

From Global Search to Local Lookup: Scalable Knowledge Base Completion via Differentiable Guarded Logic

Yizheng Zhao

National Key Laboratory for Novel Software Technology, Nanjing University

Nanjing, China

zhaoyz@nju.edu.cn

ABSTRACT

State-of-the-art Knowledge Base Completion (KBC) models, ranging from geometric embeddings to Graph Neural Networks (GNNs), achieve impressive predictive performance but share a critical limitation: they act as black boxes that often ignore data consistency, producing predictions that violate rigorous ontological constraints. While Neural-Symbolic (NeSy) frameworks attempt to bridge this gap by integrating logical rules, they face a prohibitive scalability barrier. Existing approaches rely on unguarded universal quantification that necessitates global search, causing combinatorial explosions and execution failures on large-scale knowledge bases.

In this paper, we propose GUARDNET, a differentiable reasoning framework designed to break this expressivity-scalability deadlock. Our core contribution is leveraging the Guarded Fragment (GF) of first-order logic to fundamentally restructure the computational graph of reasoning. We demonstrate that the GF’s syntactic “guard” acts as a topological constraint that transforms intractable global quantification into efficient neighborhood-restricted lookups. For sparse knowledge bases, this paradigm shift eliminates the quadratic grounding overhead that plagues traditional NeSy systems, reducing complexity to linear in the number of edges. Extensive experiments on massive benchmarks demonstrate that GUARDNET is the first NeSy framework to scale to such large knowledge bases, succeeding where NeSy and probabilistic baselines fail to converge within 72 hours, while significantly outperforming SOTA geometric embedding and GNN models in both accuracy and consistency.

PVLDB Reference Format:

Yizheng Zhao. From Global Search to Local Lookup: Scalable Knowledge Base Completion via Differentiable Guarded Logic. PVLDB, 14(1): XXX-XXX, 2026.

doi:XX.XX/XXX.XX

PVLDB Artifact Availability:

The source code, data, and proofs of all theorems have been made available for review at <https://github.com/anonymous-ai-researcher/vldb2026>.

1 INTRODUCTION

Knowledge Base Completion (KBC) is a fundamental task in data management, requiring systems to infer missing information from

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit <https://creativecommons.org/licenses/by-nc-nd/4.0/> to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

large-scale, incomplete knowledge bases [9, 18, 26, 27]. In the Description Logic (DL) context, a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is typically defined as the union of a TBox (aka ontology) \mathcal{T} (terminological axioms defining concept hierarchies and relationships) and an ABox \mathcal{A} (assertional facts about individuals). Equivalently, from a knowledge graph perspective, the TBox corresponds to the ontological schema while the ABox corresponds to the instance-level graph of entities and relations. Modern biomedical ontologies such as SNOMED CT [43] (377K concepts) and Gene Ontology [3] (44K concepts; abbreviation: GO) encode rich taxonomic hierarchies in their TBoxes, while protein interaction networks demand reasoning over millions of relational facts in their ABoxes [47]. These applications require *multi-hop logical inference*—deducing that a drug treats a disease by composing intermediate relationships—rather than shallow pattern matching [19, 24, 32, 37, 38, 45, 65].

Prominent approaches to KBC fall into two camps, each with fundamental limitations. *Geometric embedding methods* [9, 28, 46, 51, 59, 64] and their ontology-aware extensions [1, 20, 23, 61, 67] scale linearly but sacrifice logical fidelity, learning implicit patterns rather than explicit rules. *Neural-Symbolic (NeSy) frameworks* [6, 11, 40, 53] preserve logical semantics but suffer catastrophic scalability failures on real-world knowledge bases. This creates an **expressivity-scalability dilemma** that has stymied progress toward production-grade logical reasoning systems [12, 33].

The Symbol Grounding Bottleneck. We identify the root cause of this dilemma as the *grounding bottleneck* inherent in existing NeSy architectures, as illustrated in **Figure 1**. Consider evaluating a universally quantified formula $\forall x, y. R(x, y) \rightarrow S(x, y)$ over a domain of N entities. Standard frameworks such as LTN [6] and Neural LP [60] implement this via *dense tensor broadcasting*: they materialize the full Cartesian product $N \times N$, allocating $O(N^2)$ memory to represent all possible variable bindings. For SNOMED CT with $N = 377K$, this requires 1.4×10^{11} tensor elements—far exceeding any GPU’s capacity.

This is analogous to executing a nested-loop join without indices in a database system. The vast majority of computed entries correspond to *vacuous truths*—pairs (x, y) where $R(x, y)$ is false, making the implication trivially satisfied. These computations are semantically meaningless yet consume identical resources to genuine inferences, creating massive waste that triggers out-of-memory (OOM) failures at modest scales. As we demonstrate empirically, LTN and Neural LP crash at merely 5K entities, while production knowledge bases contain two orders of magnitude more.

The Landscape of Existing Solutions. The field of NeSy AI has developed three major paradigms to integrate neural learning with symbolic reasoning [8, 13], each representing a different trade-off:

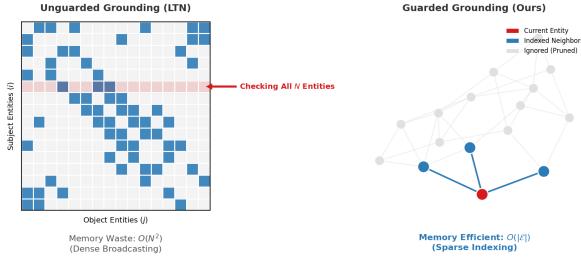


Figure 1: The Grounding Bottleneck. Left (Unguarded): Standard NeSy approaches materialize the full $N \times N$ Cartesian product, with gray cells representing vacuous truths that consume $O(N^2)$ memory. Right (Guarded): GUARDNET uses guard atoms as topological indices, restricting computation to the guard extension $\text{ext}(\alpha)$.

- *Probabilistic Logic Programming.* Systems such as ProbLog [36], DeepProbLog [30], NeurASP [62], DeepStochLog [57], and NeuPSL [35] integrate neural networks with probabilistic inference. While principled for handling uncertainty, their reliance on explicit symbolic grounding limits scalability to knowledge bases with tens of thousands of facts.
- *Logic as Soft Constraint.* Another paradigm treats logical axioms as regularization terms during training. This includes Semantic-Based Regularization [14], Semantic Loss [58], Deep Logic Models [34], and differentiable SAT solvers [56]. These methods achieve efficiency by relaxing logical strictness, but struggle to enforce complex multi-hop dependencies.
- *Differentiable Fuzzy Logic.* Frameworks including DFL [53], LTN [6], logLTN [7], Neural Theorem Provers [40], TensorLog [11], and semantic entailment systems [48, 49] embed first-order logic (FOL) into continuous vector spaces via fuzzy semantics [21, 53]. While maximally expressive, they inherit FOL’s undecidability [10, 52] and the quadratic grounding bottleneck.

Orthogonally, GNN-based approaches [50, 54, 55, 66, 68] achieve scalability through implicit message passing, but lack explicit logical semantics and cannot express negation, disjunction, or universal quantification [25]. Recent advances in temporal reasoning [63], ontology matching [15], and KG population [27] further highlight the need for scalable yet logically grounded systems. The fundamental question remains: **can we achieve the expressivity of FOL-based reasoning with the scalability of geometric methods?**

Our Insight: The Guarded Fragment as Computational Primitive. We answer this question affirmatively by identifying a classical logical formalism that has been overlooked by the NeSy community: the **Guarded Fragment (GF)** of FOL [2]. In GF, every quantified variable must appear within a *guard atom*—a relational atom that restricts the variable’s scope to a local neighborhood. This syntactic constraint, far from being a limitation, provides exactly the inductive bias needed for efficient neighborhood-based reasoning.

The key insight is that the guard atom functions as a topological index. Rather than broadcasting over the entire domain, quantification is restricted to entities connected by existing edges in the knowledge base. For a guarded formula $\forall \bar{x}(\alpha(\bar{x}) \rightarrow \psi)$, this transforms the enumeration over all $N^{|\bar{x}|}$ possible tuples into a

sparse traversal over only those tuples that actually satisfy the guard α in the knowledge base. In sparse knowledge bases—the predominant structure in real-world applications where the number of edges $|\mathcal{E}| \ll N^2$ —this yields substantial complexity reductions. Crucially, GF is not a toy fragment: it subsumes major DLs including \mathcal{ALC} [42] and \mathcal{EL}^{++} [4], while remaining decidable [5, 17]. This means the vast repository of industrial ontologies and knowledge bases can be directly translated into GF without loss of semantics, enabling rigorous logical reasoning at unprecedented scale.

The GUARDNET Framework. We operationalize this insight in GUARDNET, the first end-to-end differentiable framework that leverages the GF as a structured inductive bias for neural reasoning over knowledge bases. Our contributions are:

- **Principled Fuzzy Semantics (Section 3).** Building on theoretical foundations of fuzzy DLs [44] and differentiable fuzzy operators [53], we present a semantics grounded in the Product t-norm with a Sigmoidal Reichenbach implication. This combination ensures smooth, non-vanishing gradients that prevent dead zones plaguing existing systems.
- **Guarded Grounding with Linear Complexity (Section 4.1).** We develop a novel symbol grounding mechanism that uses guard atoms as topological indices, transforming quantified formulas into sparse, neighborhood-restricted tensor operations. This reduces memory complexity from $O(N^2)$ to $O(|\mathcal{E}|)$, enabling—for the first time—rigorous FOL-based reasoning on knowledge bases with hundreds of thousands of concepts. Unlike parallel embedding training approaches [22] that distribute existing algorithms, our method fundamentally restructures the computation itself.
- **Hybrid Domain Strategy (Sections 4.2, 4.3, 4.4)** We propose a novel witness construction mechanism that systematically prevents the model from satisfying axioms vacuously. This addresses a fundamental learning pathology—where models learn trivial “all-false” solutions—ensuring that logical satisfaction reflects genuine semantic understanding.

Empirical Validation. We conduct extensive experiments on four challenging benchmarks. GUARDNET consistently outperforms 15 state-of-the-art baselines spanning five paradigms. Notably, while SOTA NeSy methods fail with OOM at 5K entities, GUARDNET scales linearly to the full 377K concepts—a **75x improvement in scalability** with simultaneous gains in accuracy. Our ablation studies further validate the critical role of Guarded Grounding and the Hybrid Domain Strategy in achieving these results.

2 PRELIMINARIES: GUARDED FRAGMENT

In this section, we introduce the **Guarded Fragment (GF)** [2], a decidable fragment of first-order logic (FOL) that strikes a balance between expressivity and complexity (decidability).

Let \mathcal{P} , \mathcal{C} , and \mathcal{V} be countably infinite and pairwise disjoint sets of **predicate**, **constant**, and **variable** symbols, respectively. A **signature** $\Sigma = (\mathcal{P}, \mathcal{C})$ specifies the non-logical vocabulary used to construct well-formed GF formulas. Following the standard definition of GF [2], **terms** are restricted to constants and variables only, excluding function symbols present in full FOL.

An **atom** has the form $P(t_1, \dots, t_n)$, where $P \in \mathcal{P}$ is an n -ary predicate and t_1, \dots, t_n are terms. For any atom $\alpha = P(t_1, \dots, t_n)$,

we define $\text{Vars}(\alpha) = \{t_i \mid t_i \in \mathcal{V}\}$ as the set of variables occurring in α . In a quantified formula $\forall x \phi / \exists x \phi$, we say the quantifier **binds** the variable x , and ϕ is the **scope** of the quantifier. An **occurrence** of a variable x in a formula ϕ is **bound** if it lies within the scope of a quantifier that binds x ; otherwise, the occurrence is **free**. A variable x is a **free variable** of ϕ if x has at least one free occurrence in ϕ . We denote by $\text{FV}(\phi)$ the set of all free variables of ϕ . A GF formula with no free variables is called a **sentence**.

DEFINITION 1 (SYNTAX OF GF). *GF formulas are defined inductively as follows:*

- Every atom is a GF formula;
- If ϕ and ψ are GF formulas, then so are $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, and $\phi \rightarrow \psi$;
- If ψ is a GF formula, \bar{x} is a tuple of variables, and α is an atom satisfying the **guard condition**:

$$\text{FV}(\psi) \cup \bar{x} \subseteq \text{Vars}(\alpha),$$

then $\exists \bar{x} (\alpha \wedge \psi)$ and $\forall \bar{x} (\alpha \rightarrow \psi)$ are GF formulas.

In guarded quantification, the atom α serves as a **guard** for the **body** ψ . The guard condition requires that all variables involved in the quantification—both the free variables of the body and the quantified variables—appear in the guard. This fundamentally relativizes quantification: instead of ranging over the entire domain, quantified variables are restricted to the local neighborhood defined by the guard. For instance, $\exists x (R(x, y) \wedge \psi(x, y))$ restricts x to objects that are R -related to y , effectively performing a one-hop neighbor lookup on a knowledge base, directly mirroring the message-passing paradigm in GNNs.

Based on this syntax, we define the central data structure used in our framework. A **knowledge base** \mathcal{K} is a finite set of GF sentences constructed over the signature Σ . Following the DL tradition, \mathcal{K} can be decomposed into a **TBox** \mathcal{T} (terminological axioms, e.g., concept subsumptions $C \sqsubseteq D$) and an **ABox** \mathcal{A} (assertional facts, e.g., role assertions $R(a, b)$). The **size** of the knowledge base, denoted $|\mathcal{K}|$, is defined as the total number of symbols required to write down all formulas in \mathcal{K} .

DEFINITION 2 (SEMANTICS OF GF). *A **structure** for a signature Σ is a pair $\mathfrak{A} = \langle \Delta, \cdot^{\mathfrak{A}} \rangle$, where Δ is a non-empty set called the **domain**, and $\cdot^{\mathfrak{A}}$ is an **interpretation function** that maps each constant $c \in C$ to an element $c^{\mathfrak{A}} \in \Delta$ and each n -ary predicate $P \in \mathcal{P}$ to a relation $P^{\mathfrak{A}} \subseteq \Delta^n$. A **variable assignment** is a function $\rho : \mathcal{V} \rightarrow \Delta$. For a tuple of variables $\bar{x} = (x_1, \dots, x_k)$ and elements $\bar{a} = (a_1, \dots, a_k) \in \Delta^k$, we write $\rho[\bar{x} \mapsto \bar{a}]$ for the assignment that maps each x_i to a_i and agrees with ρ elsewhere. The **interpretation** of a term t under \mathfrak{A} and ρ , denoted $t^{\mathfrak{A}, \rho}$, is defined by: $x^{\mathfrak{A}, \rho} = \rho(x)$ for variables, and $c^{\mathfrak{A}, \rho} = c^{\mathfrak{A}}$ for constants.*

The **satisfaction relation** $\mathfrak{A}, \rho \models \phi$ is defined inductively:

- $\mathfrak{A}, \rho \models P(t_1, \dots, t_n)$ iff $(t_1^{\mathfrak{A}, \rho}, \dots, t_n^{\mathfrak{A}, \rho}) \in P^{\mathfrak{A}}$.
- $\mathfrak{A}, \rho \models \neg\phi$ iff $\mathfrak{A}, \rho \not\models \phi$.
- $\mathfrak{A}, \rho \models \phi \wedge \psi$ iff $\mathfrak{A}, \rho \models \phi$ and $\mathfrak{A}, \rho \models \psi$.
- $\mathfrak{A}, \rho \models \exists \bar{x} (\alpha \wedge \psi)$ iff there exists $\bar{a} \in \Delta^{|\bar{x}|}$ such that $\mathfrak{A}, \rho[\bar{x} \mapsto \bar{a}] \models \alpha \wedge \psi$.
- $\mathfrak{A}, \rho \models \forall \bar{x} (\alpha \rightarrow \psi)$ iff for all $\bar{a} \in \Delta^{|\bar{x}|}$, $\mathfrak{A}, \rho[\bar{x} \mapsto \bar{a}] \models \alpha$ implies $\mathfrak{A}, \rho[\bar{x} \mapsto \bar{a}] \models \psi$.

The remaining Boolean connectives (\vee, \rightarrow) are interpreted standardly.

GF is decidable [2], and enjoys the finite model property [17], i.e., every satisfiable GF formula has a model of a finite domain. These properties make GF a fragment where expressive logical reasoning remains computationally feasible.

3 FUZZY GF: A DIFFERENTIABLE SEMANTICS FOR LEARNING

Classical GF provides decidable reasoning [2], yet real-world applications demand the ability to handle the uncertainty and vagueness inherent in noisy data. To address this, we extend GF with fuzzy semantics, mapping atomic formulas like $\text{Fever}(x)$ to continuous truth values in $[0, 1]$. However, the choice of fuzzy operators is a critical hyperparameter defining the **geometry of the optimization landscape** [53]. A poor choice can lead to vanishing gradients or stagnation. We analyze these dynamics below (summarized in Table 1) to justify our configuration.

3.1 Fuzzy Operator Selection Rationale

Based on the gradient dynamics summarized in Table 1, we provide the theoretical justification for our operator configuration.

T-norm: Product. Among the five t-norm families, **Lukasiewicz** and **Gödel** suffer from fundamental gradient pathologies: the former creates dead zones where optimization completely stalls, while the latter exhibits winner-take-all sparsity that ignores non-bottleneck features. **Hamacher** aggressively freezes confident inputs, which can accelerate convergence but risks premature fixation. **Yager** offers tunability but introduces hyperparameter sensitivity. The **Product** t-norm uniquely provides *adaptive* gradient scaling—each input receives signal proportional to its partner’s confidence—balancing bottleneck correction with holistic representation learning.

T-conorm: Probabilistic Sum. As the dual of the Product t-norm, the **Probabilistic Sum** inherits its balanced gradient properties. Unlike **Maximum** (sparse, single-winner updates) or **Bounded Sum** (saturation-induced gradient vanishing), it distributes gradients to all disjuncts while softly penalizing redundant evidence.

Implication: Sigmoidal Reichenbach. R-implications like **Goguen** satisfy strict logical adjointness but create zero-gradient plateaus when rules are satisfied, providing no margin signal for confident predictions. The **Reichenbach** S-implication offers robust, symmetric gradients via its loss form $a(1 - c)$, but treats all violations linearly. The **Sigmoidal** variant [53] addresses this by applying a scaled sigmoid transformation that concentrates gradient mass on decision boundaries—amplifying penalties for ambiguous violations while dampening gradients for trivial cases.

3.2 Guarded Quantifiers as Parameterized Local Pooling

In GF, quantification is always restricted to a local neighborhood defined by the guard atom. To support our framework’s logic-agnostic design—where operators may range from strict (Gödel) to smooth (Product)—the aggregation mechanism must be equally adaptable. We adopt the **Generalized Mean** (p -mean) as a universal, parameterizable pooling operator. By adjusting the exponent $p \in [1, \infty)$,

Table 1: Comparison of fuzzy operators for differentiable reasoning. T-norms/T-conorms evaluated on $(a, b) = (0.9, 0.1)$; Implications evaluated on $(a, c) = (0.9, 0.2)$; Quantifiers evaluated on set $\mathbf{z} = \{0.9, 0.1\}$ for \forall (target=low). Note: Gradients for T-norms/T-conorms are w.r.t output score; for Implications w.r.t loss.

Operator	Definition	Score	Gradients	Gradient Characteristics & Learning Implications
T-norms (Conjunction \wedge): Generalizes AND. Determines credit assignment between conjuncts.				
Product (Recommended)	$T_P(a, b) = a \cdot b$	0.09	$\nabla_a=0.1$ $\nabla_b=0.9$	Adaptive signal. $\partial T / \partial a = b$: gradient scales with partner value. Stronger signal to limiting factor; near-zero values attenuate gradients, preventing destabilization.
Gödel	$T_G(a, b) = \min(a, b)$	0.10	$\nabla_a=0$ $\nabla_b=1$	Single-passing. Gradient flows only to minimum input. Winner-take-all: learning halts for non-minimum inputs, creating vast zero-gradient plateaus.
Lukasiewicz	$T_L(a, b) = \max(0, a+b-1)$	0	$\nabla_a=0$ $\nabla_b=0$	Dead zones. Gradients vanish when $a+b \leq 1$. Optimization stalls if model not “confident enough”; extremely brittle to initialization.
Yager ($p=2$)	$\max(0, 1 - \ 1-\mathbf{x}\ _p)$	0.10	$\nabla_a=-0.10$ $\nabla_b=0.90$	Tunable. p interpolates: $p=1 \rightarrow$ Lukasiewicz, $p \rightarrow \infty \rightarrow$ Gödel. Euclidean penalty without dead zones. Caution: $p < 1$ risks exploding gradients.
Hamacher	$\frac{ab}{a+b-ab}$	0.10	$\nabla_a=0.09$ $\nabla_b=0.83$	Coupled dynamics. Each gradient depends on <i>both</i> inputs non-linearly. Own truth value modulates sensitivity; may accelerate high-confidence convergence.
T-conorms (Disjunction \vee): Generalizes OR. Dual to t-norms: $S(a, b) = 1 - T(1-a, 1-b)$.				
Prob. Sum (Recommended)	$S_P(a, b) = a+b-ab$	0.91	$\nabla_a=0.9$ $\nabla_b=0.1$	Balanced. Updates all operands; penalizes redundancy softly. Dual to Product t-norm.
Maximum	$S_G(a, b) = \max(a, b)$	0.90	$\nabla_a=1$ $\nabla_b=0$	Sparse. Updates only the strongest evidence. Dual to Gödel t-norm.
Bounded Sum	$S_L(a, b) = \min(1, a+b)$	1.0	$\nabla_a=0$ $\nabla_b=0$	Saturation. Gradients vanish when sum ≥ 1 . Dual to Lukasiewicz.
Fuzzy Implications (\rightarrow): Models rules “if a then c”. Loss = $1 - I(a, c)$. Critical for rule-based learning.				
Goguen (R-impl.)	$I_G(a, c) = \min(1, c/a)$	0.78	$\nabla_a \propto c/a^2$ $\nabla_c \propto 1/a$	Zero plateau. Satisfies adjointness but gradient vanishes when satisfied ($a \leq c$). No margin signal. Explodes as $a \rightarrow 0$.
Reichenbach (S-impl.)	$I_R(a, c) = 1 - a + ac$ $\equiv \neg a \vee c$	0.72	$\nabla_a=0.8$ $\nabla_c=-0.9$	Robust. Loss = $a(1-c)$: symmetric, non-vanishing. Satisfies <i>contrapositive differentiability</i> : robust for Modus Ponens & Tollens.
Sigmoidal (Recommended)	scale($\sigma(s \cdot (I+b_0))$) scale maps σ range to $[0, 1]$	—	Boundary-focused	Corner fix [53]. Concentrates gradient on decision boundary; dampens extremes. Soft attention on hard examples.
Quantifier Aggregation (\forall, \exists): Pools evidence \mathbf{z} from neighborhood. Evaluated on $\mathbf{z} = \{0.9, 0.1\}$ for \forall (target=low).				
Strict (Min/Max)	$\min(\mathbf{z})$ for \forall $\max(\mathbf{z})$ for \exists	0.10	$\nabla_{\min}=1$ others 0	Sparse. Gradient flows only to the single “bottleneck” neighbor. Efficient but discards all supporting evidence; mimics Gödel logic.
Mean (Linear)	$\frac{1}{k} \sum z_i$	0.50	$\nabla_i=1/k$ for all i	Dilution. Outliers are washed out by the majority. A single failure (0.1) is masked by successes, making it poor for logical validity.
Gen. Mean (Recommended)	$\mathcal{A}_p^\forall(\mathbf{z})$ (Eq. 2) $\mathcal{A}_p^\exists(\mathbf{z})$ (Eq. 1)	0.36 ($p=2$)	Weighted by error	Tunable Focus. p controls sensitivity to extremes. Low p aggregates uniformly; high p focuses on violations. $p \approx 2$ offers optimal trade-off.

it provides a continuous spectrum: reducing to the arithmetic mean at $p = 1$ (matching Lukasiewicz linearity) and converging to the hard extremum as $p \rightarrow \infty$ (matching Gödel). Let $\mathbf{z} = (z_1, \dots, z_k)$ be the truth values evaluated over k guard-satisfying neighbors.

Existential quantifier (\exists). We employ the standard Generalized Mean to aggregate evidence for satisfaction:

$$\mathcal{A}^\exists(\mathbf{z}; p) = \left(\frac{1}{k} \sum_{i=1}^k z_i^p \right)^{\frac{1}{p}} \quad (1)$$

Higher p focuses on the best evidence (approaching max); lower p aggregates uniformly.

Universal quantifier (\forall). We employ the **Generalized Mean Error** to aggregate deviation from perfect truth [6]:

$$\mathcal{A}^\forall(\mathbf{z}; p) = 1 - \left(\frac{1}{k} \sum_{i=1}^k (1 - z_i)^p \right)^{\frac{1}{p}} \quad (2)$$

This formulation prevents gradient vanishing when the majority of neighbors satisfy the condition. As $p \rightarrow \infty$, the result converges to the minimum (the “weakest link”).

Numerical stability. As noted by Badreddine et al. [6], the gradient of the p -mean can explode when inputs approach boundaries (0 or 1) for large p . To ensure stability across arbitrary operator combinations, we implement the *Stable Product* strategy, applying a projection $\pi(x) = (1 - \epsilon)x + \epsilon$ to inputs before aggregation.

3.3 Generalized Fuzzy Satisfaction

Finally, we integrate the operator choices discussed above into a unified semantic framework. GUARDNET treats the logical semantics as a hyperparameter, allowing the system to be configured based on domain constraints (e.g., noise levels, graph sparsity). We define the generalized fuzzy satisfaction degree $\llbracket \phi \rrbracket_{\mathfrak{A}, \rho} \in [0, 1]$ as follows.

DEFINITION 3 (GENERALIZED FUZZY SATISFACTION). Let \mathfrak{A} be a fuzzy structure, ρ a variable assignment, and $O = \{T, S, I, \mathcal{A}_p^{\exists}, \mathcal{A}_p^{\forall}\}$ be a selected set of differentiable fuzzy operators parameterized by p . The satisfaction degree is defined recursively:

$$\begin{aligned} \llbracket P(\bar{t}) \rrbracket_{\mathfrak{A}, \rho} &= P^{\mathfrak{A}}(\bar{t} | \mathfrak{A}, \rho) \\ \llbracket \neg \phi \rrbracket_{\mathfrak{A}, \rho} &= 1 - \llbracket \phi \rrbracket_{\mathfrak{A}, \rho} \\ \llbracket \phi \wedge \psi \rrbracket_{\mathfrak{A}, \rho} &= T(\llbracket \phi \rrbracket_{\mathfrak{A}, \rho}, \llbracket \psi \rrbracket_{\mathfrak{A}, \rho}) \\ \llbracket \phi \vee \psi \rrbracket_{\mathfrak{A}, \rho} &= S(\llbracket \phi \rrbracket_{\mathfrak{A}, \rho}, \llbracket \psi \rrbracket_{\mathfrak{A}, \rho}) \\ \llbracket \exists \bar{x} (\alpha \wedge \psi) \rrbracket_{\mathfrak{A}, \rho} &= \mathcal{A}_p^{\exists} \left(\left\{ \llbracket \alpha \wedge \psi \rrbracket_{\mathfrak{A}, \rho'} \mid \bar{a} \in \Delta^{|\bar{x}|} \right\} \right) \\ \llbracket \forall \bar{x} (\alpha \rightarrow \psi) \rrbracket_{\mathfrak{A}, \rho} &= \mathcal{A}_p^{\forall} \left(\left\{ \llbracket \alpha \rightarrow \psi \rrbracket_{\mathfrak{A}, \rho'} \mid \bar{a} \in \Delta^{|\bar{x}|} \right\} \right) \end{aligned}$$

where $\rho' = \rho[\bar{x} \mapsto \bar{a}]$ denotes the assignment update.

This formulation provides GUARDNET with continuous adaptability: it can interpolate between **strict, rule-bound logic** (using Gödel operators and large p) for precise reasoning in clean domains, and **probabilistic, soft reasoning** (using Product operators and small p) for robust learning in noisy, uncertain knowledge bases.

4 GUARDNET: A DIFFERENTIABLE FUZZY GF FRAMEWORK

To operationalize the theoretical advantages of GF for learning, we introduce GUARDNET, a neural-symbolic framework built upon three core components: (1) a scalable neural grounding mechanism; (2) a unified fuzzy loss function; and (3) a semantically-aware domain construction strategy. The overall architecture is illustrated in [Figure 2](#).

4.1 Neural Grounding Mechanism

The bridge between logic and neural networks is the **grounding** function \mathcal{G}_{θ} . Before detailing our implementation, we formally analyze the bottleneck that necessitates our architectural departure from standard neural-symbolic approaches.

4.1.1 The Bottleneck: Unguarded Grounding and Gradient Dilution. Standard approaches (e.g., LTN) typically ground FOL rules by instantiating variables over the entire domain Δ . For a rule like $\forall x, y (R(x, y) \rightarrow C(y))$, standard semantics treat quantifiers as

unguarded, aggregating over the full Cartesian product:

$$\llbracket \forall x, y \phi(x, y) \rrbracket = \bigotimes_{u, v \in \Delta} \llbracket \phi(u, v) \rrbracket \quad (3)$$

This formulation introduces two critical failures in large-scale learning:

- **System Implementation Bottleneck (Memory):** Physically evaluating Eq. 3 in deep learning frameworks requires **tensor broadcasting** to materialize all variable combinations. For a domain size of $|\Delta| \approx 3.7 \times 10^5$ (e.g., SNOMED CT), constructing a dense pairwise tensor of shape $[\Delta, |\Delta|, d]$ attempts to allocate hundreds of gigabytes of GPU memory, triggering immediate OOM errors.
- **Optimization Bottleneck (Gradient Dilution):** Even if memory permits, existing frameworks remain agnostic to the sparsity of R . They evaluate the implication $R(u, v) \rightarrow C(v)$ for every pair. In sparse knowledge bases, $R(u, v)$ is false for the vast majority of pairs. Since the logical implication *False* $\rightarrow \dots$ trivially evaluates to True (1), the model wastes $O(|\Delta|^2)$ computation verifying these tautologies. Crucially, this creates a gradient dilution problem: the learning signal from the few actual edges (where R is true) is mathematically drowned out by millions of vacuously satisfied instances, stalling convergence.

4.1.2 The Solution: Guarded Sparse Indexing. GUARDNET resolves this by strictly enforcing the **Guarded Grounding** paradigm. Instead of broadcasting over the domain, we leverage the GF syntax to treat the guard atom α as a topological index.

Complexity Analysis. Let $\phi = \forall \bar{x} (\alpha(\bar{x}) \rightarrow \psi(\bar{x}))$ be a guarded axiom where α is the guard atom and $|\bar{x}| = k$ is the number of quantified variables. We define the *guard extension* as:

$$\text{ext}(\alpha) = \{(a_1, \dots, a_k) \mid \alpha(a_1, \dots, a_k) \text{ holds in } \mathcal{K}\}$$

- **Unguarded Grounding:** Standard approaches enumerate all N^k variable assignments (where $N = |\Delta|$ is the domain size), yielding $O(N^k)$ complexity per axiom. For binary predicates ($k = 2$), this is $O(N^2)$.
- **Guarded Grounding:** Our approach enumerates only tuples in $\text{ext}(\alpha)$, yielding $O(|\text{ext}(\alpha)|)$ complexity per axiom.

For a knowledge base \mathcal{K} with m axioms, the total grounding complexity per epoch is:

$$O \left(\sum_{i=1}^m |\text{ext}(\alpha_i)| \right) \leq O(m \cdot |\mathcal{E}|_{\max})$$

where $|\mathcal{E}|_{\max} = \max_i |\text{ext}(\alpha_i)|$ is the size of the largest guard extension in \mathcal{K} .

Sparsity Assumption. This complexity advantage is predicated on the *sparsity* of real-world knowledge bases. When guard predicates have bounded extension size (i.e., $|\mathcal{E}|_{\max} = O(N)$), guarded grounding achieves complexity linear in the domain size. We note two caveats:

- In dense knowledge bases where guard extensions approach $\Theta(N^2)$, guarded and unguarded grounding have equivalent asymptotic complexity.

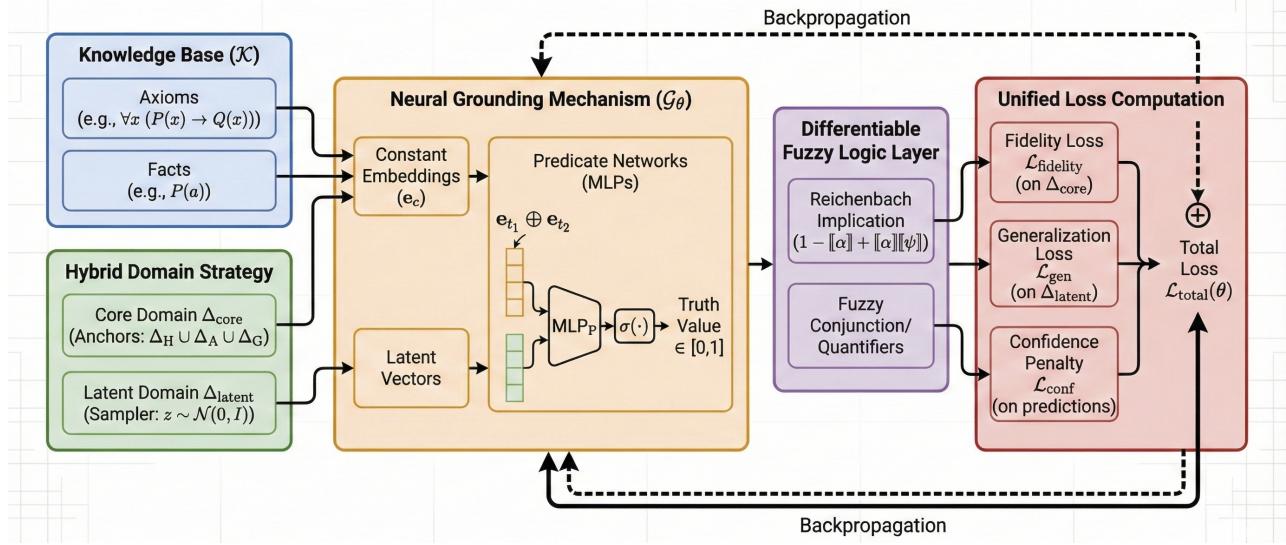


Figure 2: The GUARDNET framework, illustrating the Neural Grounding mechanism, the Hybrid Domain construction, and the unified loss computation.

- Our experimental datasets (SNOMED CT, GO, PPI networks) exhibit sparse structure with average degree $\bar{d} \ll N$, empirically validating this assumption.

Below we detail the specific components of this efficient grounding mechanism:

- **Constants as Learnable Embeddings:** Each constant symbol $c \in C$ is grounded as a dense vector embedding $e_c \in \mathbb{R}^d$. Unlike traditional symbolic reasoning where constants are rigid identifiers, here they act as malleable parameters. This optimization process allows the embeddings to capture the semantic properties of entities specifically as constrained by the logical axioms (e.g., learning that two constants appearing in similar logical contexts have similar vector representations).
- **Predicates as Differentiable Functions:** To ground our n -ary predicates in a way that is both expressive and scalable, we employ Multi-Layer Perceptrons (MLPs) as universal function approximators [41]. This choice provides the flexibility to learn arbitrary non-linear relationships without imposing strong prior assumptions on their geometric structure, which is a crucial feature for a general-purpose GF framework.

For an n -ary predicate, the input strategy to this MLP is straightforward: it is formed by concatenating the n individual term embeddings [16]. Concatenation is chosen as it is a parameter-free, information-preserving operation that makes no prior assumptions about the relationships between predicate arguments, leaving this task entirely to the learnable layers of the predicate network (see Figure 2). The truth value for an atom $P(t_1, \dots, t_n)$ is then explicitly given by:

$$\llbracket P(t_1, \dots, t_n) \rrbracket_{\mathcal{G}_\theta} = \sigma(\text{MLP}_P(e_{t_1} \oplus \dots \oplus e_{t_n})) \quad (4)$$

where $e_{t_i} \in \mathbb{R}^d$ is the embedding of term t_i and \oplus denotes concatenation. Our predicate MLPs utilize the ReLU activation function in hidden layers for its robustness against vanishing gradients.

The final output layer employs a Sigmoid function $\sigma(\cdot)$ to map the network's logit to a fuzzy truth value in the $[0, 1]$ interval, aligning with a probabilistic interpretation of satisfaction.

4.2 The Challenge: Representational Collapse

A critical vulnerability in neural-symbolic learning is the propensity for models to discover *reasoning shortcuts* [31]—devious paths to minimizing the loss function that bypass meaningful learning. We trace the origin of many such shortcuts to a foundational flaw: an improperly constructed interpretation domain. This flaw permits a catastrophic failure mode we term **representational collapse**. Consider a simple knowledge base with the axiom $\forall x (\text{Cat}(x) \rightarrow \exists y (\text{hasPart}(x, y) \wedge \text{Tail}(y)))$. Representational collapse manifests in two primary forms, as depicted in Figure 3:

- **Collapse to Emptiness:** The model achieves zero loss by simply learning to predict $\llbracket \text{Cat}(c) \rrbracket \approx 0$ for all constants c . In this degenerate solution, the axiom is *vacuously satisfied* (since False \rightarrow Anything is True), and the model learns nothing about the actual relationship between cats and tails.
- **Collapse to Ambiguity:** The neural networks learn to output a constant value of 0.5 for all inputs, representing maximum uncertainty. This strategy exploits the properties of certain fuzzy loss landscapes to minimize penalties without the risk of making incorrect decisive predictions.

While prior frameworks such as LTN [6] can be effective given dense factual data, real-world ontologies often consist primarily of universal rules (axioms) with sparse factual assertions. This lack of explicit counter-examples makes them highly vulnerable to the aforementioned collapse modes.

Two natural and standard approaches initially seem plausible to address this data scarcity, but they ultimately fail:

- **Failure of Naive Sampling:** First, one might propose relying solely on randomly sampled vectors (“latent constants”) from a

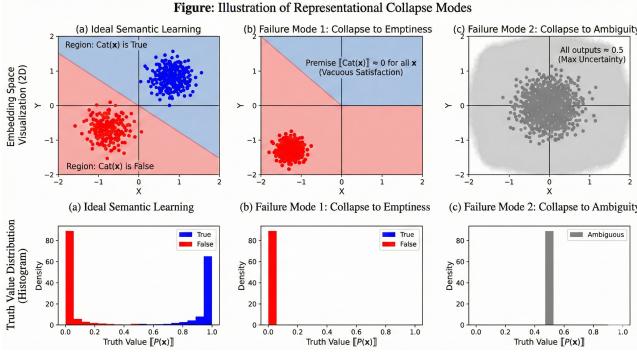


Figure 3: Illustration of Representational Collapse Modes: Collapse to Emptiness (vacuous satisfaction) and Collapse to Ambiguity (maximum uncertainty).

prior distribution to drive learning. However, when faced with random inputs producing ambiguous outputs (e.g., $\llbracket \text{Cat}(\text{latent}_1) \rrbracket \approx 0.5$), the model quickly discovers a trivial optimal strategy: setting $\llbracket \text{Cat}(x) \rrbracket \approx 0$ for all inputs. This allows the model to achieve zero loss by making the axiom vacuously satisfied, without learning any semantic structure.

- **Failure of Standard Regularization:** Second, one might employ standard regularization techniques [16] to mitigate collapse. We do actually incorporate such methods: a truth confidence penalty $\mathcal{L}_{\text{conf}} = \sum_i p_i (1 - p_i)$ that softly encourages predicate outputs p_i to be closer to 0 or 1, along with standard L2 weight decay. However, these techniques prove fundamentally insufficient. The confidence penalty is “direction-agnostic”—it encourages confident predictions but provides no guidance on what those predictions should be. Even with $\mathcal{L}_{\text{conf}}$, the model can achieve perfect satisfaction by confidently predicting $\llbracket \text{Cat}(x) \rrbracket \approx 0$ for all inputs. This simultaneously satisfies the confidence penalty (since 0 is a decisive prediction) and makes the axiom vacuously satisfied (since the premise is never met), while the model learns nothing meaningful about the relationship between cats and tails.

4.3 The Solution: Hybrid Domain Strategy

These failures reveal that robust learning requires explicit semantic anchors that force non-trivial engagement with logical structure. To address this, GUARDNET implements a **Hybrid Domain Strategy** that synergistically combines a Core Domain with a Latent Domain, as visualized in **Figure 4**.

4.3.1 Core Domain: Ensuring Fidelity via Semantically-Aware Construction. The Core Domain, Δ_{core} , provides a set of evolving “semantic anchors” that prevent representational collapse through a principled, multi-layered construction process addressing distinct failure modes.

- **Herbrand Domain (Δ_H):** A natural and canonical starting point rooted in classical logic is the **Herbrand Universe** [39]. In the context of a function-free language like GF, this domain consists of all constant symbols that explicitly appear as arguments to predicates within the knowledge base \mathcal{K} :

$$\Delta_H = \{c \mid c \in C \text{ and } c \text{ appears as an argument in } \mathcal{K}\} \quad (5)$$

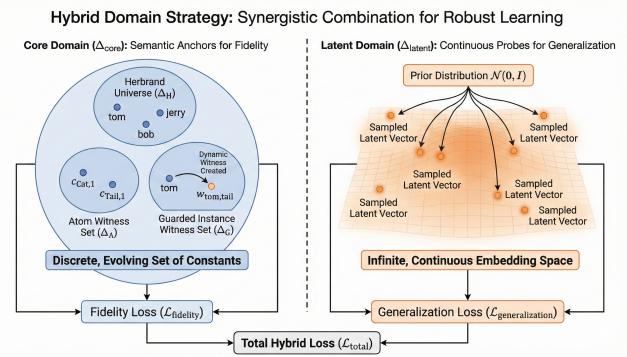


Figure 4: Hybrid Domain Strategy: The Core Domain (Δ_{core}) provides semantic anchors to ensure fidelity; the Latent Domain (Δ_{latent}) drives generalization via continuous sampling.

This domain serves to ground the model’s reasoning in the concrete entities for which prior factual knowledge exists. However, relying solely on Δ_H is theoretically insufficient for robust learning. Real-world knowledge bases often consist primarily of universal rules without specific instances, rendering Δ_H sparse or even empty. Consequently, a domain constrained only to these explicit constants remains highly vulnerable to the representational collapse modes described previously.

- **Atom Witness Set (Δ_A):** This component acts as our primary defense against global collapse to emptiness, addressing the foundational problem of predicate existence in the absence of explicit facts. For every n -ary predicate $P \in \mathcal{P}$ that is not currently grounded by factual assertions in \mathcal{K} , we introduce a unique tuple of n fresh witness constants, $(c_{P,1}, \dots, c_{P,n})$. Through dedicated loss terms, we actively enforce a high truth value for these grounded atoms, i.e., $\llbracket P(c_{P,1}, \dots, c_{P,n}) \rrbracket \approx 1$. This mechanism guarantees that every concept in the theory—such as $\text{Cat}(c_{\text{Cat},1})$ or $\text{Tail}(c_{\text{Tail},1})$ —is interpreted as a non-empty set from the outset, thereby rendering a total collapse to emptiness impossible.
- **Guarded Instance Witness Set (Δ_G):** While atom witnesses break the initial stalemate of emptiness, they are insufficient on their own. A model might satisfy atom witnesses while still learning only vague statistical associations—maintaining moderate truth values that achieve acceptable loss through quantifier aggregation without learning precise relationships. To combat this, we introduce a dynamic mechanism designed to make such ambiguity computationally untenable for complex axioms. This mechanism creates specific evidence “on-demand” when guarded rules are triggered. Its dynamic nature is crucial: the set of witnesses grows as the model learns. Consider the sample axiom $\forall x (\text{Cat}(x) \rightarrow \exists y (\text{hasPart}(x, y) \wedge \text{Tail}(y)))$.
 - **Early Training:** At epoch 1, model predictions are poor. The premise $\llbracket \text{Cat}(c) \rrbracket$ is low for all constants, so no witnesses are created.
 - **Triggering Event:** As training progresses (e.g., epoch 50), the model learns to recognize specific entities, such as ‘tom’, as cats with high confidence ($\llbracket \text{Cat}(\text{tom}) \rrbracket \approx 0.95$).

- **Dynamic Creation:** This high-confidence guard triggers the creation of a *fresh* witness constant, $w_{\text{tom,tail}}$, which is dynamically added to the domain.

Each such event immediately augments the learning objective with a strong enforcement loss specific to that instance:

$$\begin{aligned} \mathcal{L}_{\text{witness}}(\text{axiom}, \text{tom}) = \\ 1 - [\![\text{hasPart}(\text{tom}, w_{\text{tom,tail}}) \wedge \text{Tail}(w_{\text{tom,tail}})]\!] \end{aligned} \quad (6)$$

Critically, this mechanism renders the **collapse to ambiguity** a prohibitively costly strategy.

Since our framework interprets conjunction (\wedge) with the Product t-norm, an ambiguous output where both atoms in the conclusion have a truth value of 0.5 would yield a conjunction value of $0.5 \times 0.5 = 0.25$. With a high-confidence guard ($[\![\text{Cat}(\text{tom})]\!] \approx 1$), the witness loss becomes approximately $1 - 0.25 = 0.75$, a substantial penalty that the optimizer must eliminate. To minimize this loss, the model is mathematically compelled to push the truth values of *both* atoms towards 1, enforcing high-confidence predictions. These dynamically generated witnesses are fully integrated into the learning process. Their vector embeddings are initialized randomly (via Xavier) and become trainable parameters. Once added to Δ_{core} , they are available for sampling across all axioms, ensuring their semantic properties are constrained by the entire knowledge base.

The entire **Core Domain** is thus the dynamically evolving set $\Delta_{\text{core}} = \Delta_H \cup \Delta_A \cup \Delta_G$.

4.3.2 Latent Domain Δ_{latent} : Driving Generalization. While the Core Domain Δ_{core} ensures logical fidelity by grounding the model in specific entities, relying on it alone risks overfitting. The model might memorize the properties of these particular constants without capturing the universal nature of the logical rules. To address this and ensure robust generalization, we introduce the **Latent Domain** Δ_{latent} , which represents the infinite, continuous embedding space itself, rather than a fixed set of discrete points.

Our strategy for utilizing the latent domain involves a continuous sampling mechanism. In each training step, we dynamically generate a mini-batch of “latent constants” by sampling vectors from a standard prior distribution (e.g., $\mathcal{N}(0, I)$ in \mathbb{R}^d). These temporary vectors act as random probes of the learned geometric space, used exclusively to evaluate universal axioms. This process defines our **generalization loss**, which forces the predicate networks to learn decision boundaries that are logically coherent everywhere in the embedding space, not just at the locations of known constants in Δ_{core} . Effectively, this ensures the model learns the general concept of what “a cat is”, rather than merely learning that specific entities like “Tom is a cat”.

4.4 The Hybrid Training Objective

The central learning objective in GUARDNET is to find the optimal parameters θ that maximize the satisfiability of the knowledge base \mathcal{K} . We derive our training objective directly from our principled fuzzy semantics defined in Section 3, constructing a loss function that minimizes the dissatisfaction of each axiom using our selected differentiable operators.

Loss for a Single Axiom Instance. For any universally quantified axiom of the form $\forall \bar{x}(\alpha \rightarrow \psi)$, the loss for a grounded instance is derived using the **Sigmoidal Reichenbach implication** recommended in Section 3.1. Let $I_R = 1 - [\![\alpha]\!] + [\![\alpha]\!][\![\psi]\!]$ be the standard Reichenbach implication score. The instance loss is defined as:

$$\begin{aligned} \mathcal{L}_{\text{instance}}(\forall \bar{x}(\alpha \rightarrow \psi); \bar{a}) &= 1 - [\![\alpha \rightarrow \psi]\!]_{\rho[\bar{x} \mapsto \bar{a}]} \\ &= 1 - \text{scale}(\sigma(s \cdot (I_R(\bar{a}) + b_0))) \end{aligned} \quad (7)$$

where $\sigma(\cdot)$ is the Sigmoid function with steepness s and bias b_0 . While this formulation introduces non-linearity to sharpen gradients around the decision boundary, it fundamentally preserves the deductive structure of the underlying Reichenbach logic (I_R). The core term I_R dictates that the penalty is driven by $[\![\alpha]\!](1 - [\![\psi]\!])$, aligning with the classical inference rules of Modus Ponens and Modus Tollens. By wrapping this logic in the Sigmoidal transformation, we ensure robust, non-vanishing gradients even for satisfied or violated constraints, as analyzed in Table 1.

Theoretical Total Loss. The total loss aggregates the dissatisfaction over all axioms. Consistent with our choice of the **Generalized Mean** for quantifier aggregation (Section 3.2), we define the theoretical objective not as a simple linear expectation, but using the p -mean error aggregator \mathcal{A}_p^\vee :

$$\mathcal{L}_{\text{theoretical}}(\theta) = \sum_{\phi_i \in \mathcal{K}} \mathcal{A}_p^\vee \left(\left\{ \mathcal{L}_{\text{instance}}(\phi_i; \bar{a}) \mid \bar{a} \in \Delta^{|\text{FV}(\phi_i)|} \right\} \right) \quad (8)$$

This formulation ensures that the model focuses on the most violated instances (behaving like a soft-max) rather than being diluted by the majority of satisfied cases, a critical feature for learning from sparse signals in knowledge bases.

The Practical Hybrid Loss Function. To align with our **Hybrid Domain Strategy**, we realize this objective via stratified sampling. The total loss is a weighted combination:

$$\mathcal{L}_{\text{total}}(\theta) = \lambda \cdot \mathcal{L}_{\text{fidelity}} + (1 - \lambda) \cdot \mathcal{L}_{\text{gen}} + \gamma \cdot \mathcal{L}_{\text{conf}} \quad (9)$$

where $\lambda \in (0, 1]$ balances fidelity and generalization, and γ controls the confidence regularization $\mathcal{L}_{\text{conf}} = \sum_i p_i(1 - p_i)$. The two main loss terms are explicitly defined as follows:

- **Fidelity Loss ($\mathcal{L}_{\text{fidelity}}$):** Computed on the Core Domain Δ_{core} , this term enforces consistency with known facts and dynamic witnesses:

$$\mathcal{L}_{\text{fidelity}} = \mathcal{A}_p^\vee (\{\mathcal{L}_{\text{instance}}(\phi; \bar{c}) \mid \bar{c} \sim \Delta_{\text{core}}\}) + \sum \mathcal{L}_{\text{witness}} \quad (10)$$

It acts as “textbook examples”, grounding the model in the given theory and explicitly preventing collapse via the witness term.

- **Generalization Loss (\mathcal{L}_{gen}):** Computed on the Latent Domain Δ_{latent} , this term enforces universal consistency:

$$\mathcal{L}_{\text{gen}} = \mathcal{A}_p^\vee (\{\mathcal{L}_{\text{instance}}(\phi; \bar{z}) \mid \bar{z} \sim \mathcal{N}(0, I)\}) \quad (11)$$

It acts as the “final exam”, testing the model with countless unseen inputs to ensure it has learned the universal principles rather than memorizing specific constants.

By jointly optimizing these complementary loss terms, GUARDNET learns a model that is both faithful to the provided data and robustly generalizable.

THEOREM 1 (SOUNDNESS OF GUARDNET). Let \mathcal{K} be a knowledge base consisting of a finite set of GF formulas. If a GUARDNET model trained on \mathcal{K} achieves $\mathcal{L}_{\text{fidelity}}(\theta) = 0$, then the learned fuzzy interpretation \mathcal{G}_θ satisfies all axioms on the Core Domain Δ_{core} . That is, for every axiom $\phi \in \mathcal{K}$ and every grounding $\bar{c} \in \Delta_{\text{core}}^{|\text{FV}(\phi)|}$, we have $\llbracket \phi \rrbracket_{\mathcal{G}_\theta, \bar{c}} = 1$.

Complementing its semantic soundness, GUARDNET also enjoys efficient training dynamics due to the syntactic restrictions of GF.

THEOREM 2 (TRAINING COMPLEXITY). Let \mathcal{K} be a knowledge base with m guarded axioms $\{\forall \bar{x}_i (\alpha_i \rightarrow \psi_i)\}_{i=1}^m$ over a domain of N entities. Let d denote the embedding dimension, L the number of layers in predicate MLPs, B the batch size, and $|\mathcal{P}|$ the number of predicates. Recall that $\text{ext}(\alpha)$ denotes the guard extension (defined in Section 4.1) and Δ_{core} the Core Domain (Section 4.3). A single training iteration of GUARDNET has:

Time Complexity:

$$O\left(\sum_{i=1}^m |\text{ext}(\alpha_i)| \cdot L \cdot d^2\right) = O\left(m \cdot |E|_{\max} \cdot L \cdot d^2\right) \quad (12)$$

where $|E|_{\max} = \max_i |\text{ext}(\alpha_i)|$ is the largest guard extension. The d^2 factor arises from matrix operations in the MLP forward pass.

Space Complexity:

$$O\left(|\Delta_{\text{core}}| \cdot d + |\mathcal{P}| \cdot L \cdot d^2 + B \cdot |E|_{\max}\right) \quad (13)$$

where the three terms account for entity embeddings, predicate MLP parameters, and the batch of grounded instances, respectively.

In sparse knowledge bases where $|E|_{\max} = O(N)$, both complexities are **linear in the domain size** N , circumventing the $O(N^2)$ bottleneck of unguarded approaches.

5 EMPIRICAL EVALUATION

In the previous sections, we theoretically argued that GUARDNET resolves the fundamental conflict between expressivity and scalability. This is achieved through our novel **Guarded Inductive Bias**, which restricts computation to valid topological neighborhoods, and our **Hybrid Domain Strategy**, which ensures robust generalization. This section provides the **empirical substantiation** of these claims. We structure our evaluation to progress from system-level feasibility to task-level performance, and finally to architectural component analysis. Specifically, we design our experiments to answer three core Research Questions (RQs):

- **RQ1 (Scalability & System Efficiency):** Can GUARDNET physically overcome the “tensor broadcasting” bottleneck analyzed in Section 4.1 and scale to large knowledge bases (e.g., SNOMED CT) where existing neural-symbolic baselines fail?
- **RQ2 (Comparative Performance):** Once feasibility is established, does GUARDNET outperform state-of-the-art Geometric Embeddings, GNNs, and NeSy models across diverse reasoning tasks, demonstrating robust generalization capabilities?
- **RQ3 (Ablation & Mechanism Analysis):** What are the sources of these performance gains? We conduct a comprehensive ablation study to verify the impact of the **Hybrid Domain Strategy** in preventing representational collapse, and analyze the sensitivity of learning dynamics to different **Fuzzy Operators**.

Data Translation Strategy. Since no native benchmark datasets exist specifically for the GF, we construct our experimental testbed by converting standard ontologies into GF formulas. This approach is theoretically well-grounded due to the deep syntactic connections among these formalisms. GF was originally introduced in [2] as a natural generalization of modal logic to the polyadic (multi-variable) setting, preserving key properties such as decidability and the finite model property. DLs, in turn, are well-known to be notational variants of modal logics [5]. As a result, standard DL axioms translate directly into GF sentences. This principled translation ensures that our experiments faithfully evaluate reasoning capabilities on semantically equivalent logical structures.

Evaluation Tasks and Datasets. We evaluate GUARDNET on two reasoning tasks. **TBox Completion** tests concept subsumption: given a test axiom $C \sqsubseteq D$, we rank the true axiom against corruptions generated by replacing C (or D) with all candidate concepts. **ABox Completion** tests link prediction: given a test triple $r(h, t)$, we rank candidates for head prediction ($?, r, t$) and tail prediction ($h, ?, ?$). Both tasks employ the standard **filtered setting** [9], removing known true facts from the candidate list.

For TBox completion, we use **SNOMED CT** (377K concepts) [43] as a scalability benchmark in medical terminology and **Gene Ontology (GO)** (44K concepts) [3] for its hierarchically rich biological taxonomy. These ontologies are expressed in the DL \mathcal{EL}^{++} [4], a syntactic fragment of GF [5]. For ABox completion, we use two protein-protein interaction (PPI) datasets (**Yeast PPI**: 6.4K entities, 110K edges, **Human PPI**: 17.6K entities, 75K edges), combining factual interactions from the STRING database [47] with TBox constraints from GO. Following standard conventions, all datasets are randomly split into 80% training, 10% validation, and 10% test sets.

Baselines. We benchmark GUARDNET against 15 baselines spanning five distinct paradigms in KBC tasks:

- **Geometric Ontology Embeddings:** EL Embeddings (ELEM) [23], Box²EL [20], and TransBox [61]. As state-of-the-art approaches that are specifically designed to capture the semantics of the DL \mathcal{EL}^{++} , these constitute the strongest and most direct baselines for ontology-based reasoning tasks.
- **Graph Neural Networks (GNNs):** CompGCN [54], GRAIL [50], and NBFNet [68]. These represent the leading GNN-based approaches for the KBC task, whose reliance on implicit message propagation provides a crucial contrast to GUARDNET’s explicit, logic-defined neighborhoods.
- **Neural-Symbolic (FOL-based):** LTN [6], logLTN [7], and Neural LP [60]. Rooted in more general, unguarded FOL, these frameworks highlight the computational complexity that GUARDNET’s principled restriction to GF is designed to overcome.¹
- **Probabilistic Frameworks:** NeurASP [62], DeepProbLog [30], and DeepStochLog [57]. These are prominent probabilistic logic programming systems that integrate neural networks with probabilistic inference, representing an orthogonal approach to neural-symbolic integration.

¹Note that to enable these baselines to run on large datasets for Table 2, we applied stochastic mini-batching. As discussed in Sec. 5.1, this engineering compromise avoids OOM errors but severely degrades logical reasoning performance.

- **Standard KG Embedding Models:** TransE [9], RotatE [46], and ComplEx [51]. These widely adopted embedding methods learn logical patterns *implicitly* through geometric formulations (e.g., RotatE for composition, ComplEx for symmetry), providing a clear baseline to quantify the performance gains from explicit logical reasoning.

Evaluation Metrics. For both completion tasks, we report **Hits@K** ($K \in \{1, 10, 100\}$) and **Mean Reciprocal Rank (MRR)**. GUARDNET ranks candidates by fuzzy satisfaction degree $\llbracket \alpha \rrbracket \in [0, 1]$ in descending order; geometric ontology embedding baselines rank by ascending distance, and KG embedding models use their respective scoring functions.

Implementation Details. All experiments were run on NVIDIA RTX 4090 GPUs using PyTorch 2.0 with CUDA 11.8. For all models, the AdamW optimizer [29] was used with an early stopping strategy (patience of 15 epochs) based on validation MRR.

For training, we employ **negative sampling** with $\omega = 64$ corrupted samples per positive instance. Within our fuzzy semantics, observed facts $P(a, b)$ are assigned target truth value 1, while corrupted facts $P(a', b)$ or $P(a, b')$ are assigned target truth value 0. This discriminative signal complements the axiom-based losses in Eq. 9: while $\mathcal{L}_{\text{fidelity}}$ and \mathcal{L}_{gen} enforce logical consistency across the embedding space, the negative sampling loss provides explicit supervision for individual facts. We adopt a self-adversarial margin-based loss (margin $\delta = 1.0$) to sharpen the discrimination boundary. L2 regularization ($\lambda_{\text{reg}} = 10^{-5}$) is applied to all embeddings.

All hyperparameters were selected via grid search with sensitivity analysis on validation sets. For baselines, search spaces included learning rates $\in \{10^{-4}, 2 \times 10^{-4}, 5 \times 10^{-4}\}$, embedding dimensions $d \in \{100, 200, 400\}$, and batch sizes $\in \{256, 512, 1024\}$. For GUARDNET, the optimal configuration is: learning rate 2×10^{-4} , embedding dimension $d = 200$, batch size 512 (SNOMED CT) or 1024 (others), predicate MLPs with two 256-unit ReLU layers. For fuzzy operators, we use steepness $s = 10$ and bias $b_0 = -0.5$ in the Sigmoidal implication, and aggregation exponent $p = 2$ for quantifiers. The hybrid loss weights are $\lambda = 0.7$ (fidelity-generalization balance) and $\gamma = 0.01$ (confidence penalty). These values were determined through systematic sensitivity analysis; the details can be found at <https://github.com/anonymous-ai-researcher/vldb2026>.

To guarantee reproducibility, all reported metrics are the **mean \pm standard deviation over 5 independent runs** with different random seeds. Paired t-tests between GUARDNET and the strongest baseline confirm statistical significance ($p < 0.01$) across all benchmarks and metrics, except for H@1 on SNOMED CT ($p < 0.05$).

5.1 Scalability and System Efficiency (RQ1)

We begin by addressing the most critical bottleneck in NeSy reasoning: physical scalability. To answer **RQ1**, we conducted a stress test comparing GUARDNET against representative NeSy baselines (LTN [6], logLTN [7], and Neural LP [60]) on increasingly large subsets of SNOMED CT.

The results (**Figure 5**) offer definitive validation of our polynomial complexity advantages. Crucially, we must distinguish between computational feasibility and logical validity. While baselines can technically run on smaller-scale ABox tasks (as reported

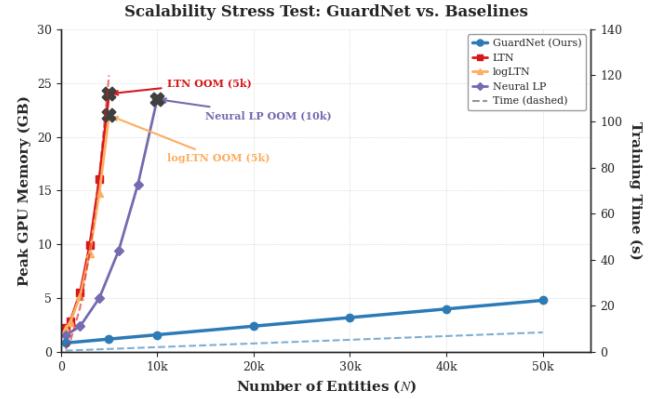


Figure 5: We compare the peak GPU memory usage (solid lines) and training time (dashed) as the number of entities N increases. The “OOM Wall”: Standard NeSy models LTN (red) and logLTN (orange) hit the memory limit (24GB) at just 5K entities due to their $O(N^2)$ tensor broadcasting bottleneck. Even matrix-based approaches like NEURAL LP (purple) fail at 10K. The Guarded Advantage: In contrast, GUARDNET (blue) maintains a linear memory footprint $O(|\mathcal{E}|)$ via topological indexing, scaling effortlessly to 50K+ entities.

in Table 2) by employing aggressive stochastic mini-batching, this strategy fundamentally breaks the global dependency chains required for logical inference. As shown in Table 2, even with such engineering compromises, NeSy baselines completely fail (DNF) on large-scale TBox reasoning tasks such as SNOMED CT and GO, while achieving only moderate performance on the smaller PPI datasets (e.g., LTN attains MRR of 0.285 on Yeast PPI, far below GUARDNET’s 0.405). **Figure 5** reveals the root cause: when forced to perform *rigorous* full grounding to preserve logic, these models hit a quadratic memory wall at just 5K entities. In sharp contrast, GUARDNET maintains linear scalability without sacrificing logical fidelity, achieving the unique combination of SOTA accuracy and linear efficiency.

5.2 Comparative Performance (RQ2)

Having established physical feasibility, we evaluate the predictive performance of GUARDNET on full-scale datasets. Table 2 presents the “Grand Slam” benchmark results, comparing our model against 15 state-of-the-art baselines across TBox (Concept Subsumption) and ABox (Link Prediction) tasks.

GUARDNET demonstrates consistent superiority across all datasets and reasoning tasks. On the massive **SNOMED CT** ontology, GUARDNET attains an MRR of 0.125, surpassing the strongest geometric baseline TRANSBox (0.119) and Box²EL (0.114), while the leading GNN baseline NBFNet lags significantly behind (0.055) due to its inability to capture strict logical entailment in deep hierarchies. This advantage extends to the **GO**, where GUARDNET achieves an MRR of 0.133, outperforming TransBox (0.126) by a notable margin, while NeSy baselines like LTN fail to complete (DNF) on this scale.

The primary reason for this success lies in our framework’s **high logical fidelity**. Unlike geometric models that approximate logical

Table 2: Overall KBC Performance Comparison. We evaluate GUARDNET against SOTA baselines across four benchmarks spanning TBox reasoning (concept subsumption) and ABox reasoning (link prediction). GUARDNET (Product t-norm + Sigmoidal Reichenbach implication) achieves the best performance across all metrics. DNF: Did Not Finish (OOM/Timeout after 72h).

Model / Configuration	SNOMED CT (377K) TBox Reasoning (Concept Subsumption)				GO (44K)				Yeast PPI (110K edges) ABox Reasoning (Link Prediction)				Human PPI (75K edges)					
	H@1		H@10		MRR		H@1		H@10		MRR		H@1		H@10		MRR	
	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
GUARDNET (Prod+Sig)	5.1±.1	28.3±.4	70.5±.3	.125±.00	5.5±.2	29.8±.3	73.4±.2	.133±.00	30.5±.4	60.2±.5	91.1±.3	.405±.00	29.2±.5	57.9±.6	88.9±.4	.388±.00		
Geometric Ontology Embedding Models																		
ELEM	2.1±.1	20.1±.6	38.9±.5	.078±.00	2.4±.1	23.8±.5	43.4±.4	.089±.00	20.5±.6	45.0±.8	74.8±.6	.301±.00	18.2±.8	40.1±1.1	69.7±.8	.268±.00		
Box ² EL	3.4±.2	25.5±.6	68.1±.5	.114±.00	3.9±.2	26.5±.5	70.9±.4	.120±.00	27.1±.5	55.1±.6	86.9±.5	.368±.00	25.5±.6	52.3±.8	82.7±.6	.346±.00		
TransBox	3.6±.2	26.8±.5	69.2±.4	.119±.00	4.1±.2	27.8±.4	71.8±.3	.126±.00	28.5±.5	57.5±.6	88.5±.4	.385±.00	26.8±.6	54.8±.7	85.2±.5	.365±.00		
Graph Neural Networks																		
CompGCN	0.8±.1	8.1±.5	20.3±.4	.041±.00	1.2±.1	9.9±.4	24.1±.3	.052±.00	23.5±.6	50.5±.7	78.3±.5	.328±.00	21.5±.7	45.2±.8	72.1±.6	.302±.00		
GRAIL	1.0±.1	9.8±.5	24.0±.5	.050±.00	1.5±.1	11.9±.5	28.1±.4	.063±.00	25.8±.6	52.6±.7	80.2±.5	.345±.00	23.8±.7	48.0±.8	75.5±.6	.321±.00		
NBFNet	1.2±.1	10.5±.4	25.8±.4	.055±.00	1.8±.1	13.0±.4	30.2±.3	.071±.00	26.5±.5	53.5±.6	81.1±.5	.351±.00	24.5±.6	49.1±.7	76.9±.6	.332±.00		
Expressive NeSy Models (Un.guarded, FOL-based)																		
LTN (Prod+Sig)	DNF			DNF			DNF			DNF			DNF			DNF		
logLTN (Prod+Sig)	DNF			DNF			DNF			DNF			DNF			DNF		
Neural LP	DNF			DNF			DNF			DNF			DNF			DNF		
Probabilistic Frameworks																		
NeurASP	DNF			DNF			DNF			DNF			DNF			DNF		
DeepProbLog	DNF			DNF			DNF			DNF			DNF			DNF		
DeepStochLog	DNF			DNF			DNF			DNF			DNF			DNF		
Standard KG Embedding Models																		
TransE	0.2±.1	2.1±.3	8.7±.2	.018±.00	0.3±.1	2.8±.3	12.3±.2	.023±.00	15.5±.4	38.0±.7	64.9±.5	.238±.00	14.0±.5	35.0±.8	60.7±.6	.219±.00		
RotatE	0.3±.1	3.2±.4	11.8±.3	.025±.00	0.4±.1	4.1±.4	16.7±.3	.032±.00	18.5±.5	42.6±.8	68.1±.6	.268±.00	16.0±.6	38.3±.9	64.3±.7	.245±.00		
ComplEx	0.3±.1	2.6±.3	10.1±.3	.022±.00	0.4±.1	3.5±.3	14.9±.3	.028±.00	17.0±.5	40.5±.8	66.4±.6	.252±.00	15.5±.6	36.1±.9	62.8±.7	.232±.00		

constraints through spatial relationships (e.g., box containment or vector rotation), GUARDNET’s loss function is derived directly from the **GF semantics**. This forces the learned embeddings to maintain strict consistency with the ontological structure, effectively pruning the search space to valid neighborhoods. This is particularly evident in precision-oriented metrics, where GUARDNET excels by avoiding the false positives that plague purely statistical methods.

On ABox reasoning tasks (Yeast and Human PPI), the dramatic performance differential with standard KG embedding models confirms the fundamental importance of incorporating structured logical reasoning. GUARDNET achieves an MRR of 0.405 on Yeast PPI, significantly outperforming the leading GNN baseline NBFNet (0.351) by over 15%. This confirms that the *Guarded Inductive Bias* is not limited to taxonomic reasoning but also generalizes robustly to inductive link prediction, effectively capturing multi-hop logical dependencies that standard message-passing schemes miss.

5.3 Ablation and Mechanism Analysis (RQ3)

To isolate the source of the performance gains, we perform a comprehensive component analysis and investigate the model’s sensitivity to operator choices.

Hybrid Domain Strategy. Table 3 reveals that the **Hybrid Domain Strategy** is critical for preventing representational collapse.

• **Preventing Emptiness:** The base model (Row 2) relying solely on the Herbrand domain (Δ_H) suffers from an 82% collapse rate on SNOMED CT, as the model learns to predict logical falsity to vacuously satisfy axioms. Introducing **Atom Witnesses** (Δ_A ,

Row 3) provides the necessary static anchors, drastically reducing collapse to 12% and nearly doubling the MRR.

- **Enforcing Decisiveness:** The addition of **Guarded Witnesses** (Δ_G , Row 4) further refines the decision boundaries, handling dynamic triggering of rules. Finally, the full model with **Latent Generalization** (Δ_{latent} , Row 5) eliminates collapse almost entirely (<1%), proving that continuous sampling is essential for learning universal logical principles.

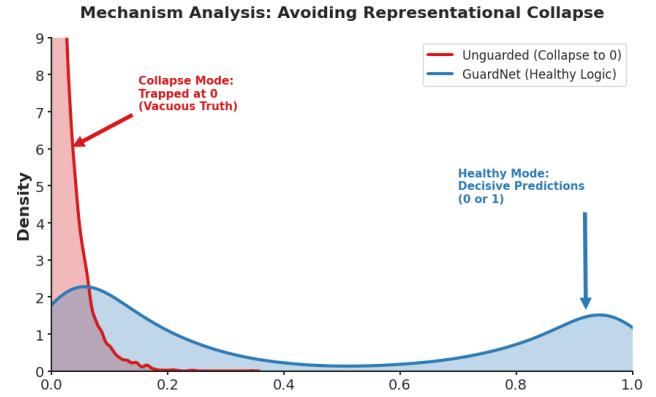


Figure 6: Unguarded model (red line) collapses to vacuous truths (trapped at 0), whereas GUARDNET (blue line) maintains a healthy bimodal distribution with decisive predictions (near 0 or 1), effectively avoiding the collapse modes.

Table 3: Comprehensive Component Analysis (RQ3). We systematically dissect GUARDNET’s performance gains. Col. (Collapse Rate): % of predictions stuck in trivial states. Low is good. RSR (Rule Satisfaction Rate): Average truth value of axioms. High is good. Note: On PPI (Graph tasks), Collapse Rate is consistently low (<1%) for all GuardNet variants due to explicit positive edges, but RSR steadily improves, showing that the model learns to satisfy logical rules more strictly. Percentage improvements are computed relative to Row 2 (GuardNet Base) for TBox tasks and Row 1 (LTN Optimized) for ABox tasks.

Model Configuration	Domain	SNOMED CT						GO						Yeast PPI						Human PPI								
		H@1	H@10	H@100	MRR	Col.	RSR	H@1	H@10	H@100	MRR	Col.	RSR	H@1	H@10	H@100	MRR	RSR	H@1	H@10	H@100	MRR	RSR	H@1	H@10	H@100	MRR	RSR
1. LTN (Optimized) (Product + Sigmoidal) (Δ_H)	Unguarded	DNF (OOM)						DNF (OOM)						17.5	42.5	70.1	.285	.82	16.0	39.8	65.4	.265	.80					
2. GuardNet (Base) (Product + Sigmoidal) (Δ_H)	Guarded	1.2 <i>(High Collapse due to Emptiness)</i>	11.5	24.8	.055	82%	.65	1.4 <i>(High Collapse due to Emptiness)</i>	12.8	26.5	.062	79%	.68	22.1 +26% +17%	49.8 +14% +14%	80.5 +17% +11%	.335	.91	20.5 +28% +16%	46.5 +15% +15%	75.2 +17% +11%	.312	.89					
3. + Atom Witness (+ Static Anchors) ($\cup\Delta_A$)	Hybrid	3.8 +216% +104%	23.5 +148%	61.5 +91%	.105 ↓70pt +.27	12% +91%	.92	4.2 +200% +101%	25.8 +146%	65.4 +85% ↓69pt +.25	.115 +24%	10% +10%	.93	27.5 +24% +10%	55.2 +7.8% +12%	86.8 +12% +.05	.375	.96	26.2 +27% +15%	53.5 +9.7% +9.7%	82.5 +13% +13%	.352	.94					
4. + Guarded Witness (+ Dynamic Anchors) ($\cup\Delta_G$)	Hybrid	4.8 +26%	26.8 +14%	68.2 +11%	.118 ↓12%	5% ↓7pt	.96	5.1 +21%	28.2 +9.3%	71.5 +9.3%	.126 +9.6% ↓6pt	4% +.04	.97	29.8 +8.3% +7.1%	59.1 +4.2%	90.5 +5.6% +.02	.396	.98	28.5 +8.7% +6.5%	57.0 +6.0% +6.0%	87.5 +7.3% +7.3%	.378	.96					
5. Full GUARDNET (+ Latent Generalization) ($\cup\Delta_{\text{latent}}$)	Hybrid	5.1 +6.2%	28.3 +5.6%	70.5 +3.4%	.125 +5.9%	<1% ↓4pt	.99 +.03	5.5 +7.8%	29.8 +5.7%	73.4 +2.7%	.133 +5.6% ↓3pt	<1% +.02	.99 +.02	30.5 +2.3% +1.9%	60.2 +0.7% +0.7%	91.1 +0.7% +2.3%	.405 +.02	.99 +.01	29.2 +2.4% +1.6%	57.9 +1.6% +1.6%	88.9 +2.6% +2.6%	.388 +.02	.98 +.02					

Mechanism Analysis: Visualizing Collapse. To visually confirm the elimination of representational collapse, we plotted the density distribution of predicted truth values on the test set in **Figure 6**.

The **Unguarded** baseline (red curve) exhibits a pathological distribution heavily skewed towards 0. This empirically validates our theoretical concern that without proper grounding, the model minimizes loss by learning “vacuous truths” (i.e., predicting all atoms as False to trivially satisfy implications like $\text{False} \rightarrow \dots$). In contrast, GUARDNET (blue curve) displays a healthy **bimodal distribution** with distinct peaks near 0 (False) and 1 (True). This indicates that our Hybrid Domain Strategy successfully forces the model to make decisive, semantically meaningful predictions, effectively avoiding the collapse modes that plague standard NeSy approaches.

Sensitivity to Fuzzy Operators. Finally, **Figure 7** highlights the pivotal role of the **Sigmoidal Implication**. While the choice of T-norm (rows) shows marginal variance, the implication operator (columns) is decisive. Standard R-implications (Goguen) frequently lead to vanishing gradients (“Dead Zones”), causing models to fail (DNF) or underperform on large datasets. The Sigmoidal Implication (Red Box) consistently resolves this, providing the robust gradient signal required for deep reasoning architectures to converge.

6 CONCLUSIONS AND FUTURE WORK

In this work, we resolved the longstanding dichotomy between logical expressivity and computational scalability in neural-symbolic learning. By reinterpreting the **Guarded Fragment** of full FOL not as a restriction, but as a powerful topological inductive bias, we fundamentally restructured the computational graph of reasoning: transforming intractable global search into efficient, neighborhood-constrained lookups. GUARDNET demonstrates that rigorous logical constraints need not be computational bottlenecks; on the contrary, the “guard” acts as a semantic index that guides neural optimization, enabling linear scalability ($O(|\mathcal{E}|)$) on massive ontologies like SNOMED CT where prior methods fail. Furthermore, our Hybrid

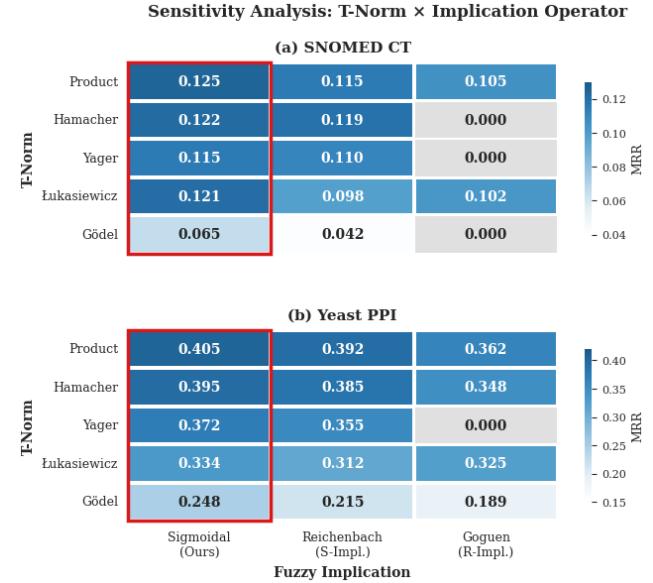


Figure 7: The heatmap isolates the impact of fuzzy operators on MRR. While the choice of T-norm (rows) has marginal variance, the choice of Implication (columns) is decisive.

Domain Strategy successfully reconciles data fidelity with generalization, effectively preventing representational collapse.

For future work, we envision two strategic directions:

- **In-Database Reasoning:** Integrating GUARDNET’s sparse grounding kernels directly into graph database engines (e.g., logic-aware stored procedures) to support real-time, zero-shot logical query answering without external model export.
- **Beyond Static Graphs:** Extending the guarded mechanism to *Temporal Guarded Logic* to handle dynamic knowledge graphs where validity intervals act as temporal guards.

REFERENCES

- [1] Ralph Abboud, İsmail İlkan Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. BoxE: A Box Embedding Model for Knowledge Base Completion. In *Advances in Neural Information Processing Systems 33 (NeurIPS'20)*.
- [2] Hajnal Andréka, István Németi, and Johan van Benthem. 1998. Modal Languages and Bounded Fragments of Predicate Logic. *J. Philos. Log.* 27, 3 (1998), 217–274.
- [3] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, Midori A Harris, David P Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanne Lewis, John C Matese, Joel E Richardson, Martin Ringwald, Gerald M Rubin, and Gavin Sherlock. 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature Genetics* 25, 1 (2000), 25–29.
- [4] Franz Baader, Sebastian Brandt, and Carsten Lutz. 2005. Pushing the EL Envelope. In *Proc. IJCAI'05*. Professional Book Center, 364–369.
- [5] Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. 2017. *An Introduction to Description Logic*. Cambridge University Press.
- [6] Samy Badreddine, Artur S. d’Avila Garcez, Luciano Serafini, and Michael Spranger. 2022. Logic Tensor Networks. *Artif. Intell.* 303 (2022), 103649.
- [7] Samy Badreddine, Luciano Serafini, and Michael Spranger. 2023. logLTN: Differentiable Fuzzy Logic in the Logarithm Space. *CoRR* abs/2306.14546 (2023).
- [8] Tarek R. Besold, Artur S. d’Avila Garcez, Sebastian Bader, Howard Bowman, Pedro M. Domingos, Pascal Hitzler, Kai-Uwe Kühnberger, Luís C. Lamb, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas, Hoifung Poon, and Gerson Zaverucha. 2021. Neural-Symbolic Learning and Reasoning: A Survey and Interpretation. In *Neuro-Symbolic Artificial Intelligence: The State of the Art*. Frontiers in Artificial Intelligence and Applications, Vol. 342. IOS Press, 1–51.
- [9] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, 2787–2795.
- [10] Alonzo Church. 1936. An Unsolvable Problem of Elementary Number Theory. *American Journal of Mathematics* 58 (1936), 345.
- [11] William W. Cohen, Fan Yang, and Kathryn Mazaitis. 2020. TensorLog: A Probabilistic Database Implemented Using Deep-Learning Infrastructure. *J. Artif. Intell. Res.* 67 (2020), 285–325.
- [12] Artur S. d’Avila Garcez and Luís C. Lamb. 2023. Neurosymbolic AI: the 3rd wave. *Artif. Intell. Rev.* 56, 11 (2023), 12387–12406.
- [13] Artur S. d’Avila Garcez, Krysia Broda, and Dov M. Gabbay. 2002. *Neural-symbolic learning systems - foundations and applications*. Springer.
- [14] Michelangelo Diligenti, Marco Gori, and Claudio Saccà. 2017. Semantic-based regularization for learning and inference. *Artif. Intell.* 244 (2017), 143–165.
- [15] Jonathan Fürst, Mauricio Fadel Argerich, and Bin Cheng. 2023. VersaMatch: Ontology Matching with Weak Supervision. *Proc. VLDB Endow.* 16, 6 (2023), 1305–1318.
- [16] Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. 2016. *Deep Learning*. MIT Press.
- [17] Erich Grädel. 1999. On The Restraining Power of Guards. *J. Symb. Log.* 64, 4 (1999), 1719–1742.
- [18] Víctor Gutiérrez-Basulto and Steven Schockaert. 2018. From Knowledge Graph Embedding to Ontology Embedding? An Analysis of the Compatibility between Vector Space Representations and Rules. In *Proc. KR'18*. AAAI Press, 379–388.
- [19] William L. Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. 2018. Embedding Logical Queries on Knowledge Graphs. In *Advances in Neural Information Processing Systems 31 (NeurIPS'18)*, 2030–2041.
- [20] Mathias Jackermeier, Jiaoyan Chen, and Ian Horrocks. 2024. Dual Box Embeddings for the Description Logic \mathcal{EL}^+ . In *Proc. WWW'24*. ACM, 2250–2258.
- [21] Erich-Peter Klement, Radko Mesiar, and Endre Pap. 2000. *Triangular Norms*. Trends in Logic, Vol. 8. Springer.
- [22] Adrian Kochsiek and Rainer Gemulla. 2021. Parallel Training of Knowledge Graph Embedding Models: A Comparison of Techniques. *Proc. VLDB Endow.* 15, 3 (2021), 633–645.
- [23] Maxat Kulmanov, Wang Liu-Wei, Yuan Yan, and Robert Hoehndorf. 2019. EL Embeddings: Geometric Construction of Models for the Description Logic \mathcal{EL}^+ . In *Proc. IJCAI'19*. ijcai.org, 6103–6109.
- [24] Brenden M. Lake and Marco Baroni. 2018. Generalization without Systematicity: On the Compositional Skills of Sequence-to-Sequence Recurrent Networks. In *Proc. ICML'18 (Proceedings of Machine Learning Research)*, Vol. 80. PMLR, 2879–2888.
- [25] Na Li, Thomas Bailleux, Zied Bouraoui, and Steven Schockaert. 2024. Ontology Completion with Natural Language Inference and Concept Embeddings: An Analysis. *CoRR* abs/2403.17216 (2024).
- [26] Na Li, Zied Bouraoui, and Steven Schockaert. 2019. Ontology Completion Using Graph Convolutional Networks. In *Proc. ISWC'19 (LNCS)*, Vol. 11778. Springer, 435–452.
- [27] Xueling Lin, Haoyang Li, Hao Xin, Zijian Li, and Lei Chen. 2020. KBPearl: A Knowledge Base Population System Supported by Joint Entity and Relation Linking. *Proc. VLDB Endow.* 13, 7 (2020), 1035–1049.
- [28] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *Proc. AAAI'15*. AAAI Press, 2181–2187.
- [29] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *Proc. ICLR'19*. OpenReview.net.
- [30] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. 2018. DeepProbLog: Neural Probabilistic Logic Programming. In *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, 3753–3763.
- [31] Emanuele Marconato, Stefano Teso, Antonio Vergari, and Andrea Passerini. 2023. Not All Neuro-Symbolic Concepts Are Created Equal: Analysis and Mitigation of Reasoning Shortcuts. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*.
- [32] Gary Marcus. 2018. Deep Learning: A Critical Appraisal. *CoRR* abs/1801.00631 (2018).
- [33] Gary Marcus. 2020. The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence. *CoRR* abs/2002.06177 (2020).
- [34] Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, and Marco Gori. 2019. Integrating Learning and Reasoning with Deep Logic Models. In *Proc. ECML-PKDD'19 (LNCS)*, Vol. 11907. Springer, 517–532.
- [35] Connor Pryor, Charles Dickens, Eric Augustine, Alon Albala, William Yang Wang, and Lise Getoor. 2023. NeuPSL: Neural Probabilistic Soft Logic. In *Proc. IJCAI'23*. ijcai.org, 4145–4153.
- [36] Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. 2007. ProbLog: A Probabilistic Prolog and Its Application in Link Discovery. In *Proc. IJCAI'07*. 2462–2467.
- [37] Hongyu Ren, Hanjun Dai, Bo Dai, Xinyun Chen, Michihiro Yasunaga, Haitian Sun, Dale Schuurmans, Jure Leskovec, and Denny Zhou. 2021. LEGO: Latent Execution-Guided Reasoning for Multi-Hop Question Answering on Knowledge Graphs. In *Proc. ICML'21 (Proceedings of Machine Learning Research)*, Vol. 139. PMLR, 8959–8970.
- [38] Hongyu Ren, Hanjun Dai, Bo Dai, Xinyun Chen, Denny Zhou, Jure Leskovec, and Dale Schuurmans. 2022. SMORE: Knowledge Graph Completion and Multi-hop Reasoning in Massive Knowledge Graphs. In *Proc. KDD'22*. ACM, 1472–1482.
- [39] John Alan Robinson and Andrei Voronkov (Eds.). 2001. *Handbook of Automated Reasoning (in 2 volumes)*. Elsevier and MIT Press.
- [40] Tim Rocktäschel and Sebastian Riedel. 2017. End-to-end Differentiable Proving. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 3788–3800.
- [41] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323, 6088 (1986), 533–536.
- [42] Manfred Schmidt-Schauss and Gert Smolka. 1991. Attributive Concept Descriptions with Complements. *Artif. Intell.* 48, 1 (1991), 1–26.
- [43] Kent A. Spackman, Keith E. Campbell, and Roger A. Côté. 1997. SNOMED RT: a reference terminology for health care. In *Proc. AMIA'97*. AMIA.
- [44] Umberto Straccia. 2001. Reasoning within Fuzzy Description Logics. *J. Artif. Intell. Res.* 14 (2001), 137–166.
- [45] Ye Sun, Lei Shi, and Yongxin Tong. 2025. eXpath: Explaining Knowledge Graph Link Prediction with Ontological Closed Path Rules. *Proc. VLDB Endow.* 18, 9 (2025), 2818–2830.
- [46] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *Proc. ICLR'19*. OpenReview.net.
- [47] Damian Szklarczyk, Annika L. Gable, David Lyon, Alexander Junge, Stefan Wyder, Jaime Huerta-Cepas, Milan Simonovic, Nadezhda T. Doncheva, John H. Morris, Peer Bork, Lars Juhl Jensen, and Christian von Mering. 2019. STRING v11: protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Res.* 47, Database-Issue (2019), D607–D613.
- [48] Zhenwei Tang, Tilman Hinnerichs, Xi Peng, Xiangliang Zhang, and Robert Hoehndorf. 2022. \mathcal{FALCON} : Sound and Complete Neural Semantic Entailment over \mathcal{ALC} Ontologies. *CoRR* abs/2208.07628 (2022).
- [49] Zhenwei Tang, Shichao Pei, Xi Peng, Fuzhen Zhuang, Xiangliang Zhang, and Robert Hoehndorf. 2022. Joint Abductive and Inductive Neural Logical Reasoning. *CoRR* abs/2205.14591 (2022).
- [50] Komal K. Teru, Etienne G. Denis, and William L. Hamilton. 2020. Inductive Relation Prediction by Subgraph Reasoning. In *Proc. ICML'20 (Proceedings of Machine Learning Research)*, Vol. 119. PMLR, 9448–9457.
- [51] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *Proc. ICML'16 (JMLR Workshop and Conference Proceedings)*, Vol. 48. JMLR.org, 2071–2080.
- [52] Alan M. Turing. 1937. On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.* s2-42, 1 (1937), 230–265.
- [53] Emile van Krieken, Erman Acar, and Frank van Harmelen. 2022. Analyzing Differentiable Fuzzy Logic Operators. *Artif. Intell.* 302 (2022), 103602.
- [54] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. Composition-based Multi-Relational Graph Convolutional Networks. In *Proc. ICLR'20*. OpenReview.net.

- [55] Kai Wang, Yuwei Xu, and Siqiang Luo. 2024. TIGER: Training Inductive Graph Neural Network for Large-scale Knowledge Graph Reasoning. *Proc. VLDB Endow.* 17, 10 (2024), 2459–2472.
- [56] Po-Wei Wang, Priya L. Donti, Bryan Wilder, and J. Zico Kolter. 2019. SATNet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In *Proc. ICML’19 (Proceedings of Machine Learning Research)*, Vol. 97. PMLR, 6545–6554.
- [57] Thomas Winters, Giuseppe Marra, Robin Manhaeve, and Luc De Raedt. 2022. DeepStochLog: Neural Stochastic Logic Programming. In *Proc. AAAI’22*. AAAI Press, 10090–10100.
- [58] Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. 2018. A Semantic Loss Function for Deep Learning with Symbolic Knowledge. In *Proc. ICML’18 (Proceedings of Machine Learning Research)*, Vol. 80. PMLR, 5498–5507.
- [59] Zezhong Xu, Yincen Qu, Wen Zhang, Lei Liang, and Huajun Chen. 2024. InBox: Recommendation with Knowledge Graph using Interest Box Embedding. *Proc. VLDB Endow.* 17, 13 (2024), 4641–4654.
- [60] Fan Yang, Zhilin Yang, and William W. Cohen. 2017. Differentiable Learning of Logical Rules for Knowledge Base Reasoning. In *Advances in Neural Information Processing Systems 30 (NIPS’17)*. 2319–2328.
- [61] Hui Yang, Jiaoyan Chen, and Uli Sattler. 2025. TransBox: \mathcal{EL}^{++} -closed Ontology Embedding. In *Proc. WWW’25*. ACM, 22–34.
- [62] Zhun Yang, Adam Ishay, and Joohyung Lee. 2020. NeurASP: Embracing Neural Networks into Answer Set Programming. In *Proc. IJCAI’20*. ijcai.org, 1755–1762.
- [63] Jiasheng Zhang, Deqiang Ouyang, Shuang Liang, and Jie Shao. 2025. Towards Pattern-aware Data Augmentation for Temporal Knowledge Graph Completion. *Proc. VLDB Endow.* 18, 10 (2025), 3573–3586.
- [64] Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. 2020. Learning Hierarchy-Aware Knowledge Graph Embeddings for Link Prediction. In *Proc. AAAI’20*. AAAI Press, 3065–3072.
- [65] Zhanqiu Zhang, Jie Wang, Jiajun Chen, Shuiwang Ji, and Feng Wu. 2021. ConE: Cone Embeddings for Multi-Hop Reasoning over Knowledge Graphs. In *Advances in Neural Information Processing Systems 34 (NeurIPS’21)*. 19172–19183.
- [66] Zhanqiu Zhang, Jie Wang, Jieping Ye, and Feng Wu. 2022. Rethinking Graph Convolutional Networks in Knowledge Graph Completion. In *Proc. WWW’22*. ACM, 798–807.
- [67] Fernando Zhapa-Camacho and Robert Hohndorf. 2024. Lattice-Preserving \mathcal{ALC} Ontology Embeddings. In *Proc. NeSy’24 (LNCS)*, Vol. 14979. Springer, 355–369.
- [68] Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal A. C. Xhonneux, and Jian Tang. 2021. Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*. 29476–29490.

A THEORETICAL PROOFS AND ANALYSIS

In this appendix, we provide the formal definitions, rigorous mathematical proofs, and detailed complexity analysis for the theorems presented in the main text. We begin by summarizing the notation used throughout the paper.

A.1 Summary of Notation

Table 4: Summary of Mathematical Notation

Symbol	Description
$\mathcal{K} = (\mathcal{T}, \mathcal{A})$	Knowledge Base composed of TBox \mathcal{T} and ABox \mathcal{A}
$\Sigma = (\mathcal{P}, C)$	Signature containing predicates \mathcal{P} and constants C
$\Delta, \Delta_{\text{core}}$	The domain of interpretation and the Core Domain subset
$\text{ext}(\alpha)$	The extension of a guard atom α (set of satisfying tuples in \mathcal{K})
$N, \mathcal{E} $	Number of entities ($ \Delta $) and number of edges/facts in \mathcal{K}
d, L	Embedding dimension and number of MLP layers
\mathbf{e}_c	Vector embedding of constant $c \in \mathbb{R}^d$
$[\phi] \in [0, 1]$	Fuzzy truth value of formula ϕ
$T_P(a, b)$	Product T-norm (Fuzzy Conjunction): $a \cdot b$
$I_{\sigma R}(a, b)$	Sigmoidal Reichenbach Implication
\mathcal{A}_p^\vee	Universal Quantifier (Truth) Aggregator (as defined in Main Text Eq. 2)
\mathcal{E}_p^\vee	Error Aggregator (Loss counterpart used in proofs, $\mathcal{E} = 1 - \mathcal{A}$)

A.2 Formal Definitions of Fuzzy Operators

To support the soundness proof, we explicitly define the properties of the differentiable fuzzy operators used in GUARDNET. These definitions are strictly aligned with the formulations in Section 3.2 of the main text.

DEFINITION 4 (PRODUCT T-NORM). *The conjunction of two fuzzy truth values $a, b \in [0, 1]$ is defined as:*

$$T_P(a, b) = a \cdot b \quad (14)$$

Property: $T_P(a, b) = 1$ if and only if $a = 1$ and $b = 1$.

DEFINITION 5 (SIGMOIDAL REICHENBACH IMPLICATION). *The standard Reichenbach implication is $I_R(a, c) = 1 - a + a \cdot c$. The **Sigmoidal Reichenbach** implication used in Eq. 7 is defined as:*

$$I_{\sigma R}(a, c) = \text{scale}(\sigma(s \cdot (I_R(a, c) + b_0))) \quad (15)$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. The function $\text{scale}(\cdot)$ linearly maps the range $[\sigma(s \cdot b_0), \sigma(s \cdot (1 + b_0))]$ to the unit interval $[0, 1]$ to ensure strict boundary satisfaction:

$$\text{scale}(y) = \frac{y - \sigma(s \cdot b_0)}{\sigma(s \cdot (1 + b_0)) - \sigma(s \cdot b_0)} \quad (16)$$

Soundness Condition: Due to the strict monotonicity of $\sigma(\cdot)$ and the linear scaling, minimizing the loss $\mathcal{L} = 1 - I_{\sigma R}(a, c)$ to 0 strictly implies $I_R(a, c) = 1$.

DEFINITION 6 (GENERALIZED MEAN ERROR AGGREGATOR). *In the main text (Eq. 2), we defined the **Truth Aggregator** $\mathcal{A}^\vee(z)$ for the universal quantifier. For the purpose of the loss analysis in proofs, it is convenient to define the corresponding **Error Aggregator** \mathcal{E}_p^\vee over the set of instance losses $\mathbf{l} = \{l_1, \dots, l_k\}$ (where $l_i = 1 - z_i$):*

$$\mathcal{E}_p^\vee(\mathbf{l}) = \left(\frac{1}{k} \sum_{i=1}^k l_i^p \right)^{\frac{1}{p}} \quad (17)$$

Relation to Main Text: The two concepts are duals: $\mathcal{A}^\vee(z) = 1 - \mathcal{E}_p^\vee(1 - z)$.

Crucial Property (Strict Zero-Condition): For any $p \geq 1$, $\mathcal{E}_p^\vee(\mathbf{l}) = 0$ if and only if $l_i = 0$ for all $i \in \{1, \dots, k\}$.

A.3 Proof of Soundness (Theorem 1)

THEOREM (SOUNDNESS OF GUARDNET). Let \mathcal{K} be a knowledge base consisting of a finite set of GF formulas. If a GUARDNET model trained on \mathcal{K} achieves $\mathcal{L}_{\text{fidelity}}(\theta) = 0$, then the learned fuzzy interpretation \mathcal{G}_θ satisfies all axioms on the Core Domain Δ_{core} . That is, for every axiom $\phi \in \mathcal{K}$ and every grounding $\bar{c} \in \Delta_{\text{core}}^{|\text{FV}(\phi)|}$, we have $\llbracket \phi \rrbracket_{\mathcal{G}_\theta, \bar{c}} = 1$.

PROOF. The proof proceeds by analyzing the structure of the Fidelity Loss function and the properties of the constituent operators.

Step 1: Decomposition of Fidelity Loss. Recall the definition of the Fidelity Loss from Eq. 10:

$$\mathcal{L}_{\text{fidelity}} = \underbrace{\mathcal{E}_p^{\vee} (\{\mathcal{L}_{\text{instance}}(\phi; \bar{c}) \mid \phi \in \mathcal{K}, \bar{c} \in \Delta_{\text{core}}\})}_{\text{Axiom Term}} + \underbrace{\sum \mathcal{L}_{\text{witness}}}_{\text{Witness Term}} \quad (18)$$

Note that here we express the total loss using the error aggregator \mathcal{E}_p^{\vee} defined in Definition A.2, which sums the individual instance losses.

Step 2: Strict Zero-Property. The loss components are non-negative:

- The instance loss $\mathcal{L}_{\text{instance}} \in [0, 1]$ because fuzzy truth values are in $[0, 1]$.
- The aggregator \mathcal{E}_p^{\vee} (Generalized Mean of errors) preserves non-negativity.

Therefore, $\mathcal{L}_{\text{fidelity}} = 0$ implies that every term in the aggregation must be exactly 0. Specifically, using the **Strict Zero-Condition** of the generalized mean:

$$\mathcal{E}_p^{\vee}(\mathbf{L}) = 0 \iff \forall l \in \mathbf{L}, l = 0 \quad (19)$$

This implies that for every axiom $\phi \in \mathcal{K}$ and every witness tuple $\bar{c} \in \Delta_{\text{core}}^{|\text{FV}(\phi)|}$:

$$\mathcal{L}_{\text{instance}}(\phi; \bar{c}) = 0 \quad (20)$$

Step 3: From Loss to Truth Value. The instance loss is defined as $\mathcal{L}_{\text{instance}} = 1 - \llbracket \phi \rrbracket_{\mathcal{G}_\theta, \bar{c}}$ (Eq. 7). Consequently:

$$1 - \llbracket \phi \rrbracket_{\mathcal{G}_\theta, \bar{c}} = 0 \implies \llbracket \phi \rrbracket_{\mathcal{G}_\theta, \bar{c}} = 1 \quad (21)$$

Step 4: Satisfaction of Logical Structure. We must verify that $\llbracket \phi \rrbracket = 1$ implies the logical satisfaction of the formula structure. Let $\phi = \forall \bar{x}(\alpha \rightarrow \psi)$. The truth value is calculated via the Sigmoidal Reichenbach implication $I_{\sigma R}$. From Definition A.2, $I_{\sigma R} = 1$ implies (via the inverse of the scaling and sigmoid functions) that the underlying Reichenbach score $I_R(\llbracket \alpha \rrbracket, \llbracket \psi \rrbracket) = 1$.

- $I_R(a, b) = 1 - a + ab$.
- $1 - a + ab = 1 \implies a(1 - b) = 0$.
- This equation holds if $a = 0$ (premise false, vacuous truth) or $b = 1$ (conclusion true).

This aligns exactly with the classical logic definition of material implication ($P \rightarrow Q \equiv \neg P \vee Q$).

Conclusion. Since $\mathcal{L}_{\text{fidelity}} = 0$ forces the instance loss to be 0 for all axioms grounded over all tuples in the Core Domain Δ_{core} , it follows that the learned interpretation \mathcal{G}_θ assigns a truth value of 1 to all such groundings. Thus, the model satisfies the knowledge base \mathcal{K} on Δ_{core} . \square

A.4 Proof of Training Complexity (Theorem 2)

THEOREM (TRAINING COMPLEXITY). Let \mathcal{K} be a knowledge base with m guarded axioms $\{\forall \bar{x}_i(\alpha_i \rightarrow \psi_i)\}_{i=1}^m$. Let d be the embedding dim, L the MLP layers, B the batch size. A single training iteration has:

- **Time Complexity:** $O(m \cdot |E|_{\max} \cdot L \cdot d^2)$
- **Space Complexity:** $O(|\Delta_{\text{core}}| \cdot d + |\mathcal{P}| \cdot L \cdot d^2 + B \cdot |E|_{\max})$

where $|E|_{\max} = \max_i |\text{ext}(\alpha_i)|$.

PROOF. We analyze the computational graph constructed by GUARDNET for a single training step (forward and backward pass).

A.4.1 Time Complexity Analysis. Consider the processing of a single guarded axiom $\phi = \forall \bar{x}(\alpha(\bar{x}) \rightarrow \psi(\bar{x}))$.

1. **Guarded Indexing (Lookup):** GUARDNET first retrieves the valid variable bindings (tuples) that satisfy the guard atom α . This corresponds to querying the ABox for the extension $\text{ext}(\alpha)$. The number of such tuples is $|\text{ext}(\alpha)|$. This is a sparse lookup operation (gathering indices). Cost: $O(|\text{ext}(\alpha)|)$.

2. **Neural Grounding (Feature Construction):** For each tuple $\bar{a} \in \text{ext}(\alpha)$, we retrieve the embeddings of the constants. If $\bar{a} = (c_1, \dots, c_k)$, we fetch e_{c_1}, \dots, e_{c_k} . Cost: $O(|\text{ext}(\alpha)| \cdot k \cdot d)$. Since k (arity) is small and constant, this is $O(|\text{ext}(\alpha)| \cdot d)$.

3. **Predicate Evaluation (MLP Forward Pass):** The truth value of the body ψ involves evaluating predicate MLPs. Let's assume ψ contains constant number of atoms. For each atom, we pass the concatenated embeddings through an MLP. * Input dimension: $k \cdot d$. * Hidden dimension: d (for simplicity). * Number of layers: L . * Matrix-Vector Multiplication cost per layer: $\mathbf{W} \in \mathbb{R}^{d \times d}, \mathbf{h} \in \mathbb{R}^d$. Cost is d^2 . * Total MLP cost per tuple: $O(L \cdot d^2)$. * Total cost for all guard-satisfying tuples: $O(|\text{ext}(\alpha)| \cdot L \cdot d^2)$.

4. **Aggregation:** We aggregate the $|\text{ext}(\alpha)|$ truth values using the p -mean. This is a linear reduction. Cost: $O(|\text{ext}(\alpha)|)$.

Total Time Complexity: Summing over m axioms in the knowledge base (or a batch of axioms):

$$T_{\text{total}} = \sum_{i=1}^m O(|\text{ext}(\alpha_i)| \cdot L \cdot d^2) \quad (22)$$

Let $|E|_{\max} = \max_i |\text{ext}(\alpha_i)|$. Then:

$$T_{\text{total}} \approx O(m \cdot |E|_{\max} \cdot L \cdot d^2) \quad (23)$$

In sparse graphs, $|E|_{\max} \approx O(N)$ (linear in entities) or $O(|\mathcal{E}|)$ (linear in edges), whereas unguarded approaches require $O(N^2)$ or $O(N^k)$.

A.4.2 Space Complexity Analysis. We analyze the memory footprint required to store parameters and intermediate activation maps for backpropagation.

1. **Model Parameters:** * **Entity Embeddings:** We store embeddings for all constants in the Core Domain. Cost: $O(|\Delta_{\text{core}}| \cdot d)$. * **Predicate Networks:** For each of the $|\mathcal{P}|$ predicates, we store an MLP with L layers of size $d \times d$. Cost: $O(|\mathcal{P}| \cdot L \cdot d^2)$.

2. **Intermediate Activations (The Bottleneck):** To compute gradients, we must store the activations for every node in the computational graph *for the current batch*. * Unlike unguarded approaches that instantiate a tensor of size N^k (all possible tuples), GUARDNET only instantiates the computational graph for valid guard extensions. * For a batch of axioms B (or processing the whole KB if full-batch), the number of active computation nodes is proportional to the number of satisfying tuples. * Number of tuples $\leq B \cdot |E|_{\max}$. * For each tuple, we store activations of size $O(L \cdot d)$. * Cost: $O(B \cdot |E|_{\max} \cdot L \cdot d)$. Assuming L, d are constants relative to graph size, the scaling factor is $B \cdot |E|_{\max}$.

Total Space Complexity:

$$S_{\text{total}} = \underbrace{O(|\Delta_{\text{core}}| \cdot d)}_{\text{Embeddings}} + \underbrace{O(|\mathcal{P}| \cdot L \cdot d^2)}_{\text{Weights}} + \underbrace{O(B \cdot |E|_{\max})}_{\text{Activations}} \quad (24)$$

Comparison: * **Unguarded (LTN/NeuralLP):** The activation term is $O(N^2)$ (for binary predicates), which dominates and causes OOM.

* **GuardNet:** The activation term is $O(|E|_{\max})$. For sparse graphs, $|E| \ll N^2$, often $|E| \approx cN$ (linear).

Thus, the complexity is linear w.r.t the number of edges/entities in sparse KGs. \square