OrLog: Resolving Complex Queries with LLMs and Probabilistic Reasoning

Mohanna Hoveyda[1(✉)], Jelle Piepenbrock[2], Arjen P. de Vries[1], Maarten de Rijke[3], and Faegheh Hasibi[1]

[1] Radboud University, Nijmegen, The Netherlands
{mohanna.hoveyda,arjen.devries,faegheh.hasibi}@ru.nl
[2] Eindhoven University of Technology, Eindhoven, The Netherlands
j.h.piepenbrock@tue.nl
[3] University of Amsterdam, Amsterdam, The Netherlands
m.derijke@uva.nl

Abstract. Resolving complex information needs that come with multiple constraints should consider enforcing the logical operators encoded in the query (i.e., conjunction, disjunction, negation) on the candidate answer set. Current retrieval systems either ignore these constraints in neural embeddings or approximate them in a generative reasoning process that can be inconsistent and unreliable. Although well-suited to structured reasoning, existing neuro-symbolic approaches remain confined to formal logic or mathematics problems as they often assume unambiguous queries and access to complete evidence, conditions rarely met in information retrieval. To bridge this gap, we introduce **OrLog**, a neuro-symbolic retrieval framework that decouples predicate-level plausibility estimation from logical reasoning: a large language model (LLM) provides plausibility scores for atomic predicates in one decoding-free forward pass, from which a probabilistic reasoning engine derives the posterior probability of query satisfaction. We evaluate OrLog across multiple backbone LLMs, varying levels of access to external knowledge, and a range of logical constraints, and compare it against base retrievers and LLM-as-reasoner methods. Provided with entity descriptions, OrLog can significantly boost top-rank precision compared to LLM reasoning with larger gains on disjunctive queries. OrLog is also more efficient, cutting mean tokens by ∼90% per query–entity pair. These results demonstrate that generation-free predicate plausibility estimation combined with probabilistic reasoning enables constraint-aware retrieval that outperforms monolithic reasoning while using far fewer tokens.

Keywords: Complex queries · Large language models · Probabilistic logic programming · Neuro-symbolic methods

1 Introduction

Many real-world information needs involve multiple interacting constraints [24, 35]. Consider, e.g., the query *"Museums in Paris or Amsterdam but not art*

R. Campos et al. (Eds.): ECIR 2026, LNCS 16483, pp. 98–114, 2026.
https://doi.org/10.1007/978-3-032-21289-4_7

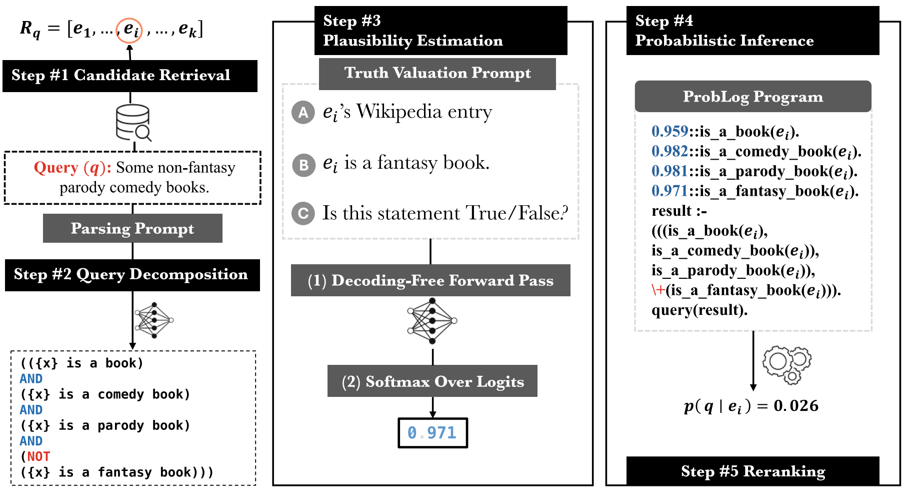

Fig. 1. Overview of **OrLog**, a neuro-symbolic framework for resolving complex entity-seeking queries. For a given query five steps are followed. *Step 1*: a retriever narrows the search space to candidate entities (R_q). *Step 2*: a semantic parser decomposes the query into atomic predicates and a logical form. *Step 3*: predicate priors are elicited from an LLM via a truth-valuation prompt, using a single decoding-free forward pass. *Step 4*: these priors instantiate a probabilistic program in ProbLog, which computes posterior query satisfaction for each entity. *Step 5*: Candidates are reranked by their inferred probabilities.

museums", which requires not only identifying semantically related entities but also enforcing how conjunction, disjunction, and negation restrict the answer space [44,50]. In formal terms, a valid entity e must satisfy $museum(e) \wedge (paris(e) \vee amsterdam(e)) \wedge \neg art(e)$. Resolving such queries involves evaluating whether candidate entities satisfy a logical constraint, rather than merely ranking by semantic similarity. Current retrieval systems are ill-equipped for addresssing such queries [11,23,24]. Sparse and dense models approximate relevance by lexical overlap or embedding similarity. Such relevance scores do not compose in ways that respect logical operators: the outcome for logical constraints over a query's atomic predicates cannot necessarily be derived from token- or embedding-level similarities between a document and predicates [32,35,43].

End-to-end reasoning with LLMs that attempts to enforce constraints directly in token space [42] can also be unreliable and computationally expensive [2,49], despite being capable of producing plausible answers. Neuro-symbolic approaches could, in principle, enforce constraints faithfully and provide a viable path for modeling reasoning in complex retrieval settings. However, existing systems assume unambiguous queries and complete evidence at inference time, conditions that may fit logical puzzles and mathematical problems, but are rarely satisfied in information retrieval [29,31,47].

Approach. We introduce **OrLog**, a neuro-symbolic retrieval framework that decouples *predicate-level plausibility estimation* from *logical reasoning.* OrLog uses a standard retriever to narrow the search space, then invokes an LLM to decompose the natural language query into atomic predicates and a logical form. Rather than performing end-to-end reasoning with an LLM, in our framework, the LLM acts as an *uncertainty oracle*: given an entity and a predicate, the model returns a plausibility score in a single forward pass without free-form generation. This score is a scalar between 0 and 1 obtained by transforming the model's logits for designated truth labels; it quantifies the model's degree of support that the predicate holds for the entity in the context of the query. These plausibility scores are then used as predicate-level priors in ProbLog [34], to compute the posterior probability of an entity satisfying the full query. By shifting compositional reasoning to a probabilistic reasoning engine, OrLog enforces constraints efficiently, avoiding the verbosity and unreliability of monolithic LLM reasoning. Our experiments address three research questions:

(RQ1) How does *OrLog* compare to *LLM-as-reasoner* on resolving complex set-compositional queries?
(RQ2) How does the choice of the underlying LLM affect OrLog's performance and that of the LLM?
(RQ3) How does query structure (e.g., conjunction, disjunction, and negation) affect the relative performance of OrLog and monolithic LLMs?

Contributions. The four main contributions of this work are:

1. A neuro-symbolic framework, OrLog, that combines retrieval, LLM-based predicate plausibility estimation, and probabilistic reasoning for constraint-aware re-ranking for complex queries.
2. An efficient predicate-level plausibility estimation method that elicits plausibility scores from LLM logits in a decoding-free forward-pass.
3. Empirical evidence that OrLog achieves significant performance gains over LLM-as-reasoner while reducing token usage by ∼90%.
4. A structural analysis of how logical operators in queries affect the comparative performance of OrLog versus monolithic LLM reasoning.

2 Related Work

2.1 Logic-Based Information Retrieval

The 'logical approach to information retrieval (IR)' is an early line of research to ground retrieval models in formal axioms and logical theories [6,35], taking the stand that logic would provide a necessary conceptual foundation for modeling retrieval [36]. Documents and queries would be represented as logical sentences, relevance defined by logical implication [15], and uncertainty through logical imaging [7]. While theoretically rigorous, these frameworks faced practical obstacles in their application across large collections. Also, relative performance gains (if any) over previous approaches never outweighed their increased system and modeling complexity. As a result, the pursuit of logically grounded retrieval systems has remained unresolved [1,8].

2.2 Reasoning: Monolithic and LLM-Modulo Solutions

Contemporary approaches to complex, logic-intensive natural language tasks increasingly rely on LLMs as general-purpose monolithic reasoners. These models are incentivized to produce verbose sequences of intermediate steps, so-called chain-of-thought (CoT), to approximate structured reasoning [42]. While this strategy often improves final accuracy, it provides no guarantee of logical coherence: the generated chain may obscure inconsistencies in intermediate steps or rationalize an incorrect answer [2,21,48]. Reinforcement learning techniques that promote longer traces amplify the risk of confabulated content [14,49], as seen in recent OpenAI models compared to earlier ones [18,30]. Beyond the concerns of faithful reasoning, verbose generation incurs non-trivial computational cost, particularly in settings requiring extended or branching reasoning paths [4,12]. An alternative line of work externalizes reasoning: LLMs act as stochastic components that excel in semantic understanding and parsing, while symbolic backends enforce structured reasoning [9]. Within this LLM-modulo paradigm [13,19], systems diverge in the choice of backend. LINC [29] employs a first-order prover, achieving high logical fidelity for logical puzzles but requiring exact translations and complete information at inference time. Some cognitive science-inspired approaches to modeling reasoning extend this principle further, casting LLMs as front-ends that synthesize bespoke probabilistic programs, with reasoning delegated to probabilistic program interpreters to support reasoning in novel situations [45,46]. Closer to our setting, Nafar et al. [26] use probabilistic logic programming as a principled substrate for uncertainty, outperforming LLMs in probabilistic reasoning but assuming predicate probabilities are given.

2.3 LLM Uncertainty Estimation

Since OrLog treats the LLM as a probabilistic oracle, its effectiveness hinges on how reliably we can estimate predicate-level uncertainty from the model. UE methods for LLMs can be divided into white-box and black-box approaches [39]. White-box methods exploit internal logits to quantify predictive uncertainty [3,10,17,20]. Despite access to logits, these methods typically rely on multiple decoding passes or sampled generations to approximate uncertainty over full responses, leading to high computational cost. Black-box methods treat the model as an opaque generator, inferring uncertainty from similarity or dispersion across sampled outputs [22,27]. Both paradigms assume free-form generation tasks and become unstable with suboptimal external evidence, as in retrieval-augmented generation (RAG) [40]. OrLog departs from both lines as predicate-level uncertainty is elicited in a decoding-free setting. For each predicate, a single forward pass yields the logits for tokens `True`/`False`, which are normalized into a final score, indicating the truth probability of that predicate. This approach avoids repeated sampling, reduces token cost, and yields uncertainty estimates tailored for probabilistic logical retrieval.

3 The OrLog Reasoning Framework

3.1 Problem Definition

Let $\mathcal{E}$ be the corpus of entities, where each entity is characterized by a textual description and an identifier. The textual representation of an entity $e \in \mathcal{E}$ may be a document (e.g., a Wikipedia article) or a set of (semi-)structured facts (e.g., a Wikidata item). We focus on Wikipedia documents. The user's natural-language query q implicitly specifies logical constraints over $\mathcal{E}$. E.g., the query *"Some non-fantasy parody comedy books"* contains atomic predicates A, B, C, and D combined through formal connectives as $A \cap B \cap C \cap \neg D$, as illustrated in Fig. 1. Our goal is to return a ranked list of entities $E = [e_1^*, e_2^*, \ldots, e_n^*]$ that contains relevant entities to the query q, satisfying the logical constraints expressed in the query.

3.2 Framework Overview

We propose a neuro-symbolic framework that combines LLMs and formal probabilistic logic to improve trustworthiness and retrieval effectiveness. Figure 1 sketches our *neuro-symbolic retrieval* framework, **OrLog**, in five steps: (i) *Candidate retrieval* retrieves the top-k entities via a sparse or dense model. (ii) *Query decomposition* extracts atomic predicates from the natural-language query and constructs its Boolean logical representation. (iii) *Predicate Plausibility estimation* estimates plausibility scores for atomic predicates. (iv) *Probabilistic reasoning* computes the posterior score of an entity satisfying the query over the weighted predicates. (v) *Reranking* sorts the candidates by their posterior probabilities in descending order. Below, we detail each component.

3.3 Candidate Retrieval

Reasoning over all entities in the entity corpus $\mathcal{E}$ for each incoming query Q is intractable. We therefore narrow down the search space to the top-k ranked results using a base retrieval model. This cost-effective filtering ensures that the subsequent reasoning procedure is applied only to a promising list of candidate entities $\mathcal{R}_q$, likely to contain entities relevant to the query.

3.4 Query Decomposition

We next derive a *formal representation* of natural language query q to enable symbolic reasoning over its candidate entities, by decomposing q into a finite set of atomic predicates $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$. Each predicate is a function of entity e that represents a factual statement, taking the value of TRUE or FALSE. For example, predicate $P_1(e) =$"$\{e\}$ is a comedy book" is applied to every candidate entity (illustrated in Fig. 1). These predicates are combined using Boolean connectives (conjunction AND, disjunction OR, and negation NOT) to represent the constraints of the query. Figure 1 illustrates how the formal representation of query *"Some non-fantasy parody comedy books"* consists of four predicates combined with logical connectives AND and NOT.

To perform the transformation from a natural language query to a formal proposition, an LLM is prompted with a one-shot example. The LLM produces: (i) a set of atomic predicates $\mathcal{P}$ extracted from query q, and (ii) a logical proposition of the query q', such that q' approximates the logical semantics of q. The fidelity of this translation directly affects downstream reasoning in OrLog. In the absence of ground-truth translations, it is difficult to quantify translation accuracy and its effect on OrLog's performance, but manual inspection of a sample of queries suggests that the translated representations are of satisfactory quality.[1]

3.5 Predicate Plausibility Estimation

In probabilistic logic, every predicate (or atom) is annotated with a prior probability to indicate the likelihood of the predicate being true. Here, we introduce our method to estimate these scores. We utilize the logits produced by an LLM for the tokens *True* and *False* in response to a truth-valuation prompt. The plausibility score, then, is a scalar value between 0 and 1, indicating model's degree of support that the predicate holds true for the entity. While we will treat these scores as prior probabilities in the downstream probabilistic program, we emphasize that they originate as plausibility estimates of predicate truth, i.e., calibrated degrees of support in $[0, 1]$ rather than probabilities in the strict axiomatic sense [16,33].

Truth Valuation Prompt. Given an entity e, we construct a truth-valuation prompt $T = \langle c, t_e, P(e), \sigma \rangle$ consisting of four elements: (i) context c, which is a description of the entity (e.g., from Wikipedia); (ii) the entity title t_e; (iii) the atomic predicate $P(e)$, indicating a factual statement about entity e; and (iv) a truth inquiry suffix σ (e.g., "Is this predicate True or False?") to facilitate eliciting LLM's probability output. The predicate and entity title are mandatory parts of the prompt, the rest is optional. These elements are concatenated and fed into an LLM in a single forward pass.

Computing Plausibility Score. LLM's estimated plausibility that the atomic predicate $P(e)$ holds is calculated as the softmax score assigned to the token *True*. Let z_{True}, z_{False} be the LLM final layer's logits for the tokens *True* and *False*, respectively, given the truth valuation prompt T. Following prior work that derives relevance estimates from LLMs' logits over target tokens [28,51], here we estimate the truth plausibility of each atomic predicate from the LLM as:

$$\pi(P(e)) = \frac{\exp(z_{\text{True}})}{\exp(z_{\text{True}}) + \exp(z_{\text{False}})}.$$

[1] We do not experiment with the choice of LLM for this step, using *Llama-3.3-70B-instruct* for all transformations in our experiments.

3.6 Probabilistic Reasoning

For each e in $\mathcal{R}_q$, we assemble a ProbLog program that includes the weighted predicates $\{\langle \pi(P(e)), P(e)\rangle \mid P \in \mathcal{P}\}$ and the logical proposition of the query q'. ProbLog reasoning then computes the posterior probability of e satisfying the query e, denoted as $\rho(q, e)$. ProbLog interprets the program as a distribution over truth assignments to the predicates of e, and returns the posterior probability $\rho(q, e)$ by summing the weights of all assignments in which q' holds. For conjunctive queries, this reduces to a product of predicate probabilities, while disjunctive and mixed queries require marginalization over multiple satisfying worlds. As shown in step 4 in Fig. 1, this yields the final score used for reranking.

3.7 Reranking

Finally, candidates are ranked in descending order of their posterior probability $\rho(q, e)$, yielding the final list $[e_{(1)}, \ldots, e_{(k)}]$ (final step in Fig. 1). In cases where two entities receive identical posteriors, ties are resolved by preserving their relative order from the originally retrieved ranking.

4 Experiments

4.1 Dataset

Evaluating whether retrieval systems can resolve complex queries with logical constraints requires test queries that go beyond topical or semantic relevance and involve interacting logical constraints expressed in natural language. The QUEST dataset [24] is well-aligned with this need: it consists of queries mapped to sets of Wikipedia entities defined through set-theoretic operations, intersection, union, and difference, and their combinations. Queries are derived from Wikipedia category compositions (e.g., *science-fiction films shot in England*) and span seven templates, covering multiple constraint types and logical complexities. We run experiments on the 1,727 query test split released by the authors.

4.2 Retrievers

For candidate retrieval, the first step of OrLog, we use two models to retrieve the top-k entities per query: *BM25* [37] as a sparse lexical model, and *E5-base-v2* [41] as a dense model optimized for semantic similarity that is shown to perform reasonably well on QUEST [38]. We set $k = 20$ in all experiments to ensure a tractable pool for downstream reasoning. Out of the 1,727 test queries in QUEST, BM25 retrieves at least one gold entity for 644 queries, while E5 does so for 981 queries. Our reported results are based on the complete test set in QUEST.

4.3 LLMs

In our work, LLMs are used in three distinct roles:

1. **Reasoning** Serving as *LLM-as-reasoner* baseline, the LLM receives the query and entity information and is prompted to decide whether each entity satisfies the query, and to output the final result in True/False.
2. **Translation** Parsing the natural-language query into atomic predicates and logical forms, which will later be used in OrLog.
3. **Plausibility estimation** eliciting predicate-level plausibility scores later assigned as priors to weighted predicates in ProbLog.

For translation, we fix the model across all experiments to *Llama-3.3-70B-Instruct* to isolate the variability in parsing quality. Both the LLM-as-reasoner and OrLog frameworks are evaluated across three backbone LLM families: *Mistral-v1* (7B-Instruct, 8×7B-Instruct), *Qwen-2.5* (7B-Instruct, 72B-Instruct), and *Llama-3* (8B-Instruct, 70B-Instruct). These models vary in scale, architecture, and training approach, allowing us to analyze the effect of different model families and sizes on both predicate plausibility estimation and monolithic LLM reasoning. Due to limited local compute resources, we offload LLM reasoning experiments to the OpenRouter API rather than hosting large LLMs on-premises. Since OpenRouter does not provide access to model logits, we perform all OrLog-related experiments on our H100 or A100 GPU instances, wherever relevant. Larger backbones are 4-bit quantized to fit within a single GPU.[2]

4.4 Knowledge Availability Setups

To examine the impact of information access during reasoning process in each method, we evaluate them under two access levels: (i) **Parametric**: only the entity name and the query are provided; the LLM relies exclusively on its internal (parametric) knowledge to estimate predicate truth plausibility or assess query satisfaction (ii) **Parametric+**: in addition to the query and entity name, we provide a Wikipedia description of the candidate entity (explicit external evidence). This text may or may not contain a direct answer, but is intended to elicit more relevant parametric knowledge in OrLog's setup or to enhance reasoning in the LLM-as-reasoner setup. This comparison allows us to assess how the absence of contextual information affects performance across frameworks.

4.5 Baselines

We evaluate three baseline configurations. The first is the **Retriever-Only** setup, where we directly evaluate the rankings produced by the base retrievers without applying any further reasoning. The second is the **LLM-as-reasoner** setup, where LLMs are prompted in an end-to-end fashion with both the query

[2] Our repository containing the implementation, prompt templates, and experiments is available at github.com/informagi/RSN.

and candidate entity information (± evidential knowledge) to decide whether each entity satisfies the query. The third baseline, **Logic-Augmented LLM**, is based on the LLM-as-reasoner setup, with the prompt being augmented by the logical decompositions of the query to help the LLM reasoning process. These decompositions are the ones used in OrLog, and the motivation for adding this baseline is to have comparative setups between OrLog and LLM reasoning.

4.6 Evaluation Metrics

We evaluate retrieval performance using Precision@K (P@K), Recall@K (R@K), F1@K, and NDCG@K at $K \in \{1, 10\}$, as well as Mean Reciprocal Rank (MRR). All metrics are averaged over the test set and, where relevant, reported per query type. To assess statistical significance, we perform paired two-tailed sign tests and report the p-values. For cost evaluation, we compare the systems in terms of the number of tokens generated by LLMs. For the *LLM-as-reasoner* setup, we measure the total number of tokens generated by the language model across all query–entity pairs and report the average per pair. For *OrLog*, the token count per query–entity pair comprises (1) the number of atomic predicates in the parsed query, since each predicate triggers one forward pass to the LLM, and (2) the one-off token cost to produce the parsing. Final average is reported over all query–entity pairs.

5 Experimental Results

We answer our questions from the introduction: (i) Does isolating predicate-level uncertainty to an LLM and delegating reasoning to ProbLog yield higher retrieval accuracy than monolithic LLM reasoning? (ii) How does the choice of backbone LLM affect reasoning quality in both OrLog and LLM-as-reasoner frameworks? And (iii) which classes of logical constraints most clearly reveal the performance gap between OrLog and LLM-as-reasoner approaches? We examine each question in sequence below.

5.1 OrLog vs. LLM for Complex Query Resolution

Table 1 presents the evaluation results for OrLog and the LLM-as-reasoner baseline across two base retrievers (BM25 and E5) and two knowledge access settings (*Parametric* and *Parametric+*).

Effect of Access to External Knowledge. We notice at the first glance that OrLog*(Param+)* significantly outperforms all baselines with respect to all metrics in all setups, except P, R, and F1 at rank 10 with the E5 retriever. In the *Parametric* setting, where only the entity name is provided, OrLog significantly underperforms the LLM-as-reasoner, with the largest deficit under E5. Lacking contextual grounding, the LLM's predicate plausibility estimates are poorly calibrated; ProbLog receives miscalibrated priors and performance degrades.

Table 1. Results for BM25 and E5 base retrievers (shaded) and reranked variants using *Llama-3.3-70B-Instruct* across three reasoning strategies: **LLM**, **Logic-augmented LLM** (LLM*), and **OrLog**. Significance of OrLog (better or worse) is tested against the comparable LLM baseline (Param or Param+);$^{\bigcirc}$ $p<0.05$, $\square$ $p<0.01$, $\triangle$ $p<0.001$. Bold indicates the best score per base retriever and metric.

System	P@K		R@K		F1@K		NDCG@K		MRR
	1	10	1	10	1	10	1	10	
BM25	0.116	0.067	0.014	0.073	0.022	0.062	0.116	0.089	0.175
+ LLM*(Param)	0.158	0.083	0.019	0.088	0.031	0.077	0.158	0.114	0.219
+ LLM(Param)	0.155	0.079	0.020	0.085	0.032	0.074	0.155	0.110	0.212
+ OrLog(Param)	0.138$^{\square}$	0.075	0.015	0.080$^{\square}$	0.026	0.070$^{\square}$	0.138	0.101$^{\bigcirc}$	0.199
+ LLM*(Param+)	0.144	0.076	0.019	0.083	0.030	0.071	0.144	0.106	0.203
+ LLM(Param+)	0.185	0.086	0.023	0.093	0.038	0.080	0.185	0.125	0.244
+ **OrLog(Param+)**	**0.221**$^{\triangle}$	**0.089**$^{\bigcirc}$	**0.026**$^{\triangle}$	**0.095**$^{\bigcirc}$	**0.044**$^{\triangle}$	**0.082**$^{\bigcirc}$	**0.221**$^{\triangle}$	**0.134**$^{\triangle}$	**0.267**$^{\square}$
E5	0.182	0.105	0.021	0.116	0.035	0.098	0.182	0.142	0.269
+ LLM*(Param)	0.204	0.119	0.024	0.132	0.041	0.111	0.204	0.163	0.299
+ LLM(Param)	0.192	0.115	0.023	0.126	0.038	0.107	0.192	0.156	0.287
+ OrLog(Param)	0.161$^{\square}$	0.104$^{\triangle}$	0.019$^{\square}$	0.114$^{\triangle}$	0.031$^{\square}$	0.097$^{\triangle}$	0.161$^{\square}$	0.136$^{\triangle}$	0.260$^{\square}$
+ LLM*(Param+)	0.241	0.127	0.031	0.138	0.050	0.118	0.241	0.180	0.333
+ LLM(Param+)	0.263	**0.133**	0.034	**0.145**	0.056	**0.123**	0.263	0.193	0.355
+ **OrLog(Param+)**	**0.308**$^{\triangle}$	0.131	**0.039**$^{\triangle}$	0.142	**0.063**$^{\triangle}$	0.121	**0.308**$^{\triangle}$	**0.197**$^{\bigcirc}$	**0.384**$^{\square}$

The pattern also demonstrates that OrLog gains more from external evidence than LLM-as-reasoner, suggesting more effective integration of factual grounding into the reasoning process. Interestingly, the *Logic-Augmented LLM* experiments shows that augmenting the prompt with atomic decompositions and a logical form does not consistently improve performance. While such structure is intended to guide more faithful inference, in the *Param+* setting it leads to degradation relative to the plain LLM. A likely cause is that the additional instructions and representations introduce redundancy and cognitive load for the model, which struggles to integrate them effectively. In contrast, decomposition is one of the primary sources of OrLog's performance gains.

Effect of the Base Retriever. The gains with BM25 are more uniform than with E5. While in the *parametric+* setup, OrLog substantially outperforms the LLM baseline on NDCG, its gains are less consistent in other metrics. A likely cause is our fallback mechanism: when the LLM-as-reasoner fails to emit a valid binary label, we revert to the base-retriever score. Because E5 already yields strong initial rankings, this disproportionately benefits the baseline. Still, OrLog shows sharper top-rank precision; consistently higher P@1 and NDCG@1. Overall, improvements correlate with retriever weakness: OrLog delivers larger gains with BM25 and only modest lift on an already-strong E5.

Cost Comparison. Beyond ranking efficacy, practical deployment demands reasoning efficiency as well. Table 2 reports the mean token count per query–entity invocation; end-to-end LLM reasoning uses $35-55$ tokens on average (higher under *Parametric+* due to longer prompts and outputs), whereas OrLog remains below 6 tokens. This reduction stems from (i) decoding-free predicate plausibility estimation (a single forward pass per predicate) and (ii) one-off query parsing with composition delegated to ProbLog, whose runtime and computational cost are negligible relative to an LLM call.

Table 2. Average generated token count per query-entity for each method.OrLog reduces token generation by ∼90% compared to LLM-as-reasoner approaches.

Retriever	Method	Avg. generated tokens
BM25	LLM (Param)	50.2
	LLM (Param+)	55.0
	OrLoG	**5.80**
E5	LLM (Param)	35.3
	LLM (Param+)	46.6
	OrLoG	**5.79**

5.2 Effective Predicate Plausibility Estimation May Offset Model Size

Table 3 depicts the performance variation of OrLog and the LLM-as-reasoner pipeline with various backbone models. In the parametric knowledge setting, LLM-as-reasoner consistently outperforms OrLog across all families of models (Mistral, Qwen, and Llama). This, again, confirms that, absent any grounding information, predicate plausibility estimates from the LLM are too noisy to support robust probabilistic reasoning, causing OrLog's rankings to lag. Following the same trend in Table 1, introducing Wikipedia descriptions (+Parametric) boosts both frameworks, but OrLog benefits more effectively from the provided context. In this table, the Qwen-2.5-7B results introduce an important nuance: despite its modest scale, the small Qwen backbone supplies predicate estimates of sufficient fidelity to enable OrLog to outperform even the 72B variant. Concretely, OrLog (Parametric+) with Qwen-7B achieves P@1 $= 0.303$ and MRR $=$ 0.381, slightly surpassing both its large-model OrLog counterpart (0.300/0.374) and the monolithic LLM 72 B (0.294/0.380). This indicates that various factors, such as training and post-training quality, can compensate for parameter count when generating calibrated plausibility estimates.

5.3 Disjunction Amplifies OrLog's Gains over LLM-Based Reasoning

Figure 2 depicts mean ΔP@1 between OrLog and LLM-reasoner, stratified by query structural template taken from QUEST, evaluated on Llama-3.3-70B and

Qwen-2.5-7B. Three clear patterns emerge. First, OrLog's benefit is negligible for purely *conjunctive* queries, as for $A \wedge B$ and $A \wedge B \wedge C$, gains are negligible (typically ≈ 0.02 or lower), indicating that well-prompted LLMs compose simple conjunctive constraints nearly as well as a symbolic backend if not better.

Table 3. Effect of various backbone LLMs on the performance of OrLog vs. LLM-as-reasoner with E5 as the candidate retriever. Each panel reports results for two size variants of Mistral-v1, Qwen-2.5, and Llama-3. Significance of OrLog (better or worse) is tested against the comparable LLM baseline (Param or Param+);$^{\bigcirc}$ $p<0.05$, $\square$ $p<0.01$, $\triangle$ $p<0.001$. Boldface denotes the best score.

(a) Mistral-v1

	7B			8 × 7B		
System	@1	@10	MRR	@1	@10	MRR
LLM(Param)	0.176	0.142	0.271	0.191	0.149	0.281
OrLog(Param)	0.122$^{\triangle}$	0.112$^{\triangle}$	0.221$^{\triangle}$	0.163$^{\square}$	0.134$^{\triangle}$	0.259
LLM(Param+)	**0.260**	**0.180**	**0.347**	0.249	0.181	0.340
OrLog(Param+)	0.160$^{\triangle}$	0.139$^{\triangle}$	.263$^{\triangle}$	**0.266**	**0.185**$^{\square}$	**0.352**

(b) Qwen-2.5

	7B			72B		
System	@1	@10	MRR	@1	@10	MRR
LLM(Param)	0.181	0.146	0.273	0.204	0.160	0.298
OrLog(Param)	0.145$^{\triangle}$	0.126$^{\triangle}$	0.240$^{\triangle}$	0.217	0.154$^{\bigcirc}$	0.304
LLM(Param+)	0.271	0.180	0.350	0.294	**0.201**	**0.380**
OrLog(Param+)	**0.303**$^{\square}$	**0.199**$^{\triangle}$	**0.381**$^{\triangle}$	**0.300**	0.196	0.374

(c) Llama-3

	8B			70B		
System	@1	@10	MRR	@1	@10	MRR
LLM(Param)	.177	.144	.269	.192	.156	.287
OrLog(Param)	.138$^{\triangle}$	.124$^{\triangle}$	.234$^{\triangle}$	.161$^{\square}$	.136$^{\triangle}$	.260$^{\triangle}$
LLM(Param+)	.244	.182	.339	.263	.193	.355
OrLog(Param+)	**.263**	**.186**	**.351**	**.308**$^{\triangle}$	**.197**$^{\bigcirc}$	**.384**$^{\square}$

Negation shows moderate gains for OrLog. Queries of the form $A \wedge \neg$ B and $A \wedge B \wedge \neg$ C show mean gains in the 0.02–0.03 range, with some structures achieving statistical significance ($p < 0.05$) with the Qwen backbone. This suggests that negation, while tractable for LLMs, still imposes a mild burden on their end-to-end reasoning, which OrLog alleviates by delegating to ProbLog. The last and most notable is the case of *disjunction*. Simple disjunctions ($A \vee B$) exhibit the largest performance gap: OrLog improves NDCG@1 on Llama and on Qwen, both at high significance ($p < 0.05$ and $p < 0.01$, respectively). Even extended disjunctions ($A \vee B \vee C$) benefit substantially. These results demonstrate that monolithic LLM reasoning struggles to balance alternative predicate truth-assignments, whereas OrLog's symbolic engine effectively integrates multiple probabilistic hypotheses to satisfy at least one of the disjuncts. Qualitative analysis of LLM's results showed that LLMs tend to treat disjunction as conjunction, rejecting a fitting entity for a disjunction by saying that this does not satisfy both conditions.

6 Discussion and Future Work

We have proposed OrLog, a framework to resolve complex entity-seeking queries through separation of concerns; decoupling predicate-level plausibility estimation from logical reasoning, using LLMs for the former and probabilistic logic

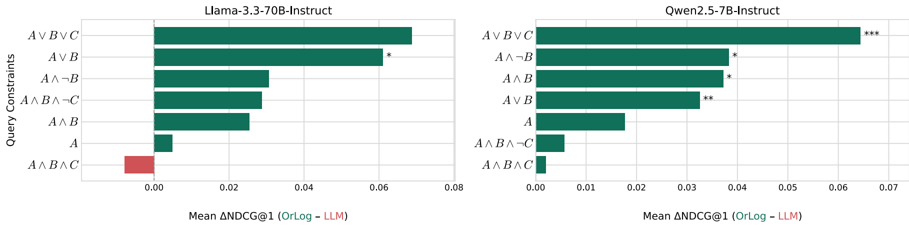

Fig. 2. Mean ΔP@1 between *OrLog* and *LLM-as-reasoner* baseline across query-template structures in QUEST, in the *informed* knowledge access setting. Each bar shows the average per-query gain or loss for a specific template, with green indicating OrLog outperforming the LLM-as-reasoner baseline, and red indicating the reverse. Left and right panels correspond to different underlying LLMs: *Llama-3.3-70B-Instruct* and *Qwen-2.5-7B-Instruct*. Larger deviations highlight structural patterns where symbolic probabilistic reasoning offers a clear advantage relative to reasoning purely with LLMs.

for the latter. We showed empirically that OrLog can achieve significantly better performance than previous approaches using base retrievers or monolithic LLM reasoning. With entity descriptions (Parametric+), OrLog consistently improves top-rank results and shows its largest advantages on disjunctive structures, where the LLM treats disjunctive constraints as conjunction. At the same time, OrLog cuts average tokens by one decoding-free forward pass per predicate and offloading logical reasoning to ProbLog. These results indicate that effective reasoning does not require ever larger models or longer traces; smaller backbones suffice when used as calibrated uncertainty oracles, provided their plausibility outputs are reliable [5]. We have referred to the LLM as an *Oracle*, a deliberate abstraction that assumes access to representative predicate plausibility estimation. However, this assumption is only partially borne out today; the elicited plausibilities from LLMs can be miscalibrated, especially under weak or conflicting evidence. Progress requires improving plausibility elicitation through more sophisticated methods or strengthening the backend model to output more representative scores. To strengthen this component, we see three promising paths for future work; (1) jointly learn plausibility estimation and logical inference via DeepProbLog [25], a differentiable probabilistic logic programming framework; (2) replace point plausibilities with distributions and marginalize uncertainty during inference; and (3) improve elicitation through better truth valuation prompts and alternate logit-space aggregation. Finally, robustness analysis of OrLog vs LLMs under perturbations of queries or entity descriptions and extending OrLog to other types of complex queries (e.g., multi-hop) constitute the immediate next steps to test whether OrLog's performance gains sustain as noise and task complexity grow. In sum, OrLog demonstrates that principled separation between plausibility estimation and logical reasoning is not only feasible in information retrieval with the tools we have available today, but also a promising foundation for building scalable, reliable reasoning systems.

Acknowledgements. This research was (partially) supported by the Dutch Research Council (NWO), under project numbers 024.004.022, NWA.1389.20.183, and KICH3.LTP.20.006, and the European Union under grant agreements No. 101070212 (FINDHR) and No. 101201510 (UNITE). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Abdulahhad, K., Berrut, C., Chevallet, J., Pasi, G.: Modeling information retrieval by formal logic: a survey. ACM Comput. Surv. **52** (2019)
2. Arcuschin, I., Janiak, J., Krzyzanowski, R., Rajamanoharan, S., Nanda, N., Conmy, A.: Chain-of-thought reasoning in the wild is not always faithful (2025)
3. Bakman, Y.F., Yaldiz, D.N., Buyukates, B., Tao, C., Dimitriadis, D., Avestimehr, S.: MARS: meaning-aware response scoring for uncertainty estimation in generative LLMs. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics ACL, pp. 7752–7767 (2024)
4. Belcak, P., et al.: Small language models are the future of agentic AI. CoRR (2025)
5. Belcak, P., et al.: Small language models are the future of agentic ai. arXiv preprint arXiv:2506.02153 (2025)
6. Bruza, P., Huibers, T.W.C.: Investigating aboutness axioms using information fields. In: Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. (Special Issue of the SIGIR Forum) (1994)
7. Crestani, F.: Logical Imaging and Probabilistic Information Retrieval, pp. 247–279. Springer US, Boston, MA (1998). ISBN 978-1-4615-5617-6, https://doi.org/10.1007/978-1-4615-5617-6_10
8. Crestani, F., Ruthven, I., Sanderson, M., van Rijsbergen, C.J.: The troubles with using a logical model of IR on a large collection of documents. In: Proceedings of The Fourth Text REtrieval Conference, TREC 1995, Gaithersburg, Maryland, USA, November 1-3, 1995, National Institute of Standards and Technology (NIST) (1995)
9. DeepMind: Ai achieves silver-medal standard solving international mathematical olympiad problems (2024). https://deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level/. Accessed 01 Oct 2025
10. Duan, J., et al.: Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 5050–5063. ACL (2024)
11. van den Elsen, C., Barkhof, F., Nijdam, T., Lupart, S., Aliannejadi, M.: Reproducing nevir: negation in neural information retrieval. In: Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR 25 (2025)
12. Feng, S., Fang, G., Ma, X., Wang, X.: Efficient reasoning models: a survey. CoRR **abs/2504.10903** (2025)

13. Hoveyda, M., Oosterhuis, H., de Vries, A.P., de Rijke, M., Hasibi, F.: Adaptive orchestration of modular generative information access systems. In: Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 25, ACM (2025)
14. Huang, L., et al.: A survey on hallucination in large language models: principles, taxonomy, challenges, and open questions. ACM Trans. Inf. Syst. (2025)
15. Huibers, T.W.C., Lalmas, M., van Rijsbergen, C.J.: Information retrieval and situation theory. SIGIR Forum (1996)
16. Jaynes, E.T.: Probability Theory. The Logic of Science. Cambridge University Press, Cambridge (2002)
17. Kadavath, S., et al.: Language models (mostly) know what they know (2022). https://arxiv.org/abs/2207.05221
18. Kalai, A.T., Nachum, O., Vempala, S.S., Zhang, E.: Why language models hallucinate (2025)
19. Kambhampati, S., et al.: Position: Llms can t plan, but can help planning in llm-modulo frameworks. In: Forty-first International Conference on Machine Learning (2024)
20. Kuhn, L., Gal, Y., Farquhar, S.: Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In: The Eleventh International Conference on Learning Representations ICLR (2023)
21. Lanham, T., et al.: Measuring faithfulness in chain-of-thought reasoning. arXiv preprint arXiv:2307.13702 (2023)
22. Lin, Z., Trivedi, S., Sun, J.: Generating with confidence: uncertainty quantification for black-box large language models. Trans. Mach. Learn. Res. **2024** (2024)
23. Mai, Q., Gauch, S., Adams, D.: Setbert: enhancing retrieval performance for boolean logic and set operation queries. In: Proceedings of the 2024 8th International Conference on Natural Language Processing and Information Retrieval (2024)
24. Malaviya, C., Shaw, P., Chang, M., Lee, K., Toutanova, K.: QUEST: a retrieval dataset of entity-seeking queries with implicit set operations. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics ACL 2023, Association for Computational Linguistics (2023)
25. Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., De Raedt, L.: Deepproblog: neural probabilistic logic programming. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
26. Nafar, A., Venable, K.B., Kordjamshidi, P.: Reasoning over uncertain text by generative large language models. In: AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence. AAAI Press (2025)
27. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Advances in Neural Information Processing Systems 14 [Neural InformationcProcessing Systems: Natural and Synthetic, NIPS], pp. 849–856 (2001)
28. Nogueira, R., Jiang, Z., Pradeep, R., Lin, J.: Document ranking with a pretrained sequence-to-sequence model. In: Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics (2020)
29. Olausson, T., et al.: Linc: a neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2023)
30. OpenAI: OpenAI o3 and o4?mini System Card. Technical report, OpenAI (2025). https://openai.com/index/o3-o4-mini-system-card/

31. Pan, L., Albalak, A., Wang, X., Wang, W.Y.: Logic-lm: empowering large language models with symbolic solvers for faithful logical reasoning. In: Findings of the Association for Computational Linguistics: EMNLP, Association for Computational Linguistics (2023)
32. Petcu, R., Bhargav, S., de Rijke, M., Kanoulas, E.: A comprehensive taxonomy of negation for NLP and neural retrievers. arXiv preprint arXiv:2507.22337 (2025)
33. Po?lya, G.: Mathematics and Plausible Reasoning. Princeton University Press, Princeton (1968)
34. Raedt, L.D., Kimmig, A., Toivonen, H.: Problog: a probabilistic prolog and its application in link discovery. In: IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence (2007)
35. van Rijsbergen, C.: A new theoretical framework for information retrieval. SIGIR Forum (1986)
36. van Rijsbergen, C.J., Lalmas, M.: Information calculus for information retrieval. J. Am. Soc. Inf, Sci (1996)
37. Robertson, S.E., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Found. Trends Inf. Retr. **3**(4) (2009)
38. Shen, Y., Chen, S., Xu, X., Zhang, Y., Malaviya, C., Roth, D.: LogiCoL: logically-informed contrastive learning for set-based dense retrieval. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2025)
39. Shorinwa, O., Mei, Z., Lidard, J., Ren, A.Z., Majumdar, A.: A survey on uncertainty quantification of large language models: taxonomy, open research challenges, and future directions. CoRR **abs/2412.05563** (2024), https://doi.org/10.48550/ARXIV.2412.05563
40. Soudani, H., Kanoulas, E., Hasibi, F.: Why uncertainty estimation methods fall short in RAG: an axiomatic analysis. In: Findings of the Association for Computational Linguistics: ACL 2025, pp. 16596–16616, Association for Computational Linguistics (2025)
41. Wang, L., et al.: Text embeddings by weakly-supervised contrastive pre-training. ArXiv (2022)
42. Wei, J., et al.: Chain-of-thought prompting elicits reasoning in large language models. In: Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022 (2022)
43. Weller, O., Boratko, M., Naim, I., Lee, J.: On the theoretical limitations of embedding-based retrieval (2025)
44. Weller, O., Lawrie, D.J., Durme, B.V.: Nevir: negation in neural information retrieval. In: Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL, Association for Computational Linguistics (2024)
45. Wong, L., et al.: Modeling open-world cognition as on-demand synthesis of probabilistic models. CoRR **abs/2507.12547** (2025)
46. Wong, L., et al.: From word models to world models: translating from natural language to the probabilistic language of thought (2023)
47. Yang, K., et al.: Leandojo: theorem proving with retrieval-augmented language models. Adv. Neural. Inf. Process. Syst. **36**, 21573–21612 (2023)
48. Yee, E., Li, A., Tang, C., Jung, Y.H., Paturi, R., Bergen, L.: Faithful and unfaithful error recovery in chain of thought. In: First Conference on Language Modeling (COLM) (2024)

49. Yue, Y., et al.: Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In: 2nd AI for Math Workshop @ ICML 2025 (2025)
50. Zhang, W., et al.: Excluir: exclusionary neural information retrieval. In: AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence (2025)
51. Zhuang, H., et al.: Beyond yes and no: improving zero-shot LLM rankers via scoring fine-grained relevance labels. In: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), Association for Computational Linguistics (2024)