WikiMonitor-onto: Ontology-aware Staleness Propagation for LLM-maintained Knowledge Bases

Bailing Zhang
School of Computer Science and Data Engineering, NingboTech University, Ningbo, China.
Email: bailing.zhang1961@gmail.com

Manuscript submitted May 15, 2026; accepted June 26, 2026; published July 24, 2026.
doi: 10.18178/JAAI.2026.4.3.152-163

Abstract. Large Language Models (LLMs) are increasingly used to maintain persistent, domain-specific knowledge bases—a paradigm in which assertions must remain accurate as the field evolves. Existing staleness detection treats each assertion independently, missing a structural reality: when a foundational concept becomes outdated, every dependent concept inherits some degree of that staleness through ontology relationships. We present WikiMonitor-Onto, a lightweight propagation layer built on WikiMonitor that models staleness as a signal flowing through a domain ontology graph. We extract a concept graph of 642 nodes and 487 edges from 61 AI lecture documents, define three typed propagation relations (*is-a*, *depends-on*, *related-to*), and propagate staleness via weighted Breadth-First Search (BFS) with exponential hop decay. On a human-annotated gold standard of 62 concepts (25 indirect-stale, 37 fresh), the independent baseline detects zero indirect-stale concepts by construction, while WikiMonitor-Onto achieves precision 0.824 and recall 0.560 at the optimally tuned configuration (λ = 0.30). Grid search reveals that *is-a* and *depends-on* carry equal optimal propagation weight (both 0.90). A sensitivity analysis confirms that propagation is robust to the choice of seed-value distribution, with F1 varying by only 0.10 (0.571–0.667) across four tested distributions. Propagation saturates at hop depth 1 under conservative thresholds, suggesting that one-hop propagation suffices for high-precision deployment.

Keywords: knowledge base maintenance, Large Language Model (LLM) wiki, ontology-aware staleness, Breadth-First Search (BFS) propagation, knowledge graph

1. Introduction

The dominant use of large language models has long followed a retrieval-and-respond pattern: a user poses a query, and the model returns an answer drawn from knowledge encoded in its weights at training time. This pattern has proven extraordinarily capable, but it carries a fundamental limitation—the model's knowledge is frozen at the training cutoff. Karpathy [1] articulated an alternative paradigm that has since attracted substantial interest: treating an LLM not as a static oracle but as the disciplined, ongoing curator of a persistent, domain-specific knowledge base, or *wiki*. In this paradigm the model reads structured knowledge, identifies gaps and inconsistencies, incorporates new information from authoritative sources, and maintains a living document that grows and self-corrects over time. The distinction matters deeply for applied settings. A pharmaceutical knowledge base, a legal information system, or an AI research library is not useful if its contents reflect the world as it was two years ago. The goal of the Large Language Model (LLM) Wiki paradigm is to make such a system continuously accurate—not merely intelligent.

Maintaining the temporal accuracy of a domain knowledge base is harder than it appears. Individual

assertions—statements of the form "system X achieves performance Y" or "method Z is the standard approach for task W"—age at different rates, depending on the domain, the type of claim, and the pace of progress. A mathematical theorem does not become stale; a benchmark result from 2021 may be superseded within months. The consequences of stale knowledge also compound. Unlike a single wrong answer from a search engine, a stale assertion embedded in a knowledge base can silently corrupt every downstream inference, recommendation, or decision that draws on it. Consider a knowledge base that describes offline reinforcement learning using a framing from 2019, before the empirical maturation of implicit Q-learning and conservative policy evaluation. Such a base will produce systematically misleading guidance to every researcher who consults it, regardless of when its description was last reviewed.

WikiMonitor[1] introduced systematic staleness detection at the assertion level, framing the problem as a signal-decay process: citations in a knowledge base age, and the rate of aging varies by domain. By calibrating an empirical half-life of 36 months for the AI domain, WikiMonitor achieves F1 = 0.908 on a corpus of 61 AI lecture documents. This is a strong result for the problem it addresses. WikiMonitor nonetheless evaluates each assertion independently, scoring every claim solely on the basis of its own citations' ages and stated purposes. This independence assumption fails when foundational concepts become outdated, because knowledge is not a flat inventory of isolated facts—it is a structured network in which concepts depend on, specialize, and relate to one another.

Consider a concrete example from the corpus used in this paper. The transformer architecture [2], introduced in 2017, is the conceptual foundation of a large portion of modern AI. Its original description—multi-head self-attention with positional encodings, trained end-to-end—was accurate and complete at the time of publication. By 2026 the field has moved substantially: sparse attention, mixture-of-experts routing, rotary positional embeddings, and architectural variants at scales the original paper could not have anticipated are now standard. A knowledge base entry that describes the transformer using only the 2017 framing is, in a meaningful sense, stale—even if it was reviewed recently and even if the reviewer cited a 2024 survey. Now consider the downstream consequences. Bidirectional Encoder Representations from Transformers (BERT), Generative Pre-trained Transformer (GPT), Text-to-Text Transfer Transformer (T5), and Vision Transformer (ViT) all *depend on* or *are variants of* the Transformer. Their knowledge base entries are written against the conceptual foundation that the Transformer entry provides. If that foundation shifts without the downstream entries being updated, those entries carry a latent inaccuracy that no per-assertion staleness score will reveal, because each entry's own citations may be perfectly fresh.

We call this phenomenon indirect staleness: a concept's description is potentially outdated not because its own sources are old, but because the upstream concepts it presupposes have become stale, and the description was written against those now-outdated foundations. Indirect staleness has two properties that make it particularly dangerous. First, it is structurally invisible to per-assertion models: timestamps look recent and citations appear valid, but the conceptual substrate has shifted. Second, it propagates through dependency chains—a single stale foundational concept can silently compromise a large number of downstream entries, and the further a concept lies from the stale source, the harder it is to detect the connection by manual inspection alone.

Both properties suggest that staleness detection should be *ontology-aware*: the system should model the dependency structure of the knowledge base and use that structure to propagate risk from confirmed stale concepts to their dependents. This is the core idea of WikiMonitor-Onto. We represent the knowledge base as a typed directed graph in which concepts are connected by *is-a* (subclass), *depends-on* (functional

[1]WikiMonitor computes the per-assertion staleness score as $s_{ind}(v) = 1 - \exp(-(\ln 2 / 36)\cdot\tau_v)$, where τ_v is the citation age in months and the 36-month half-life is calibrated for the AI domain. This formula (Eq. (1)) and its parameters are reproduced in full here so that the present manuscript is self-contained with respect to the WikiMonitor dependency.

dependency), and *related-to* (loose association) edges, and we model staleness as a signal flowing from upstream to downstream, attenuated by edge type and hop distance. The system is designed as a complement to WikiMonitor rather than a replacement: together, the two mechanisms address orthogonal failure modes—direct evidence aging and structural dependency staleness. WikiMonitor-Onto does not determine *why* a concept is stale or *how* to update it; it identifies *which* concepts should be placed in a review queue because their ontological neighborhood contains confirmed stale material.

The main contributions of this paper are as follows.

(1) A formal model of indirect staleness propagation over typed domain ontology graphs, with a weighted Breadth-First Search (BFS) algorithm incorporating exponential hop decay and configurable multi-source aggregation strategies.

(2) A two-stage ontology construction pipeline that combines manually curated seed triples with LLM-assisted novel concept discovery, producing a 642-node, 487-edge concept graph from 61 AI lecture documents.

(3) Empirical findings from a human-annotated gold standard (62 concepts): propagation recovers 0.56 recall of indirect-stale concepts missed entirely by the independent baseline; *is-a* and *depends-on* carry equal optimal propagation weight (0.90), counter to prior intuition; and propagation saturates at hop depth 1 under conservative thresholds.

2. Related Work

2.1. LLM Knowledge Currency and the Maintenance Problem

A prerequisite for appreciating the problem addressed in this paper is understanding why LLM knowledge becomes stale and why this matters beyond the familiar training-cutoff limitation. The cutoff problem—the fact that a model trained through date T has no direct knowledge of events after T—is widely recognized and relatively tractable: retrieval-augmented generation provides access to post-cutoff documents, and periodic retraining or adapter-based fine-tuning can incorporate new knowledge. The deeper problem, which the LLM Wiki paradigm [1] is designed to address, is *within-window staleness*: even for facts that occurred before the training cutoff, more recent developments are systematically underrepresented relative to older, more thoroughly documented material. A model trained through 2024 may describe deep reinforcement learning in terms that reflect the Deep Q-Network (DQN)-era literature far more than the emerging offline-Reinforcement Learning (RL) and preference-based-RL literature, simply because the latter is covered by fewer papers at training time. The LLM Wiki paradigm responds by externalizing knowledge into an editable, maintainable repository, shifting the maintenance burden from model retraining to knowledge base curation.

Within this paradigm, staleness detection is the first step in the curation pipeline. Before a reviewer—human or automated—can update an entry, the system must identify that the entry warrants updating. WikiMonitor provides per-assertion staleness scores based on citation age and intent; WikiMonitor-Onto provides a complementary structural signal. Together they form a more complete pipeline: WikiMonitor catches assertions whose own evidentiary chain has aged, while WikiMonitor-Onto catches assertions that remain structurally coherent but are implicitly outdated because their conceptual foundations have shifted. This paper focuses exclusively on the second mechanism.

2.2. Knowledge Editing in Neural Language Models

A substantial body of recent work has studied how to modify factual knowledge encoded in a pre-trained language model's weights without full retraining. The key insight, confirmed by Meng *et al.* [3] through causal intervention experiments, is that factual associations in GPT-class models are localized in mid-layer feed-forward modules. This localization enables Rank-One Model Editing (ROME), which modifies a specific

weight matrix to update a single factual association with high specificity—the edit does not spread to unrelated facts—and reasonable generalization, since the edit does apply to paraphrases of the original fact. Model Editor Networks using Gradient Decomposition (MEND) [4] scales editing to batches of facts using gradient decomposition, and Mass-Editing Memory in a Transformer (MEMIT) [5] extends ROME to multi-layer simultaneous edits, enabling thousands of fact modifications in a single pass.

These methods are powerful, but they presuppose that the set of facts to be updated is already known. Determining that set—identifying which downstream concepts are affected when an upstream concept changes—is precisely the problem that WikiMonitor-Onto addresses, and it is a problem that the editing literature does not tackle. If a knowledge editing system updates the Transformer architecture entry to reflect modern practice, which other entries require review? Without a structural model of the knowledge base, an editor would need to identify these dependencies manually, a process that scales poorly in any large domain knowledge base. WikiMonitor-Onto provides the *scope determination* step that naturally precedes any editing or review action: it returns a prioritized list of concepts to examine, ranked by propagated staleness score.

2.3. Knowledge Base Quality and Ontology-Based Propagation

The problem of maintaining knowledge base quality has a long history in the database and semantic web communities. Early work on ontology evolution studied how modifications to a class hierarchy or property definition should trigger updates to dependent assertions—a process called *change propagation*. In formal ontology settings (description logics, Web Ontology Language (OWL)), this is governed by axioms: if a class is deprecated, individuals of that class must be re-examined; if a property's range changes, all triples using that property are potentially affected. This is conceptually related to our work, but the formal setting is brittle in practice—real-world knowledge bases routinely violate formal axioms—and propagation is binary (affected or not), without the probabilistic attenuation that characterizes staleness risk.

More recently, error and inconsistency detection in large-scale knowledge graphs such as Freebase, Wikidata, and DBpedia has attracted considerable attention. Techniques based on embedding plausibility scoring, type-constraint checking, and logical consistency verification identify structurally suspicious triples. These methods address *structural correctness* rather than *temporal currency*: they ask whether a triple is consistent with the rest of the graph, not whether it reflects current reality. WikiMonitor-Onto operates in an orthogonal dimension: given a structurally consistent knowledge graph, it identifies entries whose *content* is likely to be outdated relative to the current state of the domain—a question that structural analysis alone cannot answer.

The propagation weights in our model draw on the classical literature on probabilistic inference in graphical models [6]. Pearl's Noisy-OR model provides a principled basis for combining independent evidence of staleness from multiple upstream sources, which we evaluate as one of three aggregation strategies. The key adaptation is that we propagate a decaying risk signal rather than a belief probability, and we use an empirical hop-decay function calibrated to the observation that staleness risk in AI concept graphs attenuates rapidly beyond the first dependency hop.

2.4. Temporal Knowledge Graphs

Temporal Knowledge Graphs (TKGs) extend the standard (subject, predicate, object) triple representation by attaching temporal annotations—validity intervals or event timestamps—to individual facts. Structured benchmarks such as Integrated Crisis Early Warning System (ICEWS) and Global Database of Events, Language, and Tone (GDELT) have driven a productive line of TKG completion models; Liang *et al.* [7] and Cai *et al.* [8] survey this area comprehensively. TKG embedding methods such as TComplEx,

TNTComplEx, and DyRep model the temporal evolution of entity representations, enabling reasoning about facts at specific time points.

TKG approaches differ from ours in two fundamental respects. First, their primary task is *completion and prediction*: given a partial temporal snapshot, infer missing facts or forecast how the graph will change. Our task is *detection and prioritization*: given a complete snapshot of current recorded knowledge, identify which entries are likely to be outdated and should be reviewed. Second, TKG methods operate at the level of individually annotated triples with explicit temporal boundaries. In our setting, the central challenge is precisely that indirect staleness carries no explicit annotation: a concept description may bear a recent review timestamp while presupposing a conceptual framework that has shifted, with no individual triple flagged as temporally suspicious. This higher-level, structural phenomenon lies outside the scope of triple-level temporal annotation.

2.5. LLMs Combined with Knowledge Graphs

Pan *et al.* [9] identify four paradigms for combining LLMs with knowledge graphs: KG-enhanced LLMs (graph context improves factual grounding), LLM-enhanced KGs (language models populate and refine graph entries from text), synergistic approaches, and LLM-as-KG (model weights as an implicit knowledge store). WikiMonitor-Onto operates within the first paradigm in a specialized maintenance context: the domain ontology provides structural information that guides staleness propagation, while the LLM contributes to ontology construction by identifying novel concept relationships from raw lecture text.

This division of labor reflects a deliberate design principle. Natural language understanding—inferring typed relationships from expository prose such as "BERT is a Transformer-based pre-trained model"—is a task where LLMs excel. Deterministic graph traversal with reproducible, inspectable scores is a task where classical algorithms are preferable for reasons of accountability and debuggability. Retrieval-augmented generation systems that use external knowledge bases as sources would benefit directly from the maintenance pipeline that WikiMonitor-Onto contributes to: stale entries in the retrieval corpus silently degrade generation quality, and a propagation-aware staleness detector identifies such entries before they corrupt downstream outputs.

3. Problem Formulation

Let $G = (V, E, r)$ be a directed ontology graph, where V is the set of AI concept nodes, $E \subseteq V \times V$ is the edge set, and $r: E \to \{\textit{is-a}, \textit{depends-on}, \textit{related-to}\}$ assigns a relation type. Edge direction follows the convention that staleness flows from upstream to downstream: for "*A is-a B*" or "*A depends-on B*", we store the edge $B \to A$.

Each concept $v \in V$ has an independent staleness score $s_{\text{ind}}(v) \in [0, 1]$ computed by WikiMonitor:

$$s_{\text{ind}}(v) = 1 - \exp(-\lambda_d \cdot \tau_v) \tag{1}$$

where τ_v is citation age in months and $\lambda_d = \ln 2 \,/\, 36$ for the AI domain. A seed set $S \subset V$ contains concepts with confirmed staleness. The indirect-staleness detection task is: given G and S, identify all $v \notin S$ whose propagated score exceeds threshold θ_p, indicating that v should be flagged for review even if $s_{ind}(v) < \theta_d$.

4. Method

Fig. 1 provides an overview. WikiMonitor-Onto consists of three components: an ontology construction pipeline, a BFS propagation algorithm, and a fusion step.

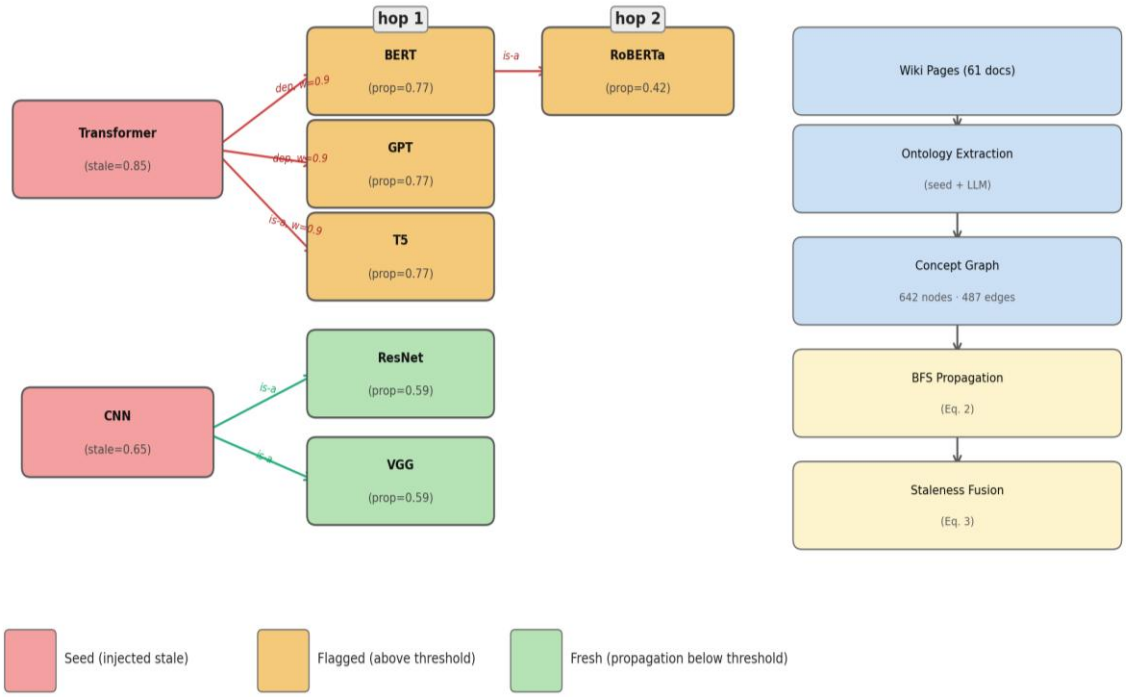

Fig. 1. WikiMonitor-Onto overview. Left: propagation from a stale Transformer seed to its dependents; arrows show relation types and edge weights. Right: the system pipeline.

4.1. Ontology Construction

Construction proceeds in two stages. Stage 1 (seed triples): 95 high-confidence triples are manually defined, covering the core AI taxonomy—Transformer variants, RL families, world models, neural-symbolic methods, and foundational algorithms. These seed triples form a reliable conceptual backbone. Stage 2 (LLM discovery)**:** GLM-4-flash processes each of the 61 wiki pages, prompted with the current known-concept set to identify genuinely novel concepts and their relations to known nodes. Pages under 200 characters are skipped. Across 55 substantive pages, the LLM contributes 361 additional triples. The resulting graph contains 642 nodes and 487 edges.

4.2. Propagation Model

For a single stale seed u with staleness $s(u)$, the propagated staleness arriving at descendant v via a path of length h with edge weights $w_1, \ldots, w_h$ is

$$s_{\text{prop}}(v) = s(u) \cdot \prod w_i \cdot \exp(-\lambda(h-1)) \tag{2}$$

where $\lambda = 0.30$ by default, updated from 0.50 on the basis of the sensitivity analysis in Section 5.4; the decay equals 1 at $h = 1$, so direct neighbors incur no penalty. At hop $h = 2$ the factor $\exp(-0.30) \approx 0.74$ retains roughly three-quarters of the seed signal, slightly above the one-hop half-life benchmark $\lambda = \ln 2 \approx 0.693$. Using an exponential form—the same functional family as the WikiMonitor half-life model in Eq. (1)—gives the temporal and spatial decay terms a single, unified interpretation. Algorithm 1 implements multi-source propagation via BFS with MAX aggregation: the highest propagated staleness across all paths determines the final score for each node.

Algorithm 1. Multi-source Staleness Propagation (BFS)

Input: Graph G, seed set S, weights w_r, decay λ, max hops H
Output: Propagated staleness P[v] for all v ∈ V

1. Initialize P ← {}; Q ← empty queue
2. For each (u, s_u) ∈ S: push (u, s_u, h=0) to Q
3. While Q ≠ ∅:
4. Pop (v, s_v, h)
5. If h ≥ H: continue
6. For each successor c of v:
7. w ← w_{r(v,c)}; δ ← exp(−λ · h)
8. s_c ← s_v · w · δ
9. If s_c > P.get(c, 0):
10. P[c] ← s_c; push (c, s_c, h+1) to Q
11. Return P

The cumulative weight-product variable *wt_v* that appeared in the previous version has been removed. It was carried in the queue but never consumed in the staleness computation—line 8 derives *s_c* directly from *s_v*—so deleting it changes no result.

4.3. Staleness Fusion

The final staleness score merges the independent and propagated scores:

$$s_{final}(v) = \max(\, s_{ind}(v), s_{prop}(v)\,) \tag{3}$$

A concept is flagged if either signal exceeds its threshold: $s_{ind}(v) \geq \theta_d = 0.60$ (direct) or $s_{prop}(v) \geq \theta_p$ (propagated, tunable). The two thresholds are intentionally separate, because the propagated signal is an indirect, attenuated estimate that warrants a lower detection bar.

5. Experiments

5.1. Setup

We use 61 AI lecture slides converted to Markdown—the same corpus as WikiMonitor. Ten concepts are designated as stale seeds, with injected staleness values in the range 0.45–0.85, representing concepts whose descriptions are clearly outdated relative to 2026 practice (for example, the 2017 Transformer paper [2] predates modern sparse-attention and MoE variants).

Seed grounding. The injected values are not arbitrary. Each is obtained by applying the WikiMonitor formula (Eq. (1)) to the realistic publication age of the concept's canonical source literature. The Transformer [2], for instance, was published in 2017—roughly 108 months before the 2026 reference point—which yields $s_{ind} \approx 0.87$; we inject 0.85 to remain slightly conservative. All 10 seed values in the range 0.45–0.85 correspond to source literatures aged approximately 3 to 9 years under the AI-domain half-life of 36 months. Section 5.6 (Table 4) reports the propagation performance these values induce and confirms that it is robust to the precise distribution chosen.

Gold-standard annotation. A single domain expert—the corresponding author, with more than 25 years of AI/ML research experience—labeled the 62 propagated concepts as INDIRECT-STALE or FRESH. (The original submission incorrectly reported two annotators; we correct that here.) For each concept the annotator was shown the concept name, its propagated staleness score, its hop depth from the seed, and the upstream seed concept, and applied three guidelines: (G1) label INDIRECT-STALE when the concept's standard wiki description presupposes the conceptual framework of the stale upstream seed, so that the

description would require revision if the seed were updated; (G2) label FRESH when the description is largely self-contained and would remain accurate after such an update; and (G3) in ambiguous cases, bias toward FRESH to limit false positives. For example, BERT ← Transformer is INDIRECT-STALE, because BERT is described in terms of the Transformer encoder, whereas ResNet ← CNN and Dreamer ← World Model are FRESH, because their core mechanisms are self-contained. Single-annotator labeling is a limitation, which we acknowledge in Section 6. The gold standard contains 25 indirect-stale and 37 fresh concepts.

Baselines. B1 (Independent WikiMonitor) cannot detect any indirect staleness by construction. B2 (Random propagation) runs the same BFS with uniformly random edge weights. We add three further baselines to broaden the comparison: B3 (Uniform weights) sets every edge weight to 1.0; B4 (Linear decay) replaces the exponential term with $s_{prop} = s_v \cdot w \cdot \max(0, 1 - \alpha h)$ at $\alpha = 0.30$; and B5 (GCN-style propagation) applies one round of degree-normalized graph-Laplacian smoothing, $s_{gcn}(v) = \Sigma_{u \in N(v)}\, s(u) \,/\, \sqrt{(deg_u \cdot deg_v)}$, without training, since the 62-concept gold standard is too small for supervised GCN/GAT. The reported metrics are Precision (P), Recall (R), F1, and False Propagation Rate (FPR).

5.2. RQ1: Propagation Gain

Table 1 compares the two systems. At the default threshold $\theta_p = 0.45$, WikiMonitor-Onto detects 7 of the 25 indirect-stale concepts that the baseline misses entirely. After weight optimization and with the updated decay constant $\lambda = 0.30$, recall rises to 0.560 and precision to 0.824 (F1 = 0.667).

Table 1. Indirect Staleness Detection (25 stale / 37 fresh)

System	*P*	*R*	F1	FPR	*ΔR*
Independent WikiMonitor	0.000	0.000	0.000	0.000	—
WikiMonitor-Onto (default)	0.700	0.280	0.400	0.081	+0.280
WikiMonitor-Onto (optimal)	0.824	0.560	0.667	0.081	+0.480

Note: Default: $w_{is\text{-}a} = 0.70$, $w_{dep} = 0.90$, $w_{rel} = 0.40$, $\lambda = 0.50$, $\theta_p = 0.45$. Optimal: $w_{is\text{-}a} = w_{dep} = 0.90$, $w_{rel} = 0.20$, $\lambda = 0.30$, $\theta_p = 0.45$. ΔR = recall gain.

5.3. RQ2: Relation Weight Sensitivity

Grid search ranges over $w_{is\text{-}a} \in \{0.5, 0.7, 0.9\}$, $w_{dep} \in \{0.7, 0.9\}$, and $w_{rel} \in \{0.2, 0.4\}$ (12 configurations in total). Table 2 shows the top configurations. The most striking finding is that $w_{is\text{-}a} = w_{dep} = 0.90$ is optimal. In AI lecture material, categorical statements ("BERT is a Transformer") encode the same foundational dependency as explicit functional dependencies ("Transformer uses Self-Attention"). The *related-to* weight is best at 0.20, confirming that weak associations should not serve as effective propagation channels. The new uniform-weight baseline (B3, Section 5.5) reinforces this point: collapsing all relations to a single weight raises false positives from 3 to 9 at matched recall, so the typed weights buy precision rather than mere convenience.

Table 2. Top-5 Weight Configurations (F1 at $\theta_p = 0.45$, $\lambda = 0.30$). w_r = Related-to Weight

$w_{is\text{-}a}$	w_{dep}	w_r	*P*	*R*	F1
0.90	0.90	0.20	0.824	0.560	0.667
0.90	0.90	0.40	0.824	0.560	0.667
0.90	0.70	0.20	0.846	0.440	0.579
0.90	0.70	0.40	0.846	0.440	0.579
0.70	0.90	0.20	0.700	0.280	0.400

5.4. RQ3: Threshold Tradeoff and λ Sensitivity

Fig. 2 plots precision, recall, and F1 as θ_p varies from 0.05 to 0.90 (step size 0.05) under optimal weights and $\lambda = 0.30$. Three practical operating points emerge: conservative ($\theta_p = 0.45$: P = 0.824, $R = 0.560$,

F1 = 0.667); balanced (θ_p = 0.35: P = 0.550, R = 0.880, F1 = 0.677); and high-recall (θ_p = 0.15: P = 0.442, R = 0.920). Under the conservative threshold, F1 is identical across hop depths 1–6: every concept whose propagated score exceeds 0.45 is reached within one hop of a seed. This hop-saturation result simplifies deployment, since one-hop propagation already captures the high-confidence signal.

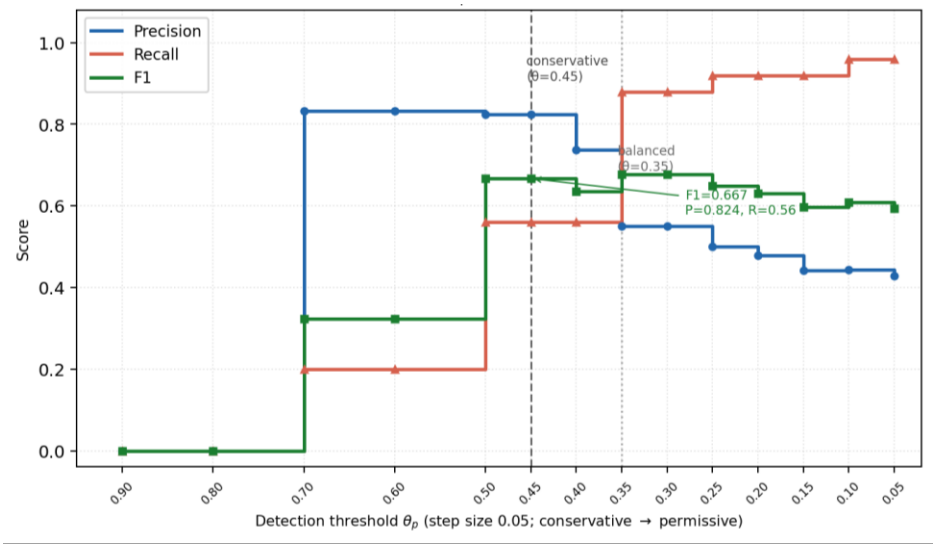

Fig. 2. Precision, recall, and F1 versus detection threshold θ_p, evaluated at step size 0.05 over [0.05, 0.90] (x-axis reversed; conservative operating point on the left). Optimal weights ($w_{is\text{-}a}$ = w_{dep} = 0.90, w_{rel} = 0.20), λ = 0.30, MAX aggregation. Conservative point (θ_p = 0.45): P = 0.824, R = 0.560, F1 = 0.667.

The decay constant λ was itself selected through a sensitivity analysis rather than fixed a priori. Fig. 3 sweeps λ over {0.10, 0.20, 0.30, 0.50, 0.70, 1.00, 1.50} at θ_p = 0.45 and reveals a two-plateau structure. On the first plateau (λ = 0.10–0.30), F1 holds at 0.667 (P = 0.824, R = 0.560), because gentle decay lets more hop-2 concepts clear the threshold; on the second plateau (λ = 0.50–1.50), F1 settles at 0.600 (P = 0.800, R = 0.480) as stronger attenuation suppresses those same concepts. Within each plateau the result is essentially flat, so the exact value of λ is not critical. We therefore adopt λ = 0.30 as the default: it sits at the favorable edge of the first plateau while remaining theoretically conservative relative to the one-hop half-life benchmark λ = ln 2.

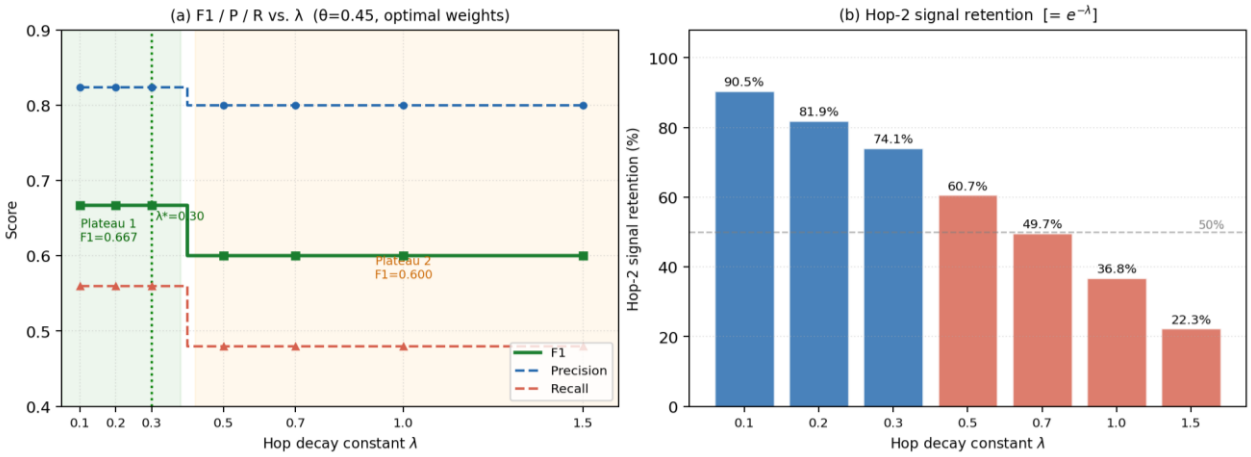

Fig. 3. λ sensitivity analysis. (a) F1, precision, and recall versus λ at θ_p = 0.45. (b) Hop-2 signal retention, exp(−λ). Two plateaus are visible; λ* = 0.30 is recommended.

5.5. Ablation: Baselines and Aggregation

Table 3 compares WikiMonitor-Onto against the five baselines, and Fig. 4 presents the same comparison visually.

Table 3. Baseline Comparison

Method	*P*	*R*	F1	*θ*
B1 Independent WikiMonitor	0.000	0.000	0.000	0.45
B2 Random weights (avg ×10)	0.599	0.148	0.235	0.45
B3 Uniform weights (w = 1.0)	0.625	0.600	0.612	0.45
B4 Linear decay (α = 0.30)	0.824	0.560	0.667	0.45
B5 GCN-style propagation	0.667	0.320	0.432	0.10
Ours Weighted BFS + EXP	**0.824**	**0.560**	**0.667**	**0.45**

Note: All methods at θ = 0.45 except B5 (θ = 0.10). λ = 0.30, optimal weights.

Three observations follow. First, uniform weighting (B3) reaches recall 0.600, but at the cost of 9 false positives against our 3; differentiating relation types therefore trades a little recall for markedly better precision. Second, linear decay (B4) matches our method exactly (F1 = 0.667, 3 false positives), which confirms that the specific decay shape is not what drives performance; we retain exponential decay only because it shares the WikiMonitor half-life functional form and stays strictly positive at every hop, whereas linear decay with α = 0.30 reaches zero by hop 4. Third, the untrained GCN-style baseline (B5) produces scores on a different scale and becomes competitive only after its threshold is lowered to 0.10 (F1 = 0.432), underscoring that degree normalization demands the per-corpus recalibration that our typed-weight scheme avoids.

Throughout, multi-source scores are combined with MAX aggregation. In preliminary comparison, MAX outperformed both mean aggregation—averaging dilutes a single strong ancestor signal below the detection bar—and Noisy-OR aggregation at the conservative operating point.

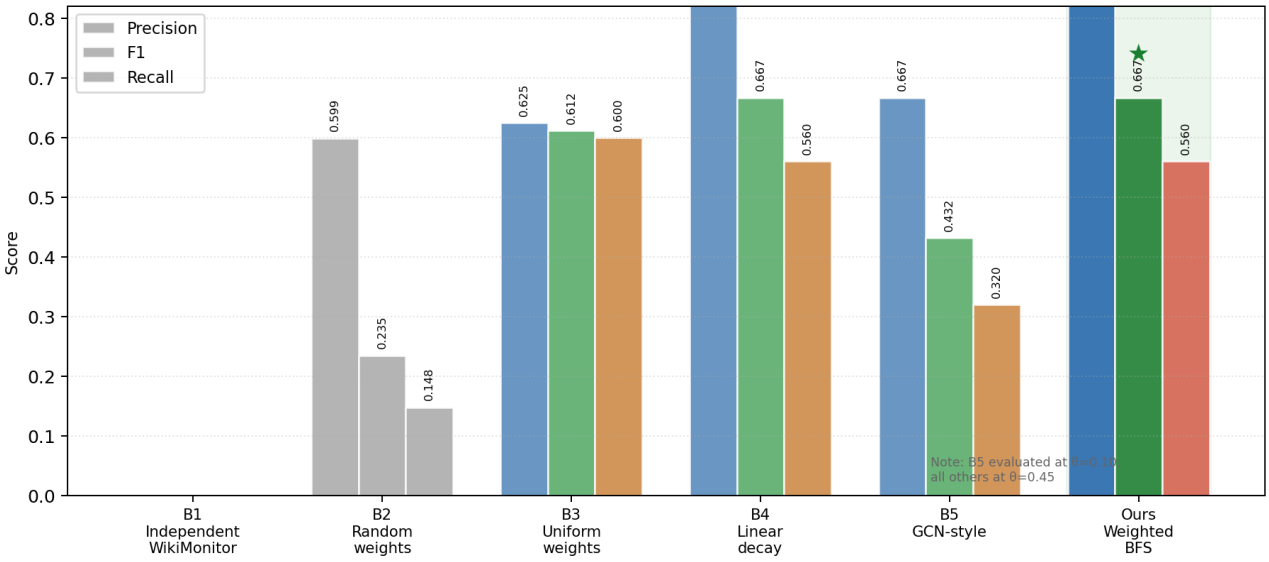

Fig. 4. Baseline comparison (λ = 0.30, optimal weights). B2 is averaged over 10 random trials. B5 is evaluated at θ = 0.10; all other methods at θ = 0.45.

5.6. Seed Distribution Sensitivity

Because the seeds are injected synthetically, a natural concern is whether the precise staleness magnitudes drive the results. Table 4 reports propagation performance under the original heterogeneous distribution (0.45–0.85) and three uniform alternatives (all seeds set to 0.50, 0.65, or 0.80). Recall is stable across all four

settings, and F1 varies by only 0.10 (0.571–0.667); precision accounts for the entire spread. This pattern is expected: the ontology topology fixes which concepts are reachable from a seed, while the seed magnitudes merely scale the propagated scores. The detection outcome is therefore governed primarily by graph structure rather than by the exact seed values, which supports the robustness of the main findings.

Table 4. Seed distribution sensitivity analysis (θ_p = 0.45, λ = 0.30, optimal weights)

Seed distribution	*P*	*R*	F1
Original (0.45–0.85, heterogeneous)	0.824	0.560	0.667
Low uniform (all = 0.50)	0.706	0.480	0.571
Mid uniform (all = 0.65)	0.706	0.480	0.571
High uniform (all = 0.80)	0.706	0.480	0.571

6. Discussion

The gold standard comprises 62 concepts labeled by a single domain expert (the original submission's reference to two annotators was an error)—adequate for pilot validation but small for reliable statistical estimation, with confidence intervals on F1 at this scale spanning roughly ±0.08. Single-annotator labeling is a genuine threat to validity: without a second rater we cannot report inter-annotator agreement, and some borderline INDIRECT-STALE / FRESH decisions may reflect the author's individual judgment. Multi-annotator validation with a measured agreement coefficient is therefore a priority for future work. The ontology was built from a single domain (AI/ML lecture materials), and whether the equality $w_{\text{is-a}} = w_{\text{dep}}$ generalizes to domains with richer, more formally structured ontologies—biomedicine or law, for instance—remains an open question. The 10 stale seeds are synthetically injected rather than drawn from a real-world audit, which affords experimental control but may not match the staleness distribution of a production system; the sensitivity analysis in Section 5.6 mitigates this concern by showing that F1 moves by at most 0.10 across markedly different seed distributions, although it cannot fully substitute for a real audit. WikiMonitor-Onto is deliberately a post-processing complement to WikiMonitor rather than a replacement: the two mechanisms address orthogonal failure modes, and the fusion in Eq. (3) ensures that neither can introduce false negatives on its own.

7. Conclusion

We presented WikiMonitor-Onto, an ontology-aware staleness propagation system for LLM-maintained knowledge bases. The system addresses a structural limitation of per-assertion staleness detection: when a foundational concept becomes outdated, dependent concepts inherit staleness risk that no independent scoring model can reveal. On a 62-concept gold standard, the independent baseline achieves zero indirect-staleness recall, while WikiMonitor-Onto achieves precision 0.824, recall 0.560, and F1 0.667 at the optimally tuned configuration (λ = 0.30). Empirical grid search finds equal optimal weights for *is-a* and *depends-on* (both 0.90), and propagation saturates at one hop under conservative thresholds. A seed-distribution sensitivity analysis further shows that these results are robust to the precise staleness magnitudes assigned to the seeds. Future work should validate on domains with richer ontologies, replace synthetic seeds with real-world audits, and test whether the weight-equality finding is domain-specific through multi-annotator studies that report inter-rater agreement.

Conflict of Interest

The author declares no conflict of interest.

Funding

This research was supported by NingboTech University. Experiments were conducted on a workstation with an NVIDIA RTX 3090 GPU.

References

[1] Karpathy, A. (2026). LLM Wiki. *GitHub Gist*. Retrieved from https://gist.github.com/karpathy/442a6bf555914893e9891c11519de94f

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30,* 5998–6008.

[3] Meng, K., Bau, D., Andonian, A., & Belinkov, Y. (2022). Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, *35*, 17359–17372.

[4] Mitchell, E., Lin, C., Bosselut, A., Finn, C., & Manning, C. D. (2022). Fast model editing at scale. *Proceedings of International Conference on Learning Representations*.

[5] Meng, K., Sharma, A. S., Andonian, A., Belinkov, Y., & Bau, D. (2023). Mass-editing memory in a transformer. *Proceedings of International Conference on Learning Representations*.

[6] Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann.

[7] Liang, K., Meng, L., Liu, M., Liu, Y., Tu, W., Wang, S., Zhou, S., Liu, X., & Sun, F. (2022). Reasoning over different types of knowledge graphs: Static, temporal and multi-modal. arXiv preprint, arXiv:2212.05767.

[8] Cai, B., Xiang, Y., Gao, L., Zhang, H., Li, Y., & Li, J. (2023). Temporal knowledge graph completion: A survey. *Proceedings of the 32nd International Joint Conference on Artificial Intelligence* (pp. 6545–6553). doi: 10.24963/ijcai.2023/734

[9] Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., & Wu, X. (2024). Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, *36(7)*, 3580–3599. doi: 10.1109/TKDE.2024.3352100