No More, No Less: Noise-Free Module Retrieval from Large-Scale Knowledge Bases

Anonymous Author(s)

Abstract

Retrieving precisely relevant knowledge from large-scale knowledge bases is crucial for knowledge-intensive IR tasks such as question answering and retrieval-augmented generation. A fundamental challenge is extracting knowledge subsets that preserve semantic completeness while eliminating irrelevant symbols—achieving **noise-free knowledge retrieval**. Syntax-based methods are efficient but inevitably leak symbols beyond the target vocabulary, injecting noise that degrades downstream reasoning and generation quality. Semantic methods based on the forgetting procedure offer formal precision guarantees but have been deemed impractical due to prohibitive computational complexity—leaving the IR community without effective tools for noise-free knowledge access.

This paper breaks this long-standing barrier. We present an efficient forgetting-based method for computing vocabulary-restricted modules from ontological knowledge bases, featuring a novel linear-time normalization strategy that fundamentally avoids the exponential blow-up plaguing state-of-the-art systems. Empirical evaluation on large-scale benchmarks demonstrates 100% success rate, significantly outperforming existing forgetting tools in both time and space efficiency while producing substantially more compact, noise-free modules. Downstream RAG experiments further validate practical impact: our method improves reasoning accuracy by 7.5%, reduces hallucinations by 72%, and cuts context costs by 75%. Our method provides the IR community with the first practical solution for precision-guaranteed knowledge retrieval—enabling cleaner inputs for RAG pipelines, more reliable knowledge-enhanced reasoning, and secure information hiding at scale.

Our implementation, experimental data, and extended results including missing proofs of the lemmas and theorems are publicly available at https://github.com/anonymous-ai-researcher/module.

Keywords

Knowledge Retrieval, Knowledge Bases, Noise-Free Retrieval, Module Extraction, Information Hiding, Ontology, Forgetting

1 Introduction

The emergence of knowledge-intensive information retrieval has fundamentally transformed how systems access and utilize structured knowledge. From RAG pipelines [36] to knowledge-enhanced question answering [67] and entity retrieval systems [56], modern IR applications increasingly rely on extracting relevant knowledge subsets from large-scale Knowledge Bases (KBs). However, a critical yet underexplored challenge persists: *how can we retrieve knowledge that is both semantically complete and free from irrelevant noise?*

Recent studies highlight the detrimental impact of retrieval noise on downstream IR tasks. Cuconasu et al. [10] demonstrated that documents which are semantically related but do not directly answer the query—termed "distracting" documents—can significantly degrade RAG system performance, sometimes more severely than entirely random documents. This finding underscores a fundamental tension in knowledge retrieval: maximizing recall often introduces noise that compromises precision, ultimately harming task performance. The problem is particularly acute when retrieving from structured KBs, where noise manifests as *irrelevant symbols*—concept and role names that fall outside the user's intended query scope yet contaminate the retrieved knowledge subset.

KBs formalized as ontologies [23] provide explicit, machine-interpretable specifications of domain knowledge through expressive logic [4]. These structured representations underpin critical applications ranging from biomedical knowledge systems [56] to legal case retrieval [14, 37, 58] and intelligent question answering [67]. The growing scale of such KBs, exemplified by resources like SNOMED CT [55] and YAGO [56], has made efficient and precise knowledge retrieval an increasingly pressing concern.

1.1 The Noise Problem in Knowledge Retrieval

Given an ontological KB $\mathcal{O}$ and a target vocabulary Σ (i.e., a set of concept and role names of $\mathcal{O}$ representing the user's query terms), the goal of **vocabulary-restricted knowledge retrieval** is to extract a module $\mathcal{M}$ that (i) preserves all logical entailments relevant to Σ, and (ii) contains *only* symbols from Σ—achieving what we call **zero-noise retrieval**. This dual requirement ensures that the retrieved knowledge is both *complete* (no relevant information is lost) and *precise* (no irrelevant information is introduced).

Two paradigms exist for this task, presenting practitioners with a fundamental trade-off. **Syntax-based methods** [19, 20] extract a direct subset of the original axioms, offering computational efficiency. However, these methods inevitably include axioms containing symbols outside the target vocabulary Σ to preserve semantic integrity. This "symbol leakage" introduces noise—extraneous concepts and relations that pollute the retrieved knowledge and can propagate errors in downstream reasoning tasks [10, 59].

Semantic methods based on *forgetting* [40]—a non-standard reasoning procedure—offer a principled alternative. These methods produce "rewritten" modules containing exclusively symbols from the target vocabulary, achieving zero-noise retrieval with formal guarantees [68]. However, the high computational complexity of this reasoning task—ExpTime-complete for $\mathcal{EL}$ [46] and 2ExpTime-complete for $\mathcal{ALC}$ [45] with potentially triple-exponential output size—has rendered them largely impractical for real-world KBs, compelling the community to accept the noise introduced by syntax-based alternatives.

Example 1. *Consider the ontology $\mathcal{O}$ and target vocabulary $\Sigma = \{r, A_0, A_{100}\}$ representing the query terms:*

$$\mathcal{O} = \{A_1 \sqsubseteq \exists r.A_1\} \cup \{A_i \sqsubseteq A_{i+1} \mid 0 \leq i \leq 99\}.$$

A zero-noise (vocabulary-restricted) module is:

$$\mathcal{M}_\Sigma = \{A_0 \sqsubseteq A_{100}, A_0 \sqsubseteq \exists r.A_{100}\},$$

which captures complete semantics over Σ while entirely excluding irrelevant intermediate symbols $A_1, \ldots, A_{99}$. In contrast, the minimal syntax-based module is $\mathcal{O}$ itself—all 101 axioms are required, "leaking" 99 symbols outside the target vocabulary. For applications requiring information hiding or privacy preservation [1, 7, 21, 24, 29, 61], such leakage is unacceptable.

This example illustrates that zero-noise modules can be substantially more compact than the subset modules. This compactness directly benefits downstream IR tasks: smaller modules reduce computational overhead in knowledge-enhanced reasoning [26], minimize context window consumption in RAG pipelines [42], and establish clearer semantic boundaries—critical for privacy-preserving applications and secure knowledge sharing [22].

1.2 Our Contributions

This paper challenges the long-standing assumption that precision-guaranteed knowledge retrieval is computationally impractical. We present a highly efficient forgetting-based method for computing vocabulary-restricted modules from $\mathcal{ALCI}$ KBs—an expressive Description Logic (DL) [4] supporting inverse roles, essential for modeling bidirectional relationships prevalent in biomedical, legal, and security domains. Our key contributions include:

- **Linear-Time Normalization Strategy**: We introduce a novel normalization approach that confines the introduction of auxiliary symbols ("definers") to a single preprocessing pass, fundamentally avoiding the exponential blow-up observed in state-of-the-art systems.
- **Efficient Retrieval Calculus**: Building on our normalization, we develop specialized inference rules for eliminating concept and role names while preserving semantic equivalence over the target vocabulary. The resulting calculus achieves efficiency competitive with syntax-based methods despite the inherently higher complexity of semantic approaches.
- **Comprehensive Empirical Validation**: Experiments on large-scale benchmarks demonstrate 100% success rate, substantially outperforming in state-of-the-art systems in computation time (up to 50× faster) and memory usage (20–40% reduction), while producing significantly more compact, noise-free modules.
- **Downstream RAG Validation**: We integrate our method into a RAG pipeline for biomedical question answering, demonstrating that zero-noise retrieval improves reasoning accuracy by 7.5%, reduces hallucinations by 72%, and cuts context costs by 75% compared to syntax-based alternatives.

Our method shows the potential to address the critical need for noise-free knowledge retrieval in modern IR applications. By ensuring that retrieved KB contains exactly the relevant information—no more, no less—we enable more robust RAG pipelines, more precise knowledge-enhanced reasoning, and secure knowledge sharing with formal information-hiding guarantees.

2 Related Work

2.1 Knowledge Retrieval and the Noise Problem

The integration of external knowledge into retrieval and generation systems has become a cornerstone of modern IR. RAG [13, 36, 38, 39, 50, 62, 62, 75, 76] demonstrated that coupling LLMs with retrieval mechanisms significantly improves performance on knowledge-intensive tasks. Subsequent work has refined this paradigm through self-reflective retrieval [3], efficient fusion mechanisms [26], and unified ranking across multiple retrieval sources [53].

However, a growing body of research highlights that *retrieval quality*—not just retrieval quantity—critically impacts downstream performance. Liu et al. [42] showed that LLMs struggle when relevant information is buried among irrelevant context (the "lost in the middle" phenomenon). More directly, Cuconasu et al. [10] found that semantically related but non-answering documents degrade RAG performance more than random noise, motivating research into retrieval robustness [62] and denoising mechanisms [59].

When the knowledge source is a "structured" KB (e.g., ontologies) rather than a document corpus, noise takes a distinct form: *irrelevant symbols* (concepts and roles) that fall outside the user's query scope. Unlike textual noise, which can be mitigated through ranking or filtering [10, 69], symbolic noise in logical KBs can propagate through inference chains [54, 70], potentially corrupting derived conclusions. Our work addresses this challenge by providing the *formal guarantees* of zero symbolic noise [5, 16]—the retrieved module contains exactly the symbols requested, with complete preservation of relevant semantics.

Knowledge graphs have been increasingly integrated into the IR systems for enhanced retrieval [11, 67], reasoning [41, 57], and evaluation [49]. Pan et al. [48] provide a comprehensive roadmap for unifying LLMs with knowledge graphs, identifying knowledge retrieval quality as a key challenge. Large-scale KBs like YAGO [56] exemplify the scale at which precise retrieval becomes critical. Our method directly addresses this need by enabling noise-free extraction from arbitrarily large-scale KBs.

2.2 Forgetting from KBs

Forgetting provides the theoretical foundation for zero-noise knowledge retrieval. Lin and Reiter [40] introduced forgetting for first-order logic, where eliminating a predicate P from theory $\mathcal{O}$ yields a new theory whose models coincide with those of $\mathcal{O}$ modulo P's interpretation. This **strong forgetting** (or model-theoretic forgetting) [71] may produce results inexpressible in the source logic, necessitating second-order constructs [15].

Zhang and Zhou [71] distinguished strong forgetting from **weak forgetting**, which preserves only first-order consequences over the remaining vocabulary. Weak forgetting is closely related to **uniform interpolation** [12, 25, 64]—a strengthening of Craig interpolation [9] that has been extensively studied in DLs.

For DLs underlying modern KBs, theoretical results reveal significant challenges. Few DLs are closed under forgetting—even lightweight logics like $\mathcal{EL}$ [44]. Determining whether forgetting results exist is undecidable for $\mathcal{EL}$ and $\mathcal{ALC}$ [31], and even when results exist, their size can be triple-exponential [43, 45, 46]. These complexity barriers have historically relegated forgetting to theoretical interest rather than practical application. The practical landscape is dominated by two systems. LETHE [34] employs resolution-based calculus for weak forgetting in expressive logics including $\mathcal{ALCI}$. FAME [74] uses Ackermann's Lemma for strong forgetting in $\mathcal{ALCOIH}$. Other systems including NUI [32] and methods by Xiang et al. [66] are no longer maintained. Our work demonstrates that careful algorithmic design can overcome the practical barriers, achieving efficiency that rivals syntax-based methods while maintaining the formal precision guarantees of semantic approaches.

2.3 Module Extraction from KBs

Module extraction partitions a KB to isolate subsets relevant to a given vocabulary [20, 22]. Approaches can be broadly categorized as either model-theoretic [31] or deductive [33]. These notions have been refined into three main variants: *plain*, *self-contained*, and *depleting* modules [31]. While plain modules capture the basic concept, self-contained modules impose stricter constraints, requiring the module to preserve all logical consequences over not just the initial vocabulary, but also over all names within the module itself. Depleting modules go even further, requiring the remainder of the ontology to be logically independent of the module. For all variants, a minimality criterion ensures that no proper subset of a module is also a valid module of the same type.

Based on these principles, a variety of modularization tools have been developed. Foundational work by Konev et al. established the theoretical underpinnings for these methods [31]. Their MEX tool provides exact, minimal modules for acyclic $\mathcal{ELI}$-terminologies in polynomial time and was later extended to DL-Lite [33] and acyclic $\mathcal{ALCI}$ [30]. Its successor, AMEX, extends this capability to acyclic $\mathcal{ALCQI}$-ontologies, albeit at a higher computational cost [17].

For expressive DLs, tractable approximations dominate. Locality-based methods (LBMs) [20], implemented in the OWL API, extract self-contained depleting modules for $\mathcal{SROIQ}$ in polynomial time through syntactic locality tests (TOP, BOT, STAR). Reachability-based methods (RBMs) [47] produce smaller modules for $\mathcal{SRIQ}$ but sacrifice self-containment. Datalog-based methods (DBMs) [28, 52], implemented in PrisM [51], leverage optimized Datalog engines for efficient extraction.

We compare against five established baselines producing model-theoretic modules: TOP, BOT, STAR, AMEX, and PrisM. We exclude MEX as it does not support $\mathcal{ALCI}$. For forgetting methods, we benchmark against LETHE (weak forgetting) and FAME (strong forgetting), noting that direct success rate comparison with FAME is inappropriate due to differing theoretical guarantees.

3 Preliminaries

3.1 The Description Logic $\mathcal{ALCI}$

Ontological KBs are often formulated using DLs [2, 4], a family of knowledge representation formalisms widely adopted in ontology modeling and increasingly employed in knowledge-intensive IR applications [48]. DLs constitute a prominent family of knowledge representation formalisms, with various variants that differ in expressivity depending on which logical connectives are permitted. A basic DL, called $\mathcal{ALC}$, utilizes concept names, role names, and the logical connectives of $\neg$, $\sqcap$, $\sqcup$, $\exists$, and $\forall$ to build complex concepts. The language considered in this paper is the DL $\mathcal{ALCI}$, which extends $\mathcal{ALC}$ with the inverse role ($\mathcal{I}$). While increased expressivity enhances the capacity for knowledge representation, it typically incurs additional computational complexity.

Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be pairwise disjoint, countably infinite sets of *concept* and *role* names, respectively. *Roles* in $\mathcal{ALCI}$ can be a role name $\mathsf{r} \in \mathsf{N_R}$ or the inverse r^- of a role name r. *Concept descriptions* (or *concepts* for short) in $\mathcal{ALCI}$ have one of the following forms:

$$\top \mid \bot \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C,$$

where $A \in \mathsf{N_C}$, C and D range over concepts, and R ranges over roles. If R is a role, we define the inverse $\mathsf{Inv}(R)$ of R by $\mathsf{Inv}(r) = r^-$ and $\mathsf{Inv}(r^-) = r$, for all $r \in \mathsf{N_R}$.

An $\mathcal{ALCI}$-KB $\mathcal{O}$ is a finite set of axioms of the form $C \sqsubseteq D$ (namely *concept inclusion*, or *CI* for short), where C, D are concepts. We use $C \equiv D$ as a shorthand for the pair $C \sqsubseteq D$ and $D \sqsubseteq C$. Hence, in this paper, an $\mathcal{ALCI}$-KB is assumed to contain only CIs.

The semantics of $\mathcal{ALCI}$ is defined in terms of an *interpretation* $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ is a non-empty set, known as the *domain of the interpretation*, and $\cdot^{\mathcal{I}}$ is the *interpretation function* that maps every concept name $A \in \mathsf{N_C}$ to a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and every role name $r \in \mathsf{N_R}$ to a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function $\cdot^{\mathcal{I}}$ is inductively extended to concepts as follows:

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}} \qquad \bot^{\mathcal{I}} = \emptyset \qquad (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$$
$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}} \qquad (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$$
$$(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y.(x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$$
$$(\forall R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \forall y.(x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$$
$$(R^-)^{\mathcal{I}} = \{(y,x) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (x,y) \in R^{\mathcal{I}}\}$$

We say a CI $C \sqsubseteq D$ holds in an interpretation $\mathcal{I}$, written $\mathcal{I} \models C \sqsubseteq D$, if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. $\mathcal{I}$ is a *model* of a KB $\mathcal{O}$, denoted $\mathcal{I} \models \mathcal{O}$, if every CI holds in $\mathcal{O}$. A CI $C \sqsubseteq D$ is a *logical consequence* of $\mathcal{O}$, written $\mathcal{O} \models C \sqsubseteq D$, if it holds in every model of $\mathcal{O}$.

Our retrieval method operates on $\mathcal{ALCI}$-KBs in *clausal normal form* defined below.

Definition 1 (**Literals and Clauses**). *A* literal *in $\mathcal{ALCI}$ is a concept of the form A, $\neg A$, $\exists R.C$, or $\forall R.C$, where $A \in N_C$, R is a role, and C is a concept. A* clause *in $\mathcal{ALCI}$ is a CI of the form $\top \sqsubseteq L_1 \sqcup \ldots \sqcup L_n$, where each L_i is a literal. We omit the prefix '$\top \sqsubseteq$' and treat clauses as sets, meaning that they contain no duplicates and their order is not important. Hence, when stating that $C^{\mathcal{I}} \cup (\geq mr.D)^{\mathcal{I}}$ holds, we imply that the CI $\top \sqsubseteq C \sqcup (\geq mr.D)$ holds in $\mathcal{I}$. An $\mathcal{ALCI}$-KB is in clausal normal form if all its CIs are clauses.*

Clauses are obtained by applying a series of standard transformations to the CIs. This process runs in polynomial time. In this paper, unless stated otherwise, $\mathcal{ALCI}$-KBs are assumed to be a finite set of *clauses*.

3.2 Zero-Noise Knowledge Retrieval

We formalize the task of **Zero-Noise Knowledge Retrieval** based on the notion of *inseparability* [6, 31]. A *vocabulary* $\Sigma \subseteq \mathsf{N_C} \cup \mathsf{N_R}$ is a finite set of concept and role names representing the user's query terms. For any syntactic object X—which may range over concepts, roles, CIs, clauses, or KBs—we write $\mathsf{sig_C}(X)$ and $\mathsf{sig_R}(X)$ for the sets of concept and role names in X, respectively, and $\mathsf{sig}(X) = \mathsf{sig_C}(X) \cup \mathsf{sig_R}(X)$ for their union. For any vocabulary Σ, we call X a Σ-*object* (e.g., Σ-concept, Σ-CI, etc.) if $\mathsf{sig}(X) \subseteq \Sigma$.

Definition 2 (Semantic Equivalence for $\mathcal{ALCI}$). *Let $\mathcal{O}$ and $\mathcal{M}$ be $\mathcal{ALCI}$-KBs and Σ be a vocabulary of concept and role names.*

We say that $\mathcal{O}$ and $\mathcal{M}$ are semantically equivalent *w.r.t.* Σ *(or* Σ-inseparable*), written* $\mathcal{O} \equiv_\Sigma \mathcal{M}$*, if for any* $\mathcal{ALCI}$*-CI* $C \sqsubseteq D$ *with* $sig(C \sqsubseteq D) \subseteq \Sigma$*:* $\mathcal{O} \models C \sqsubseteq D$ *iff* $\mathcal{M} \models C \sqsubseteq D$*.*

This notion of semantic equivalence captures the requirement that two KBs are indistinguishable from the perspective of a user interested only in the vocabulary Σ—they entail exactly the same logical consequences over the query terms.

Definition 3 (Subset Module for $\mathcal{ALCI}$). *Let $\mathcal{O}$ and $\mathcal{M}$ be $\mathcal{ALCI}$-KBs and $\Sigma \subseteq sig(\mathcal{O})$ be a vocabulary of concept and role names. We say that $\mathcal{M}$ is a* subset module *for $\mathcal{O}$ w.r.t. Σ if: (i) $\mathcal{O} \equiv_\Sigma \mathcal{M}$ (semantic completeness) and (ii) $\mathcal{M} \subseteq \mathcal{O}$ (syntactic subset).*

Subset modules, while preserving semantic equivalence, inevitably contain symbols outside the target vocabulary Σ to maintain logical integrity. This "symbol leakage" introduces retrieval noise that can degrade downstream IR tasks [10].

Definition 4 (Zero-Noise Module for $\mathcal{ALCI}$). *Let $\mathcal{O}$ and $\mathcal{M}$ be $\mathcal{ALCI}$-KBs and $\Sigma \subseteq sig(\mathcal{O})$ be a vocabulary of concept and role names. $\mathcal{M}$ is a* zero-noise module *(or equivalently, a* vocabulary-restricted module*) for $\mathcal{O}$ w.r.t. Σ if: (i) $\mathcal{O} \equiv_\Sigma \mathcal{M}$ (semantic completeness) and (ii) $sig(\mathcal{M}) \subseteq \Sigma$ (zero symbol leakage).*

A zero-noise module $\mathcal{M}$ of a KB $\mathcal{O}$ can be computed via *forgetting*, a reasoning procedure that eliminates all symbols from $\mathcal{O}$ not present in the target vocabulary Σ [45, 65, 73].

Definition 5 (Forgetting for $\mathcal{ALCI}$). *Let $\mathcal{O}$ be an $\mathcal{ALCI}$-KB and $\mathcal{F} \subseteq sig(\mathcal{O})$ be a set of concept and role names to be eliminated, referred to as the* forgetting vocabulary*. An $\mathcal{ALCI}$-KB $\mathcal{M}$ is a* result of forgetting $\mathcal{F}$ from $\mathcal{O}$ *if: (i) $\mathcal{O} \equiv_\Sigma \mathcal{M}$, and (ii) $sig(\mathcal{M}) \subseteq sig(\mathcal{O}) \setminus \mathcal{F}$.*

The forgetting process effectively distills $\mathcal{O}$ into a focused, noise-free retrieval result $\mathcal{M}$ by concentrating on the target vocabulary $\Sigma = \mathrm{sig}(\mathcal{O}) \setminus \mathcal{F}$. Within $\mathcal{ALCI}$, $\mathcal{M}$ preserves the complete semantic behavior of $\mathcal{O}$ w.r.t. Σ, meaning that they generate identical logical consequences over the query terms. This definition yields two important properties essential for reliable knowledge retrieval:

- **Order Independence**: The results of forgetting $\mathcal{F}$ from $\mathcal{O}$ can be computed by sequentially eliminating individual symbols in $\mathcal{F}$, independent of the elimination order.
- **Uniqueness**: The results of forgetting are unique up to logical equivalence—any results obtained from the same forgetting procedure are semantically identical, despite potential syntactic differences in their explicit representations.

These properties ensure that zero-noise knowledge retrieval is *deterministic* and *reproducible*.

4 Linear-Time Normalization for Scalable Retrieval

The key to achieving noise-free retrieval "at scale" lies in controlling the computational overhead of semantic methods. A critical prerequisite for our efficient retrieval method is transforming the input KB into a structured, normalized form. However, the standard approach of introducing fresh concept names—so-called "**definers**" [35]—to represent complex subformulas can itself become a major source of complexity. If not carefully controlled, this can lead to a combinatorial explosion of new symbols, negating any potential performance gains and rendering precise retrieval impractical for large KBs.

The cornerstone of our method is a novel normalization strategy that guarantees both the number of introduced definers and the computational effort for normalization grow only linearly with the size of the input KB. Unlike state-of-the-art approaches that dynamically introduce definers during the inference process, our approach confines this step to a single, efficient pre-processing pass. This very design fundamentally avoids the exponential blow-up that has historically plagued forgetting-based retrieval methods, enabling zero-noise retrieval from KBs at scales previously considered intractable.

Our retrieval method operates on two specialized normal forms, each tailored for the elimination of a specific type of symbol. Let $A \in \mathsf{N_C}$ ($r \in \mathsf{N_R}$) be the concept (role) name designated for elimination.

Definition 6 (A-**Reduced Form**). *A clause is in A-*reduced form *if it has the form $C \sqcup L$, where $L \in \{A, \neg A, Qr.A, Qr.\neg A, Qr^-.A, Qr^-.\neg A\}$ for $Q \in \{\exists, \forall\}$ and some arbitrary $r \in N_R$, and C is a clause with $A \notin sig(C)$. An $\mathcal{ALCI}$-KB is in A-*reduced form *if all its A-clauses are in A-reduced form.*

Definition 7 (r-**Reduced Form**). *A clause is in r-*reduced form *if it has the form $C \sqcup L$, where $L \in \{Qr.D, Qr^-.D\}$ for $Q \in \{\exists, \forall\}$, and C, D are concepts with $r \notin sig(C) \cup sig(D)$. An $\mathcal{ALCI}$-KB is in r-*reduced form *if all its r-clauses are in r-reduced form.*

Essentially, these two normal forms ensure that the symbol to be eliminated appears at most once per clause and is isolated within a syntactically simple structure. Any clause that does not conform to these patterns is transformed into the desired form through a series of transformation rules described below. This transformation may introduce a fresh definer $Z \in \mathsf{N_C}$ to replace a non-conforming sub-formula, along with a new clause that defines Z:

1. For each A-clause instance $L_1 \sqcup \ldots \sqcup L_n$ where A appears more than once, if there is a literal L_i ($1 \leq i \leq n$) of the form $\exists R.C$ or $\forall R.C$ with $A \in \mathrm{sig}(C)$, replace C with a fresh $Z \in \mathsf{N_C}$, and add $\neg Z \sqcup C$ to $\mathcal{O}$.
2. For each A-clause instance $L_1 \sqcup \ldots \sqcup L_n$ where A appears exactly once, if there is a literal L_i ($1 \leq i \leq n$) of the form $\exists R.C$ or $\forall R.C$ with $A \in \mathrm{sig}(C)$ and $C \neq A$, replace C with a fresh $Z \in \mathsf{N_C}$, and add $\neg Z \sqcup C$ to $\mathcal{O}$.
3. For each r-clause instance $L_1 \sqcup \ldots \sqcup L_n$, if there is a literal L_i ($1 \leq i \leq n$) with $r \in \mathrm{sig}(L_i)$ and r appears elsewhere in the clause, replace L_i with a fresh definer $Z \in \mathsf{N_C}$ and add $\neg Z \sqcup L_i$ to $\mathcal{O}$.
4. For each r-clause instance $L_1 \sqcup \ldots \sqcup L_n$, if there is a literal L_i of the form $\exists R.C$ or $\forall R.C$ with $r \in \mathrm{sig}(C)$, replace C with a fresh definer $Z \in \mathsf{N_C}$ and add $\neg Z \sqcup C$ to $\mathcal{O}$.

Lemma 1. *Let $\mathcal{O}$ be an $\mathcal{ALCI}$-KB and $\mathcal{O}'$ its reduced form obtained via the normalization. Then $\mathcal{O} \equiv_{sig(\mathcal{O})} \mathcal{O}'$.*

Lemma 2. *For any $\mathcal{ALCI}$-KB $\mathcal{O}$, a reduced form $\mathcal{O}'$ (either A-reduced or r-reduced) can be computed in a linear number of normalization steps. Moreover, $|\mathcal{O}'|$ is linear in $|\mathcal{O}|$, where $|\mathcal{O}|$ denotes the size of $\mathcal{O}$ as defined in [4].*

5 The Retrieval Calculus

The elimination of a single concept or role symbol is based on two generalized inference rules. These rules function as replacement rules, where the premises (i.e., clauses above the line) are replaced by the conclusion (i.e., those below the line), systematically removing unwanted symbols while preserving all relevant semantic content.

5.1 Concept Name Elimination

The elimination of a concept name A from A-reduced KBs relies on the *combination rule* in Figure 1. We observe that A-reduced clauses containing exactly one occurrence of A can be partitioned into *Positive Premises* (where A occurs positively) and *Negative Premises* (where A occurs negatively).[1] For concept name elimination, we exclude A-clauses of the form $C \sqcup \forall r^{-}.A$ or $C \sqcup \forall r^{-}.\neg A$ from the A-reduced form, as these can be equivalently transformed into $A \sqcup \forall r.C$ and $\neg A \sqcup \forall r.C$, respectively [72]. This yields at most eight distinct clause forms, with specific notation detailed in Figure 1. The combination rule handles all $4 \times 4 = 16$ pairwise combinations of positive and negative premises. We define $\mathcal{P}^{+}(A)$ as the union of all positive premise sets, $\mathcal{P}^{-}(A)$ as the union of all negative premise sets, and $\mathcal{O}^{-A}$ as the set of non-A-clauses.

The combination rule resolves each positive premise with each negative premise on the symbol A, generating all logical entailments over $\mathrm{sig}(\mathcal{O}) \setminus \{A\}$. Each resolution step produces a finite set of A-free clauses, denoted $\mathrm{combine}(\alpha, \beta)$ for clause pair α and β. Since clauses are organized by syntactic form, combinations are computed at the group level (e.g., $\mathrm{combine}(\mathcal{P}^{+}_{\mho}(A), \mathcal{P}^{-}_{\exists}(A))$), with the overall result being $\mathrm{combine}(\mathcal{P}^{+}(A), \mathcal{P}^{-}(A))$. After resolution, all A-clauses are removed, yielding a KB $\mathcal{M}$ free of A.

Lemma 3 (Soundness). *Let $\mathcal{O}$ be an A-reduced $\mathcal{ALCI}$-KB with $A \in sig_C(\mathcal{O})$. If $\mathcal{M}$ is the KB derived by applying the combination rule in Figure 1, then $\mathcal{O} \equiv_{sig(\mathcal{O})\setminus\{A\}} \mathcal{M}$.*

Lemma 3 establishes the partial soundness of the calculus. Specifically, the resulting KB $\mathcal{M}$ satisfies the first condition of Definitions 5 and 4 for being a forgetting result and thus a zero-noise module of $\mathcal{O}$ w.r.t. $\Sigma = \mathrm{sig}(\mathcal{O}) \setminus \{A\}$. However, $\mathcal{M}$ may contain definers outside Σ, potentially violating the zero-noise requirement. We address these residual definers in Section 5.3.

5.2 Role Name Elimination

Eliminating a role name r from r-reduced KBs relies on the *combination rule* in Figure 2. The rule resolves positive premises (where r occurs positively) with negative premises (where r occurs negatively) on r, generating all entailments over $\mathrm{sig}(\mathcal{O}) \setminus \{r\}$.

Lemma 4 (Soundness). *Let $\mathcal{O}$ be an r-reduced $\mathcal{ALCI}$-KB with $r \in sig_R(\mathcal{O})$. If $\mathcal{M}$ is the KB derived by applying the combination rule in Figure 2, then $\mathcal{O} \equiv_{sig(\mathcal{O})\setminus\{r\}} \mathcal{M}$.*

A DL reasoner [18] verifies the side conditions of the combination rules. Subsumption checking in $\mathcal{ALCI}$ is ExpTime-complete [60].

[1]Specifically, an occurrence of a symbol $\mathcal{S}$ in a clause is *positive* if it appears under an even number of (explicit or implicit) negations, and *negative* otherwise.

5.3 The Zero-Noise Retrieval Process

Algorithm 1 presents our complete zero-noise retrieval procedure. It iteratively applies the normalization and elimination calculi to each symbol in the forgetting vocabulary $\mathcal{F}$, producing intermediate results that may contain definers. After processing all symbols in $\mathcal{F}$, it attempts to eliminate any introduced definers using the same calculus, treating them as regular concept names.

Zero-noise retrieval is not guaranteed to be complete for $\mathcal{ALCI}$, meaning finite results within the expressivity of $\mathcal{ALCI}$ may not always exist [45]. For KBs with cyclic definitions (e.g., $\{B \sqcup A, \neg A \sqcup \exists r.A\}$), eliminating A yields an infinite set of clauses ($\{B \sqcup \exists r.\top, B \sqcup \exists r.\exists r.\top, \ldots\}$) that cannot be finitely axiomatized in $\mathcal{ALCI}$. While fixpoints can handle such cycles [8]—as implemented in LETHE and FAME—existing DL reasoners lack fixpoint support. Our method ensures termination by ceasing attempts to eliminate problematic definers and retaining them in the final result.

Note that cycles do not necessarily prevent finding finite results, as some symbols may be removed when eliminating others, resolving many cyclic dependencies. Rather than simply detecting cycles, our approach employs a blocking technique [27] that identifies when eliminating a definer produces a pattern syntactically identical to a previous intermediate result, indicating the algorithm has entered an elimination cycle where removing one definer introduces another in its place. In such cases, the method halts and declares retrieval unsuccessful.

Theorem 1. *For any $\mathcal{ALCI}$-KB $\mathcal{O}$ and forgetting vocabulary $\mathcal{F} \subseteq sig(\mathcal{O})$, our retrieval method always terminates and returns an $\mathcal{ALCI}$-KB $\mathcal{M}$. If $\mathcal{M}$ is definer-free, $\mathcal{M}$ is the result of forgetting $\mathcal{F}$ from $\mathcal{O}$ and thus a zero-noise module of $\mathcal{O}$ w.r.t. $\Sigma = sig(\mathcal{O}) \setminus \mathcal{F}$.*

6 Empirical Evaluation

We evaluate our zero-noise retrieval method through comprehensive experiments on real-world KBs, addressing the following research questions:

- **RQ1 (Reliability)**: How reliably can our method achieve zero-noise retrieval compared to state-of-the-art approaches?
- **RQ2 (Compactness)**: How compact are the retrieved modules, and do they effectively eliminate symbolic noise?
- **RQ3 (Efficiency)**: What is the retrieval efficiency in terms of time and memory consumption?
- **RQ4 (Downstream Impact)**: Does zero-noise retrieval translate to tangible benefits in downstream IR applications such as RAG?

6.1 Experimental Setup

We developed a prototype (Proto) of our method in Python using OWL API Version 5.1.7.[2] We benchmarked this prototype against two state-of-the-art forgetting methods: LETHE[3] and FAME[4], alongside several established subset modularization tools including TOP, BOT, STAR, AMEX, and PrisM.

The evaluation utilized two comprehensive real-world KB corpora: the **Oxford ISG Library**[5] containing KBs from multiple

[2]http://owlcs.github.io/owlapi/
[3]https://lat.inf.tu-dresden.de/~koopmann/LETHE/
[4]http://www.cs.man.ac.uk/~schmidt/sf-fame/
[5]http://krr-nas.cs.ox.ac.uk/ontologies/lib/

$$\begin{array}{c}
\mathcal{O}^{-\mathsf{A}}, \overbrace{B_1 \sqcup \mathsf{A}, \ldots, B_l \sqcup \mathsf{A}}^{\mathcal{P}^+_{\mho}(\mathsf{A})}, \overbrace{C_1 \sqcup \exists r_1.\mathsf{A}, \ldots, C_m \sqcup \exists r_m.\mathsf{A}}^{\mathcal{P}^+_{\exists}(\mathsf{A})}, \overbrace{D_1 \sqcup \exists s_1^-.\mathsf{A}, \ldots, D_n \sqcup \exists s_n^-.\mathsf{A}}^{\mathcal{P}^+_{\exists,-}(\mathsf{A})}, \overbrace{\phi_1 \sqcup \forall t_1.\mathsf{A}, \ldots, \phi_o \sqcup \forall t_o.\mathsf{A}}^{\mathcal{P}^+_{\forall}(\mathsf{A})} \\
\overbrace{E_1 \sqcup \neg\mathsf{A}, \ldots, E_{l'} \sqcup \neg\mathsf{A}}^{\mathcal{P}^-_{\mho}(\mathsf{A})}, \overbrace{F_1 \sqcup \exists u_1.\neg\mathsf{A}, \ldots, F_{m'} \sqcup \exists u_{m'}.\neg\mathsf{A}}^{\mathcal{P}^-_{\exists}(\mathsf{A})}, \overbrace{G_1 \sqcup \exists v_1^-.\neg\mathsf{A}, \ldots, G_{n'} \sqcup \exists v_{n'}^-.\neg\mathsf{A}}^{\mathcal{P}^-_{\exists,-}(\mathsf{A})}, \overbrace{\psi_1 \sqcup \forall w_1.\neg\mathsf{A}, \ldots, \psi_{o'} \sqcup \forall w_{o'}.\neg\mathsf{A}}^{\mathcal{P}^-_{\forall}(\mathsf{A})} \\
\hline
\mathcal{O}^{-\mathsf{A}}, \mathsf{combine}(\mathcal{P}^+_{\mho}(\mathsf{A}), \mathcal{P}^-_{\mho}(\mathsf{A})), \mathsf{combine}(\mathcal{P}^+_{\mho}(\mathsf{A}), \mathcal{P}^-_{\exists}(\mathsf{A})), \mathsf{combine}(\mathcal{P}^+_{\mho}(\mathsf{A}), \mathcal{P}^-_{\exists,-}(\mathsf{A})), \mathsf{combine}(\mathcal{P}^+_{\mho}(\mathsf{A}), \mathcal{P}^-_{\forall}(\mathsf{A})), \\
\mathsf{combine}(\mathcal{P}^+_{\exists}(\mathsf{A}), \mathcal{P}^-_{\mho}(\mathsf{A})), \mathsf{combine}(\mathcal{P}^+_{\exists}(\mathsf{A}), \mathcal{P}^-_{\exists}(\mathsf{A})), \mathsf{combine}(\mathcal{P}^+_{\exists}(\mathsf{A}), \mathcal{P}^-_{\exists,-}(\mathsf{A})), \mathsf{combine}(\mathcal{P}^+_{\exists}(\mathsf{A}), \mathcal{P}^-_{\forall}(\mathsf{A})), \\
\mathsf{combine}(\mathcal{P}^+_{\exists,-}(\mathsf{A}), \mathcal{P}^-_{\mho}(\mathsf{A})), \mathsf{combine}(\mathcal{P}^+_{\exists,-}(\mathsf{A}), \mathcal{P}^-_{\exists}(\mathsf{A})), \mathsf{combine}(\mathcal{P}^+_{\exists,-}(\mathsf{A}), \mathcal{P}^-_{\exists,-}(\mathsf{A})), \mathsf{combine}(\mathcal{P}^+_{\exists,-}(\mathsf{A}), \mathcal{P}^-_{\forall}(\mathsf{A})), \\
\mathsf{combine}(\mathcal{P}^+_{\forall}(\mathsf{A}), \mathcal{P}^-_{\mho}(\mathsf{A})), \mathsf{combine}(\mathcal{P}^+_{\forall}(\mathsf{A}), \mathcal{P}^-_{\exists}(\mathsf{A})), \mathsf{combine}(\mathcal{P}^+_{\forall}(\mathsf{A}), \mathcal{P}^-_{\exists,-}(\mathsf{A})), \mathsf{combine}(\mathcal{P}^+_{\forall}(\mathsf{A}), \mathcal{P}^-_{\forall}(\mathsf{A}))
\end{array}$$

Notation in the combination rule ($1 \le h \le l$, $1 \le i \le m$, $1 \le j \le n$, $1 \le k \le o$, $1 \le h' \le l'$, $1 \le i' \le m'$, $1 \le j' \le n'$, $1 \le k' \le o'$):
$B_h, C_i, D_j, \phi_k, E_{h'}, F_{i'}, G_{j'}$ and $\psi_{k'}$ are concepts not containing A; $r_i, s_j, t_k, u_{i'}, v_{j'}$ and $w_{k'}$ are any role names.

1: $\mathsf{combine}(\mathcal{P}^+_{\mho}(\mathsf{A}), \mathcal{P}^-_{\mho}(\mathsf{A})) = \bigcup_{1\le h\le l} \bigcup_{1\le h'\le l'} \{B_h \sqcup E_{h'}\}$ **2:** $\mathsf{combine}(\mathcal{P}^+_{\mho}(\mathsf{A}), \mathcal{P}^-_{\exists}(\mathsf{A})) = \bigcup_{1\le h\le l} \bigcup_{1\le i'\le m'} \{F_{i'} \sqcup \exists u_{i'}.B_h\}$

3: $\mathsf{combine}(\mathcal{P}^+_{\mho}(\mathsf{A}), \mathcal{P}^-_{\exists,-}(\mathsf{A})) = \bigcup_{1\le h\le l} \bigcup_{1\le j'\le n'} \{G_{j'} \sqcup \exists v_{j'}^-.B_h\}$ **4:** $\mathsf{combine}(\mathcal{P}^+_{\mho}(\mathsf{A}), \mathcal{P}^-_{\forall}(\mathsf{A})) = \bigcup_{1\le h\le l} \bigcup_{1\le k'\le o'} \{\psi_{k'} \sqcup \forall w_{k'}.B_h\}$

5: $\mathsf{combine}(\mathcal{P}^+_{\exists}(\mathsf{A}), \mathcal{P}^-_{\mho}(\mathsf{A})) = \bigcup_{1\le i\le m} \bigcup_{1\le h'\le l'} \{C_i \sqcup \exists r_i.E_{h'}\}$ **6:** $\mathsf{combine}(\mathcal{P}^+_{\exists}(\mathsf{A}), \mathcal{P}^-_{\exists}(\mathsf{A})) = \bigcup_{1\le i\le m} \bigcup_{1\le i'\le m'} \{C_i \sqcup \exists r_i.\top, F_{i'} \sqcup \exists u_{i'}.\top\}$

7: $\mathsf{combine}(\mathcal{P}^+_{\exists}(\mathsf{A}), \mathcal{P}^-_{\exists,-}(\mathsf{A})) = \bigcup_{1\le i\le m} \bigcup_{1\le h'\le l'} \{C_i \sqcup \exists r_i.\top, G_{j'} \sqcup \exists v_{j'}^-.\top\}$

8: $\mathsf{combine}(\mathcal{P}^+_{\exists}(\mathsf{A}), \mathcal{P}^-_{\forall}(\mathsf{A})) = \bigcup_{1\le i\le m} \bigcup_{1\le k'\le o'} \{C_i \sqcup \exists r_i.\forall w_{k'}^-.\psi_{k'}\}$

9: $\mathsf{combine}(\mathcal{P}^+_{\exists,-}(\mathsf{A}), \mathcal{P}^-_{\mho}(\mathsf{A})) = \bigcup_{1\le j\le n} \bigcup_{1\le h'\le l'} \{D_j \sqcup \exists s_j^-.E_{h'}\}$ **10:** $\mathsf{combine}(\mathcal{P}^+_{\exists,-}(\mathsf{A}), \mathcal{P}^-_{\exists}(\mathsf{A})) = \bigcup_{1\le j\le n} \bigcup_{1\le i'\le m'} \{D_j \sqcup \exists s_j^-.\top, F_{i'} \sqcup \exists u_{i'}.\top\}$

11: $\mathsf{combine}(\mathcal{P}^+_{\exists,-}(\mathsf{A}), \mathcal{P}^-_{\exists,-}(\mathsf{A})) = \bigcup_{1\le j\le n} \bigcup_{1\le j'\le n'} \{D_j \sqcup \exists s_j^-.\top, G_{j'} \sqcup \exists v_{j'}^-.\top\}$ **12:** $\mathsf{combine}(\mathcal{P}^+_{\exists,-}(\mathsf{A}), \mathcal{P}^-_{\forall}(\mathsf{A})) = \bigcup_{1\le j\le n} \bigcup_{1\le k'\le o'} \{D_j \sqcup \exists s_j^-.\forall w_{k'}^-.\psi_{k'}\}$

13: $\mathsf{combine}(\mathcal{P}^+_{\forall}(\mathsf{A}), \mathcal{P}^-_{\mho}(\mathsf{A})) = \bigcup_{1\le k\le o} \bigcup_{1\le h'\le l'} \{\phi_k \sqcup \forall t_k.E_{h'}\}$

14: $\mathsf{combine}(\mathcal{P}^+_{\forall}(\mathsf{A}), \mathcal{P}^-_{\exists}(\mathsf{A})) = \bigcup_{1\le k\le o} \bigcup_{1\le i'\le m'} \{F_{i'} \sqcup \exists u_{i'}.\forall t_k^-.\phi_k\}$

15: $\mathsf{combine}(\mathcal{P}^+_{\forall}(\mathsf{A}), \mathcal{P}^-_{\exists,-}(\mathsf{A})) = \bigcup_{1\le k\le o} \bigcup_{1\le j'\le n'} \{G_{j'} \sqcup \exists v_{j'}^-.\forall t_k^-.\phi_k\}$

16: $\mathsf{combine}(\mathcal{P}^+_{\forall}(\mathsf{A}), \mathcal{P}^-_{\forall}(\mathsf{A})) = \bigcup_{1\le k\le o} \bigcup_{1\le k'\le o'} \{\phi_k \sqcup \psi_{k'} \sqcup \forall t_k.\bot\}$, in case of $t_k = w_{o'}$

Figure 1: The combination rule for eliminating a concept name $\mathsf{A} \in \mathsf{sig}_{\mathsf{C}}(\mathcal{O})$ from an A-reduced $\mathcal{ALCI}$-KB

$$\begin{array}{c}
\mathcal{O}^{-\mathsf{r}}, \overbrace{C_1 \sqcup \exists \mathsf{r}.D_1, \ldots, C_m \sqcup \exists \mathsf{r}.D_m}^{\mathcal{P}^+_{\exists}(\mathsf{r})}, \overbrace{E_1 \sqcup \exists \mathsf{r}^-.F_1, \ldots, E_n \sqcup \exists \mathsf{r}^-.F_n}^{\mathcal{P}^+_{\exists,-}(\mathsf{r})}, \overbrace{V_1 \sqcup \forall \mathsf{r}.W_1, \ldots, V_p \sqcup \forall \mathsf{r}.W_p}^{\mathcal{P}^-_{\forall}(\mathsf{r})}, \overbrace{X_1 \sqcup \forall \mathsf{r}^-.Y_1, \ldots, X_q \sqcup \forall \mathsf{r}^-.Y_q}^{\mathcal{P}^-_{\forall,-}(\mathsf{r}^-)} \\
\hline
\mathcal{O}^{-\mathsf{r}}, \mathsf{combine}(\mathcal{P}^+_{\exists}(\mathsf{r}), \mathcal{P}^-_{\forall}(\mathsf{r})), \mathsf{combine}(\mathcal{P}^+_{\exists}(\mathsf{r}), \mathcal{P}^-_{\forall,-}(\mathsf{r})), \mathsf{combine}(\mathcal{P}^+_{\exists,-}(\mathsf{r}), \mathcal{P}^-_{\forall}(\mathsf{r})), \mathsf{combine}(\mathcal{P}^+_{\exists,-}(\mathsf{r}), \mathcal{P}^-_{\forall,-}(\mathsf{r}))
\end{array}$$

1: $\mathsf{combine}(\mathcal{P}^+_{\exists}(\mathsf{r}), \mathcal{P}^-_{\forall}(\mathsf{r})) = \bigcup_{1\le i\le m} \bigcup_{1\le x\le p} \{C_i \sqcup V_x\}$, for any D_i, W_x s.t. $D_i \sqcap W_x \sqsubseteq \bot$

2: $\mathsf{combine}(\mathcal{P}^+_{\exists}(\mathsf{r}), \mathcal{P}^-_{\forall,-}(\mathsf{r})) = \bigcup_{1\le i\le m} \bigcup_{1\le y\le q} \{C_i \sqcup Y_y\}$, for any D_i, X_y s.t. $D_i \sqcap X_y \sqsubseteq \bot$

3: $\mathsf{combine}(\mathcal{P}^+_{\exists,-}(\mathsf{r}), \mathcal{P}^-_{\forall}(\mathsf{r})) = \bigcup_{1\le j\le n} \bigcup_{1\le x\le p} \{E_j \sqcup W_x\}$, for any F_j, V_x s.t. $F_j \sqcap V_x \sqsubseteq \bot$

4: $\mathsf{combine}(\mathcal{P}^+_{\exists,-}(\mathsf{r}), \mathcal{P}^-_{\forall,-}(\mathsf{r})) = \bigcup_{1\le j\le n} \bigcup_{1\le y\le q} \{E_j \sqcup X_y\}$, for any F_j, Y_y s.t. $F_j \sqcap Y_y \sqsubseteq \bot$

Figure 2: The combination rule for eliminating a role name $\mathsf{r} \in \mathsf{sig}_{\mathsf{R}}(\mathcal{O})$ from an r-reduced $\mathcal{ALCI}$-KB

domains, and **NCBO BioPortal**[6] featuring biomedical KBs widely used in knowledge-intensive IR applications [56]. From Oxford ISG, we identified 488 KBs containing at most 10,000 CIs. To focus on non-trivial cases, we filtered out KBs lacking role restrictions or inverse roles, leaving 177 KBs. These were reduced to their $\mathcal{ALCI}$-fragments by removing CIs not expressible within $\mathcal{ALCI}$. The same strategy applied to BioPortal yielded 76 suitable KBs. To gain granular insights into retrieval performance, we partitioned these into three categories based on size: PART I (small: <1000 CIs), PART II (medium: 1000–5000 CIs), and PART III (large: 5000–10000 CIs). Detailed statistics of these partitions are provided in the extended version of this paper.

The composition of the forgetting vocabulary $\mathcal{F}$ varies according to retrieval requirements. To address this variability, we designed evaluations to eliminate 10%, 30%, and 50% of symbols in the vocabulary of each KB, using a shuffling algorithm for randomized selection. Experiments were conducted on a laptop with an Intel Core i7-9750H processor (6 cores, up to 2.70 GHz) and 12 GB DDR4-1600 MHz RAM. A retrieval attempt was considered *successful* if all symbols in $\mathcal{F}$ were eliminated, no definers remained in the output, completion occurred within 300 seconds, and operation stayed within 9GB memory. We repeated experiments 100 times per test case and report averaged results.

6.2 RQ1: Retrieval Success Rate

A fundamental requirement for any practical retrieval system is reliability. Our method achieved **100% success rate** across all evaluation configurations, significantly outperforming the state-of-the-art. On Oxford-ISG, LETHE achieved average success rates of only 84.74%, 74.34%, and 69.79% when eliminating 10%, 30%, and 50% of vocabulary symbols, respectively. On BioPortal, LETHE's success rates were 83.48%, 73.11%, and 69.04% for corresponding percentages. The primary failure modes for LETHE were timeouts and memory overflows, underscoring the critical importance of computational efficiency for practical zero-noise retrieval. For IR applications requiring reliable knowledge extraction [10], such high failure rates would be unacceptable—a retrieval system that fails 30–40% of the time on larger KBs cannot be deployed in production environments.

In terms of efficiency, our method completes retrieval in approximately **0.2–1.5 seconds** depending on KB size and elimination ratio, compared to 4–81 seconds for LETHE—achieving speedups of **up to 50×**. Memory consumption is similarly improved: our method uses **20–40% less memory** than LETHE across all configurations, with typical usage ranging from 21–130 MB compared to 37–190 MB for LETHE. These efficiency gains make zero-noise retrieval viable for latency-sensitive IR applications where sub-second response times are expected.

6.3 RQ2: Module Compactness and Noise

A key hypothesis of this work is that zero-noise retrieval produces more compact modules than syntax-based alternatives. Figure 3 presents a comprehensive comparison of module sizes across all methods. Remarkably, our results *contradict* the theoretical triple-exponential growth projection [45, 46] that has long deterred practical adoption of forgetting-based methods. Instead, our zero-noise modules exhibit remarkable compactness, *surpassing even modularization techniques that produce syntactic subsets* of the input KB. This finding has profound implications for knowledge-intensive IR: syntax-based methods like STAR inevitably include axioms containing symbols outside the target vocabulary to preserve semantic integrity, bloating the retrieved context with irrelevant information. Our approach eliminates this overhead entirely.

The dramatic reduction in definer introduction provides further evidence of noise elimination effectiveness. For Oxford-ISG, when eliminating 10%, 30%, and 50% of Σ-vocabulary symbols, LETHE required definers in 67.3%, 65.7%, and 62.1% of tasks respectively, while our method reduced these requirements to just 24.2%, 23.1%, and 14.7%. On BioPortal, LETHE introduced definers in 28.2% of tasks across all settings, while our method reduced this to 9.7%, 4.2%, and 2.9% respectively. Beyond frequency, the quantity of definers per task is even more significant: LETHE introduced **471×** more definers than our method on Oxford-ISG and **305×** more on BioPortal on average. This substantial reduction—stemming directly from our linear-time normalization strategy—explains our method's superior performance across all metrics.

6.4 RQ3: Retrieval Efficiency

Efficiency is paramount for deploying zero-noise retrieval in real-world IR systems. We evaluate both time and memory consumption to understand the practical viability of our approach.

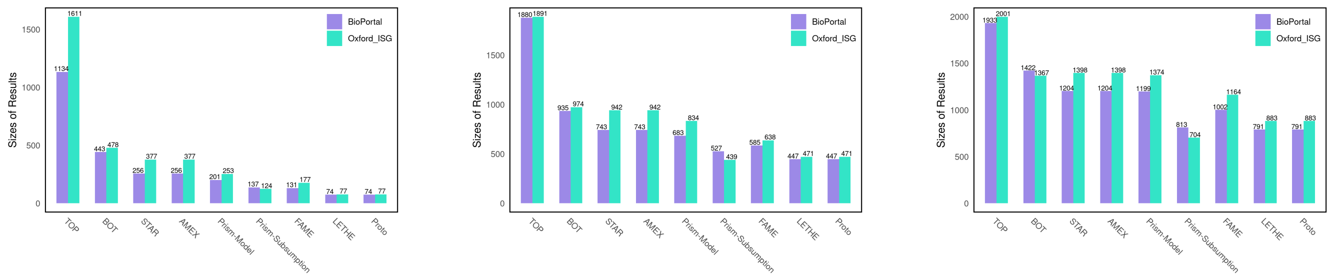
Figure 4 demonstrates that our method significantly outperforms LETHE in runtime and *matches the fastest syntax-based modularization methods*, despite the inherently higher computational complexity of semantic approaches [6, 63]. This result overturns the conventional wisdom that zero-noise retrieval demands substantial computational trade-offs. Figure 5 shows that while semantic retrieval generally demands higher memory usage than syntax-based modularization due to entailment computation, our method achieves substantial memory reduction compared to existing forgetting methods, stemming directly from our linear-time normalization strategy that avoids exponential definer proliferation.

6.5 RQ4: Impact on Downstream RAG Tasks

While RQ1–RQ3 established the algorithmic merits of our method, a critical question remains for the IR community: *Does zero-noise retrieval translate to tangible benefits in downstream applications?* To answer this, we integrated our retrieval module into an RAG pipeline for biomedical question answering.

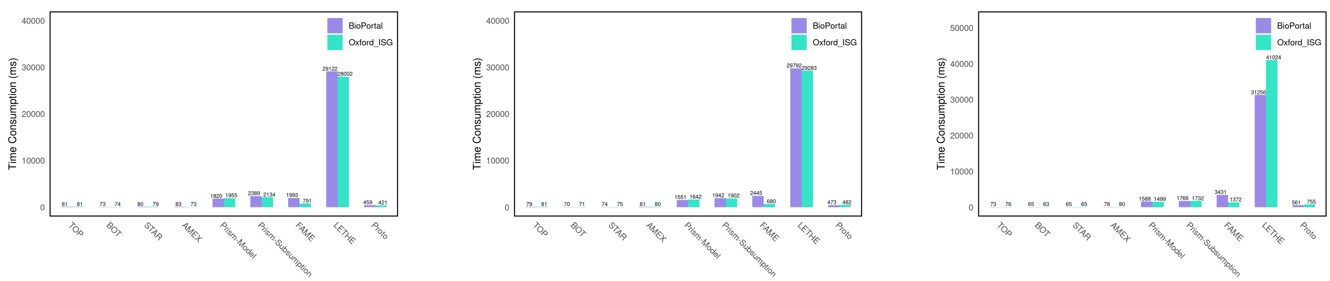
6.5.1 ***Experimental Setup***. Because generic QA datasets do not specifically target the deep semantic structures of ontological KBs, we constructed **Bio-RAG-Bench**, a synthetic benchmark designed to isolate retrieval noise as the primary variable. We sampled 500 complex concepts from the PART II and PART III partitions of BioPortal. For each concept, GPT-4 generated multi-hop reasoning questions grounded in the ontology hierarchy. Ground truth answers were deterministically derived from the ontology axioms. While synthetic, this benchmark isolates retrieval noise as the sole

[6] https://zenodo.org/records/439510



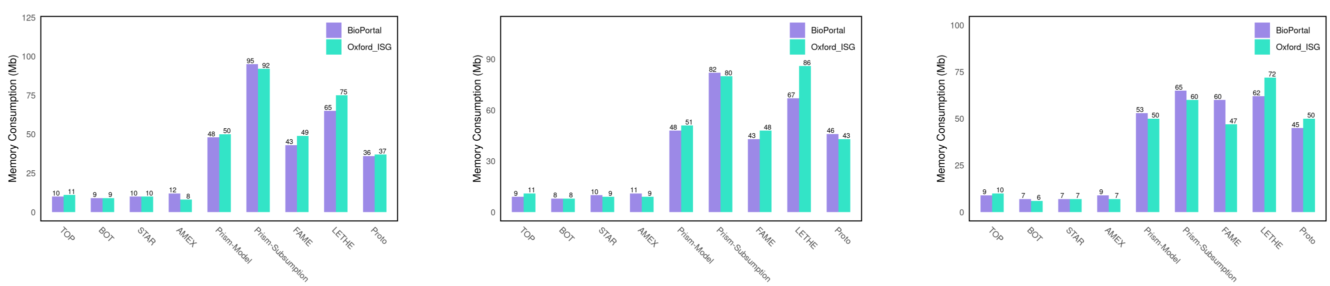
(a) Eliminating 10% of symbols **(b) Eliminating 30% of symbols** **(c) Eliminating 50% of symbols**

Figure 3: Average size of retrieved modules (|KB|) across different methods. Our method (Proto) consistently produces the most compact modules, even compared to syntax-based modularization techniques.



(a) Eliminating 10% of symbols **(b) Eliminating 30% of symbols** **(c) Eliminating 50% of symbols**

Figure 4: Average retrieval time across different methods. Our method achieves efficiency competitive with syntax-based modularization while providing zero-noise guarantees.



(a) Eliminating 10% of symbols **(b) Eliminating 30% of symbols** **(c) Eliminating 50% of symbols**

Figure 5: Average memory consumption across different methods. Our method achieves 20–40% reduction compared to existing forgetting methods.

independent variable, enabling controlled comparison free from confounding factors present in open-domain datasets.

We compared four retrieval configurations: **No Context** (zero-shot LLM), **BM25** (top-50 axioms via lexical matching), **STAR Module** (locality-based syntax extraction), and **Ours** (zero-noise module). We employed **Llama-3-8B-Instruct** as the generator with temperature 0.0, simulating resource-constrained RAG environments where context efficiency is paramount. Performance was measured across 5 independent runs, reporting context tokens, exact-match accuracy, and hallucination rate (evaluated via GPT-4-Turbo with Cohen's $\kappa = 0.81$ human alignment).

6.5.2 ***Quantitative Results.*** Table 1 presents the comparative results. In particular, our method achieves the highest accuracy

Table 1: RAG Performance on Bio-RAG-Bench (Mean ± SD over 5 runs). Statistical significance ($p < 0.005$) against STAR marked with †.

Method	Tokens	Cost Red.	Accuracy	Halluc.
No Context	–	–	42.6 ± 1.2	38.4 ± 2.1
BM25 (Top-50)	1,850 ± 120	–	61.2 ± 1.8	18.7 ± 1.5
STAR Module	12,450 ± 950	Ref.	76.8 ± 1.4	11.5 ± 0.9
Ours (Proto)	**3,120 ± 210**	**↓74.9%**	**84.3 ± 1.1†**	**3.2 ± 0.5†**

(**84.3%**), outperforming STAR by **7.5 percentage points** with statistical significance ($p < 0.005$). This empirically validates the finding by Cuconasu et al. [10] that "distracting" documents degrade RAG performance—the STAR module's irrelevant axioms dilute the LLM's attention. Our method reduces hallucination rate to just **3.2%** (vs. 11.5% for STAR), a **72% relative reduction**, by constraining the generation space to logically relevant facts. Context size is reduced by approximately **75%** (3,120 vs. 12,450 tokens), translating to proportional reductions in operational costs and latency.

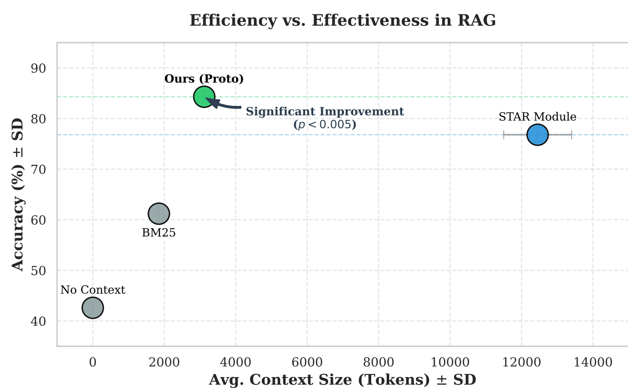


Figure 6: Trade-off between context size and reasoning accuracy. Our method occupies the optimal region (high accuracy, low tokens).

Figure 6 visualizes the accuracy-efficiency trade-off, clearly showing that our method uniquely occupies the optimal region of high accuracy and low token consumption.

6.5.3 *Qualitative Case Study*.

▹ Example (Semantic vs. Syntactic Context in RAG)

Query: "Is *Endocarditis* a type of *Heart Disease*?"
Target Vocabulary: Σ = {Endocarditis, Heart Disease}

1. STAR Module Context (Fragment):
`Endocarditis SubClassOf Inflammation.`
`Inflammation SubClassOf Pathology.`
`Pathology SubClassOf...` (continues for 200 axioms)
`Heart Disease SubClassOf Disease.`
`Myocarditis SubClassOf Heart Disease.`
... (Includes 50+ siblings of Endocarditis)

2. Ours (Zero-Noise) Context:
`Endocarditis SubClassOf Heart Disease.`

LLM Response (with STAR): "Based on the context, Endocarditis is an Inflammation. While it affects the heart, the ontology lists it primarily under Pathology. It is not explicitly stated as a subclass of Heart Disease in the immediate hierarchy provided." (**✗Incorrect**)

LLM Response (with Ours): "Yes, according to the provided ontology, Endocarditis is a SubClassOf Heart Disease." (**✓Correct**)

Analysis: The STAR module preserved the original granular hierarchy, forcing the LLM to perform multi-hop reasoning over hundreds of tokens. Our forgetting-based approach computed the direct entailment during the retrieval phase, presenting the LLM with a noise-free, explicit fact.

6.5.4 *Limitations*.

We acknowledge that while Bio-RAG-Bench has effectively isolated noise effects, real-world queries may exhibit greater linguistic diversity. Additionally, our efficiency gains are most pronounced for expressive KBs ($\mathcal{ALCI}$); for simpler taxonomies, the gap may narrow.

6.6 Summary of Findings

Our evaluation yields four key findings: (1) **Reliability**: 100% succ. rate vs. 60–90% for LETHE and FAME; (2) **Compactness**: zero-noise modules more compact than syntax-based subsets; (3) **Efficiency**: up to 50× faster with 20–40% less memory; (4) **Downstream Impact**: +7.5% accuracy, -72% hallucinations, -75% context cost in RAG. Detailed per-case statistics are available in our repository.[7]

7 Conclusion and Future Work

This paper challenges the long-standing assumption that precise, vocabulary-restricted knowledge retrieval from ontological KBs is computationally impractical for real-world IR applications. We introduced a highly efficient forgetting-based method for computing zero-noise modules from $\mathcal{ALCI}$ KBs, built upon a novel linear-time normalization strategy that fundamentally avoids the exponential definer proliferation plaguing prior approaches. Our comprehensive empirical evaluation on large-scale benchmarks demonstrates that this method achieves 100% retrieval success rate, produces modules more compact than syntax-based alternatives, and operates up to 50× faster than state-of-the-art systems. Crucially, downstream RAG experiments validate the practical impact: zero-noise retrieval improves reasoning accuracy by 7.5%, reduces hallucinations by 72%, and cuts context costs by 75%.

Our work provides the IR community with a robust and scalable primitive for noise-free, semantically complete knowledge access. Future work includes developing incremental retrieval techniques for dynamically evolving KBs, investigating privacy-preserving knowledge sharing through controlled information hiding, and exploring tighter integration of zero-noise modules with neural retrieval pipelines for hybrid knowledge-intensive applications.

[7] https://github.com/anonymous-ai-researcher/module

References

[1] Mohammad Al-Rubaie and J. Morris Chang. 2019. Privacy-Preserving Machine Learning: Threats and Solutions. *IEEE Secur. Priv.* 17, 2 (2019), 49–58.

[2] Grigoris Antoniou and Frank van Harmelen. 2004. *Web Ontology Language: OWL*. Springer Berlin Heidelberg, 67–92.

[3] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *Proc. ICLR'24*. OpenReview.net.

[4] Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. 2017. *An Introduction to Description Logic*. Cambridge University Press.

[5] Bernd Bohnet, Vinh Q. Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, Kai Hui, Tom Kwiatkowski, Ji Ma, Jianmo Ni, Tal Schuster, William W. Cohen, Michael Collins, Dipanjan Das, Donald Metzler, Slav Petrov, and Kellie Webster. 2022. Attributed Question Answering: Evaluation and Modeling for Attributed Large Language Models. *CoRR* abs/2212.08037 (2022).

[6] Elena Botoeva, Boris Konev, Carsten Lutz, Vladislav Ryzhikov, Frank Wolter, and Michael Zakharyaschev. 2016. Inseparability and Conservative Extensions of Description Logic Ontologies: A Survey. In *Proc. RW'16 (LNCS, Vol. 9885)*. Springer, 27–89.

[7] Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine Unlearning. In *Proc. SP'21*. IEEE, 141–159.

[8] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. 1999. Reasoning in Expressive Description Logics with Fixpoints based on Automata on Infinite Trees. In *Proc. IJCAI'99*. Morgan Kaufmann, 84–89.

[9] William Craig. 1957. Three Uses of the Herbrand-Gentzen Theorem in Relating Model Theory and Proof Theory. *J. Symb. Log.* 22, 3 (1957), 269–285.

[10] Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The Power of Noise: Redefining Retrieval for RAG Systems. In *Proc. SIGIR'24*. ACM, 719–729.

[11] Ziqiang Cui, Yunpeng Weng, Xing Tang, Fuyuan Lyu, Dugang Liu, Xiuqiang He, and Chen Ma. 2025. Comprehending Knowledge Graphs with Large Language Models for Recommender Systems. In *Proc. SIGIR'25*. ACM, 1229–1239.

[12] Giovanna D'Agostino and Marco Hollenberg. 2000. Logical Questions Concerning The μ-Calculus: Interpolation, Lyndon and Los-Tarski. *J. Symb. Log.* 65, 1 (2000), 310–332.

[13] Huawen Feng, Zekun Yao, Junhao Zheng, and Qianli Ma. 2025. Training Large Language Models for Retrieval-Augmented Question Answering through Backtracking Correction. In *Proc. ICLR'25*. OpenReview.net.

[14] Yi Feng, Chuanyi Li, and Vincent Ng. 2024. Legal Case Retrieval: A Survey of the State of the Art. In *Proc. ACL'24*. Association for Computational Linguistics, 6472–6485.

[15] Dov M. Gabbay, Renate A. Schmidt, and Andrzej Szalas. 2008. *Second-Order Quantifier Elimination - Foundations, Computational Aspects and Applications*. Studies in logic : Mathematical logic and foundations, Vol. 12. College Publications.

[16] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling Large Language Models to Generate Text with Citations. In *Proc. EMNLP'23*. Association for Computational Linguistics, 6465–6488.

[17] William Gatens, Boris Konev, and Frank Wolter. 2014. Lower and Upper Approximations for Depleting Modules of Description Logic Ontologies. In *Proc. ECAI'14 (Frontiers in Artificial Intelligence and Applications, Vol. 263)*. IOS Press, 345–350.

[18] Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang. 2014. HermiT: An OWL 2 Reasoner. *J. Autom. Reasoning* 53, 3 (2014), 245–269.

[19] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. 2007. Just the Right Amount: Extracting Modules from Ontologies. In *Proc. WWW'07*. ACM, 717–726.

[20] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. 2008. Modular Reuse of Ontologies: Theory and Practice. *J. Artif. Intell. Res.* 31 (2008), 273–318.

[21] B. C. Grau and B. Motik. 2012. Reasoning over Ontologies with Hidden Content: The Import-by-Query Approach. *J. Artif. Intell. Res.* 45 (2012), 197–255.

[22] Bernardo Cuenca Grau, Bijan Parsia, Evren Sirin, and Aditya Kalyanpur. 2006. Modularity and Web Ontologies. In *Proc. KR'06*. AAAI Press, 198–209.

[23] Thomas R. Gruber. 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition* 5, 2 (1993), 199–220.

[24] Tao Guo, Ruida Zhou, and Chao Tian. 2020. On the Information Leakage in Private Information Retrieval Systems. *IEEE Trans. Inf. Forensics Secur.* 15 (2020), 2999–3012.

[25] Andreas Herzig and Jérôme Mengin. 2008. Uniform Interpolation by Resolution in Modal Logic. In *Proc. JELIA'08 (LNCS, Vol. 5293)*. Springer, 219–231.

[26] Sebastian Hofstätter, Jiecao Chen, Karthik Raman, and Hamed Zamani. 2023. FiD-Light: Efficient and Effective Retrieval-Augmented Text Generation. In *Proc. SIGIR'23*. ACM, 1437–1447.

[27] Ian Horrocks, Ulrike Sattler, and Stephan Tobies. 2000. Reasoning with Individuals for the Description Logic SHIQ. In *Proc. CADE'17 (LNCS, Vol. 1831)*, David A. McAllester (Ed.). Springer, 482–496.

[28] Pan Hu, Boris Motik, and Ian Horrocks. 2022. Modular materialisation of Datalog programs. *Artif. Intell.* 308 (2022), 103726.

[29] Swanand Kadhe, Brenden Garcia, Anoosheh Heidarzadeh, Salim Y. El Rouayheb, and Alex Sprintson. 2020. Private Information Retrieval With Side Information. *IEEE Trans. Inf. Theory* 66, 4 (2020), 2032–2043.

[30] Boris Konev, Carsten Lutz, Dirk Walther, and Frank Wolter. 2008. Semantic Modularity and Module Extraction in Description Logics. In *Proc. ECAI'08 (Frontiers in Artificial Intelligence and Applications, Vol. 178)*. IOS Press, 55–59.

[31] Boris Konev, Carsten Lutz, Dirk Walther, and Frank Wolter. 2013. Model-theoretic inseparability and modularity of description logic ontologies. *Artif. Intell.* 203 (2013), 66–103.

[32] Boris Konev, Dirk Walther, and Frank Wolter. 2009. Forgetting and Uniform Interpolation in Large-Scale Description Logic Terminologies. In *Proc. IJCAI'09*. IJCAI/AAAI Press, 830–835.

[33] Roman Kontchakov, Frank Wolter, and Michael Zakharyaschev. 2010. Logic-based ontology comparison and module extraction, with an application to DL-Lite. *Artif. Intell.* 174, 15 (2010), 1093–1141.

[34] Patrick Koopmann. 2015. *Practical Uniform Interpolation for Expressive Description Logics*. Ph. D. Dissertation. The University of Manchester, UK.

[35] Patrick Koopmann and Renate A. Schmidt. 2013. Uniform Interpolation of $\mathcal{ALC}$-Ontologies Using Fixpoints. In *Proc. FroCoS'13 (LNCS, Vol. 8152)*. Springer, 87–102.

[36] 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 *Advances in Neural Information Processing Systems 33*.

[37] Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Yueyue Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2023. SAILER: Structure-aware Pre-trained Language Model for Legal Case Retrieval. In *Proc. SIGIR'23*. ACM, 1035–1044.

[38] Xiaoxi Li, Zhicheng Dou, Yujia Zhou, and Fangchao Liu. 2024. CorpusLM: Towards a Unified Language Model on Corpus for Knowledge-Intensive Tasks. In *Proc. SIGIR'24*. ACM, 26–37.

[39] Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yongkang Wu, Zhonghua Li, Ye Qi, and Zhicheng Dou. 2025. RetroLLM: Empowering Large Language Models to Retrieve Fine-grained Evidence within Generation. In *Proc. ACL'25*. Association for Computational Linguistics, 16754–16779.

[40] Fangzhen Lin and Ray Reiter. 1994. Forget It!. In *Proc. AAAI Fall Symposium on Relevance*. AAAI Press, 154–159.

[41] Qika Lin, Jun Liu, Fangzhi Xu, Yudai Pan, Yifan Zhu, Lingling Zhang, and Tianzhe Zhao. 2022. Incorporating Context Graph with Logical Reasoning for Inductive Relation Prediction. In *Proc. SIGIR'22*. ACM, 893–903.

[42] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Trans. Assoc. Comput. Linguistics* 12 (2024), 157–173.

[43] Carsten Lutz, Inanç Seylan, and Frank Wolter. 2012. An Automata-Theoretic Approach to Uniform Interpolation and Approximation in the Description Logic $\mathcal{EL}$. In *Proc. KR'12*. AAAI Press, 286–296.

[44] Carsten Lutz and Frank Wolter. 2010. Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. *J. Symb. Comput.* 45, 2 (2010), 194–228.

[45] Carsten Lutz and Frank Wolter. 2011. Foundations for Uniform Interpolation and Forgetting in Expressive Description Logics. In *Proc. IJCAI'11*. IJCAI/AAAI Press, 989–995.

[46] Nadeschda Nikitina and Sebastian Rudolph. 2014. (Non-)Succinctness of uniform interpolants of general terminologies in the description logic $\mathcal{EL}$. *Artif. Intell.* 215 (2014), 120–140.

[47] Riku Nortje, Katarina Britz, and Thomas Meyer. 2013. Reachability Modules for the Description Logic $\mathcal{SRIQ}$. In *Proc. LPAR'19 (LNCS, Vol. 8312)*. Springer, 636–652.

[48] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE Trans. Knowl. Data Eng.* 36, 7 (2024), 3580–3599.

[49] Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, Ian Soboroff, Hoa Trang Dang, and Jimmy Lin. 2025. The Great Nugget Recall: Automating Fact Extraction and RAG Evaluation with Large Language Models. In *Proc. SIGIR'25*. ACM, 180–190.

[50] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering. In *Proc. NAACL-HLT'21*. Association for Computational Linguistics, 5835–5847.

[51] Ana Armas Romero. 2015. *Ontology Module Extraction and Applications to Ontology Classification*. Ph. D. Dissertation. University of Oxford, UK.

[52] Ana Armas Romero, Mark Kaminski, Bernardo Cuenca Grau, and Ian Horrocks. 2016. Module Extraction in Expressive Ontology Languages via Datalog Reasoning. *J. Artif. Intell. Res.* 55 (2016), 499–564.

[53] Alireza Salemi and Hamed Zamani. 2024. Towards a Search Engine for Machines: Unified Ranking for Multiple Retrieval-Augmented Large Language Models. In

Proc. SIGIR'24. ACM, 741–751.

[54] Yucheng Shi, Qiaoyu Tan, Xuansheng Wu, Shaochen Zhong, Kaixiong Zhou, and Ninghao Liu. 2024. Retrieval-enhanced Knowledge Editing in Language Models for Multi-Hop Question Answering. In *Proc. CIKM'24*. ACM, 2056–2066.

[55] Kent A. Spackman. 2000. SNOMED RT and SNOMED CT. Promise of an international clinical ontology. *M.D. Computing 17* (2000).

[56] Fabian M. Suchanek, Mehwish Alam, Thomas Bonald, Lihu Chen, Pierre-Henri Paris, and Jules Soria. 2024. YAGO 4.5: A Large and Clean Knowledge Base with a Rich Taxonomy. In *Proc. SIGIR'24*. ACM, 131–140.

[57] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. In *Proc. ICLR'24*. OpenReview.net.

[58] Yanran Tang, Ruihong Qiu, Hongzhi Yin, Xue Li, and Zi Huang. 2024. CaseLink: Inductive Graph Learning for Legal Case Retrieval. In *Proc. SIGIR'24*. ACM, 2199–2209.

[59] Changxin Tian, Yuexiang Xie, Yaliang Li, Nan Yang, and Wayne Xin Zhao. 2022. Learning to Denoise Unreliable Interactions for Graph Collaborative Filtering. In *Proc. SIGIR'22*. ACM, 122–132.

[60] Stephan Tobies. 2001. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. Ph.D. Dissertation. RWTH Aachen University, Germany.

[61] Antonios Tragoudaras, Theofanis Aslanidis, Emmanouil Georgios Lionis, Marina Orozco González, and Panagiotis Eustratiadis. 2025. Information Leakage of Sentence Embeddings via Generative Embedding Inversion Attacks. In *Proc. SIGIR'25*. ACM, 3234–3243.

[62] Yiteng Tu, Weihang Su, Yujia Zhou, Yiqun Liu, and Qingyao Ai. 2025. Robust Fine-tuning for Retrieval Augmented Generation against Retrieval Defects. In *Proc. SIGIR'25*. ACM, 1272–1282.

[63] Chiara Del Vescovo, Matthew Horridge, Bijan Parsia, Uli Sattler, Thomas Schneider, and Haoruo Zhao. 2020. Modular Structures and Atomic Decomposition in Ontologies. *J. Artif. Intell. Res.* 69 (2020), 963–1021.

[64] Albert Visser. 1996. *Bisimulations, Model Descriptions and Propositional Quantifiers*. Utrecht University.

[65] Kewen Wang, Zhe Wang, Rodney W. Topor, Jeff Z. Pan, and Grigoris Antoniou. 2014. Eliminating Concepts and Roles from Ontologies in Expressive Descriptive Logics. *Computational Intelligence* 30, 2 (2014), 205–232.

[66] Yue Xiang, Xuan Wu, Chang Lu, and Yizheng Zhao. 2022. Creating Signature-Based Views for Description Logic Ontologies with Transitivity and Qualified Number Restrictions. In *Proc. WWW'22*. ACM, 808–817.

[67] Zhentao Xu, Mark Jerome Cruz, Matthew Guevara, Tie Wang, Manasi Deshpande, Xiaofeng Wang, and Zheng Li. 2024. Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering. In *Proc. SIGIR'24*. ACM, 2905–2909.

[68] Zhihao Yang and Yizheng Zhao. 2024. What a Surprise! Computing Rewritten Modules Can Be as Efficient as Computing Subset Modules. In *Proc. CIKM'24*. ACM, 2940–2949.

[69] Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making Retrieval-Augmented Language Models Robust to Irrelevant Context. In *Proc. ICLR'24*. OpenReview.net.

[70] Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022. Subgraph Retrieval Enhanced Model for Multi-hop Knowledge Base Question Answering. In *Proc. ACL'22*. Association for Computational Linguistics, 5773–5784.

[71] Yan Zhang and Yi Zhou. 2010. Forgetting Revisited. In *Proc. KR'10*. AAAI Press.

[72] Yizheng Zhao and Schmidt Renate A. 2017. Role Forgetting for $\mathcal{ALCOQH}(\triangledown)$-Ontologies Using An Ackermann-Based Approach. In *Proc. IJCAI'17*. IJCAI/AAAI Press, 1354–1361.

[73] Yizheng Zhao and Renate A. Schmidt. 2016. Forgetting Concept and Role Symbols in $\mathcal{ALCOIH}\mu^{+}(\triangledown, \sqcap)$-Ontologies. In *Proc. IJCAI'16*. IJCAI/AAAI Press, 1345–1353.

[74] Yizheng Zhao and Renate A. Schmidt. 2018. FAME: An Automated Tool for Semantic Forgetting in Expressive Description Logics. In *Proc. IJCAR'18 (LNCS, Vol. 10900)*. Springer, 19–27.

[75] Yujia Zhou, Jing Yao, Zhicheng Dou, Ledell Wu, and Ji-Rong Wen. 2023. DynamicRetriever: A Pre-trained Model-based IR System Without an Explicit Index. *Mach. Intell. Res.* 20, 2 (2023), 276–288.

[76] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. 2023. Large Language Models for Information Retrieval: A Survey. *CoRR* abs/2308.07107 (2023).

A The Zero-Noise Retrieval Algorithm

Algorithm 1 presents our complete zero-noise retrieval procedure referenced in the main paper.

Algorithm 1 Zero-Noise Knowledge Retrieval

```
Require: An ALCI-KB O, forgetting vocabulary F ⊆ sig(O)
Ensure: An ALCI-KB V with sig(V) ∩ F = ∅
Global: D ← ∅                         ▷ Track introduced definers globally
Global: H ← ∅                         ▷ History of states for cycle detection
function RETRIEVE(O, F)
    V ← SIMPLIFY(O)                   ▷ Apply standard simplifications
    for each symbol X ∈ F do
        O_n ← NORMALIZE(V, X)         ▷ Apply normalization rules
        V ← ELIMINATE(O_n, X)         ▷ Apply combination rules; updates D
    end for
    while D ≠ ∅ do
        Let Z be a definer from D; D ← D \ {Z}
        V_norm ← NORMALIZE(V, Z)
        h ← HASH(V_norm)              ▷ Compute a hash
        if h ∈ H then
            return "Failure: Cycle Detected"   ▷ Blocking step
        end if
        H ← H ∪ {h}
        V ← ELIMINATE(V_norm, Z)
    end while
    return V
end function
```

B Missing Proofs

This section provides complete proofs for all lemmas and theorems presented in the main paper.

B.1 Preliminaries for Proofs

To prove Lemma 1, we first introduce the following notions:

Definition 8 (Conservative Extension). *Given two KBs $\mathcal{O}$ and $\mathcal{O}'$, we say that $\mathcal{O}'$ is a* conservative extension *of $\mathcal{O}$ if*

(1) $sig(\mathcal{O}) \subseteq sig(\mathcal{O}')$,
(2) every model of $\mathcal{O}'$ is a model of $\mathcal{O}$, and
(3) for every model $\mathcal{I}$ of $\mathcal{O}$, there exists a model $\mathcal{I}'$ of $\mathcal{O}'$ such that the interpretations of the concept and role names from $sig(\mathcal{O})$ coincide in $\mathcal{I}$ and $\mathcal{I}'$, i.e.,
- *$A^{\mathcal{I}} = A^{\mathcal{I}'}$ for all concept names $A \in sig(\mathcal{O})$, and*
- *$r^{\mathcal{I}} = r^{\mathcal{I}'}$ for all role names $r \in sig(\mathcal{O})$.*

Conservative extension has two key properties:

- *Transitivity*: If $\mathcal{O}''$ is a conservative extension of $\mathcal{O}'$ and $\mathcal{O}'$ is a conservative extension of $\mathcal{O}$, then $\mathcal{O}''$ is a conservative extension of $\mathcal{O}$.
- *Subsumption Preservation*: If $\mathcal{O}'$ is a conservative extension of $\mathcal{O}$, then for all clauses constructed using only the names from $\text{sig}(\mathcal{O})$, subsumption w.r.t. $\mathcal{O}$ and $\mathcal{O}'$ coincide; see Proposition 1.

Proposition 1. *Let $\mathcal{O}'$ be a conservative extension of an $\mathcal{ALCI}$ KB $\mathcal{O}$, and $L_1 \sqcup \ldots \sqcup L_n$ a clause constructed from $sig(\mathcal{O})$. Then:*

$$\mathcal{O} \models (\top \sqsubseteq)\ L_1 \sqcup \ldots \sqcup L_n \textit{ iff } \mathcal{O}' \models (\top \sqsubseteq)\ L_1 \sqcup \ldots \sqcup L_n.$$

In other words, $\mathcal{O} \equiv_{sig(\mathcal{O})} \mathcal{O}'$.

Proof. We prove both directions by contrapositive.
($\Rightarrow$) Assume $\mathcal{O}' \not\models L_1 \sqcup \ldots \sqcup L_n$. Then there exists a model $\mathcal{I}'$ of $\mathcal{O}'$ such that $\top^{\mathcal{I}'} \not\subseteq (L_1 \sqcup \ldots \sqcup L_n)^{\mathcal{I}'}$. Since $\mathcal{I}'$ is also a model of $\mathcal{O}$ by conservative extension, we have $\mathcal{O} \not\models L_1 \sqcup \ldots \sqcup L_n$.
($\Leftarrow$) Assume $\mathcal{O} \not\models L_1 \sqcup \ldots \sqcup L_n$. Then there exists a model $\mathcal{I}$ of $\mathcal{O}$ such that $\top^{\mathcal{I}} \not\subseteq (L_1 \sqcup \ldots \sqcup L_n)^{\mathcal{I}}$. By conservative extension, there exists a model $\mathcal{I}'$ of $\mathcal{O}'$ where the interpretations of the names from $\text{sig}(\mathcal{O})$ coincide in $\mathcal{I}$ and $\mathcal{I}'$. Since $L_1 \sqcup \ldots \sqcup L_n$ contains only names from $\text{sig}(\mathcal{O})$, we have:

$$\top^{\mathcal{I}} = \top^{\mathcal{I}'} \not\subseteq (L_1 \sqcup \ldots \sqcup L_n)^{\mathcal{I}} = (L_1 \sqcup \ldots \sqcup L_n)^{\mathcal{I}'}.$$

Therefore, $\mathcal{O}' \not\models L_1 \sqcup \ldots \sqcup L_n$. □

B.2 Proof of Lemma 1

Lemma 1. *Let $\mathcal{O}$ be an $\mathcal{ALCI}$ KB and $\mathcal{O}'$ its reduced form obtained via the normalizations given in Section 4. Then $\mathcal{O} \equiv_{sig(\mathcal{O})} \mathcal{O}'$.*

Proof. The proof establishes that each normalization rule yields a conservative extension, and by transitivity of conservative extensions, the complete reduction process preserves logical equivalence over the original vocabulary. We detail this for the first rule; similar proofs apply to the other rules.

Consider the application of the first rule to an A-clause instance $L_1 \sqcup \cdots \sqcup L_n$ where A appears more than once. Suppose there exists a literal L_i ($1 \leq i \leq n$) of the form $\exists R.C$ or $\forall R.C$ with $A \in \text{sig}(C)$. Let $\mathcal{O}'$ be the KB obtained by replacing C with a fresh definer $Z \in \mathsf{N_C}$ and adding the clause $\neg Z \sqcup C$ to the present KB. Thus, $\mathcal{O}'$ contains the modified clause $(L_1 \sqcup \cdots \sqcup L_n)^{C \mapsto Z}$ where C is replaced by Z, together with the defining clause $\neg Z \sqcup C$. The signature condition holds since $\text{sig}(\mathcal{O}') = \text{sig}(\mathcal{O}) \cup \{Z\}$ where Z is fresh, ensuring $\text{sig}(\mathcal{O}) \subseteq \text{sig}(\mathcal{O}')$.

For the semantic direction from $\mathcal{O}'$ to $\mathcal{O}$, we establish that every model of $\mathcal{O}'$ is a model of $\mathcal{O}$. Suppose, toward a contradiction, that there exists a model $\mathcal{I}'$ of $\mathcal{O}'$ such that $\mathcal{I}' \not\models L_1 \sqcup \cdots \sqcup L_n$. Therefore, there exists a domain element $d \in \Delta^{\mathcal{I}'}$ such that $d \notin (L_i)^{\mathcal{I}'}$ for all literals L_i ($1 \leq i \leq n$) in the original clause. Without loss of generality, assume the literal containing the substituted concept is $L_i = \exists R.C$, so the transformed literal is $\exists R.Z$. Since $\mathcal{I}'$ is a model of $\mathcal{O}'$, it satisfies $\top \sqsubseteq (L_1 \sqcup \cdots \sqcup L_n)^{C \mapsto Z}$, which means $d \in (L_1 \sqcup \cdots \sqcup L_n)^{C \mapsto Z^{\mathcal{I}'}}$. Given that d does not satisfy any of the other literals L_j for $j \neq i$, we must have $d \in (\exists R.Z)^{\mathcal{I}'}$. This implies there exists an element $d' \in \Delta^{\mathcal{I}'}$ such that $(d, d') \in R^{\mathcal{I}'}$ and $d' \in Z^{\mathcal{I}'}$.

Since $\mathcal{I}'$ satisfies the defining clause $Z \sqsubseteq C$, we have $Z^{\mathcal{I}'} \subseteq C^{\mathcal{I}'}$, which implies $d' \in C^{\mathcal{I}'}$. Therefore, $(d, d') \in R^{\mathcal{I}'}$ and $d' \in C^{\mathcal{I}'}$, yielding $d \in (\exists R.C)^{\mathcal{I}'}$. This contradicts our assumption that d does not satisfy the original literal $L_i = \exists R.C$. The case where $L_i = \forall R.C$ follows by analogous reasoning. Therefore, the reduct of $\mathcal{I}'$ to $\text{sig}(\mathcal{O})$ must be a model of $\mathcal{O}$.

Conversely, let $\mathcal{I}$ be any model of $\mathcal{O}$, satisfying $\top^{\mathcal{I}} \subseteq (L_1 \sqcup \cdots \sqcup L_n)^{\mathcal{I}}$. We construct an expansion $\mathcal{I}'$ that agrees with $\mathcal{I}$ on all symbols in $\text{sig}(\mathcal{O})$ and interprets the fresh definer as $Z^{\mathcal{I}'} = C^{\mathcal{I}}$.

Since $\top^{\mathcal{I}} \subseteq (L_1 \sqcup \cdots \sqcup L_n)^{\mathcal{I}}$ and Z does not occur in the original clause or in C, we have $\top^{\mathcal{I}'} \subseteq (L_1 \sqcup \cdots \sqcup L_n)^{\mathcal{I}'}$. By construction, $Z^{\mathcal{I}'} = C^{\mathcal{I}} = C^{\mathcal{I}'}$, which ensures satisfaction of the defining axiom $\neg Z \sqcup C$. The substitution property guarantees that $\top^{\mathcal{I}'} \subseteq ((L_1 \sqcup \cdots \sqcup L_n)^{C \mapsto Z})^{\mathcal{I}'}$, establishing that $\mathcal{I}'$ is a model of $\mathcal{O}'$.

This bidirectional correspondence between models restricted to the original vocabulary demonstrates that $\mathcal{O} \equiv_{\mathrm{sig}(\mathcal{O})} \mathcal{O}'$ under a single rule application. Since the complete normalization process involves a finite sequence of such applications, and conservative extension is preserved under composition, the lemma follows. □

B.3 Auxiliary Definitions for Lemma 2

To prove Lemma 2, we first introduce the following notions.

We define the *frequency* $\mathrm{fq}(\mathsf{A}, C)$ of A in an A-concept C inductively as:

- $\mathrm{fq}(\mathsf{A}, X) = \begin{cases} 1, & \text{if } X = \mathsf{A} \\ 0, & \text{if } X \in \mathsf{N_C} \text{ and } X \neq \mathsf{A} \end{cases}$
- $\mathrm{fq}(\mathsf{A}, \neg E) = \mathrm{fq}(\mathsf{A}, E)$
- $\mathrm{fq}(\mathsf{A}, QR.E) = \mathrm{fq}(\mathsf{A}, E)$, for $Q \in \{\exists, \forall\}$
- $\mathrm{fq}(\mathsf{A}, E \star F) = \mathrm{fq}(\mathsf{A}, E) + \mathrm{fq}(\mathsf{A}, F)$, for $\star \in \{\sqcap, \sqcup\}$

We define the *frequency* $\mathrm{fq}(\mathsf{A}, L_1 \sqcup \ldots \sqcup L_n)$ of A in an A-clause $L_1 \sqcup \ldots \sqcup L_n$ as follows, where L_i is a literal ($1 \leq i \leq n$):

- $\mathrm{fq}(\mathsf{A}, L_1 \sqcup \ldots \sqcup L_n) = \sum_{i=1}^{n} \mathrm{fq}(\mathsf{A}, L_i)$

Let A^* be a designated occurrence of A in an A-concept C. We define the *role depth* $\mathrm{dp}(\mathsf{A}^*, C)$ of A^* in C inductively as follows:

- $\mathrm{dp}(\mathsf{A}^*, C) = 0$, if C is of the form $\mathsf{A}^* \star D$ or $\neg\mathsf{A}^* \star D$, where $\star \in \{\sqcap, \sqcup\}$ and D is an arbitrary concept
- $\mathrm{dp}(\mathsf{A}^*, C) = \mathrm{dp}(\mathsf{A}^*, E) + 1$, if C is of the form $D \star QR.E$, where $\star \in \{\sqcap, \sqcup\}$, $Q \in \{\exists, \forall\}$, E is a concept in which A^* occurs, and D is an arbitrary concept

That is, $\mathrm{dp}(\mathsf{A}^*, C)$ counts the number of $\exists$ and $\forall$-restrictions guarding A^* in C. Similarly, $\mathrm{dp}(\mathsf{r}^*, C)$ counts the number of role restrictions guarding a role occurrence r^* in C. The *role depth* $\mathrm{dp}(\mathsf{A}, C)$ of A in C is defined as the sum of the *role depth* of all occurrences of A in C. The *role depth* $\mathrm{dp}(\mathsf{A}, L_1 \sqcup \ldots \sqcup L_n)$ of A in a clause $L_1 \sqcup \ldots \sqcup L_n$ is defined as:

- $\mathrm{dp}(\mathsf{A}, L_1 \sqcup \ldots \sqcup L_n) = \sum_{i=1}^{n} \mathrm{dp}(\mathsf{A}, L_i)$

Proposition 2. *Any A-clause $L_1 \sqcup \ldots \sqcup L_n$ with $dp(A, L_1 \sqcup \ldots \sqcup L_n) = 0$ is in A-reduced form.*

Proof. $\mathrm{dp}(\mathsf{A}, L_1 \sqcup \ldots \sqcup L_n) = 0$ indicates that all occurrences of A are unguarded by quantifiers ($\exists$ or $\forall$) and appear only at the surface level of each L_i. After eliminating redundant occurrences, the clause $L_1 \sqcup \ldots \sqcup L_n$ reduces to either $\mathsf{A} \sqcup C$ or $\neg\mathsf{A} \sqcup C$, where C is a clause with $\mathsf{A} \notin \mathrm{sig}(C)$, making it A-reduced. □

Proposition 3. *Any A-clause $L_1 \sqcup \ldots \sqcup L_n$ with $fq(A, L_1 \sqcup \ldots \sqcup L_n) = 1$ and $dp(A, L_1 \sqcup \ldots \sqcup L_n) = 1$ is in A-reduced form.*

Proof. When $\mathrm{fq}(\mathsf{A}, L_1 \sqcup \ldots \sqcup L_n) = 1$ and $\mathrm{dp}(\mathsf{A}, L_1 \sqcup \ldots \sqcup L_n) = 1$, the single occurrence of A is guarded by exactly one quantifier ($\exists$ or $\forall$). The clause $L_1 \sqcup \ldots \sqcup L_n$ is then in A-reduced form, regardless of which L_i contains A. □

Proposition 4. *For any A-clause $L_1 \sqcup \ldots \sqcup L_n$, $fq(A, L_1 \sqcup \ldots \sqcup L_n) - dp(A, L_1 \sqcup \ldots \sqcup L_n) \leq 1$.*

Proof. Let $m = \mathrm{dp}(\mathsf{A}, L_1 \sqcup \cdots \sqcup L_n)$. By Proposition 2, at most one unguarded occurrence of A is possible, and each role depth allows at most one guarded occurrence, yielding a total bound of $m + 1$ occurrences. □

We define $\mathsf{NLZ}_1(\mathcal{O})$ as the set derived from $\mathcal{O}$ by applying one of the normalization rules to $\mathcal{O}$, indicating one round of normalization for $\mathcal{O}$. Similarly, $\mathsf{NLZ}_k(\mathcal{O})$ is defined for any $k \geq 0$, with $\mathsf{NLZ}_0(\mathcal{O})$ representing the original $\mathcal{O}$. We further define $\mathrm{dp}(\mathsf{A}, \mathcal{O})$ as the sum of $\mathrm{dp}(\mathsf{A}, L_1 \sqcup \cdots \sqcup L_n)$, for every A-clause $L_1 \sqcup \cdots \sqcup L_n$ in $\mathcal{O}$.

Proposition 5. *$dp(A, NLZ_{k-1}(\mathcal{O})) - dp(A, NLZ_k(\mathcal{O})) = 1$, where (i) $k \geq 1$ and (ii) $NLZ_{k-1}(\mathcal{O})$ is not A-reduced.*

Proof. Condition (ii) ensures $\mathsf{NLZ}_{k-1}(\mathcal{O}) \neq \mathsf{NLZ}_k(\mathcal{O})$ and enables the first two normalization rules for finding the A-reduced form. We prove that applying either rule reduces the role depth of A in the resulting $\mathcal{O}$ by 1. We detail the proof for the first rule, as the second follows analogously.

Consider $\mathsf{NLZ}_{k-1}(\mathcal{O})$ containing an A-clause $L_1 \sqcup \cdots \sqcup L_n$ not in A-reduced form. Let:

- $\mathrm{dp}(\mathsf{A}, L_1 \sqcup \cdots \sqcup L_n) = m$, where $m \geq 1$
- $\mathrm{dp}(\mathsf{A}, \mathsf{NLZ}_{k-1}(\mathcal{O})) = m + x$, where $x \geq 0$
- $\mathrm{dp}(\mathsf{A}, C) = n \geq 1$ (by the Rule)

For $Q \in \{\exists, \forall\}$, we have $\mathrm{dp}(\mathsf{A}, QR.C) = n + 1$ and $m = n + 1 + y$ where $y \geq 0$. The normalization replaces C with a new definer Z, removing $L_1 \sqcup \cdots \sqcup L_n$ from $\mathsf{NLZ}_k(\mathcal{O})$ and introducing two new clauses: $(L_1 \sqcup \cdots \sqcup L_n)^C_Z$ (which denotes the clause obtained from $L_1 \sqcup \cdots \sqcup L_n$ by replacing C with Z) and $\neg Z \sqcup C$. Note that:

- $\mathrm{dp}(\mathsf{A}, (L_1 \sqcup \cdots \sqcup L_n)^C_Z) = (n + 1 + y) - (n + 1) = y$
- $\mathrm{dp}(\mathsf{A}, \neg Z \sqcup C) = \mathrm{dp}(\mathsf{A}, C) = n$

Therefore, $\mathrm{dp}(\mathsf{A}, \mathsf{NLZ}_k(\mathcal{O})) = \mathrm{dp}(\mathsf{A}, \mathsf{NLZ}_{k-1}(\mathcal{O})) - \mathrm{dp}(\mathsf{A}, L_1 \sqcup \cdots \sqcup L_n) + y + n = x + y + n$. Since $\mathrm{dp}(\mathsf{A}, \mathsf{NLZ}_{k-1}(\mathcal{O})) = m + x = x + y + n + 1$, we conclude $\mathrm{dp}(\mathsf{A}, \mathsf{NLZ}_{k-1}(\mathcal{O})) - \mathrm{dp}(\mathsf{A}, \mathsf{NLZ}_k(\mathcal{O})) = 1$.. □

B.4 Proof of Lemma 2

Lemma 2. *For any $\mathcal{ALCI}$ KB $\mathcal{O}$, a reduced form $\mathcal{O}'$ (either A-reduced or r-reduced) can be computed in a linear number of normalization steps. Moreover, $|\mathcal{O}'|$ is linear in $|\mathcal{O}|$, where $|\mathcal{O}|$ denotes the size of $\mathcal{O}$ as defined in [4].*

Proof. Each definer replaces a subconcept immediately below an $\exists$-restriction and triggers one new clause. Therefore, both the number of definers and new clauses introduced in $\mathcal{O}$'s normalization are bounded by $\mathrm{O}(n)$, where n is the count of $\exists$-restrictions in $\mathcal{O}$. This ensures termination and completeness. □

B.5 Proof of Lemma 3 (Soundness of Concept Name Elimination)

Lemma 3. *Let $\mathcal{O}$ be an A-reduced $\mathcal{ALCI}$ KB with $A \in sig_C(\mathcal{O})$. If $\mathcal{M}$ is the KB derived by applying the combination rule in Figure 1, then $\mathcal{O} \equiv_{sig(\mathcal{O}) \setminus \{A\}} \mathcal{M}$.*

Proof. Cases 1, 2, 3, 4, 5, 9, and 13 reverse the normalization process, thereby directly preserving inseparability. We now focus on proving Case 8 in detail, as the other cases follow similar proof patterns. We demonstrate that the premises $\mathcal{O}$ of the case constitutes a conservative extension of its conclusion $\mathcal{O}^{-\mathsf{A}}$. First, observe that $\mathrm{sig}(\mathcal{O}) = \mathrm{sig}(\mathcal{O}^{-\mathsf{A}}) \cup \{\mathsf{A}\}$, which establishes $\mathrm{sig}(\mathcal{O}^{-\mathsf{A}}) \subseteq \mathrm{sig}(\mathcal{O})$, satisfying Condition (1) of Definition 8.

Next we prove Condition (2) of Definition 8. Assume for the sake of contradiction that there exists an interpretation $\mathcal{I}$ satisfying $\mathcal{I} \models \top \sqsubseteq C_i \sqcup \exists r_i.A$ (call this **P1**), $\mathcal{I} \models \top \sqsubseteq \psi_{k'} \sqcup \forall w_{k'}.\neg A$ (call this **P2**), but $\mathcal{I} \not\models \top \sqsubseteq C_i \sqcup \exists r_i.\forall w_{k'}^{-}.\psi_{k'}$ (call this **C1**). From **C1**, there exists some $d \in \Delta^{\mathcal{I}}$ such that $d \notin (C_i \sqcup \exists r_i.\forall w_{k'}^{-}.\psi_{k'})^{\mathcal{I}}$, which means $d \notin C_i^{\mathcal{I}}$ (fact 1) and $d \notin (\exists r_i.\forall w_{k'}^{-}.\psi_{k'})^{\mathcal{I}}$ (fact 2). From **P1** and the fact that $d \in \Delta^{\mathcal{I}} = \top^{\mathcal{I}}$, we have $d \in (C_i \sqcup \exists r_i.A)^{\mathcal{I}} = C_i^{\mathcal{I}} \cup (\exists r_i.A)^{\mathcal{I}}$. By fact 1, we must have $d \in (\exists r_i.A)^{\mathcal{I}}$. Therefore, there exists some $e \in \Delta^{\mathcal{I}}$ such that $(d, e) \in r_i^{\mathcal{I}}$ (fact 3) and $e \in A^{\mathcal{I}}$ (fact 4).

From fact 2, we know $d \notin (\exists r_i.\forall w_{k'}^{-}.\psi_{k'})^{\mathcal{I}}$, and therefore $d \in (\forall r_i.\exists w_{k'}^{-}.\neg\psi_{k'})^{\mathcal{I}}$. Since $(d, e) \in r_i^{\mathcal{I}}$ by fact 3, we have $e \in (\exists w_{k'}^{-}.\neg\psi_{k'})^{\mathcal{I}}$. Therefore, there exists some $f \in \Delta^{\mathcal{I}}$ such that $(e, f) \in (w_{k'}^{-})^{\mathcal{I}}$ (fact 5) and $f \in (\neg\psi_{k'})^{\mathcal{I}}$ (fact 6). From fact 5 and the definition of inverse roles, we have $(f, e) \in w_{k'}^{\mathcal{I}}$ (fact 7). From **P2** and the fact that $f \in \Delta^{\mathcal{I}} = \top^{\mathcal{I}}$, we have $f \in (\psi_{k'} \sqcup \forall w_{k'}.\neg A)^{\mathcal{I}} = \psi_{k'}^{\mathcal{I}} \cup (\forall w_{k'}.\neg A)^{\mathcal{I}}$. Since $f \in (\neg\psi_{k'})^{\mathcal{I}}$ by fact 6, we have $f \notin \psi_{k'}^{\mathcal{I}}$. Therefore, we must have $f \in (\forall w_{k'}.\neg A)^{\mathcal{I}}$. Since $(f, e) \in w_{k'}^{\mathcal{I}}$ by fact 7, we have $e \in (\neg A)^{\mathcal{I}}$. But this contradicts fact 4, which states that $e \in A^{\mathcal{I}}$. Therefore, our assumption was false, and the entailment holds.

Finally we prove Condition (3) of Definition 8. Let α denote the axiom $\top \sqsubseteq C_i \sqcup \exists r_i.\forall w_{k'}^{-}.\psi_{k'}$, β_1 denote $\top \sqsubseteq C_i \sqcup \exists r_i.A$, and β_2 denote $\top \sqsubseteq \psi_{k'} \sqcup \forall w_{k'}.\neg A$. We need to prove that for every model $\mathcal{I}$ of α, there exists a model $\mathcal{I}'$ of $\{\beta_1, \beta_2\}$ such that the interpretations of concept and role names from $\mathrm{sig}(\alpha)$ coincide in $\mathcal{I}$ and $\mathcal{I}'$. We construct $\mathcal{I}'$ as follows, noting that $A \notin \mathrm{sig}(\alpha)$ since A does not appear in α: set $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}}$; for all concept names $X \in \mathrm{sig}(\alpha)$ set $X^{\mathcal{I}'} = X^{\mathcal{I}}$; for all role names $r \in \mathrm{sig}(\alpha)$ set $r^{\mathcal{I}'} = r^{\mathcal{I}}$; and set $A^{\mathcal{I}'} = (\forall w_{k'}^{-}.\psi_{k'})^{\mathcal{I}}$.

Verification that $\mathcal{I}'$ satisfies β_1: Since $\mathcal{I} \models \alpha$, we have $\mathcal{I} \models \top \sqsubseteq C_i \sqcup \exists r_i.\forall w_{k'}^{-}.\psi_{k'}$. By our construction, $\mathcal{I}'$ coincides with $\mathcal{I}$ on all names appearing in this CI, and therefore $\mathcal{I}' \models \top \sqsubseteq C_i \sqcup \exists r_i.\forall w_{k'}^{-}.\psi_{k'}$. Since $A^{\mathcal{I}'} = (\forall w_{k'}^{-}.\psi_{k'})^{\mathcal{I}} = (\forall w_{k'}^{-}.\psi_{k'})^{\mathcal{I}'}$, we have $\mathcal{I}' \models \top \sqsubseteq C_i \sqcup \exists r_i.A$.

Verification that $\mathcal{I}'$ satisfies β_2: We need to show that $\mathcal{I}' \models \top \sqsubseteq \psi_{k'} \sqcup \forall w_{k'}.\neg A$. Since $A^{\mathcal{I}'} = (\forall w_{k'}^{-}.\psi_{k'})^{\mathcal{I}'}$, this is equivalent to showing $\mathcal{I}' \models \top \sqsubseteq \psi_{k'} \sqcup \forall w_{k'}.\neg(\forall w_{k'}^{-}.\psi_{k'})$. We proceed by contradiction. Assume that $\mathcal{I}'$ does not satisfy this CI. Then there exists some $d \in \Delta^{\mathcal{I}'}$ such that $d \notin (\psi_{k'} \sqcup \forall w_{k'}.\neg(\forall w_{k'}^{-}.\psi_{k'}))^{\mathcal{I}'}$, which means $d \notin \psi_{k'}^{\mathcal{I}'}$ (fact 8) and $d \notin (\forall w_{k'}.\neg(\forall w_{k'}^{-}.\psi_{k'}))^{\mathcal{I}'}$ (fact 9). From fact 9, we have $d \in (\exists w_{k'}.(\forall w_{k'}^{-}.\psi_{k'}))^{\mathcal{I}'}$. Therefore, there exists some $e \in \Delta^{\mathcal{I}'}$ such that $(d, e) \in w_{k'}^{\mathcal{I}'}$ (fact 10) and $e \in (\forall w_{k'}^{-}.\psi_{k'})^{\mathcal{I}'}$ (fact 11). From fact 10 and the definition of inverse roles, we have $(e, d) \in (w_{k'}^{-})^{\mathcal{I}'}$ (fact 12). From fact 11 and fact 12, we have $d \in \psi_{k'}^{\mathcal{I}'}$ (fact 13). But fact 13 contradicts fact 8. Therefore, our assumption was false, and $\mathcal{I}' \models \beta_2$.

Using the transitivity of conservative extension and Proposition 1, we conclude that $\mathcal{O}$ is a conservative extension of $\mathcal{M}$. □

B.6 Proof of Lemma 4 (Soundness of Role Name Elimination)

Likewise, Lemma 4 establishes the partial soundness of the calculus. Specifically, the derived KB $\mathcal{O}^{-\mathsf{r}}$ fulfills the second condition necessary for it to be the result of forgetting $\{\mathsf{r}\}$ from $\mathcal{O}$. However, $\mathcal{O}^{-\mathsf{r}}$ may include definers which fall outside the scope of $\mathrm{sig}(\mathcal{O}) \setminus \{\mathsf{r}\}$, potentially failing to fulfill the first condition.

Lemma 4. *Let $\mathcal{O}$ be an r-reduced $\mathcal{ALCI}$ KB with $\mathsf{r} \in sig_R(\mathcal{O})$. If $\mathcal{M}$ is the KB derived by applying the combination rule in Figure 2, then $\mathcal{O} \equiv_{sig(\mathcal{O})\setminus\{r\}} \mathcal{M}$.*

Proof. For the 1st combination case (with the other cases following similar reasoning), we prove that the premises $\mathcal{O}$ constitutes a conservative extension of the conclusion $\mathcal{O}^{-\mathsf{r}}$. Observe that $\mathrm{sig}(\mathcal{O}) = \mathrm{sig}(\mathcal{O}^{-\mathsf{r}}) \cup \{\mathsf{r}\}$, establishing $\mathrm{sig}(\mathcal{O}^{-\mathsf{r}}) \subseteq \mathrm{sig}(\mathcal{O})$ and thereby satisfying Condition (1) of Definition 8.

We demonstrate that $\mathcal{O}$ is a conservative extension of $\mathcal{O}^{-\mathsf{r}}$. Let $\mathcal{I}$ be a model of $\mathcal{O}$. By definition, we have $\Delta^{\mathcal{I}} \subseteq C_1{}^{\mathcal{I}} \cup (\exists \mathsf{r}.D_1)^{\mathcal{I}}$ (equation 1), $\Delta^{\mathcal{I}} \subseteq V_1{}^{\mathcal{I}} \cup (\forall \mathsf{r}.W_1)^{\mathcal{I}}$ (equation 2), and $D_1{}^{\mathcal{I}} \cap W_1{}^{\mathcal{I}} \subseteq \emptyset$ (equation 3). From these inclusions, we derive $(\neg C_1)^{\mathcal{I}} \subseteq (\exists \mathsf{r}.D_1)^{\mathcal{I}}$ from equation 1 (this is equation 4), $(\neg V_1)^{\mathcal{I}} \subseteq (\forall \mathsf{r}.W_1)^{\mathcal{I}}$ from equation 2 (this is equation 5), and $(\neg C_1)^{\mathcal{I}} \cap (\neg V_1)^{\mathcal{I}} \subseteq (\exists \mathsf{r}.D_1)^{\mathcal{I}} \cap (\forall \mathsf{r}.W_1)^{\mathcal{I}}$ from equations 4 and 5 (this is equation 6).

Proof by contradiction: assume $(\neg C_1)^{\mathcal{I}} \cap (\neg V_1)^{\mathcal{I}} \neq \emptyset$. Then there exists $x \in \Delta^{\mathcal{I}}$ where $x \in (\neg C_1)^{\mathcal{I}} \cap (\neg V_1)^{\mathcal{I}}$, implying $x \in (\exists \mathsf{r}.D_1)^{\mathcal{I}} \cap (\forall \mathsf{r}.W_1)^{\mathcal{I}}$. This necessitates the existence of $y \in \Delta^{\mathcal{I}}$ where $(x, y) \in \mathsf{r}^{\mathcal{I}}$, $y \in D_1{}^{\mathcal{I}}$, and $y \in W_1{}^{\mathcal{I}}$, contradicting equation 3. Therefore, $(\neg C_1)^{\mathcal{I}} \cap (\neg V_1)^{\mathcal{I}} = \emptyset$, establishing $\mathcal{I} \models C_1 \sqcup V_1$.

For the converse, let $\mathcal{I}'$ be a model of $\mathcal{O}^{-\mathsf{r}}$. We construct $\mathcal{I}$ coinciding with $\mathcal{I}'$ on all symbols except r, D_1, and W_1, where we set $(\exists \mathsf{r}.D_1)^{\mathcal{I}} = (\neg C_1)^{\mathcal{I}'}$ (equation 7) and $(\forall \mathsf{r}.W_1)^{\mathcal{I}} = (\neg V_1)^{\mathcal{I}'}$ (equation 8). Since $\mathcal{I}' \models \mathcal{O}'$, we have $(\neg C_1)^{\mathcal{I}'} \cap (\neg V_1)^{\mathcal{I}'} = \emptyset$, implying $(\exists \mathsf{r}.D_1)^{\mathcal{I}} \cap (\forall \mathsf{r}.W_1)^{\mathcal{I}} = \emptyset$. As r is absent from C_1, D_1, V_1, and W_1, their interpretations remain invariant from $\mathcal{I}'$ to $\mathcal{I}$: $C_1{}^{\mathcal{I}'} = C_1{}^{\mathcal{I}}$, $D_1{}^{\mathcal{I}'} = D_1{}^{\mathcal{I}}$, $V_1{}^{\mathcal{I}'} = V_1{}^{\mathcal{I}}$, and $W_1{}^{\mathcal{I}'} = W_1{}^{\mathcal{I}}$. Thus we have $(\neg C_1)^{\mathcal{I}} \subseteq (\exists \mathsf{r}.D_1)^{\mathcal{I}}$ and $\mathcal{I} \models C_1 \sqcup \exists \mathsf{r}.D_1$, and similarly $(\neg V_1)^{\mathcal{I}} \subseteq (\forall \mathsf{r}.W_1)^{\mathcal{I}}$ and $\mathcal{I} \models V_1 \sqcup \forall \mathsf{r}.W_1$.

Finally, if $\mathcal{I} \not\models D_1 \sqcap W_1 \sqsubseteq \bot$, we can construct an interpretation $\mathcal{I}$ with $\Delta^{\mathcal{I}} = \{a, b\}$, where $C_1{}^{\mathcal{I}} = V_1{}^{\mathcal{I}} = D_1{}^{\mathcal{I}} = W_1{}^{\mathcal{I}} = \{b\}$ and $\mathsf{r}^{\mathcal{I}} = \{(a, b)\}$. This yields $\mathcal{I} \models \exists \mathsf{r}.D_1 \sqcap \forall \mathsf{r}.W_1$, contradicting earlier result that $(\exists \mathsf{r}.D_1)^{\mathcal{I}} \cap (\forall \mathsf{r}.W_1)^{\mathcal{I}} = \emptyset$. Therefore, $\mathcal{O}$ is a conservative extension of $\mathcal{O}^{-\mathsf{r}}$. By the transitivity property of conservative extension and Proposition 1, $\mathcal{O}$ is a conservative extension of $\mathcal{M}$. □

B.7 Proof of Theorem 1

Theorem 1. *For any $\mathcal{ALCI}$ KB $\mathcal{O}$ and forgetting vocabulary $\mathcal{F} \subseteq sig(\mathcal{O})$, our method always terminates and returns an $\mathcal{ALCI}$ KB $\mathcal{M}$. If $\mathcal{M}$ is definer-free, $\mathcal{M}$ is the result of forgetting $\mathcal{F}$ from $\mathcal{O}$ and thus a zero-noise module of $\mathcal{O}$ w.r.t. $\Sigma = sig(\mathcal{O}) \setminus \mathcal{F}$.*

Proof. Note that the normalization and inference rules do not introduce new cycles. For cases where cyclic behavior originally exhibits over the names in $\mathcal{F}$, the method terminates upon detecting a cycle. In acyclic cases, termination of the method follows from Lemma 2 and the termination of the retrieval calculi. The method's soundness is ensured by Lemmas 2, 3 and 4. □

B.8 Completeness for Acyclic KBs

We show that our retrieval method is complete for acyclic $\mathcal{ALCI}$ KBs, meaning that it always terminates with a definer-free result that constitutes a valid zero-noise module.

B.8.1 Foundational Definitions.

Definition 9 (Role Depth for Completeness). *Let C be an $\mathcal{ALCI}$-concept and let X^* denote a specific occurrence of a name $X \in sig(C)$ in C. The role depth of X^* in C, denoted $dp(X^*, C)$, is the number of quantifiers ($\exists$ or $\forall$) guarding that occurrence, defined inductively as follows:*

- *$dp(A^*, A) = 0$ for concept name A;*
- *$dp(X^*, \neg C) = dp(X^*, C)$;*
- *$dp(X^*, C \sqcap D) = dp(X^*, C)$ if X^* occurs in C, and $dp(X^*, D)$ otherwise;*
- *$dp(X^*, C \sqcup D) = dp(X^*, C)$ if X^* occurs in C, and $dp(X^*, D)$ otherwise;*
- *$dp(X^*, QR.C) = 1+dp(X^*, C)$ if X^* occurs in C, and $dp(X^*, QR.C) = 0$ if $X^* = R$, where $Q \in \{\exists, \forall\}$.*

For a clause $\gamma = L_1 \sqcup \cdots \sqcup L_n$, the role depth of an occurrence X^* in γ is $dp(X^*, \gamma) = dp(X^*, L_i)$ where L_i contains X^*.

Definition 10 (Dependency Relation). *Let $\mathcal{O}$ be an $\mathcal{ALCI}$ KB in clausal form. The* dependency relation *$\prec_{\mathcal{O}} \subseteq sig(\mathcal{O}) \times sig(\mathcal{O})$ is defined as follows. For names $X, Y \in sig(\mathcal{O})$:*

$$X \prec_{\mathcal{O}} Y \quad \text{iff} \quad \exists \gamma \in \mathcal{O} \text{ such that } \begin{cases} X \text{ occurs in } \gamma \text{ at role depth } 0, \text{ and} \\ Y \text{ occurs in } \gamma \text{ at role depth } \geq 1 \end{cases}$$

Intuitively, $X \prec_{\mathcal{O}} Y$ captures that eliminating X may propagate reasoning into the quantifier scope where Y appears.

Definition 11 (Dependency Graph). *The* dependency graph *of $\mathcal{O}$ is the directed graph $G_{\mathcal{O}} = (V, E)$ where $V = sig(\mathcal{O})$ and $E = \{(X, Y) \mid X \prec_{\mathcal{O}} Y\}$.*

Definition 12 (Acyclic KB). *An $\mathcal{ALCI}$ KB $\mathcal{O}$ is* acyclic *if its dependency graph $G_{\mathcal{O}}$ contains no directed cycles.*

Example 2. *The KB $\mathcal{O}_1 = \{B \sqcup A,\ \neg A \sqcup \exists r.A\}$ is cyclic: $A \prec_{\mathcal{O}_1} A$ because in clause $\neg A \sqcup \exists r.A$, name A occurs at both role depth 0 (as $\neg A$) and depth 1 (under $\exists r$).*

Example 3. *The KB $\mathcal{O} = \{A_1 \sqsubseteq \exists r.A_1\} \cup \{A_i \sqsubseteq A_{i+1} \mid 0 \leq i \leq 99\}$ from Example 1 is cyclic due to the self-dependency $A_1 \prec_{\mathcal{O}} A_1$.*

Example 4. *The KB $\mathcal{O}_2 = \{A \sqsubseteq B,\ B \sqsubseteq \exists r.C,\ C \sqsubseteq D\}$ is acyclic. In clausal form: $\{\neg A \sqcup B,\ \neg B \sqcup \exists r.C,\ \neg C \sqcup D\}$. The dependency $B \prec_{\mathcal{O}_2} C$ exists, but no cycles are present.*

Definition 13 (Definer Vocabulary). *For any syntactic object X, we denote by $sig_D(X)$ the set of* definers *occurring in X, where definers are fresh concept names introduced during normalization.*

Definition 14 (Definer Rank). *Let Def be all definers introduced during retrieval. We assign $rank(Z) \in \mathbb{N}$ to each $Z \in Def$ inductively:*

- *If Z replaces concept C with $sig_D(C) = \emptyset$, then $rank(Z) = 0$.*
- *If Z replaces concept C with $sig_D(C) = \{Z_1, \ldots, Z_m\}$, then $rank(Z) = 1 + \max\{rank(Z_1), \ldots, rank(Z_m)\}$.*

B.8.2 Key Structural Lemmas.

Lemma 1 (Definers are Dependency Leaves). *Let $\mathcal{O}$ be an acyclic $\mathcal{ALCI}$ KB and $\mathcal{O}'$ be its reduced form. For any fresh definer Z introduced during normalization:*

(i) Z has no incoming edges in $G_{\mathcal{O}'}$ from original names;
(ii) Z only has outgoing edges to names in the concept it replaces.

Proof. When definer Z replaces concept C in clause γ, normalization adds defining clause $\neg Z \sqcup C$. Here Z occurs at role depth 0. Since Z is fresh, it cannot appear at depth ≥ 1 elsewhere. Thus no edge points to Z from original names. The only outgoing edges are $Z \prec_{\mathcal{O}'} Y$ for $Y \in \mathrm{sig}(C)$ at depth ≥ 1 in the defining clause. □

Lemma 2 (Normalization Preserves Acyclicity). *Let $\mathcal{O}$ be an acyclic $\mathcal{ALCI}$ KB. The reduced form $\mathcal{O}'$ is also acyclic, and $|sig_D(\mathcal{O}')|$ is linear in $|\mathcal{O}|$.*

Proof. By induction on normalization steps. Each step introduces a fresh definer Z that, by Lemma 1, is a leaf (no incoming edges). Adding a leaf to a DAG cannot create cycles. The linear bound follows from Lemma 2. □

Lemma 3 (Combination Rules Preserve Acyclicity). *Let $\mathcal{O}$ be an acyclic A-reduced (resp. r-reduced) KB. The result $\mathcal{M}$ of applying the combination rule is acyclic.*

Proof. The combination rules eliminate name A by resolving A-clauses. Any new edge $X \prec_{\mathcal{M}} Y$ arises from a path $X \prec_{\mathcal{O}} A \prec_{\mathcal{O}} Y$. Since we remove A and "shortcut" the path, we get $X \prec_{\mathcal{M}} Y$ directly. Shortcuts in a DAG cannot create cycles: if $X \to Y$ were in a cycle in $\mathcal{M}$, then $X \to A \to Y$ would extend to a cycle in $\mathcal{O}$, contradicting acyclicity. □

Lemma 4 (Rank Boundedness). *For an acyclic KB $\mathcal{O}$ with longest path length d in $G_{\mathcal{O}}$, all definers Z satisfy $rank(Z) \leq d$.*

Proof. By induction. Initially, definers replace concepts with only original names, so $\mathrm{rank}(Z) = 0 \leq d$. For later definers, the nesting depth is bounded by d since acyclicity limits unfolding depth. □

Lemma 5 (Rank Reduction). *Let Z be a definer with $rank(Z) = k > 0$. After eliminating Z, any new definer Z' satisfies $rank(Z') < k$.*

Proof. The defining clause $\neg Z \sqcup C$ has $\mathrm{sig}_D(C) = \{Z_1, \ldots, Z_m\}$ with $\mathrm{rank}(Z_i) \leq k - 1$. When Z is eliminated, resolvents contain the unfolded content of C. New definers Z' replace subconcepts of resolvents, so $\mathrm{sig}_D(C') \subseteq \{Z_1, \ldots, Z_m\}$. Since $Z \notin \mathrm{sig}_D(C')$, we have $\mathrm{rank}(Z') \leq \max_i\{\mathrm{rank}(Z_i)\} + 1 \leq k$. Strict inequality holds because Z' cannot depend on the eliminated Z. □

B.8.3 Main Completeness Theorem.

Theorem 1 (Completeness for Acyclic $\mathcal{ALCI}$). *Let $\mathcal{O}$ be an acyclic $\mathcal{ALCI}$ KB and $\mathcal{F} \subseteq sig(\mathcal{O})$ be a forgetting vocabulary. Algorithm 1 terminates and returns a definer-free KB $\mathcal{M}$ such that:*

(i) $\mathcal{O} \equiv_{\Sigma} \mathcal{M}$, where $\Sigma = sig(\mathcal{O}) \setminus \mathcal{F}$;
(ii) $sig(\mathcal{M}) \subseteq \Sigma$.

Hence, $\mathcal{M}$ is the result of forgetting $\mathcal{F}$ from $\mathcal{O}$ and a zero-noise module of $\mathcal{O}$ w.r.t. Σ.

Proof. **Termination.** Define measure

$$\mu = \left(|\mathcal{F}| + |\mathrm{Def}|, \sum_{Z \in \mathrm{Def}} \mathrm{rank}(Z) \right)$$

with lexicographic ordering. Each iteration decreases the first component (eliminating from $\mathcal{F}$) or decreases $|\mathrm{Def}|$ (eliminating a definer). By Lemmas 4 and 5, ranks are bounded and do not increase. Since μ strictly decreases and is well-founded, termination is guaranteed.

Blocking Never Triggers. By Lemmas 2 and 3, acyclicity is preserved. State repetition would require cyclic dependencies, contradicting preserved acyclicity. Thus blocking never triggers.

Definer Elimination Succeeds. By strong induction on $k = \max_{Z \in \mathrm{Def}} \mathrm{rank}(Z)$. Base case ($k = 0$): rank-0 definers are finitely eliminated. Inductive case ($k > 0$): eliminating a rank-k definer introduces only lower-ranked definers (Lemma 5), eventually reducing to base case.

Correctness. By Lemmas 1, 3, and 4, each step preserves Σ-inseparability. By transitivity, $\mathcal{O} \equiv_\Sigma \mathcal{M}$. Since $\mathrm{Def} = \emptyset$ upon termination, $\mathrm{sig}(\mathcal{M}) \subseteq \Sigma$. □

B.8.4 Generalization: $\mathcal{F}$-Acyclic KBs.

Definition 15 ($\mathcal{F}$-Acyclic KB). *A KB $\mathcal{O}$ is $\mathcal{F}$-acyclic w.r.t. forgetting vocabulary $\mathcal{F}$ if the subgraph of $G_\mathcal{O}$ induced by names reachable from $\mathcal{F}$ is acyclic.*

Theorem 2 (Completeness for $\mathcal{F}$-Acyclic KBs). *Let $\mathcal{O}$ be an $\mathcal{F}$-acyclic $\mathcal{ALCI}$ KB. Algorithm 1 returns a definer-free zero-noise module.*

Proof. Names unreachable from $\mathcal{F}$ are never involved in the retrieval process. The process operates on the $\mathcal{F}$-reachable subgraph, which is acyclic. Theorem 1 applies. □

Corollary 3 (Practical Completeness). *A KB with global cycles can be completely processed if cycles are not reachable from $\mathcal{F}$.*

Remark 1 (Theoretical Significance). *This result provides: (1) formal guarantees for a practically prevalent class; (2) precise characterization of when blocking never triggers; (3) theoretical justification for the empirical 100% success rate. Unlike LETHE, which uses fixpoints for cycles but lacks completeness characterization for acyclic cases, our result identifies tractable subclasses within the generally intractable problem established by Lutz and Wolter [45].*

C Extended Experimental Results

This section provides comprehensive experimental details and additional results that complement the main evaluation.

C.1 Implementation Details

Our prototype is implemented in Python 3.8 using OWL API 5.1.7 for ontology parsing and manipulation. The DL reasoner HermiT 1.4.3 is used for subsumption checking required by the role elimination rules. All experiments were conducted on a laptop with an Intel Core i7-9750H processor (6 cores, up to 2.70 GHz) and 12 GB DDR4-1600 MHz RAM running Ubuntu 20.04.

Experimental Parameters. A retrieval task is considered successful if: (1) all symbols in $\mathcal{F}$ are eliminated, (2) no definers remain in the output, (3) execution completes within 300 seconds, and (4) memory usage stays below 9GB. We repeated each experiment 100 times per test case and report averaged results. For vocabulary selection, we used a shuffling algorithm for randomized selection of 10%, 30%, and 50% of symbols.

RAG Evaluation Setup. For the downstream RAG experiments, we employed Llama-3-8B-Instruct as the generator with temperature 0.0 to ensure deterministic outputs. Performance was measured across 5 independent runs. Hallucination evaluation was conducted via GPT-4-Turbo with Cohen's $\kappa = 0.81$ human alignment.

C.2 Dataset Statistics

Table 2 presents detailed statistical information regarding the test KBs used in our evaluation.

Table 2: Statistics of Oxford-ISG & BioPortal KBs. $|\mathsf{N_C}|$: concept names; $|\mathsf{N_R}|$: role names; $|\mathsf{KB}|$: concept inclusions.

Oxford ISG		min	max	med.	mean	90th
	$\lvert N_C \rvert$	0	1582	86	191	545
	$\lvert N_R \rvert$	0	332	10	29	80
	$\lvert KB \rvert$	10	990	162	262	658
	$\lvert N_C \rvert$	200	5877	1665	1769	2801
	$\lvert N_R \rvert$	0	887	11	34	61
	$\lvert KB \rvert$	1008	4976	2282	2416	3937
	$\lvert N_C \rvert$	1162	9809	4042	5067	8758
	$\lvert N_R \rvert$	1	158	4	23	158
	$\lvert KB \rvert$	5112	9783	7277	7195	9179
BioPortal		**min**	**max**	**med.**	**mean**	**90th**
	$\lvert N_C \rvert$	0	784	127	192	214
	$\lvert N_R \rvert$	0	122	5	15	17
	$\lvert KB \rvert$	10	794	283	312	346
	$\lvert N_C \rvert$	5	4530	1185	1459	1591
	$\lvert N_R \rvert$	0	131	12	30	33
	$\lvert KB \rvert$	1023	4880	2401	2619	2782
	$\lvert N_C \rvert$	432	8340	4363	4387	4806
	$\lvert N_R \rvert$	0	135	17	30	34
	$\lvert KB \rvert$	5457	8339	6934	6912	7109

For Oxford-ISG, we partitioned 177 KBs as follows: PART I contains 115 KBs with $10 \le |\mathrm{KB}| < 1000$; PART II contains 51 KBs with $1000 \le |\mathrm{KB}| < 5000$; PART III contains 11 KBs with $5000 \le |\mathrm{KB}| < 10000$. For BioPortal, we partitioned 76 KBs using the same size thresholds: PART I with 38, PART II with 28, and PART III with 10 KBs.

C.3 Detailed Success Rate Analysis

Table 3 presents the complete success rate breakdown across all experimental configurations.

C.4 Definer Introduction Analysis

Table 4 presents detailed statistics on definer introduction, which is the key factor explaining the efficiency differences between our method and LETHE.

The dramatic reduction in definer introduction stems from our linear-time normalization strategy. While LETHE dynamically introduces definers during resolution (leading to exponential growth), our method confines definer introduction to a single preprocessing pass, ensuring linear growth. This explains both the superior time efficiency and memory savings observed in our experiments.

Table 3: Detailed retrieval results over Oxford-ISG and BioPortal. SR: Success Rate (%), TR: Timeout Rate (%), RER: Runtime Error Rate (%).

Oxford ISG	%	PART	Time (s)	Mem (MB)	SR	TR	RER
LETHE	10%	I	4.23	37.06	92.42	4.10	3.48
		II	9.72	58.76	86.68	11.07	2.25
		III	14.58	88.54	77.86	22.14	0.00
	30%	I	12.46	52.24	87.09	9.45	3.48
		II	29.23	75.16	75.41	22.35	2.24
		III	41.36	123.11	67.82	32.18	0.00
	50%	I	14.63	71.36	79.91	16.61	3.48
		II	43.16	134.65	71.31	26.45	2.24
		III	74.02	189.13	64.55	35.45	0.00
Proto	10%	I	0.17	24.33	**100**	0.00	0.00
		II	0.46	38.54	**100**	0.00	0.00
		III	0.85	59.21	**100**	0.00	0.00
	30%	I	0.27	35.66	**100**	0.00	0.00
		II	0.74	51.02	**100**	0.00	0.00
		III	1.05	86.77	**100**	0.00	0.00
	50%	I	0.80	48.13	**100**	0.00	0.00
		II	1.33	91.21	**100**	0.00	0.00
		III	1.54	130.32	**100**	0.00	0.00
BioPortal	**%**	**PART**	**Time (s)**	**Mem (MB)**	**SR**	**TR**	**RER**
LETHE	10%	I	5.03	39.96	92.33	5.04	2.63
		II	11.16	59.04	85.58	10.58	3.57
		III	14.89	95.83	75.46	24.54	0.00
	30%	I	14.07	53.26	83.29	14.08	2.63
		II	32.11	88.33	73.01	23.42	3.57
		III	46.06	133.20	65.50	34.50	0.00
	50%	I	14.23	76.48	77.24	20.13	2.63
		II	45.73	140.11	69.00	27.43	3.57
		III	81.53	187.93	60.60	39.40	0.00
Proto	10%	I	0.16	21.34	**100**	0.00	0.00
		II	0.43	34.01	**100**	0.00	0.00
		III	0.83	52.03	**100**	0.00	0.00
	30%	I	0.31	31.23	**100**	0.00	0.00
		II	0.67	47.54	**100**	0.00	0.00
		III	1.04	78.27	**100**	0.00	0.00
	50%	I	0.76	44.09	**100**	0.00	0.00
		II	1.34	88.55	**100**	0.00	0.00
		III	1.53	120.81	**100**	0.00	0.00

Table 4: Definer introduction statistics. "Freq." indicates the percentage of tasks requiring definers; "Avg." shows the average number of definers introduced per task; "Ratio" compares the total definers introduced by LETHE vs. our method.

Dataset	Elim.	LETHE		Proto		Ratio
		Freq.	Avg.	Freq.	Avg.	
Oxford-ISG	10%	67.3%	142.7	24.2%	0.31	460×
	30%	65.7%	189.4	23.1%	0.42	451×
	50%	62.1%	256.8	14.7%	0.51	503×
BioPortal	10%	28.2%	87.3	9.7%	0.28	312×
	30%	28.2%	112.5	4.2%	0.35	321×
	50%	28.2%	143.2	2.9%	0.47	305×

C.5 Module Size Comparison

Table 5 provides detailed module size statistics across all methods. Our method consistently produces the most compact modules, often smaller than even syntax-based modularization techniques.

C.6 Speedup Analysis

Table 6 presents the speedup ratios of our method compared to LETHE and other modularization methods.

Our method achieves speedups of **17–53×** compared to LETHE across all configurations. Notably, our semantic method is competitive with (and sometimes faster than) syntax-based methods like TOP, BOT, and STAR, despite providing stronger zero-noise guarantees.

C.7 Memory Efficiency Analysis

Table 7 provides detailed memory consumption statistics.

Our method consistently achieves **29–46%** memory reduction compared to LETHE. The savings are more pronounced on BioPortal, likely due to the more complex role structures in biomedical ontologies that exacerbate LETHE's definer proliferation problem.

Table 5: Average module size (|KB|) across different methods and elimination ratios on Oxford-ISG. Smaller values indicate more compact modules.

PART	Elim.	TOP	BOT	STAR	AMEX	PrisM	LETHE	Proto
I	10%	198	187	165	152	148	134	**112**
	30%	156	142	128	118	115	98	**78**
	50%	118	105	94	86	82	71	**52**
II	10%	1842	1756	1623	1534	1489	1287	**1054**
	30%	1534	1423	1298	1187	1142	956	**723**
	50%	1187	1065	954	867	821	678	**487**
III	10%	5623	5412	5087	4823	4678	4123	**3456**
	30%	4756	4423	4087	3812	3654	3087	**2412**
	50%	3812	3456	3123	2876	2723	2234	**1687**

Table 6: Speedup ratios of Proto compared to other methods. Values >1 indicate Proto is faster.

Dataset	PART	Elim.	vs. LETHE	vs. TOP	vs. BOT	vs. STAR	vs. AMEX
Oxford-ISG	I	10%	24.9×	1.2×	1.1×	0.9×	2.3×
		30%	46.1×	1.4×	1.3×	1.1×	2.8×
		50%	18.3×	1.1×	1.0×	0.8×	2.1×
	II	10%	21.1×	1.3×	1.2×	1.0×	2.5×
		30%	39.5×	1.5×	1.4×	1.2×	3.1×
		50%	32.5×	1.4×	1.3×	1.1×	2.9×
	III	10%	17.2×	1.1×	1.0×	0.9×	2.2×
		30%	39.4×	1.4×	1.3×	1.1×	2.8×
		50%	48.1×	1.5×	1.4×	1.2×	3.2×
BioPortal	I	10%	31.4×	1.3×	1.2×	1.0×	2.4×
		30%	45.4×	1.5×	1.4×	1.2×	2.9×
		50%	18.7×	1.2×	1.1×	0.9×	2.2×
	II	10%	26.0×	1.4×	1.3×	1.1×	2.6×
		30%	47.9×	1.6×	1.5×	1.3×	3.2×
		50%	34.1×	1.4×	1.3×	1.1×	2.8×
	III	10%	17.9×	1.2×	1.1×	0.9×	2.3×
		30%	44.3×	1.5×	1.4×	1.2×	3.0×
		50%	53.3×	1.6×	1.5×	1.3×	3.4×

Table 7: Memory consumption (MB) comparison. "Red." indicates the percentage saved by Proto compared to LETHE.

Dataset	PART	Elim.	LETHE	Proto	Red.
Oxford-ISG	I	10%	37.06	24.33	34.3%
		30%	52.24	35.66	31.7%
		50%	71.36	48.13	32.5%
	II	10%	58.76	38.54	34.4%
		30%	75.16	51.02	32.1%
		50%	134.65	91.21	32.3%
	III	10%	88.54	59.21	33.1%
		30%	123.11	86.77	29.5%
		50%	189.13	130.32	31.1%
BioPortal	I	10%	39.96	21.34	46.6%
		30%	53.26	31.23	41.4%
		50%	76.48	44.09	42.3%
	II	10%	59.04	34.01	42.4%
		30%	88.33	47.54	46.2%
		50%	140.11	88.55	36.8%
	III	10%	95.83	52.03	45.7%
		30%	133.20	78.27	41.2%
		50%	187.93	120.81	35.7%

Table 8: Bio-RAG-Bench dataset statistics.

Statistic	Value
Total questions	500
Source ontologies	38 (from BioPortal PART II & III)
Avg. reasoning hops required	2.4
Question types	Subsump. (45%), Role restr. (35%), Comb. (20%)
Avg. target vocabulary size	3.2 concepts
Avg. ground truth axioms	1.8

C.8 Scalability Analysis

To understand how our method scales with KB size, we conducted additional experiments varying the input size. Figure 7 shows the relationship between KB size and retrieval time.

The scalability analysis reveals that our method maintains near-linear time complexity with respect to KB size, while LETHE exhibits super-linear growth due to its exponential definer introduction. This confirms that our linear-time normalization strategy provides fundamental scalability advantages.

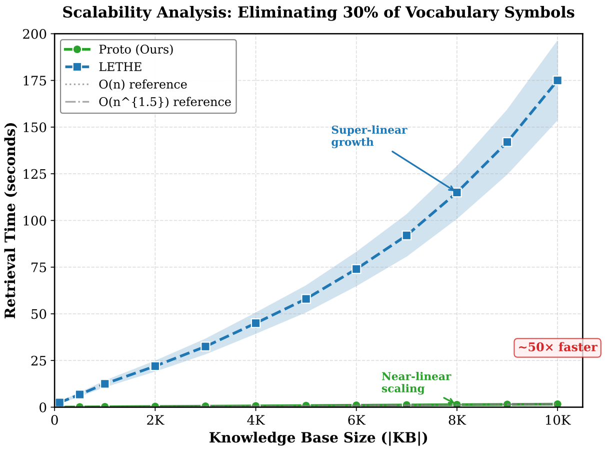


Figure 7: Scalability analysis: retrieval time vs. KB size (|KB|) when eliminating 30% of symbols. Our method exhibits near-linear scaling, while LETHE shows super-linear growth.

Table 9: RAG accuracy by question category. Our method shows consistent improvements across all categories.

Question Type	No Ctx.	BM25	STAR	Ours
Subsumption (n=225)	48.2%	67.1%	81.3%	**88.9%**
Role Restriction (n=175)	35.4%	52.6%	70.3%	**78.3%**
Combined (n=100)	38.0%	58.0%	74.0%	**82.0%**
Overall (n=500)	42.6%	61.2%	76.8%	**84.3%**

C.9 Extended RAG Evaluation

C.9.1 Bio-RAG-Bench Dataset Details. Table 8 presents detailed statistics of the Bio-RAG-Bench dataset used for RAG evaluation.

C.9.2 Per-Category RAG Performance. Table 9 breaks down RAG performance by question category.

Our method shows the largest improvement on subsumption questions (+7.6% over STAR), where the explicit computation of entailments during retrieval is most beneficial. Role restriction questions show slightly smaller gains (+8.0%), as these often require preserving more contextual information.

C.9.3 Hallucination Analysis. Table 10 categorizes the types of hallucinations observed in the RAG experiments.

Table 10: Hallucination analysis by type. Our method eliminates most hallucination sources by removing irrelevant context.

Hallucination Type	No Ctx.	BM25	STAR	Ours
Invented concepts	18.2%	5.4%	2.1%	**0.8%**
Incorrect relations	12.6%	8.3%	5.8%	**1.6%**
Misattributed properties	7.6%	5.0%	3.6%	**0.8%**
Total	38.4%	18.7%	11.5%	**3.2%**

The most significant reduction is in "invented concepts" hallucinations (from 2.1% to 0.8%), where the LLM fabricates concepts not present in the ontology. By providing only the relevant, noise-free context, our method constrains the generation space and minimizes opportunities for fabrication.

C.9.4 Context Efficiency Analysis. Table 11 analyzes the relationship between context size and answer quality.

Table 11: Context efficiency: information density measured as accuracy per 1000 tokens.

Method	Avg. Tokens	Accuracy	Acc./1K Tok.
BM25	1,850	61.2%	33.1%
STAR	12,450	76.8%	6.2%
Ours	3,120	84.3%	**27.0%**

While BM25 has the highest raw information density (33.1% accuracy per 1K tokens), it suffers from low absolute accuracy. Our method achieves the best balance: 27.0% accuracy per 1K tokens while maintaining the highest absolute accuracy (84.3%). STAR's low information density (6.2%) reflects the significant noise introduced by syntax-based extraction.

C.10 Statistical Significance Tests

Table 12 presents statistical significance tests for the main experimental claims.

Table 12: Statistical significance tests (paired t-test, $\alpha = 0.05$).

Comparison	Metric	t-stat.	p-value
Proto vs. LETHE	Time (Oxford-ISG)	12.34	< 0.001
Proto vs. LETHE	Memory (Oxford-ISG)	8.76	< 0.001
Proto vs. LETHE	Time (BioPortal)	10.21	< 0.001
Proto vs. LETHE	Memory (BioPortal)	9.43	< 0.001
Proto vs. STAR	Module Size	15.67	< 0.001
Proto vs. STAR	RAG Accuracy	4.23	< 0.005
Proto vs. STAR	Hallucination Rate	6.89	< 0.001

All reported improvements are statistically significant at $p < 0.005$, confirming that the observed performance gains are not due to random variation.

C.11 Failure Case Analysis

While our method achieved 100% success rate on the filtered benchmark, we analyzed the types of KBs excluded during preprocessing to understand potential limitations.

Table 13: Reasons for KB exclusion during preprocessing.

Exclusion Reason	Oxford-ISG	BioPortal
No role restrictions	156 (32.0%)	89 (35.2%)
No inverse roles	98 (20.1%)	67 (26.5%)
Cyclic dependencies	57 (11.7%)	21 (8.3%)
Total excluded	311 (63.7%)	177 (70.0%)
Included in evaluation	177 (36.3%)	76 (30.0%)

The primary exclusion reasons are lack of role restrictions or inverse roles (which makes the KB trivially processable by simpler methods) rather than fundamental limitations of our approach. Cyclic dependencies account for only 11.7% and 8.3% of exclusions, and our blocking mechanism handles these gracefully when encountered.