KnowledgeBerg: Evaluating Systematic Knowledge Coverage and Compositional Reasoning in Large Language Models

Xiao Zhang
University of Groningen
xiao.zhang@rug.nl

Qianru Meng
LIACS, Leiden University
q.r.meng@liacs.leidenuniv.nl

Yongjian Chen
University of Groningen
yongjian.chen@rug.nl

Yumeng Wang
LIACS, Leiden University
y.wang@liacs.leidenuniv.nl

Johan Bos
University of Groningen
johan.bos@rug.nl

Abstract

Many real-world questions appear deceptively simple yet implicitly demand two capabilities: (i) systematic coverage of a bounded knowledge universe and (ii) compositional set-based reasoning over that universe, a phenomenon we term "the tip of the iceberg." We formalize this challenge through two orthogonal dimensions: *knowledge width*, the cardinality of the required universe, and *reasoning depth*, the number of compositional set operations. We introduce KNOWLEDGEBERG, a benchmark of 4,800 multiple-choice questions derived from 1,183 enumeration seeds spanning 10 domains and 17 languages, with universes grounded in authoritative sources to ensure reproducibility. Representative open-source LLMs demonstrate severe limitations, achieving only 5.26–36.88 F1 on universe enumeration and 16.00–44.19 accuracy on knowledge-grounded reasoning. Diagnostic analyses reveal three stages of failure: *completeness*, or missing knowledge; *awareness*, or failure to identify requirements; and *application*, or incorrect reasoning execution. This pattern persists across languages and model scales. Although test-time compute and retrieval augmentation yield measurable gains—up to 4.35 and 3.78 points, respectively—substantial gaps remain, exposing limitations in how current LLMs organize structured knowledge and execute compositional reasoning over bounded domains. The dataset is available at https://huggingface.co/datasets/2npc/KnowledgeBerg.

1 Introduction

Large language models have become integral to applications spanning scientific research, industry, and everyday tasks (Raza et al., 2025). Despite the proliferation of benchmarks, most evaluations focus on isolated knowledge points, testing a single fact, definition, or concept (Yang et al., 2018; Hendrycks et al., 2021; Wang et al., 2024; Zhang et al., 2025), or on linear, step-by-step reasoning that evaluates a single chain of thought (Cobbe et al., 2021; Lightman et al., 2023; Lai and Nissim, 2024). While these benchmarks are effective for measuring point-wise recall and basic reasoning, they rarely assess whether models can systematically cover entire knowledge domains and perform compositional reasoning across them.

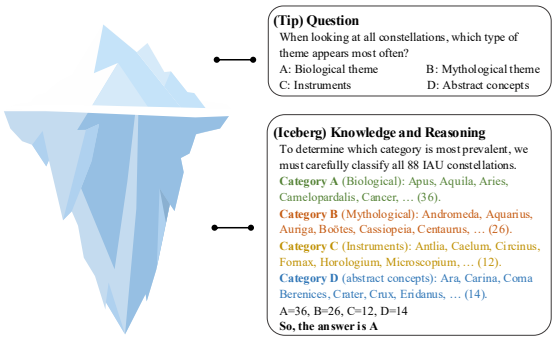

Figure 1: Illustration of the tip-of-the-iceberg phenomenon. A surface-simple question (tip) implicitly depends on a *bounded universe* and *compositional set-based reasoning* (submerged iceberg). In this example, answering requires knowledge of the 88 IAU-recognized constellations and a sequence of set operations: classify constellations, aggregate groups, count members, and compare cardinalities.

This gap has significant implications in high-stakes domains such as education, healthcare, and life sciences, where success requires structured domain knowledge and reasoning rather than basic pattern matching. For instance, clinical diagnosis requires enumerating all plausible conditions before differential analysis; missing a critical candidate can lead to incorrect treatment decisions. More broadly, many seemingly simple questions conceal substantial latent requirements: a concise prompt may depend on a large, well-defined universe of entities, together with multi-step operations over that universe. We refer to this as the *tip-of-the-iceberg* phenomenon.

Motivated by this limitation, we introduce a

Findings of the Association for Computational Linguistics: ACL 2026, pages 11272–11286

class of evaluation questions that combines knowledge coverage with compositional reasoning. Concretely, we characterize each question using two dimensions: *knowledge width*, defined as the cardinality of the implicit universe required for a correct solution, and *knowledge-grounded reasoning depth*, defined as the number of compositional set operations to derive the answer. This yields "iceberg" questions whose difficulty scales with width $\times$ depth. As illustrated in Figure 1, answering a seemingly simple "tip" question about constellations requires access to the full constellation set and the ability to execute multiple operations, such as classification, aggregation, counting, and comparison. To make these latent universes explicit and reproducible, we curate bounded domain knowledge from verifiable sources, such as domain ontologies, textbooks, governmental documents and Wikipedia. Building on these universes, we design multiple families of questions that implicitly embed the required set, so that correct answers depend on both systematic knowledge coverage and set-based compositional reasoning. We call this benchmark KNOWLEDGEBERG, inspired by the iceberg metaphor.

We compare KNOWLEDGEBERG with existing benchmarks using *Iceberg Gap*, which measures, across benchmarks, the gap between surface simplicity and latent width/depth demands. We further diagnose failure modes across representative LLMs and evaluate mitigation strategies on test-time computing. Overall, our contributions are:

- **Problem formulation and benchmark.** We formalize *tip-of-the-iceberg* evaluation, where a concise question implicitly specifies a bounded universe and demands compositional set reasoning, and introduce KNOWLEDGEBERG, a benchmark designed to jointly test bounded-universe coverage and set-based reasoning across 10 domains and 17 languages (4,800 questions) (§3).
- **Cross-benchmark metric.** We propose *Iceberg Gap*, a metric for comparing benchmarks by quantifying the gap between surface-form simplicity and latent requirements in knowledge width and reasoning depth (§3.3).
- **Diagnostics and mitigation studies.** We provide large-scale evaluations and targeted analyses that localize failures into *completeness–awareness–application*, and we quantify gains from representative approaches including test-time compute and retrieval augmentation (§4, §5).

2 Related Work

2.1 Knowledge Evaluation Benchmarks

Most LLM benchmarks evaluate knowledge via isolated fact correctness. Exam-style benchmarks—MMLU (Hendrycks et al., 2021), MMLU-Pro (Wang et al., 2024), MedQA (Jin et al., 2021), and AI2 ARC (Clark et al., 2018)—test factual recall across domains. Open-domain QA benchmarks such as TriviaQA (Joshi et al., 2017), Natural Questions (Kwiatkowski et al., 2019), WebQuestions (Berant et al., 2013), and PopQA (Mallen et al., 2023), together with knowledge-intensive suites such as KILT (Petroni et al., 2021), evaluate answering with implicit or explicit evidence. Multi-step reasoning benchmarks—HotpotQA (Yang et al., 2018), GSM8K (Cobbe et al., 2021), and CommonsenseQA (Talmor et al., 2019)—stress inference chaining, but rarely require models to systematically cover a *verifiably bounded* domain universe when forming an answer. KNOWLEDGEBERG targets precisely this setting: questions whose answers require set-based composition over a complete, implicit bounded universe.

A smaller body of work evaluates *coverage* or *sufficiency*. AmbigQA (Min et al., 2020) enumerates interpretations; ASQA (Stelmakh et al., 2022) and ELI5 (Fan et al., 2019) assess aspect coverage; and ProxyQA (Tan et al., 2024) verifies sufficiency. Related structured settings include knowledge-base and KG QA (ComplexWebQuestions (Talmor and Berant, 2018), MetaQA (Zhang et al., 2018), KQA Pro (Cao et al., 2022)) and compositional reasoning frameworks such as Chameleon (Lu et al., 2023) and analyses of compositional failures in LLMs (Li et al., 2024). These settings are closely related but typically expose the knowledge source, schema, or reasoning scaffold more explicitly. KNOWLEDGEBERG differs by focusing on bounded-universe reasoning, where the question gives no indication that complete-set coverage is required.

Recent work has also proposed benchmark-level notions of question difficulty. Retrieval Complexity (Gabburo et al., 2024) characterizes how difficult it is to retrieve the evidence needed to answer a question, whereas Iceberg Gap characterizes how strongly a question conceals its universe-coverage and compositional requirements beneath a simple surface form. These perspectives are complementary: a question may be easy to retrieve evidence for yet still exhibit a large Iceberg Gap if the prompt under-specifies the need for complete-set coverage.

2.2 Methods for Improving LLM Reasoning and Knowledge

Prior work improves LLM reasoning reliability by eliciting and validating intermediate steps, including chain-of-thought prompting (Wei et al., 2022), self-consistency (Wang et al., 2023), multidimensional consistency (Lai et al., 2025), and self-verification and refinement (Weng et al., 2023; Madaan et al., 2023). A complementary line augments parametric knowledge with external evidence via retrieval-augmented generation (Lewis et al., 2020), with related interventions including query rewriting, decomposition, and abstraction before retrieval to partially externalize latent requirements and improve evidence acquisition (Chan et al., 2024; Zheng et al., 2024).

3 KNOWLEDGEBERG

3.1 Data Sources and Construction

We manually curated structured knowledge spanning 10 domains (full taxonomy and per-domain counts in Appendix A.1) from five source types: (i) domain ontologies (e.g., Gene Ontology (Ashburner et al., 2000), NCBI Taxonomy (Schoch et al., 2020)); (ii) authoritative textbooks (e.g., *Fundamentals of Physics* (Halliday et al., 2013)); (iii) governmental and institutional documents (e.g., United Nations materials); (iv) official reference websites (e.g., the International Astronomical Union and the National Basketball Association); and (v) Wikipedia as a supplementary source when canonical lists are stable and well-cited. We exclude topics with factual disputes or lacking authoritative consensus (e.g., territorial claims or rapidly evolving scientific hypotheses) to ensure stability and reproducibility.

Bounded-set knowledge (EQ–EA). For each domain $d \in \mathcal{D}$, we construct enumeration questions in an Enumeration-Question/Enumeration-Answer (EQ–EA) format. Each enumeration question EQ_i is paired with a ground-truth answer set $\mathrm{EA}_i = \{a_1, \ldots, a_{w_i}\}$, where $w_i = |\mathrm{EA}_i|$ defines the *knowledge width*. We define the *bounded universe* required by downstream reasoning as:

$$U_i \triangleq \mathrm{EA}_i = \{a_1, \ldots, a_{w_i}\}. \tag{1}$$

From knowledge to compositional reasoning. Building on EQ–EA pairs, we generate knowledge-grounded reasoning questions (KRQs)—multiple-choice questions with 8–10 options—that require models to reason over the bounded set U_i. Each KRQ Q_j is instantiated by applying a short reasoning program $\mathbf{o}_j = \langle o_1, \ldots, o_{d_j} \rangle$ to a base enumeration question EQ_i, where each step o_t is drawn from a controlled set-operation schema including comparison, aggregation, constraint checking, filtering, counting, complementation, partitioning, ordering, union, and set equality. The program length $d_j = |\mathbf{o}_j|$ defines the reasoning depth of the KRQ.

$$Q_j = \mathrm{apply}(\mathrm{EQ}_i,\ \mathbf{o}_j). \tag{2}$$

We define *reasoning depth* as the longest-path length of an operator graph. For KNOWLEDGEBERG items, the validated operator program is a linear chain $\mathbf{o}_j = \langle o_1, \ldots, o_{d_j} \rangle$, so the longest path of the operator graph reduces to the sequence length, i.e., $\mathrm{Depth}(Q_j) = d_j = |\mathbf{o}_j|$. For example:

$$\text{Answer} = \mathrm{count}(\mathrm{filt}(U_i,\ \phi))\,, \tag{3}$$

where ϕ is a predicate. In practice, we use GPT-4o with structured prompts specifying (EQ_i, o_j) to generate candidates, and validate them through a three-stage pipeline (review $\rightarrow$ adjudication $\rightarrow$ revision), in which two annotators independently assess candidate validity and disagreements are resolved through adjudication before final revision. Examples of EQ–KRQ pairs are provided in Appendix A.2.

Iceberg structure. Each instance exposes a visible "tip" (the surface-level KRQ Q_j) while depending on a latent *iceberg* characterized by knowledge width $w_i = |U_i|$ and reasoning depth $d_j = |\mathbf{o}_j|$.

Translations. We translate all 1,183 EQs and 4,800 KRQs *from English* into 16 additional languages using Google Translate, and apply automatic sanity checks (language identification and script validation) to ensure format consistency.

3.2 Dataset Statistics

Table 1 summarizes KNOWLEDGEBERG. The benchmark contains 4,800 KRQs derived from 1,183 EQs, covering 10 domains and 17 languages. On average, each EQ yields 4.06 KRQs through different operator sequences, ensuring diverse reasoning patterns. Each KRQ has 8–10 options with exactly one correct answer, corresponding to a random-guess baseline between 10.0% and 12.5%, depending on the item.

Category	Value
Scale & Coverage	
Knowledge-grounded reasoning questions (KRQs)	4,800
Enumeration questions (EQs)	1,183
Domains	10
Languages	17
KRQs per EQ (avg)	4.06
Answer options (min / max)	8 / 10
Iceberg Structure	
Knowledge width (avg / med / min / max)	18.02 / 12 / 5 / 161
Reasoning depth (avg / med / min / max)	3.68 / 4 / 2 / 8

Table 1: Statistics of KNOWLEDGEBERG. Knowledge width is $|\mathrm{EA}_i|$. Reasoning depth is the longest-path length of the operator graph; for KNOWLEDGEBERG items this equals $|\mathbf{o}_j|$, since all validated programs are linear chains.

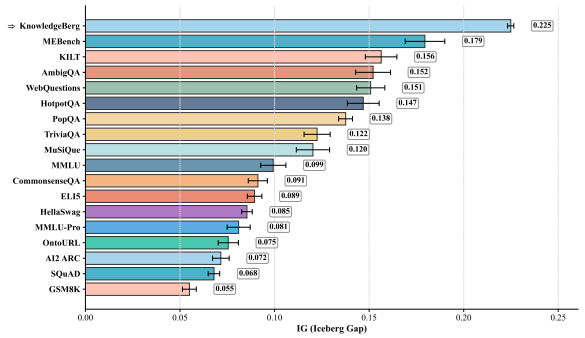

Figure 2: Iceberg Gap across benchmarks (mean $\pm$ 95% CI), estimated from $N = 500$ items sampled uniformly at random per benchmark.

3.3 Iceberg Gap: A Metric for Hidden Complexity

To quantify the "tip versus mass" contrast, we introduce **Iceberg Gap** (IG), a score that is high when a question appears surface-simple yet implicitly requires broad universe coverage and deep compositional reasoning.

Metric design. We represent each item as (U, Φ, q, a), where U is the required bounded universe, Φ the latent set-operation program, q the surface question, and a the correct answer. In KNOWLEDGEBERG, U is the ground-truth set EA_i; for benchmarks without explicit universes, we estimate $|U|$ via a uniform two-step prompting procedure (domain identification $\rightarrow$ cardinality estimation).

IG comprises three components: (1) *Surface Simplicity* S, computed from syntactic and semantic features; (2) *Knowledge Width* W, with raw width $W_{\mathrm{raw}} = \log(1 + |U|)$ to reduce sensitivity to heavy-tailed universes; and (3) *Reasoning Depth* D, defined as the longest-path length of an operator graph over a fixed operator vocabulary. We treat surface simplicity as an *item property* rather than a model-specific surprisal measure, so that IG remains comparable across both open and closed models. For KNOWLEDGEBERG, Φ is known by construction and D reduces to the validated chain length; for external benchmarks, we estimate a minimal operator graph via a fixed LLM-based extraction procedure (Appendix B). We acknowledge potential circularity in this external estimation and mitigate it by using the same prompt and model across all benchmarks, validating extracted depths on annotated subsets (Appendix B).

To enable cross-benchmark comparison, we normalize S, W, and D on a single pooled reference set Ω by mapping each raw value to its pooled percentile (mid-rank for ties), yielding scores in $(0, 1)$ (Appendix B). Within-benchmark normalization is intentionally avoided: a raw width of 10 may be extreme in a narrow benchmark but typical in a broader one, so benchmark-relative scaling would undermine cross-dataset comparability. Percentile normalization preserves relative ordering while reducing sensitivity to heterogeneous scales and heavy-tailed distributions.

Aggregation. We combine the three components via a geometric mean:

$$\mathrm{IG}(q) = \big(S(q) \cdot W(q) \cdot D(q)\big)^{1/3} \in (0, 1). \quad (4)$$

The geometric mean reflects the conjunctive nature of hidden complexity: a question should score high only when it is *simultaneously* surface-simple, broad in universe coverage, and deep in reasoning. An additive aggregation would allow a single strong component to mask deficits in the others, whereas the geometric mean penalizes imbalance.

Benchmark comparison. Figure 2 reports IG as a descriptive characterization of hidden complexity across benchmarks. KNOWLEDGEBERG attains the highest IG (0.225), reflecting the largest measured gap between surface simplicity and latent width/depth requirements. Open-domain and multi-step QA benchmarks (WebQuestions, PopQA, KILT, HotpotQA) also exhibit relatively high gaps, while exam-style and reading comprehension benchmarks (MMLU, CommonsenseQA, SQuAD) score lower, indicating weaker joint pressure on bounded-set coverage and set-based composition. GSM8K attains the lowest IG (0.055).

4 Diagnosing LLMs on KnowledgeBerg

In this section, we evaluate representative LLMs on KNOWLEDGEBERG to diagnose where failures arise in the knowledge-to-answer process. We first report overall performance and show that enumeration completeness and KRQ accuracy are only weakly coupled. Motivated by this gap, we introduce a three-stage diagnostic framework: **completeness**, **awareness**, and **application**. We examine this framework from two complementary angles: correlational analyses of structural difficulty factors (§4.3) and controlled inference-time prompt variants that selectively provide knowledge or encourage reasoning to more directly stress each stage (§4.4). We then validate the findings cross-lingually (§4.5).

4.1 Settings

We evaluate representative open-source LLMs from the Qwen (Yang et al., 2025), LLaMA (Dubey et al., 2024), Mistral, Phi, and Gemma (Team et al., 2025) families on KNOWLEDGEBERG. Models answer enumeration questions (EQs) and knowledge-grounded reasoning questions (KRQs) using a unified zero-shot prompt with greedy decoding (temperature $= 0.0$, top-$p = 1.0$; Appendix C.1).

We report Universe F1 for EQs and accuracy for KRQs. For KRQs, we deterministically extract the final answer option from the `\boxed{}` span using a regular expression and count unparseable outputs as incorrect. For EQs, we compute set-level Universe F1 using a hybrid protocol: we first apply rule-based normalization and matching, and invoke an LLM judge (Qwen3-30B-A3B-Instruct-2507; Yang et al., 2025) only for the remaining unmatched cases (Appendix C.2).

4.2 Overall Performance: A Puzzling Ceiling

Table 2 reveals two patterns across models of varying architectures and scales. First, completeness on EQs remains consistently limited. Universe F1 ranges from 5.26 to 36.88, with most models remaining below 30 and only Llama-3.3-70B-Instruct and Mistral-Small-24B-Instruct-2501 exceeding that level. This indicates that even when the relevant knowledge is constrained to a bounded, structured set, faithfully enumerating it remains difficult. Second, KRQ accuracy also remains modest overall, ranging from 16.00 to 44.19 despite the multiple-choice format of KRQs, which should in principle reduce the search space.

Model	Universe F1	KRQ Acc.
Qwen3.5-0.8B	5.26	24.38
Qwen3.5-9B	11.60	36.35
Qwen3.5-27B	12.16	44.19
Qwen3.5-35B-A3B	14.29	41.71
Mistral-Small-24B-Instruct-2501	33.26	19.88
Phi-4-mini-instruct	9.75	16.00
Phi-4-14B	25.17	38.79
Llama-3.3-70B-Instruct	36.88	35.90
Gemma-3-4B-it	16.51	29.42
Gemma-3-12B-it	23.82	31.06
Gemma-3-27B-it	27.32	32.81

Table 2: Performance on KNOWLEDGEBERG (English). Universe F1 measures completeness on EQs, and KRQ accuracy measures knowledge-grounded reasoning.

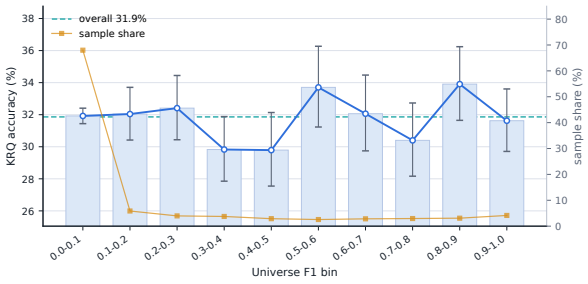

Figure 3: KRQ accuracy versus enumeration quality (Universe F1), with instances binned by F1 and aggregated across models. Instance-level correlations are near-zero (Spearman $\rho = 0.0023$; Kendall $\tau = 0.0020$).

More importantly, the two metrics are not aligned. Some models with relatively strong enumeration completeness perform poorly on KRQs; for example, Mistral-Small-24B reaches 33.26 Universe F1 but only 19.88 KRQ accuracy. Conversely, some models with limited completeness still achieve comparatively strong KRQ performance: Qwen3.5-27B attains only 12.16 Universe F1 yet reaches the highest KRQ accuracy at 44.19. This mismatch suggests that bounded-universe coverage alone is insufficient to explain model behavior on KNOWLEDGEBERG.

The Decoupling Puzzle. The weak coupling between EQ and KRQ performance becomes even clearer at the instance level. Intuitively, higher Universe F1 should translate into higher KRQ accuracy: if more of the required knowledge is available, the model should reason more reliably. However, Figure 3 shows that this relationship is essentially absent. After binning instances by Universe F1 and aggregating across models, KRQ accuracy varies only weakly and non-monotonically across bins, and the instance-level correlations are near

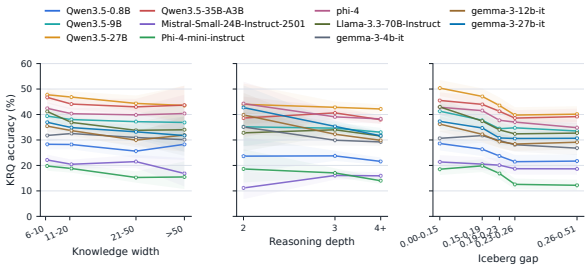

Figure 4: KRQ accuracy across bins of knowledge width, reasoning depth, and Iceberg Gap (English). Each line denotes one model.

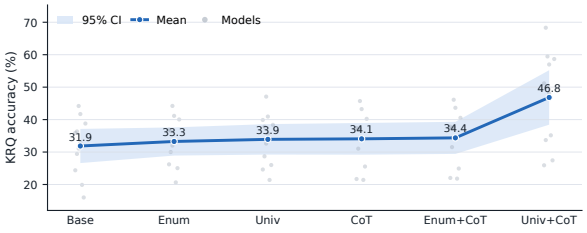

Figure 5: Effect of diagnostic prompt variants on KRQ accuracy (English). Thin lines show individual models; the highlighted curve shows the mean across models, with 95% confidence intervals.

zero (Spearman $\rho = 0.0023$; Kendall $\tau = 0.0020$). Thus, limited knowledge coverage is not the only source of failure: even when a model recovers more of the relevant universe, it may still fail to convert that knowledge into the correct answer.

A Three-Stage Hypothesis. This decoupling suggests that failures extend beyond knowledge coverage alone. Even when relevant facts are available, a model may still fail in two additional ways: it may not identify the question's implicit knowledge requirements, or it may execute the required reasoning operations incorrectly. We therefore decompose the knowledge-to-answer process into three stages: **completeness** (is the required universe knowledge available in the model?), **awareness** (does the model correctly identify what knowledge the question requires?), and **application** (does it correctly apply the identified knowledge through the necessary reasoning operations to arrive at the correct answer?). We examine this hypothesis in two ways: analyses of how accuracy varies with structural difficulty factors (§4.3) and controlled inference-time prompt variants that test whether providing knowledge or structuring reasoning can isolate each stage's contribution (§4.4).

4.3 Correlational Analysis: Structural Difficulty Factors

To localize where failures arise, we analyze how KRQ accuracy varies with three structural properties of questions (Figure 4). **Knowledge width** (the number of required universe elements) captures the size of the knowledge space and is most closely associated with *completeness*. **Reasoning depth** (the number of reasoning steps) captures the length of the inference chain over that universe and is most closely associated with *application*. **Iceberg Gap (IG)** captures the mismatch between surface simplicity and hidden complexity, and is intended to probe *awareness*: whether models recognize the extensive latent requirements beneath concise questions.

Structural patterns reveal stage-associated bottlenecks. Figure 4 shows a consistent qualitative pattern across all three axes. As knowledge width increases from 6–10 to > 50 elements, KRQ accuracy generally declines across models, with the largest-width bucket being the most challenging. This pattern is consistent with *completeness*-related bottlenecks: larger universes are harder to recover exhaustively, which aligns with the low Universe F1 values in Table 2. Accuracy also tends to decrease as reasoning depth increases from 2 to 4+, suggesting more frequent *application*-related failures in longer compositional chains. Finally, higher IG bins are associated with systematically lower KRQ accuracy, which is compatible with persistent *awareness*-related failures: when a question appears surface-simple but conceals broader latent requirements, models often fail to recognize what the task actually demands.

4.4 Diagnostic Prompt Variants

While informative, the correlational analyses in §4.3 do not by themselves isolate causal mechanisms. We therefore conduct controlled prompt interventions that selectively manipulate *universe knowledge availability* and *reasoning scaffolding* to probe which stages of the knowledge-to-answer process are most sensitive to these changes (Figure 5).

Prompt variants. We compare BASE (the default KRQ prompt) with five variants. UNIV provides the gold universe, allowing us to probe how strongly performance depends on access to the complete bounded set. ENUM asks the model to list the *candidate* universe elements it considers relevant *before* choosing an option, aiming to externalize

evidence selection and probe whether making candidate knowledge explicit helps with requirement identification. CoT encourages step-by-step reasoning to probe whether structured reasoning reduces errors in executing the required operations. Finally, ENUM+CoT and UNIV+CoT examine whether these interventions interact.

Single interventions help only modestly; combining knowledge and reasoning helps substantially more. Figure 5 shows that all single interventions yield only limited gains over the BASE prompt on the model average. Mean KRQ accuracy rises from 31.9 under BASE to 33.3 with ENUM, 33.9 with UNIV, and 34.1 with CoT; ENUM+CoT reaches 34.4. By contrast, UNIV+CoT yields a much larger improvement, reaching 46.8.

Two observations follow. First, neither externalizing candidate knowledge nor applying a single intervention in isolation is sufficient to address the benchmark. Second, the strongest gains arise when complete universe knowledge is paired with explicit reasoning scaffolding, suggesting that missing knowledge and errors in reasoning execution interact in practice rather than behaving as fully separable sources of failure.

A complementary error analysis of strong closed-source models. Although UNIV+CoT yields the strongest improvement among our prompt variants, performance remains well below ceiling for open-source models. To obtain a complementary view of residual failures under a stronger model regime, we additionally evaluate two closed-source models under CoT: DEEPSEEK-CHAT achieves 48.39%, and GEMINI-3-FLASH reaches 65.24%. We then manually inspect 100 sampled GEMINI-3-FLASH errors (Appendix D), finding **completeness**-related errors in 42% of cases, **application**-related errors in 38%, **awareness**-related errors in 17%, and **artifacts** in 3%.

Because this manual study is conducted under CoT rather than UNIV+CoT, we treat it as complementary evidence rather than as a direct decomposition of the residual gap under the strongest prompt condition. Even so, the observed distribution is broadly consistent with the prompt-variant results: completeness- and application-related failures remain dominant, whereas the limited gains from ENUM suggest that simply eliciting candidate lists does not reliably resolve requirement-identification failures.

Model	All	High	Mid	Low	Std.
Qwen3.5-0.8B	23.01	23.88	22.17	19.05	3.08
Qwen3.5-9B	31.22	32.07	29.66	28.42	2.40
Qwen3.5-27B	**38.33**	**39.07**	**36.78**	**36.19**	2.43
Qwen3.5-35B-A3B	36.62	37.17	35.52	34.97	1.65
Mistral-Small-24B-Instruct-2501	23.32	23.17	24.17	23.00	5.32
Phi-4-mini-instruct	15.91	16.00	15.11	16.58	**1.04**
Phi-4-14B	32.85	34.53	30.38	26.46	3.22
Llama-3.3-70B-Instruct	30.63	31.87	26.73	29.03	3.36
Gemma-3-4B-it	27.08	27.26	26.84	26.32	1.54
Gemma-3-12B-it	28.73	28.94	29.15	26.84	1.20
Gemma-3-27B-it	30.65	31.02	30.26	29.02	1.18

Table 3: Multilingual KRQ accuracy (%) by computational resource tier. High includes Arabic, Chinese, Dutch, English, French, German, Hindi, Italian, Japanese, Korean, Russian, and Spanish; Mid includes Bengali, Thai, and Turkish; Low includes Swahili and Telugu. Std. indicates cross-lingual consistency.

	ΔE	ΔC	ΔU	ΔE+C	$\Delta_{C\|U}$
Avg.	+1.8	+3.1	+2.8	+3.5	+13.6
Range	[+0.6, +3.0]	[+1.8, +4.6]	[+1.5, +4.2]	[+2.0, +5.0]	[+10.2, +16.9]

Table 4: Cross-lingual gains of diagnostic prompt variants, measured in accuracy points. Range is computed across languages. ΔE, ΔC, and ΔU denote gains of ENUM, CoT, and UNIV over BASE; ΔE+C denotes the gain of ENUM+CoT over BASE; and $\Delta_{C|U}$ denotes (UNIV+CoT) − UNIV.

4.5 Cross-Lingual Validation

To test whether our findings generalize beyond English, we evaluate models on KNOWLEDGEBERG's multilingual partition.

Table 3 shows that the overall ranking is broadly preserved across languages: Qwen3.5-27B remains the strongest model overall (38.33) and in each resource tier. Performance typically degrades from high- to low-resource languages, but the magnitude of the drop is usually modest rather than catastrophic. The largest degradation is observed for Phi-4-14B (34.53 → 26.46), while several models remain comparatively stable across tiers; for example, Gemma-3-27B varies only from 31.02 to 29.02, and Phi-4-mini-instruct has the lowest cross-lingual variance overall (Std. 1.04). By contrast, Mistral-Small-24B-Instruct-2501 is the least stable model in this set (Std. 5.32).

Applying the same prompt variants across languages (Table 4) reveals the same qualitative pattern as in English. ENUM again has the smallest effect, yielding an average gain of +1.8 points. CoT and UNIV each provide somewhat larger but still modest improvements (+3.1 and +2.8 points on average, respectively), while ENUM+CoT reaches +3.5 points. By far the largest gain arises when

Model	Base	Self-Consistency			Self-Refine			Self-Verifier			RAG		
		sc=4	sc=8	sc=16	k=1	k=2	k=3	n=4	n=8	n=16	dense	hybrid	rerank
Qwen3.5-0.8B	24.38	25.76	26.54	**27.18**	24.72	21.64	20.98	25.88	26.42	26.88	26.34	26.78	26.61
Qwen3.5-9B	36.35	37.82	38.34	**39.05**	36.71	33.02	32.41	37.92	38.41	38.95	38.37	38.74	38.58
Qwen3.5-27B	44.19	45.72	46.31	**47.19**	44.08	40.76	39.91	45.84	46.38	46.89	46.42	46.69	46.55
Qwen3.5-35B-A3B	41.71	43.14	43.88	**44.78**	41.52	37.95	37.22	43.21	43.91	44.46	43.88	44.18	44.02
Mistral-Small-24B	19.88	21.91	23.05	**23.98**	19.74	16.42	15.83	22.03	23.16	23.78	22.84	23.28	23.05
Phi-4-mini-instruct	16.00	18.22	19.08	**20.35**	16.38	13.85	13.12	18.35	19.42	20.06	19.31	19.78	19.56
Phi-4-14B	38.79	40.18	40.92	**41.68**	38.62	35.41	34.78	40.24	40.88	41.42	40.84	41.17	41.03
Llama-3.3-70B-Instruct	35.90	37.42	38.13	**38.91**	36.04	33.18	32.76	37.55	38.20	38.71	38.09	38.45	38.28
Gemma-3-4B-it	29.42	31.08	31.74	**33.12**	29.66	26.31	25.70	31.22	31.88	32.43	31.73	32.10	31.91
Gemma-3-12B-it	31.06	33.02	33.74	**35.11**	31.58	28.27	27.35	33.16	34.02	34.72	34.31	34.62	34.48
Gemma-3-27B-it	32.81	34.42	35.08	**36.29**	33.04	30.12	29.58	34.55	35.24	35.86	35.44	35.72	35.61

Table 5: Testing-time enhancements for KRQ across models. Boldface indicates the best configuration per model; underlined indicates the second best.

CoT is added on top of UNIV: $\Delta_{C|U}$ averages +13.6 points across languages, with gains ranging from +10.2 to +16.9. Taken together, these multilingual results support the same overall diagnosis as in English: explicit candidate enumeration contributes comparatively little on its own, whereas the strongest improvements emerge when complete universe knowledge is paired with reasoning scaffolding.

5 Improve LLMs on KnowledgeBerg

Building on the diagnosis in §4, we investigate how to improve performance on KNOWLEDGEBERG by targeting the bottlenecks identified above.

5.1 Testing-Time Compute

We evaluate four families of testing-time strategies. **Self-consistency** samples multiple trajectories and aggregates them by majority vote, primarily targeting instability in *application*. **Self-refinement** alternates *Critique* and *Revise* to iteratively edit solutions using self-generated feedback, but may become counterproductive when critiques reinforce incorrect premises. **Self-verification** decouples proposal from selection: the model first generates candidate solutions and then uses a verifier to select the most plausible answer, aiming to improve final option selection. **Retrieval-augmented generation (RAG)** adds retrieved passages to the input to alleviate *completeness* gaps. We compare dense retrieval, hybrid retrieval (BM25+dense), and cross-encoder reranking. Our retrieval corpus is a lightweight collection derived from dataset construction (ontologies, curated sources, and official documents); we intentionally avoid large general-purpose databases (e.g., Wikidata) to keep retrieval effects interpretable and enable ablations.

Table 5 reveals a clear ranking among testing-time strategies. **Self-consistency** is the strongest and most reliable intervention: performance increases monotonically with more samples, and `sc=16` is the best configuration for all 11 models. This pattern suggests that a substantial portion of KRQ errors stems from inference-time instability in executing the required reasoning operations. **Self-verification** is consistently the next strongest method: performance also improves monotonically with verifier budget, and `n=16` is the second-best configuration for all 11 models.

RAG provides robust but smaller improvements. All three retrieval variants outperform the base model for all 11 models, and hybrid retrieval is consistently the strongest RAG configuration. This result suggests that retrieval quality matters more than simply appending external passages: combining sparse and dense retrieval yields more useful evidence than dense retrieval alone, while reranking offers only limited additional benefit. Finally, **self-refinement** is the least effective strategy. A single refinement step yields only marginal gains on a subset of models, whereas additional refinement rounds consistently reduce accuracy, often substantially. This pattern is consistent with the diagnosis in §4: when the initial reasoning trajectory is grounded in incomplete or mistaken universe knowledge, iterative self-critique may amplify those errors rather than correct them.

6 Conclusion

KNOWLEDGEBERG highlights a limitation of current large language models that is easy to miss in conventional evaluations: models may answer surface-simple questions poorly not because the questions are linguistically difficult, but because

they require both systematic coverage of a bounded knowledge universe and reliable reasoning over that universe. Across our experiments, current models remain far from robust in this setting, and the gap cannot be explained by missing knowledge alone. Instead, our analyses point to failures at multiple stages of the knowledge-to-answer process, including completeness, awareness, and application. Inference-time interventions improve performance, but they do not remove the underlying difficulty. We therefore view KNOWLEDGEBERG not only as a benchmark, but also as a testbed for studying how language models organize structured knowledge and reason over explicit sets—capabilities that are especially important in high-stakes domains.

7 Limitations

Universe Construction and Domain Coverage. Although universes are grounded in authoritative sources, the choice of 10 domains and 1,183 seeds is manually curated and may introduce sampling bias. We also exclude fast-changing topics for stability, so the benchmark may underrepresent dynamic knowledge domains. Finally, most universes are small (median size 12), and very large universes (e.g., thousands of entities) are limited.

Evaluation Protocol Dependencies. Universe F1 uses rule-based normalization plus an LLM judge for unresolved matches, which can introduce model-dependent bias. Our operator set for reasoning depth is also a simplification, and depth extraction for external benchmarks relies on LLM inference, posing a risk of circular reasoning despite standardized prompting and validation.

Multilingual Evaluation Scope. The 17-language setting relies on machine translation with automatic checks but lacks comprehensive native-speaker validation. Translation artifacts may change difficulty or introduce ambiguity, especially for low-resource languages, and we do not explicitly analyze cultural/linguistic variation in how knowledge is organized or enumerated.

References

Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather L. Butler, J. Michael Cherry, Allan Peter Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna E. Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. 2000. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Shulin Cao, Jiaxin Shi, Liangming Pan, Lunyiu Nie, Yutong Xiang, Lei Hou, Juanzi Li, Bin He, and Hanwang Zhang. 2022. KQA pro: A dataset with explicit compositional programs for complex question answering over knowledge base. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6101–6119, Dublin, Ireland. Association for Computational Linguistics.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. Rq-rag: Learning to refine queries for retrieval augmented generation. *Preprint*, arXiv:2404.00610.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, and Angela Fan. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Matteo Gabburo, Nicolaas Paul Jedema, Siddhant Garg, Leonardo F. R. Ribeiro, and Alessandro Moschitti. 2024. Measuring retrieval complexity in question answering systems. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14636–14650, Bangkok, Thailand. Association for Computational Linguistics.

David Halliday, Robert Resnick, and Jearl Walker. 2013. *Fundamentals of physics*. John Wiley & Sons.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Huiyuan Lai and Malvina Nissim. 2024. mCoT: Multilingual instruction tuning for reasoning consistency in language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12012–12026, Bangkok, Thailand. Association for Computational Linguistics.

Huiyuan Lai, Xiao Zhang, and Malvina Nissim. 2025. Multidimensional consistency improves reasoning in language models. *Preprint*, arXiv:2503.02670.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Zhaoyi Li, Gangwei Jiang, Hong Xie, Linqi Song, Defu Lian, and Ying Wei. 2024. Understanding and patching compositional reasoning in LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9668–9688, Bangkok, Thailand. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023. Chameleon: plug-and-play compositional reasoning with large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Mubashar Raza, Zarmina Jahangir, Muhammad Riaz, and Muhammad Sattar. 2025. Industrial applications of large language models. *Scientific Reports*, 15.

Conrad Schoch, Stacy Ciufo, Carol Hotton, Sivakumar Kannan, Rogneda Khovanskaya, Detlef Leipe, Richard McVeigh, Kathleen O'Neill, Barbara Robbertse, Shobha Sharma, Vladimir Soussov, John Sullivan, Lu Sun, Sean Turner, and Ilene Karsch-Mizrachi. 2020. Ncbi taxonomy: A comprehensive update on curation, resources and tools. *Database*, 2020.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Haochen Tan, Zhijiang Guo, Zhan Shi, Lu Xu, Zhili Liu, Yunlong Feng, Xiaoguang Li, Yasheng Wang, Lifeng Shang, Qun Liu, and Linqi Song. 2024. ProxyQA: An alternative framework for evaluating long-form text generation with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6806–6827, Bangkok, Thailand. Association for Computational Linguistics.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, and Morgane Rivière. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

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 *ICLR 2023*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. Mmlu-pro: a more robust and challenging multi-task language understanding benchmark. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2550–2575, Singapore. Association for Computational Linguistics.

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. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

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.

Xiao Zhang, Huiyuan Lai, Qianru Meng, and Johan Bos. 2025. Ontourl: A benchmark for evaluating large language models on symbolic ontological understanding, reasoning and learning. *Preprint*, arXiv:2505.11031.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J. Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. 2024. Take a step back: Evoking reasoning via abstraction in large language models. In *The Twelfth International Conference on Learning Representations*.

A Details of KNOWLEDGEBERG

A.1 Domains

Table 6 lists the 10 domains in KNOWLEDGEBERG.

ID	Domain	#EQs	#KRQs
01	Science, Technology & Computing	130	525
02	Life Science, Health & Food	311	1,234
03	Humanities, Culture & Language	200	794
04	Geography & Geopolitics	105	434
05	Economy, Finance & Industry	98	457
06	Time & Calendar	59	228
07	Sports	75	295
08	Environment & Climate	67	259
09	Transport & Infrastructure	58	248
10	Legal & Public Policy	80	326
All	**Total**	**1,183**	**4,800**

Table 6: Domain taxonomy and per-domain counts of enumeration questions (EQs) and knowledge-grounded reasoning questions (KRQs).

A.2 Example EQ–EA and Derived KRQs

As shown in Table 7 and 8, we provide representative EQ–EA seeds and their derived KRQs to illustrate how bounded universes and operator sequences are instantiated in concrete multiple-choice questions.

Pair 1 (Life Science, Health & Food).

Enumeration seed (EQ). Which parts of the alimentary tract are listed by the National Cancer Institute SEER Training materials?
Enumeration answer (EA). Mouth; Pharynx; Esophagus; Stomach; Small intestine; Large intestine; Rectum; Anus.

Derived iceberg KRQ (Operation: Ordering + Comparison). Using that validated SEER alimentary-tract closed set, compare the items before the stomach with the items after the stomach.
Options.

A. The pre-stomach subset is larger by exactly 1.
B. The post-stomach subset is larger by exactly 1.
C. The two subsets tie.
D. The pre-stomach subset is larger by exactly 2.
E. The post-stomach subset is larger by exactly 2.
F. The pre-stomach subset has only 2 items.
G. The post-stomach subset has 5 items.
H. Together the two subsets account for only 6 items.

Gold. B

Table 7: Seed-aligned EQ–EA and a derived KRQ.

A.3 Languages

We translate all questions from English into 16 additional languages using Google Translate. We categorize the 17 languages into three computational resource tiers based on their availability in contemporary LLM pretraining corpora: High-resource (Arabic, Chinese, Dutch, English, French, German, Hindi, Italian, Japanese, Korean, Russian, and Spanish), Mid-resource (Bengali, Thai, and Turkish), and Low-resource (Swahili and Telugu).

Pair 2 (Transport & Infrastructure).

Enumeration seed (EQ). Which important parts and terms are involved in Indian Railways switch assembly terminology?
Enumeration answer (EA). Stock rail joint; Stock rail; Tongue rail; Heel of switch; Switch angle at toe; Divergence at heel; Heel block; Throw at toe; Length of switch; Bolts.

Derived iceberg KRQ (Operation: Complement + Comparison). Within the validated Indian Railways switch-assembly terminology set, compare the pure-rail subset with its complement. The pure-rail subset is stock rail and tongue rail. Which conclusion is correct?
Options.

A. The pure-rail subset is larger by exactly 5.
B. The pure-rail subset is larger by exactly 6.
C. The two sides are equal in size.
D. The complement is larger by exactly 4.
E. The complement is larger by exactly 5.
F. The complement is larger by exactly 6.
G. The pure-rail subset is smaller by exactly 8.
H. The complement is larger by exactly 7.

Gold. F

Table 8: Seed-aligned EQ–EA and a derived KRQ.

B Iceberg Gap: Implementation Details

This appendix summarizes the details needed to reproduce Iceberg Gap (IG). We describe (i) how we compute surface simplicity and its corresponding surface complexity, (ii) how we define raw width and depth, and (iii) how we normalize all three components using a pooled percentile transformation.

B.1 Surface Simplicity and Surface Complexity

We first compute a bounded **surface simplicity** score $S_{\text{surf}}(q) \in [0, 1]$ from syntactic and semantic factors, and then convert it into a bounded **surface complexity** value $SC(q) \in [0, SC_{\max}]$.

Syntactic simplicity. Let n be the number of tokens in q, $\bar{s}$ the average sentence length (in tokens), and p the punctuation density (punctuation count divided by total characters). We define

$$S_{\text{len}}(q) = 1 - \min\left(1, \frac{n}{L_{\max}}\right), \quad (5)$$

$$S_{\text{sent}}(q) = 1 - \min\left(1, \frac{\bar{s}}{S_{\max}}\right), \quad (6)$$

$$S_{\text{punct}}(q) = 1 - \min\left(1, \frac{p}{P_{\max}}\right), \quad (7)$$

with fixed caps $L_{\max} = 40$, $S_{\max} = 30$, and $P_{\max} = 0.08$. We then average the three terms:

$$S_{\text{syn}}(q) = \tfrac{1}{3}\big(S_{\text{len}}(q) + S_{\text{sent}}(q) + S_{\text{punct}}(q)\big). \quad (8)$$

Semantic simplicity. We compute GPT-2 perplexity $\text{PP}(q)$ and normalize it with a fixed cap $\text{PP}_{\max} = 1000$:

$$S_{\text{sem}}(q) = 1 - \min\left(1, \frac{\text{PP}(q)}{\text{PP}_{\max}}\right). \quad (9)$$

Combining the two. We combine syntactic and semantic simplicity with $\alpha = 0.5$:

$$S_{\text{surf}}(q) = (1-\alpha)S_{\text{syn}}(q) + \alpha S_{\text{sem}}(q). \quad (10)$$

We then convert surface simplicity into surface complexity:

$$SC(q) = SC_{\max}\big(1 - S_{\text{surf}}(q)\big), \quad (11)$$

with $SC_{\max} = 30$. Equivalently, the raw simplicity component used by IG is

$$S_{\text{raw}}(q) = 1 - \frac{\text{clip}(SC(q), 0, SC_{\max})}{SC_{\max}} = S_{\text{surf}}(q), \quad (12)$$

up to clipping at the fixed cap.

B.2 Raw Width/Depth and Pooled Percentile

Raw width and depth. We define raw knowledge width as

$$W_{\text{raw}}(q) = \log(1 + |U|), \quad (13)$$

which reduces sensitivity to heavy-tailed universe sizes. We let $D_{\text{raw}}(q)$ denote reasoning depth: the annotated depth when available, and otherwise the longest-path length of an operator DAG extracted from the question. When $|U|$ or the operator DAG is not explicitly available, we estimate them with LLM prompts (Tables 9 and 10).

Task: Estimate the total number of distinct entities that must be fully known as a complete set to answer this question. Output only an integer.

Examples:
Q: Who was the first US president?
A: 1
Q: Which of the 20 amino acids are essential?
A: 20
Q: Which domain is most common across the 88 IAU constellations?
A: 88

Question: {question_text}
Answer:

Table 9: LLM prompt for knowledge width estimation ($|U|$).

Pooled percentile normalization. Let Ω be the pooled reference set of all benchmark items and let $N = |\Omega|$. For each raw component $x \in \{S_{\text{raw}}, W_{\text{raw}}, D_{\text{raw}}\}$, we map $x(q)$ to its pooled percentile using mid-rank for ties:

$$\tilde{x}(q) = \frac{\text{rank}_{\Omega}(x(q)) - 0.5}{N} \in (0,1). \quad (14)$$

Applying this transformation separately to the three raw components yields comparable normalized scores $S(q), W(q), D(q) \in (0,1)$.

Task: Minimize the given DAG by removing redundant nodes and shortening dependency chains.

Input:
Question: {question}
Nodes: {nodes_json}
Edges: {edges_json}
Final: {final_node}

Steps:
1. Remove redundant nodes
2. Merge combinable nodes
3. Shorten chains
4. Maintain correctness

Output (JSON only):
```
{"nodes": [...], "edges": [...], "final_node": "nK", "optimized": true}
```

Table 10: LLM prompt for reasoning depth estimation via operator DAG minimization.

Iceberg Gap. Finally, we define Iceberg Gap as the geometric mean of the three normalized components:

$$\text{IG}(q) = \big(S(q) \cdot W(q) \cdot D(q)\big)^{1/3} \in (0,1). \quad (15)$$

This form ensures that IG is high only when surface simplicity, knowledge width, and reasoning depth are all high.

C Experimental Setup and Evaluation Protocol

C.1 Experimental Setup

We run two evaluation suites: (i) inference on knowledge-grounded reasoning questions (KRQs) and (ii) enumeration evaluation on enumeration questions (EQs). Both suites use the same model pool and zero-shot prompting, but differ in their scoring protocols. All experiments in this appendix, as well as all subsequent inference experiments, are conducted on $4\times$ NVIDIA H100 94GB GPUs using vLLM.

Setting	EQ	KRQ
Scoring	Universe F1	Accuracy
Judge model	Qwen3-30B-A3B-Instruct-2507	–
Decoding	greedy	greedy
Temperature	0.0	0.0
Top-p	1.0	1.0
Max tokens	10,240	512
Batch size	adaptive	adaptive

Table 11: Experimental configuration.

C.2 LLM-as-Judge for Enumeration Questions

Enumeration outputs are set-valued, and strict string matching is brittle under aliases, abbreviations, translations, and minor formatting variation.

We therefore adopt a hybrid protocol: rule-based matching first, followed by LLM judging only for unresolved cases.

Parsing. We parse gold and predicted texts into item lists $\mathcal{G}$ and $\mathcal{P}$ by splitting on common delimiters (commas, semicolons, and newlines, including Chinese variants), stripping list markers (e.g., "1)", "-", and "•"), trimming whitespace, and deduplicating.

Normalization and rule matching. We apply lightweight normalization to both gold and predicted items, including Unicode NFKC normalization, whitespace collapsing, lowercasing (when applicable), and trimming trailing punctuation. We then perform exact matching under this normalized form.

LLM judging. For predicted items not matched by rules, we invoke an LLM judge to determine whether each predicted item is semantically equivalent to *any* gold item. The judge is instructed to accept aliases, abbreviations, translations, official versus common names, minor spelling variants, and punctuation, case, or diacritic differences, while rejecting loosely related items, different entities with similar names, or partial overlaps that alter meaning. If unsure, the judge must output `false`. We use Qwen3-30B-A3B-Instruct-2507 with greedy decoding (temperature $= 0.0$) and require strictly valid JSON outputs.

One-to-one matched pairs. After obtaining judge decisions, we construct a one-to-one matching between $\mathcal{P}$ and $\mathcal{G}$ as follows: (i) rule-matched pairs are fixed; (ii) for the remaining predicted items marked as matchable by the judge, we greedily assign each prediction to at most one gold item, and each gold item to at most one prediction, prioritizing exact or normalized matches when available. Let m denote the resulting number of matched (prediction, gold) pairs.

Universe F1. We compute set-level precision, recall, and F1 as

$$\text{Precision} = \frac{m}{|\mathcal{P}|}, \qquad \text{Recall} = \frac{m}{|\mathcal{G}|}, \qquad \text{F1} = \frac{2\,\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{16}$$

C.3 LLM Judge Prompt

Table 12 shows the prompt used for the LLM judge. Predicted items are provided as an indexed list, and the judge outputs a boolean decision for each index.

Task: For each predicted item, decide whether it is equivalent to *any* gold item. Accept aliases, abbreviations, translations, minor spelling variants, and punctuation, case, or diacritic differences; reject loosely related or different entities. If unsure, output `false`.

Gold items: {gold_items}
Predicted items (indexed): {pred_items_indexed}

Output (JSON only):
`{"pred_matches":[{"pred_index":0,"match":true},...]}`

Constraints: include every `pred_index` exactly once; do not add extra keys.

Table 12: LLM-as-judge prompt for matching predicted items to a gold list.

D Error Study of Gemini-3-Flash

This appendix reports CoT results for two closed-source models and characterizes residual error types through a manual analysis.

D.1 Closed-Source CoT Evaluation

Models and data. We evaluate DeepSeek-Chat and Gemini-3-Flash on English KRQs from KNOWLEDGEBERG under the same CoT setting used in §4.4.

Results. DeepSeek-Chat achieves 48.39% accuracy, and Gemini-3-Flash achieves 65.24% accuracy.

D.2 Manual Error Taxonomy for Gemini-3-Flash

Sample. To characterize residual failures in the strongest tested setting, we randomly sample $N = 100$ incorrect predictions from Gemini-3-Flash.

Labels. Each error is assigned to exactly one category:

- **Completeness:** the model lacks, omits, or states incorrect universe elements required for the question.
- **Awareness:** the model fails to recognize the required knowledge and instead focuses narrowly on the options or surface cues.
- **Application:** the model identifies relevant knowledge but applies an incorrect set operation or composition (e.g., wrong filtering, miscounting, or invalid aggregation/comparison).
- **Artifact:** parsing errors, ill-formed answers, or other output-format issues.

Category proportions. Across 100 randomly sampled Gemini-3-Flash CoT errors, the distribution is as follows: *Completeness* 42%, *Awareness* 17%, *Application* 38%, and *Artifact* 3%.

E Testing-Time Compute: Configurations

This appendix documents the inference-time compute strategies used in our experiments: *Self-Consistency*, *Self-Refine*, *Self-Verification*, and *Retrieval-Augmented Generation* (RAG). We report their key hyperparameters, decoding settings, token budgets, aggregation or selection rules, and prompt templates.

E.1 Method Configurations

Table 13 summarizes the configurations used in all testing-time compute experiments.

Method	Key hyperparameters	Decoding / budget
Self-Consistency	$N \in \{4, 8, 16\}$; majority vote	*Gen:* T=0.8, top-p=0.95 max tokens: 4096 (N=4); 2048 ($N \in \{8, 16\}$)
Self-Refine	$K \in \{1, 2, 3\}$ rounds; Solve $\rightarrow$ Critique $\rightarrow$ Revise	*Solve/Revise:* T=0.8, top-p=0.95 max tokens: 4096 *Critique:* T=0.0, top-p=1.0 max tokens: 512
Self-Verification	Best-of-N, $N \in \{4, 8, 16\}$; verifier selects $\arg\max$	*Gen:* T=0.8, top-p=0.95 max tokens: 4096 *Verify:* T=0.0, top-p=1.0 max tokens: 64
RAG	Retriever: dense / hybrid / rerank; k=8 retrieved passages	*Gen:* T=0.1, top-p=1.0 max tokens: 4096 retrieved context $\leq$ 12k chars

Table 13: Testing-time compute configurations. "Gen" denotes answer generation; auxiliary steps use deterministic decoding unless otherwise noted. "max tokens" denotes the maximum generation length per sample or step.

Self-Consistency. For each question q, we sample $N \in \{4, 8, 16\}$ independent reasoning trajectories under stochastic decoding and aggregate the final answers by majority vote. To control compute at larger N, we cap generation length at 4096 tokens for $N = 4$ and 2048 tokens for $N \in \{8, 16\}$. Ties are resolved by selecting the earliest generated candidate.

Self-Refine. We implement iterative refinement as a fixed loop:

$$\textit{Solve} \rightarrow (\textit{Critique} \rightarrow \textit{Revise}) \times K,$$

with $K \in \{1, 2, 3\}$. Solve and Revise use stochastic decoding ($T = 0.8$, top-$p = 0.95$) with a 4096-token budget, while Critique uses deterministic decoding ($T = 0.0$, top-$p = 1.0$) with a 512-token budget.

Prompt	Template
Shared generation prompt	**Task:** Answer the following multiple-choice question by selecting the correct option. Question: {question} Options: {options} **Requirement:** Think step by step, then output `\boxed{X}`, where X is a single option label.
Self-Refine: *Critique*	**Task:** Critique the solution attempt. Identify logical mistakes, missing assumptions, or incorrect option comparisons. Provide actionable feedback. Question: {question} Options: {options} Solution attempt: {attempt}
Self-Refine: *Revise*	**Task:** Revise the solution to address the critique. Then output `\boxed{X}`. Question: {question} Options: {options} Previous attempt: {attempt} Critique: {critique}
Self-Verification: *Verifier*	**Task:** Select the single best candidate. Output only `\boxed{k}`, where $k \in \{1, \ldots, n\}$. Question: {question} Options: {options} Candidates: {candidates_block}
RAG	**Retrieved context (may be partial or irrelevant):** {context} **Task:** Answer the following multiple-choice question by selecting the correct option. Question: {question} Options: {options} **Requirement:** Think step by step, then output `\boxed{X}`.

Table 14: Prompt templates for testing-time compute methods.

Self-Verification. We decouple proposal from selection. The model first generates $N \in \{4, 8, 16\}$ candidate solutions under $T = 0.8$ and top-$p = 0.95$, with a 4096-token budget. A verifier then selects the best candidate under deterministic decoding ($T = 0.0$, top-$p = 1.0$), producing a compact 64-token output that specifies only the selected candidate index.

RAG. RAG prepends retrieved evidence to the prompt at inference time. Dense retrieval uses sentence-transformer embeddings (`all-MiniLM-L6-v2`) and retrieves $k = 8$ passages per question, prepending up to 12,000 characters of retrieved context. Hybrid retrieval combines BM25 and dense scores with equal weighting ($\alpha = 0.5$). For reranking, we first retrieve $k = 50$ candidates and then apply a cross-encoder reranker (`cross-encoder/ms-marco-MiniLM-L-6-v2`) before selecting the top 8. Documents are chunked into 800-token segments with 100-token overlap. Generation uses low temperature ($T = 0.1$, top-$p = 1.0$) to encourage evidence-grounded outputs.