, and punctuation. Treat common aliases/abbreviations as equivalent (e.g., "NYC" ≡ "New York City").

2) **Numeric expressions** — Words vs digits are equivalent ("ten" ≡ "10"). Accept $|a - b| \leq \max(0.5, 5\%$ of reference) except for calendar dates/years, which must match exactly.

3) **Lists/Sets** — If the question does not require order, treat unordered lists as equivalent when they contain the same unique items.

4) **Units & formatting** — Normalize common unit spellings/symbols ("km" ≡ "kilometers") and ignore thousands separators.

5) **Contradictions** — If any part of the prediction contradicts the gold fact, it is incorrect.

6) **Extra detail** — Extra information is acceptable if it does not contradict the gold answer.

Return exactly two lines:
Line 1: `correct: 1` or `0`
Line 2: `reasoning: <brief explanation>`

Table 13: LLM-as-judge prompt for evaluating answer correctness against ground truth.

for the number of gold hops covered by some hop in the model decomposition.

## E Additional Qualitative Examples and Failure Modes

Table 21 categorizes all inconsistent examples into three primary failure modes identified during manual analysis of 100 examples across 6 datasets. Table 22 presents eight representative inconsistency examples drawn from multiple datasets and frontier models, illustrating the qualitative behaviors underlying these failure modes.

**System prompt:**
You are an impartial pairwise string evaluator.

Given an original question (for context) and two candidate answers (A and B), decide if they convey the same essential facts with respect to the question, subject to the CRITICAL TOLERANCE RULES.

[question]: {question}
[answer_a]: {answer_a}
[answer_b]: {answer_b}

CRITICAL TOLERANCE RULES:

1) **Normalization** — Ignore case, leading/trailing spaces, punctuation, diacritics, and English articles ("the", "a", "an"). Treat common aliases/abbreviations as equivalent.

2) **Numeric expressions** — Words vs digits are equivalent ("ten" $\equiv$ "10"). Accept $|a - b| \leq \max(0.5, 5\%$ of reference$)$ except for calendar dates/years, which must match exactly.

3) **Lists/Sets** — If the question does not require order ("in order", "ranked", "first/second/third"), treat unordered lists as equivalent when they contain the same unique items.

4) **Mappings/Pairs** — Treat "A $\rightarrow$ B", "A: B", "A = B", and "A (B)" as equivalent notations for the same pair.

5) **Units & formatting** — Normalize common unit spellings/symbols ("km" $\equiv$ "kilometers") and ignore thousands separators.

6) **Yes/No** — "yes/true/correct" $\equiv$ "no/false/incorrect" only within their respective groups.

7) **Contradictions** — If any part of one answer contradicts the other regarding the same fact, they are not equivalent.

8) **Extra detail** — Extra descriptive text is acceptable if it does not change or contradict the shared fact(s).

9) **Subset allowance** — If the question does not request a specific count (e.g., "top k" or "all"), an answer that is a subset of another is acceptable provided it does not contradict any requested facts.

10) **Alternatives** — If answer_b contains semicolon-separated alternatives, compare answer_a independently to each alternative. If any single alternative is equivalent $\rightarrow$ output equivalent: 1.

Return exactly two lines:
Line 1: `equivalent: 1` or `0`
Line 2: `reasoning: <brief explanation>`

Table 14: LLM-as-judge prompt for evaluating pairwise answer equivalence (consistency).

**System prompt:**
You will be given a short Python-flavored DSL and must execute it step by step to produce intermediate answers.

Each line has the form:
`answer_k: TYPE = qa_model("QUESTION WITH {placeholders}")`

EXECUTION RULES:

- Execute strictly in sequence (answer_1, answer_2, ...). Later answers may reference earlier ones.
- Multi-item requests: return ONE string with items separated by semicolons (e.g., "A; B; C").
- Concision: values only—no extra prose; no terminal period.
- Always answer; if unsure, provide your best answer.

OUTPUT FORMAT (MANDATORY):
Your entire reply MUST be a single valid JSON object:
`{"answer_1": "<value>", "answer_2": "<value>", ...}`

Table 15: Assistive execution prompt for DSL-guided step-by-step reasoning.

**Subquestion system prompt:**
Answer the following question concisely (prefer 1–3 words when natural).

- No extra prose.
- Always answer; if unsure, provide your best single answer.
- Multi-item answers: return one string with items separated by semicolons (A; B; C).

Output: [Only the answer text]

**Aggregation system prompt:**
You will be given a reasoning chain (sequence of sub-questions and answers) that builds up to a complex answer.

- Use ONLY the provided chain to answer the original question.
- Answer concisely (prefer 1–3 words when natural).
- Return only the final answer text; do not include an explanation.

Table 16: Incremental execution prompts: subquestion prompt for each hop, and aggregation prompt for final answer synthesis.

**IC-IDK (I Don't Know) instruction:**
If you are not sure you know the answer, answer with "I don't know" only.

**AYS (Are-You-Sure) follow-up prompt:**
After receiving the model's initial answer, the following follow-up is issued:

Question: {question}
Claim: {initial_answer}

Are you sure regarding the correctness of your claim? Please answer with Yes or No.

Table 17: Baseline abstention instructions. IC-IDK is appended to the original question prompt; AYS is a separate follow-up turn after the model provides an initial answer.

| Model | Method | Prec. | Rec. | F1 | AUROC |
|---|---|---|---|---|---|
| Qwen3-8B | SC | 0.91 | 0.76 | 0.83 | 0.72 |
| | DBA-A | **0.94** | **0.91** | **0.92** | **0.74** |
| Qwen3-32B | SC | 0.93 | 0.85 | 0.89 | 0.75 |
| | DBA-A | **0.94** | **0.89** | **0.92** | **0.78** |
| Llama-3.3-70B | SC | 0.83 | 0.55 | 0.67 | 0.65 |
| | DBA-A | **0.90** | **0.88** | **0.89** | **0.81** |
| Gemini-2.5-Pro | SC | 0.67 | **0.68** | **0.67** | **0.76** |
| | DBA-A | **0.71** | 0.55 | 0.61 | 0.73 |

Table 18: Continuous-score evaluation of self-consistency.

| Model | Method | Bamboogle | | | FRAMES | | | Mintaka | | | MuSiQue | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| Gemini Flash | self-consistency | 0.50 | 0.35 | 0.41 | **0.85** | 0.49 | 0.62 | 0.28 | 0.10 | 0.15 | **0.93** | 0.32 | 0.48 |
| | DBA-A | 0.43 | **0.60** | 0.50 | 0.72 | 0.71 | 0.71 | **0.63** | **0.44** | **0.52** | 0.85 | **0.71** | **0.77** |
| | DBA-I | **0.57** | **0.60** | **0.59** | 0.69 | **0.74** | **0.71** | 0.58 | 0.35 | 0.44 | 0.82 | **0.74** | **0.78** |
| Gemini Pro | self-consistency | 0.50 | 0.17 | 0.25 | **0.90** | 0.34 | 0.49 | **0.80** | 0.09 | 0.16 | **0.85** | 0.26 | 0.40 |
| | DBA-A | 0.57 | **0.44** | **0.50** | 0.69 | 0.52 | 0.59 | 0.52 | **0.30** | **0.38** | 0.83 | **0.66** | **0.73** |
| | DBA-I | **0.67** | **0.44** | 0.53 | 0.58 | **0.58** | **0.58** | 0.65 | **0.32** | **0.43** | 0.72 | 0.62 | 0.66 |
| GPT 5.1 | self-consistency | 0.50 | 0.18 | 0.26 | **0.91** | 0.12 | 0.22 | 0.43 | 0.15 | 0.22 | **0.86** | 0.05 | 0.09 |
| | DBA-A | **0.86** | **0.63** | **0.73** | 0.66 | 0.52 | 0.58 | **0.68** | 0.32 | 0.43 | 0.75 | **0.59** | **0.66** |
| | DBA-I | 0.69 | 0.47 | 0.56 | 0.62 | **0.70** | **0.66** | 0.64 | **0.39** | **0.49** | 0.68 | 0.60 | 0.64 |
| Llama 3.1 8B | self-consistency | **1.00** | 0.07 | 0.14 | **1.00** | 0.05 | 0.09 | **1.00** | 0.08 | 0.15 | **1.00** | 0.04 | 0.08 |
| | DBA-A | 0.93 | **0.84** | **0.88** | 0.95 | 0.92 | **0.94** | 0.91 | **0.76** | **0.83** | 0.96 | **0.93** | **0.95** |
| | DBA-I | 0.93 | 0.69 | 0.79 | 0.94 | **0.93** | 0.93 | 0.91 | 0.74 | 0.81 | 0.94 | 0.91 | 0.92 |
| Llama 3.3 70B | self-consistency | **1.00** | 0.02 | 0.03 | **1.00** | 0.07 | 0.13 | 0.83 | 0.04 | 0.08 | **1.00** | 0.02 | 0.05 |
| | DBA-A | 0.94 | **0.90** | **0.92** | 0.88 | 0.89 | 0.89 | **0.83** | **0.72** | **0.77** | 0.88 | **0.85** | **0.87** |
| | DBA-I | 0.89 | 0.81 | 0.84 | 0.87 | **0.91** | **0.89** | 0.82 | **0.77** | **0.79** | 0.86 | 0.82 | 0.84 |
| Mistral 7B | self-consistency | 0.94 | 0.14 | 0.25 | **1.00** | 0.06 | 0.11 | **0.96** | 0.14 | 0.25 | **1.00** | 0.05 | 0.09 |
| | DBA-A | **0.94** | **0.43** | **0.59** | 0.93 | 0.87 | 0.90 | 0.85 | **0.38** | **0.52** | 0.93 | **0.92** | **0.93** |
| | DBA-I | 0.91 | 0.40 | 0.56 | 0.91 | **0.90** | **0.90** | 0.74 | 0.36 | 0.49 | 0.92 | **0.92** | **0.92** |
| Qwen 2.5 72B | self-consistency | 0.88 | 0.18 | 0.30 | 0.90 | 0.23 | 0.37 | 0.68 | 0.14 | 0.23 | 0.91 | 0.13 | 0.23 |
| | DBA-A | 0.95 | **0.82** | **0.88** | 0.91 | 0.85 | 0.88 | **0.87** | **0.77** | **0.81** | 0.90 | **0.83** | **0.86** |
| | DBA-I | **0.97** | 0.73 | 0.83 | **0.93** | **0.89** | **0.91** | 0.87 | **0.78** | **0.82** | 0.87 | **0.86** | **0.86** |
| Qwen3-32B | self-consistency | 0.90 | 0.25 | 0.39 | **0.97** | 0.45 | 0.61 | **1.00** | 0.12 | 0.22 | 0.93 | 0.38 | 0.54 |
| | DBA-A | **0.95** | 0.89 | **0.92** | 0.93 | 0.89 | 0.91 | 0.88 | **0.83** | **0.86** | **0.93** | **0.90** | **0.92** |
| | DBA-I | 0.94 | **0.89** | 0.91 | 0.92 | **0.94** | **0.93** | 0.87 | **0.83** | 0.85 | 0.91 | 0.86 | 0.88 |
| Qwen3-8B | self-consistency | 0.92 | 0.10 | 0.18 | **0.96** | 0.38 | 0.54 | **0.96** | 0.10 | 0.17 | **0.97** | 0.46 | 0.63 |
| | DBA-A | 0.96 | **0.90** | **0.93** | 0.94 | 0.92 | 0.93 | 0.92 | **0.82** | **0.87** | 0.92 | **0.91** | **0.91** |
| | DBA-I | **0.99** | 0.67 | 0.80 | 0.94 | **0.95** | **0.94** | 0.90 | 0.74 | 0.81 | 0.92 | 0.87 | 0.89 |

Table 19: Comparison of self-consistency vs. cross-interface disagreement for error detection. self-consistency achieves high precision when it abstains but has extremely low recall, detecting only a small fraction of errors. DBA-A and DBA-I achieve similar precision while detecting 4–5$\times$ more errors. **Bold** indicates best value in each column within each model group.

**System prompt:**
You are a careful evaluator of question decompositions. Focus on semantic coverage, not exact wording.

Output must be strict JSON only (no markdown, no commentary, no explanation).

**User prompt:**
Task:

1) Decide if the two decompositions are semantically equivalent, that is, if they would lead to the same final answer given the question. Return `equivalent_final` as 1 (yes) or 0 (no).

2) Count how many gold hops are covered by some hop in the model decomposition. Coverage means the hop asks for the same intermediate information (allow paraphrase and minor scope differences).

Return JSON with the following fields:

```
- equivalent_final: 0 or 1
- matches: integer (# gold hops covered by some model hop)
```

Dataset: {dataset}
Question: {question}

Gold DSL:
{gold_dsl}

Gold DSL lines:
{gold_block}

Model DSL:
{model_dsl}

Model DSL lines:
{model_block}

Table 20: LLM-as-judge prompt for evaluating equivalence between a model-generated DSL decomposition and the gold reference. The judge returns a strict JSON object with `equivalent_final` (0/1) and `matches` (integer hop coverage), which feeds the precision, recall, F1, and hop-ratio statistics reported in Section D.4.

| | GPT 5.1 | | | Gemini Pro | | |
|---|---|---|---|---|---|---|
| **Dataset** | **Helped** | **Hurt** | **Both** | **Helped** | **Hurt** | **Both** |
| Bamboogle | 38.1 | 19.0 | 42.9 | 25.0 | 10.0 | 65.0 |
| Mintaka | 17.0 | 14.9 | 68.1 | 17.6 | 17.6 | 64.7 |
| HotpotQA | 18.7 | 21.3 | 60.0 | 17.3 | 17.3 | 65.4 |
| CRAG | 9.7 | 11.3 | 79.0 | 10.4 | 16.4 | 73.1 |
| FRAMES | 13.9 | 18.8 | 67.3 | 17.3 | 20.2 | 62.5 |
| MuSiQue | 13.3 | 18.2 | 68.5 | 17.2 | 15.0 | 67.8 |
| **Overall** | **15.3** | **17.6** | **67.1** | **16.7** | **16.7** | **66.5** |

Table 21: Breakdown of inconsistent cases (%) for frontier models. *Helped*: Direct wrong, Assistive correct. *Hurt*: Direct correct, Assistive wrong. *Both*: Both regimes wrong (knowledge gap). Across both models, ∼67% of disagreements reflect knowledge gaps where decomposed prompting cannot help.

| Example | Case | Question and Reasoning Trace |
|---|---|---|
| **Ex. 1** | **Decomposition Helped** | **Model:** `GPT 5.1` **Dataset:** BAMBOOGLE<br>**Q:** When was the first location of the world's largest coffeehouse chain opened?<br>**Chain:** `Q1:` "What is the world's largest coffeehouse chain?" → *Starbucks*<br>`Q2:` "When was the first location of {answer_1} opened?" → *1971-03-30*<br>**Direct:** *1971-03-31* ✗ **Assistive:** *1971-03-30* ✓ **Gold:** March 30, 1971 |
| **Ex. 2** | **Decomposition Helped** | **Model:** `GPT 5.1` **Dataset:** BAMBOOGLE<br>**Q:** In what year did work begin on the second longest road tunnel in the world?<br>**Chain:** `Q1:` "What is the second longest road tunnel in the world?" → *Yamate Tunnel*<br>`Q2:` "In what year did work begin on the {answer_1}?" → *1992*<br>**Direct:** *2002* ✗ **Assistive:** *1992* ✓ **Gold:** 1992 |
| **Ex. 3** | **Decomposition Helped** | **Model:** `gemini-2.5-pro` **Dataset:** FRAMES<br>**Q:** How many more career home runs did the MLB player who had the highest slugging percentage in 1954 have than the player who was the first African American to play in MLB?<br>**Chain:** `Q1:` "Which MLB player had the highest slugging percentage in 1954?" → *Willie Mays*<br>`Q2:` "How many career home runs did {answer_1} have?" → *660*<br>`Q3:` "Who was the first African American to play in MLB?" → *Jackie Robinson*<br>`Q4:` "How many career home runs did {answer_3} have?" → *141* `Q5:` $660 - 141 = 519$<br>**Direct:** *142* ✗ **Assistive:** *519* ✓ **Gold:** 519 |
| **Ex. 4** | **Decomposition Hurt** | **Model:** `gemini-2.5-pro` **Dataset:** MINTAKA<br>**Q:** What's the tallest building in the state where Yosemite National Park is located?<br>**Chain:** `Q1:` "In which state is Yosemite National Park located?" → *California*<br>`Q2:` "What is the tallest building in {answer_1}?" → *Salesforce Tower*<br>**Direct:** *Wilshire Grand Center* ✓ **Assistive:** *Salesforce Tower* ✗ **Gold:** Wilshire Grand Center<br>*Chain retrieves outdated information; Wilshire Grand Center surpassed Salesforce Tower in 2017.* |
| **Ex. 5** | **Decomposition Hurt** | **Model:** `gemini-2.5-pro` **Dataset:** HOTPOTQA<br>**Q:** When was the artist who did the cover art of The Savage Frontier born?<br>**Chain:** `Q1:` "Who did the cover art for 'The Savage Frontier'?" → *Keith Parkinson*<br>`Q2:` "When was {answer_1} born?" → *1958-10-22*<br>**Direct:** *1948-08-05* ✓ **Assistive:** *1958-10-22* ✗ **Gold:** August 5, 1948<br>*Wrong artist in the first hop (correct artist is Larry Elmore); the birth date retrieved is for the wrong person.* |
| **Ex. 6** | **Knowledge Gap** | **Model:** `GPT 5.1` **Dataset:** BAMBOOGLE<br>**Q:** The most populous city in Punjab is how large (area wise)?<br>**Chain:** `Q1:` "What is the most populous city in Punjab?" → *Lahore*<br>`Q2:` "What is the area of {answer_1}?" → *1772 km$^2$*<br>**Direct:** *159 km$^2$* ✗ **Assistive:** *1772 km$^2$* ✗ **Gold:** 310 km$^2$<br>*Both regimes retrieve incorrect area figures; the discrepancy reflects inconsistent world knowledge about city boundaries.* |
| **Ex. 7** | **Knowledge Gap** | **Model:** `gemini-2.5-pro` **Dataset:** MUSIQUE<br>**Q:** Who is the father of the father of Anwer Ali?<br>**Chain:** `Q1:` "Who is the father of Anwer Ali?" → *Unknown*<br>`Q2:` "Who is the father of {answer_1}?" → *Unknown*<br>**Direct:** *the grandfather of Anwer Ali* ✗ **Assistive:** *Unknown* ✗ **Gold:** Ahmad Shah Bahadur<br>*Obscure historical figure; both regimes fail to retrieve the correct Mughal lineage (Anwer Ali → Muhammad Shah → Ahmad Shah Bahadur).* |

Table 22: Examples of inconsistencies from frontier models: `GPT 5.1` and `gemini-2.5-pro`.