FourCorners: A Production Knowledge Graph Unifying Thailand's Legal System

Pawitsapak Akarajaradwong[1], **Sarana Nutanong**[2], **Chompakorn Chaksangchaichot**[1]
[1]VISAI AI, Thailand, [2]Vidyasirimedhi Institute of Science and Technology
{pawitsapaka_visai,sarana.n,chompakornc_pro}@vistec.ac.th

Abstract

Jurisdictionally bound domains, such as law, often lack standardized, machine-readable data formats, requiring foundational infrastructure before downstream applications can succeed. We present FourCorners, the first unified temporal knowledge graph for Thai legal data, integrating 4,118 laws (6,561 versions) with 87,394 Supreme Court decisions, updated daily. The graph encodes hierarchy, temporal versioning, cross-references, and sequential order, all extracted from unstructured official sources where no structured representation previously existed. A five-setting comparison on NitiBench-Tax isolates data infrastructure as the sole variable: graph-structured retrieval achieves Citation F1 of 0.812 versus 0.666 for practitioner-standard web search and 0.685 for flat vector retrieval, while searching a corpus 53× larger. Trace analysis of 820 agent-issued queries reveals that hierarchy traversal and cross-reference following, capabilities absent from generic retrieval, are exercised in 50% and 16% of questions, respectively. Our system demonstrates that structured modeling of hierarchy, temporal versioning, cross-references, and sequential order can overcome structural limitations of legal data published without standardized formats.

1 Introduction

Legal data has an inherent structure that distinguishes it from other domains. Statutes are organized **hierarchically**: constitutions govern codes, which contain provisions nested within books, parts, chapters, and sections. Laws evolve **temporally** through amendments; Thailand's Revenue Code alone has 83 tracked versions since 1939. Provisions form **referencing** citation networks within and across statutes. And **sequential** ordering carries legislative intent, as citations may reference provisions relatively by position. These four structures are fundamental to both human and AI-based systems for contextualizing legal understanding.

Beyond legal structural challenges, Thai legal documents also suffer from fragmentation across web pages. For instance, the country's 20 ministries maintain separate legal repositories with no unified access point, in formats ranging from HTML to PDF with no standardization. Even when data can be located and parsed, combining it meaningfully requires understanding structural relationships that no existing resource captures, requiring practitioners to manually browse multiple web pages simultaneously. According to our private interviews with Thai legal practitioners, most rely on general web searches to gather legal information.

Unlike the medical or financial domains, where knowledge can transfer across borders, legal data is inherently domestic, requiring each jurisdiction to build its own infrastructure. To address this problem, we developed a beta version of FourCorners in September 2025, a legal QA system powered by an agentic RAG pipeline that covers all currently active Thai legislation. Based on usage and user feedback during the beta release, we empirically confirmed that recognizing the inherent structure in Thai legal data is critical for AI to reason over different types of questions. Across 13,000 sessions from 3,500 users, negative feedback focused on responses that failed to address temporal resolution, relative hierarchical context resolution, and cross-reference reasoning. These are precisely the structural failures that flat retrieval cannot address. This motivated our shift from flat retrieval to a graph-structured data infrastructure that supports all four legal structures.

In this work, we present FourCorners, the first unified temporal knowledge graph for Thai legal data, integrating two authoritative government sources: the Office of the Council of State (OCS), which maintains Thailand's primary legislation including the constitution, codes, acts, and regulations; and the Supreme Court Judgments (DEKA), which provides judicial interpretations through case

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), pages 1808–1818

law. Our graph comprises over 725,000 nodes and 1.9 million structural edges, covering 4,118 laws across 6,561 tracked versions, with 87,394 Supreme Court decisions linked to the statutes they interpret through 236,911 cross-reference edges. A daily update pipeline maintains currency with official sources. We summarize our contributions:

1. **Infrastructure Blueprint**: The structured graph described above. To our knowledge, we are the first to extract and unify Thai legislation and court decisions with temporal versioning from unstructured sources
2. **Controlled Evaluation Setups**: A comparison on NitiBench-Tax (Akarajaradwong et al., 2025b) isolating data infrastructure as the sole variable, spanning from practitioner-standard web search, through flat and generic-graph retrieval (Edge et al., 2024), to a purpose-built legal graph, with identical agent architecture and leakage-controlled web search
3. **Design Principle Analysis**: Trace analysis of 820 agent-issued Cypher queries quantifying each graph capability's contribution to retrieval, showing that hierarchy traversal and cross-reference following are exercised in 50% and 16% of questions, respectively.

Together, FourCorners demonstrates the feasibility of constructing a temporally versioned, cross-referenced legal knowledge graph directly from unstructured sources at the jurisdictional scale.

2 Related Work

Legal Knowledge Graphs Existing legal knowledge graphs (KG) focus on case law: Chinese court judgments (Dong et al., 2021) and Indian case law (Jain et al., 2022) extract entities from judicial narratives but do not model statutory hierarchy or temporal evolution. For statutory law, European standards such as ELI/EUR-Lex (Filtz et al., 2021), LKIF (Hoekstra et al., 2007), LegalRuleML (Athan et al., 2013), and Akoma Ntoso (Palmirani and Vitali, 2011) define formal representations including temporal validity. Colombo et al. (2025) build an Italian legislative property graph from Akoma Ntoso XML using LLM-assisted extraction, making it the closest comparator to our work. De Martim (de Martim, 2025) proposes separating timeless law concepts from time-specific manifestations, which we implement and extend.

These standards define interchange formats for already-structured documents; they are not construction methods. No Thai legislation exists in Akoma Ntoso or any comparable markup; official sources provide only raw HTML, PDF, or text. Our work, therefore, addresses a different problem: *constructing a structured graph from unstructured sources, rather than converting between structured representations.* The design principles underlying our graph are well established in legal informatics (Cai et al., 2023); our contribution is their unified implementation for an underserved jurisdiction using unstructured sources, with empirical validation.

Graph-Augmented Retrieval Standard RAG (Lewis et al., 2021) retrieves flat text chunks, discarding document structure. Microsoft GraphRAG (Edge et al., 2024) and subsequent work (Peng et al., 2024; Sun et al., 2024; Pan et al., 2024) show that organizing information into knowledge graphs improves retrieval. Microsoft GraphRAG constructs generic and schema-free entity-relationship graphs from text via LLM extraction and community summarization. This captures co-occurrence structure but does not produce domain-specific edge types (temporal validity periods, hierarchical containment, or sequential document order) that legal queries require. Our proposed graph structure is purpose-built from domain knowledge rather than auto-extracted from text, and we compare the two approaches empirically in §4.

Thai Legal NLP NitiBench (Akarajaradwong et al., 2025b) exposed critical limitations of flat RAG for Thai legal QA, showing failure on questions requiring hierarchical context or cross-reference following. Follow-up work improves citation accuracy through LLM fine-tuning (Akarajaradwong et al., 2025a), curriculum-structured training for citation faithfulness (Tantapong et al., 2025), comparative QA framework development with mixed legal datasets (Hanwiboonwat et al., 2025), and hybrid retrieval optimization with reranking (Promwang and Boonma, 2025). However, none of these addresses the underlying data infrastructure gap. Our work is orthogonal: we build the structured data foundation that these downstream systems can leverage.

3 FourCorners

We present a unified temporal knowledge graph that integrates Thai legal data from authoritative sources and structures it according to four design principles capturing the inherent properties.

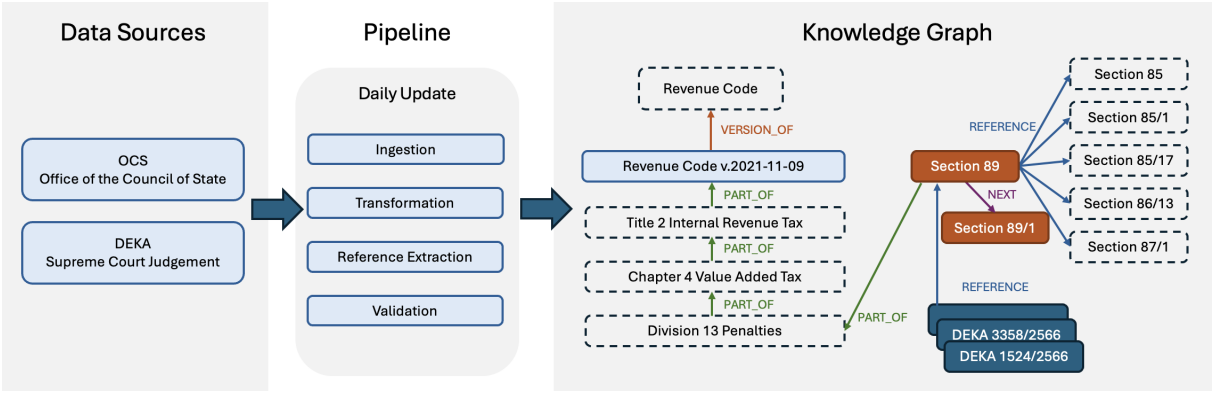

Figure 1: FourCorners system overview. **Left**: Two authoritative government sources providing primary legislation (OCS) and court decisions (DEKA). **Center**: Daily pipeline with incremental ingestion, structural parsing, LLM-assisted cross-reference extraction, and multi-stage validation. **Right**: Unified knowledge graph with four design principles encoded as edge types: **Temporal** (VERSION_OF with validity periods), **Hierarchy** (PART_OF containment), **Reference** (REFERENCES and cross-source REFERENCES_LAW), and **Sequential** (NEXT document order). Dashed borders denote timeless abstract entities; solid borders denote time-specific versions. The graph comprises 725K nodes and 1.9M structural edges.

3.1 Data Sources

Primary legislation defines the law; Supreme Court judgments (DEKA) establish how it is applied. Together, these two sources form the foundation of Thai legal practice: practitioners routinely consult both statutes and DEKA decisions to determine how a provision has been interpreted and enforced, making court judgments as essential as the written law itself. We integrate primary legislation from the Office of the Council of State (OCS) and judicial interpretations from the Supreme Court.

The OCS serves as Thailand's official repository for primary legislation, covering the constitution, codes, acts, emergency decrees, and subordinate legislation. Our graph incorporates 4,118 distinct laws (3,565 active) spanning 18 legal categories across 6,561 tracked versions (see Figure 4).

Our DEKA corpus includes 87,394 Supreme Court decisions spanning from 1920 to the present (see Figure 5). We extract 236,911 cross-reference edges linking decisions to statutory provisions using LLM-assisted extraction, enabling cross-source citation analysis, such as retrieving all Supreme Court decisions interpreting a particular provision or identifying which provisions are most contested in judicial practice.[1]

3.2 Knowledge Graph Design

Our graph implements four principles capturing the inherent structure of legal data. Figure 1 (right) visualizes these principles using a Revenue Code subgraph, and Table 1 maps graph components to each principle.

We separate *abstract* entities (timeless concepts like "the Criminal Code") from *instances* ("the Criminal Code as of January 1, 2020"), enabling both temporal traversal and point-in-time queries. **Hierarchy** edges form directed trees from components to parents across seven nesting levels (Book through Section), enabling context-aware retrieval. **Temporal** version edges carry validity timestamps, supporting point-in-time queries for any given date. **Reference** citation edges target timeless concepts rather than specific versions, enabling queries that follow citations across temporal boundaries and across sources. A sub-type, **Cross-source** edges, links citations across sources (i.e., DEKA decisions citing statutory provisions). **Sequential** order edges link consecutive components, preserving reading flow within each law. Additionally, each law carries an LLM-generated summary, enabling cross-corpus discovery: the agent can search summaries to identify relevant statutes across the full corpus before diving into individual sections.

Nodes split into two classes. *Abstract* nodes (`AbstractLaw`, `AbstractComponent`) represent timeless legal concepts such as "the Revenue Code" and serve as stable targets for citations. *Instance* nodes (`LawVersion`, `ComponentVersion`, `DekaDecision`) carry time-specific content and validity periods. Citations target abstract nodes so that references remain valid across amendments, while temporal queries resolve to instance nodes

[1]All graph statistics are as of April 2026.

Principle	Edge Type	Example (Revenue Code)
Temporal	VERSION_OF	Revenue Code v. 2021-11-09 (instance); 83 versions → Revenue Code (abstract)
Hierarchy	PART_OF	Sec 89 ∈ Div 13 ∈ Ch 4 ∈ Title 2
Reference	REFERENCES	Sec 89 → Sec 85, 85/1, 86/13 (cross-division)
Cross-source	REF._LAW	DEKA 3558/2566 → Sec 89 (118 decisions total)
Sequential	NEXT	Sec 89 → Sec 89/1 (reading order)

Table 1: Design principles instantiated with Revenue Code examples. All data verified against the live graph.

for a given date. Hierarchy spans seven nesting levels, from the outermost Book/Part down to the atomic Section/Clause: Book/Part ⊃ Title ⊃ Chapter ⊃ Division ⊃ ChapterNum ⊃ Provision ⊃ Section/Clause. Not every law uses every level; the graph preserves whichever levels appear in the source text via PART_OF edges.

3.3 Data Transformation Pipeline

A daily pipeline ingests documents incrementally from official government APIs, transforms raw data into graph structures, and loads them into a self-hosted Neo4j database.

Transformation Thai legal texts present three compounding challenges that prevent purely rule-based extraction of cross-references. First, unstandardized law titles: the same statute appears under different names across sources (abbreviated, vernacular, or formal). Second, range citations such as "Sections 10 through 13" require the cited law's section inventory to expand, because Thai provisions use ordinal suffixes (Sec 10, 10 *bis*, 10 *ter*) rather than integer sequences. Third, mixed numeral systems (Thai and Arabic) and amendment-suffix conventions make reference targets ambiguous in free text. We therefore use structural parsing with rule-based patterns for hierarchy extraction, but resolve cross-references with LLM-assisted extraction under structured output. The extraction schema is a fixed JSON object with the fields `source_component`, `target_law_title`, `target_component_type`, `section_number_or_range`, and `confidence`. Constraining the LLM to this schema prevents hallucinated relation types and confines failures to resolvable fields (unmatched law titles, phantom sections), which the validation pipeline filters.

Validation pipeline Extracted edges pass a four-stage filter: (i) a keyword pre-filter that removes candidates lacking any legal reference marker; (ii) content-hash deduplication across amendment versions, avoiding re-extraction for sections unchanged across a law's history; (iii) law-title resolution against the OCS catalog using exact-match, abbreviation expansion, and fuzzy matching; and (iv) edge validation against the cited law's section inventory, rejecting references to nonexistent sections. Across these stages, roughly 7% of candidate edges are discarded. Unresolved law titles dominate the discards, followed by *phantom law references* (laws absent from the digital repository, typically pre-digitization statutes) and *phantom sections*. A further source-side limitation is that about 30% of court-to-statute citations reference laws absent from the OCS repository entirely and therefore cannot be materialized regardless of extraction quality.

Setting	Tool(s)	Search Space
Golden Context	n/a (in-context)	Ground truth
Web Search	Web search, page fetch	Open web*
Agentic RAG	Vector search (BGE-M3)	5,127 sections
GraphRAG	Local search (gpt-4o-mini)	5,127 sections
FourCorners	Cypher query	271,893 sections

Table 2: Experimental settings. All use Claude 4.6 Opus as the reasoning LLM; only the retrieval tool and search space differ. *Leak-controlled: NitiBench and Revenue Department domains blocked.

4 Experiments

We evaluate our proposed method on the **NitiBench-Tax** benchmark (Akarajaradwong et al., 2025b) (50 questions): reasoning-intensive, multi-label questions derived from official tax rulings issued by the Revenue Department of Thailand, with section-level ground-truth citations. We fix the LLM to Claude 4.6 Opus (Anthropic, 2026) with temperature 0, isolating the data infrastructure as the only experimental variable. A temperature sensitivity analysis (Appendix A.3) confirms that FourCorners maintains stable citation accuracy across temperatures 0–1.0.

Settings We compare five settings (Table 2), all sharing the same agent prompt template and unlimited tool-call budget. Only the retrieval tool and search space differ:

1. **Golden Context**: All ground-truth provisions provided in context (upper bound, no retrieval).
2. **Web Search**: *Web search* and *page fetch*

tools, representing practitioner-standard research. Domains hosting NitiBench content[2] and Revenue Department rulings[3] are blocked to prevent leakage.

3. **Agentic RAG**: *Vector search* (BGE-M3 (Chen et al., 2024)) over NitiBench's curated pool of 5,127 sections from 35 laws[4], with query reformulation.
4. **Microsoft GraphRAG**: *Local search* from the GraphRAG pipeline (Edge et al., 2024) over the same 5,127-section pool. Entity-relationship triples are auto-extracted via gpt-4o-mini (indexing cost ≈$10), testing whether generic graph construction captures sufficient legal structure.
5. **FourCorners (Ours)**: *Cypher query* over the full knowledge graph (271,893 sections across 3,565 active laws). This is 53× larger than the curated pool, covering all active Thai legislation.

Metrics We measure **Citation Precision**, **Recall**, **F1**, and **F2** by comparing cited provisions against ground truth at the section level. F2 weights recall twice as heavily as precision, reflecting legal practice where missing a relevant provision is costlier than citing an extra one. We also report operational metrics: mean wall-clock **time**, API **cost**, and **tool call count** per question.

Output mapping All five settings share a common output format: an ordered list of (`law_title`, `section_number`) pairs emitted as structured JSON. We map these to canonical provisions with a single normalization pipeline used identically across settings, comprising exact-match, abbreviation expansion, and fuzzy matching on law title, followed by section-suffix normalization (e.g., "10/1" and "10 *bis*" resolve to the same canonical identifier). This isolates the retrieval backend as the only variable; agent prompts are byte-identical across settings.

5 Results and Analysis

Table 3 presents citation metrics and operational costs on NitiBench-Tax across all five settings. FourCorners achieves the best recall (0.983), F1 (0.812), and F2 (0.886) among all retrieval settings while searching a corpus 53× larger than the curated pool used by Agentic RAG and GraphRAG (271,893 versus 5,127 sections; Web Search is unbounded and not directly comparable on corpus size). This larger search space raises false-positive risk for FourCorners, yet it still leads on recall and F1. FourCorners is also the fastest retrieval setting (173s).

Citation accuracy FourCorners achieves the highest recall among all retrieval settings (0.983), approaching the Golden Context ceiling. The few remaining misses are cross-law references to the Civil and Commercial Code and Securities Act, provisions outside the Revenue Code that the agent did not traverse. By contrast, GraphRAG misses 13.2% of ground-truth provisions despite operating over a curated pool restricted to tax-relevant laws, and Web Search misses 16.4% due to fragmented online sources. We attribute this gap to the graph being designed around the structural requirements of legal reasoning: typed edges encode hierarchy and cross-references, allowing the agent to traverse relationships that are difficult to recover from flat text or auto-extracted entity graphs. We analyze how the agent uses these capabilities in §5.1.

FourCorners also achieves the highest F1 (0.812) and F2 (0.886), surpassing GraphRAG (0.761 / 0.811). The F2 advantage is particularly pronounced, as it weights recall more heavily, reflecting practice where missing a relevant provision carries greater risk than over-citing. A paired bootstrap test on per-question differences confirms significance for the F2 improvement over GraphRAG (ΔF2 = +0.075, $p = 0.007$), driven by recall (ΔRecall = +0.115, $p < 0.001$); the F1 difference (+0.051) is directionally positive but does not reach significance at $\alpha = 0.05$ ($p = 0.065$) given $n = 50$. The $|\hat{C}|$ column contextualizes citation volume: FourCorners cites 4.0 provisions per question from a search space of 271,893 sections, revealing high selectivity despite access to all active legislation.

Precision gap FourCorners's precision (0.751) trails its recall (0.983): false-positive citations are predominantly foundational or definitional provisions that provide necessary legal context but fall outside NitiBench-Tax's ground-truth scope, which annotates only sections directly cited in official tax rulings. This effect is amplified by FourCorners's larger search space (269,996 versus 5,127 sections). The same pattern appears in Agentic RAG (0.591 precision), where aggressive reformulation pro-

[2] https://huggingface.co/datasets/VISAI-AI/nitibench
[3] https://www.rd.go.th/
[4] https://huggingface.co/datasets/VISAI-AI/nitibench-statute

Setting	$\lvert\hat{C}\rvert$	Citation Metrics				Operational		
		Prec. ↑	Rec. ↑	F1 ↑	F2 ↑	Time ↓	Cost ↓	Tools ↓
Golden Context	2.6	0.962 ±.03	0.993 ±.01	0.977 ±.03	0.987 ±.02	60s	$0.66	n/a
Web Search	4.2	0.616 ±.06	0.836 ±.07	0.666 ±.08	0.739 ±.08	207s	$1.32	30.6
Agentic RAG	4.8	0.591 ±.06	0.952 ±.04	0.685 ±.06	0.799 ±.05	320s	$1.28	16.1
Microsoft GraphRAG	3.1	0.720 ±.07	0.868 ±.06	0.761 ±.07	0.811 ±.06	401s	$0.76†	26.0
FourCorners (Ours)	4.0	**0.751** ±.06	**0.983** ±.02	**0.812** ±.06	**0.886** ±.04	**173s**	$1.21	**18.4**

Table 3: Results on NitiBench-Tax (50 questions). $\lvert\hat{C}\rvert$: mean predicted citations per question. All citation metrics are per-question means with 95% bootstrap CIs (± half-width, 10K resamples). †Excludes gpt-4o-mini local search cost (~$0.10/question); with one-time indexing cost (~$10).

duces similar over-citation.

Operational efficiency FourCorners is the fastest retrieval setting (173s) with fewer tool calls than GraphRAG (18.4 vs. 26.0) and Web Search (30.6). Structured Cypher queries retrieve the needed subgraph in a few hops, whereas GraphRAG requires iterative community lookups (401s) and Agentic RAG's reformulation drives latency (320s). GraphRAG reports the lowest per-query cost ($0.76) due to gpt-4o-mini local search, though it additionally requires one-time indexing (≈$10). Other settings are comparable ($1.21–$1.32).[5]

Deployment validation The knowledge graph is deployed as a live Neo4j instance and currently powers a pilot version of the legal QA system under internal testing, ahead of a full rollout to the platform's existing user base. Early tester feedback is consistent with the benchmark results: structured traversal resolves the failures that motivated the transition from flat retrieval.

5.1 Graph Capability Usage

To understand *how* the agent uses the knowledge graph, we analyze all 820 Cypher queries issued across the 50 NitiBench-Tax questions (16.4 queries/question avg). Table 4 reports how often each graph capability is exercised.

Primary query strategy The agent's workflow follows a consistent pattern: (1) fulltext search on section content to discover relevant provisions (98% of questions), (2) label lookups to fetch specific sections by name (100%), and (3) document order sorting to present results in legislative sequence (88%). In 64% of questions, the agent's first query targets the Revenue Code directly, leveraging parametric knowledge to narrow the search space before querying the graph. In 38% of questions, the agent also ventures beyond the Revenue Code, searching the Civil and Commercial Code, customs law, or securities law, and uses law-level metadata (12/50 questions) as a discovery mechanism to identify relevant statutes across the full 3,565-law corpus.

Capability	Questions	Queries
Fulltext search (section content)	49/50	222
Label lookup (fetch by section name)	50/50	372
Law-level metadata search	12/50	12
DEKA content search	10/50	16
Hierarchy traversal (PART_OF)	25/50	201
Cross-reference (REFERENCES)	8/50	11
Cross-source (REFERENCES_LAW)	1/50	1
Document order sort	44/50	210
Temporal versioning (valid_from/to)	1/50	1

Table 4: Graph capabilities used by the FourCorners agent on NitiBench-Tax, measured from 820 Cypher queries across 50 questions.

Capability classification We categorize each of the 820 logged Cypher queries in Table 4 via qualitative analysis of agent behavior, inspecting the query body alongside the surrounding dialogue turn to identify which graph capability the agent is invoking. The categories are therefore behavioral, not purely syntactic: a query that combines a fulltext call with a PART_OF traversal can count toward both fulltext search and hierarchy traversal when the agent's intent exercises both capabilities. This is why the per-capability question counts in Table 4 need not sum to 50.

Structural traversal Hierarchy traversal (PART_OF edges) is the most frequently used structural capability, appearing in 50% of questions with 201 individual queries. The agent uses it to resolve scope, for example discovering all penalty sections within a division or all income

[5]All costs reflect a single agent used to isolate the data variable. Production deployments that route subtasks to smaller models can reduce per-query cost substantially.

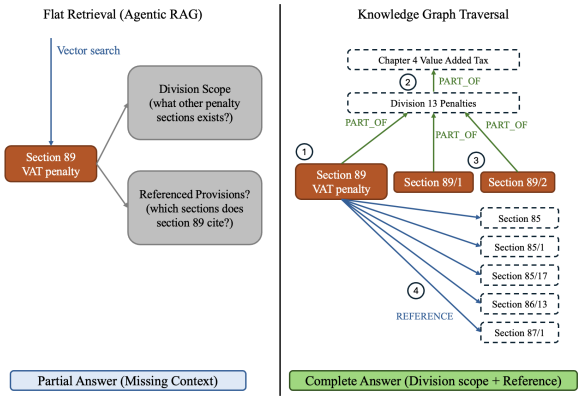

Figure 2: Multi-hop graph traversal versus flat retrieval. The agent discovers Section 89 (①), traverses PART_OF upward to the parent division (②) and downward to siblings (③), then follows REFERENCES to cross-division filing rules (④). Flat retrieval returns only the penalty text, missing structurally connected provisions.

categories within a chapter. Figure 2 illustrates this pattern: from Section 89 (VAT penalties), the agent traverses upward to identify the parent division, downward to enumerate sibling sections, and follows cross-division REFERENCES to the filing rules they enforce. Cross-reference edges appear in 16% of questions, typically when the agent reads a provision that explicitly cites another and then fetches the referenced section.

Capabilities not exercised by this benchmark Temporal versioning, sequential edges, and cross-source links are rarely or never used on NitiBench-Tax. This is expected, as the benchmark does not require historical provision retrieval or Supreme Court precedent searches. These capabilities target production queries outside the benchmark's scope.

6 Conclusion

We presented FourCorners, the first unified temporal knowledge graph for Thai legal data, integrating primary law with Supreme Court judgments into a graph of over 725K nodes and 1.9M structural edges, updated daily. A five-setting comparison on NitiBench-Tax demonstrates that purpose-built graph-structured retrieval substantially outperforms web search, flat RAG, and auto-extracted graph retrieval on citation accuracy, while trace analysis reveals how each structural capability contributes.

The four design principles (hierarchy, temporal, reference, and sequential) reflect structural properties widely observed in statutory legal systems. While FourCorners is jurisdiction-specific in content, the deeper problem we address is the absence of standardized, machine-readable legal data formats in many jurisdictions. Much prior work assumes structured markup such as XML or Akoma Ntoso, yet official legal data is often published only as HTML or PDFs with no consistent schema. FourCorners demonstrates the feasibility of constructing a temporally versioned, cross-referenced legal knowledge graph directly from such unstructured sources at the jurisdictional scale. Beyond QA, the graph supports legislative monitoring, compliance validation, judicial analytics, and structured APIs for downstream systems.

Limitations

Benchmark scope NitiBench-Tax's sample size (50 questions) is small; we report bootstrap 95% confidence intervals in Table 3 to quantify uncertainty. A paired bootstrap test confirms significance for FourCorners's F2 and Recall improvements over GraphRAG ($p = 0.007$ and $p < 0.001$), but the F1 difference does not reach significance ($p = 0.065$); a larger benchmark would be needed to resolve this. NitiBench-Tax is currently the only Thai legal benchmark with section-level ground-truth citations, making cross-benchmark validation infeasible. The small size and single-domain focus of this benchmark reflect the current state of Thai legal NLP resources rather than a methodological choice, and we hope FourCorners's release motivates broader multi-domain Thai legal benchmarks covering temporal reasoning and cross-source interpretation. The benchmark evaluates citation accuracy over a finite pool of tax-related legislation, which favors curated baselines that search only this pool. FourCorners operates over a substantially larger search space, covering all 3,565 active Thai laws with temporal versioning, yet the benchmark does not exercise capabilities such as point-in-time retrieval or cross-statute tracing of Supreme Court interpretations. The results in Table 3 should therefore be read as a lower bound on the practical advantage of a domain-specific graph over generic retrieval.

LLM reasoning bottleneck Even when correct provisions are provided directly in context, the LLM achieves Recall of only 0.993 and F1 of 0.977 on NitiBench-Tax, confirming that LLM reasoning, not retrieval, is the remaining bottleneck for citation accuracy.

Graph coverage Graph coverage is limited to OCS and DEKA; Thailand's 20 ministerial regulatory databases remain outside scope. FourCorners inherits the coverage limitations of its official sources: when OCS lacks a historical statute, typically laws predating digitization, the graph cannot provide it regardless of query quality, affecting approximately 30% of court-to-statute citations (§ 3.3). Within covered laws, our pipeline applies multi-stage validation, but errors in official sources propagate as irrecoverable phantom edges. Finally, capabilities not exercised by NitiBench-Tax, including point-in-time retrieval and cross-statute tracing of Supreme Court interpretations, are used on production traffic (Figure 2 step 4 shows the agent following court-to-statute edges during answer validation). Purpose-built benchmarks for temporal and cross-source legal reasoning are valuable future work.

Acknowledgments

We thank the beta users of FourCorners for their interest and feedback during the pilot release, which directly informed the structural design decisions described in this paper.

References

Pawitsapak Akarajaradwong, Chompakorn Chaksangchaichot, Pirat Pothavorn, Ekapol Chuangsuwanich, Attapol Rutherford, and Sarana Nutanong. 2025a. Aligning LLMs for Thai legal question answering with efficient semantic-similarity rewards. In *Proceedings of the Natural Legal Language Processing Workshop 2025*, pages 304–316, Suzhou, China. Association for Computational Linguistics.

Pawitsapak Akarajaradwong, Pirat Pothavorn, Chompakorn Chaksangchaichot, Panuthep Tasawong, Thitiwat Nopparatbundit, Keerakiat Pratai, and Sarana Nutanong. 2025b. NitiBench: Benchmarking LLM frameworks on Thai legal question answering capabilities. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34292–34315, Suzhou, China. Association for Computational Linguistics.

Anthropic. 2026. Introducing Claude Opus 4.6.

Tara Athan, Harold Boley, Guido Governatori, Monica Palmirani, Adrian Paschke, and Adam Wyner. 2013. OASIS LegalRuleML. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law (ICAIL '13)*, pages 3–12. ACM.

Borui Cai, Yong Xiang, Longxiang Gao, He Zhang, Yunfeng Li, and Jianxin Li. 2023. Temporal knowledge graph completion: A survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)*, pages 6545–6553.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Andrea Colombo, Anna Bernasconi, and Stefano Ceri. 2025. An LLM-assisted ETL pipeline to build a high-quality knowledge graph of the Italian legislation. *Information Processing and Management*, 62:104082.

Hudson de Martim. 2025. An ontology-driven graph rag for legal norms: A structural, temporal, and deterministic approach. *Preprint*, arXiv:2505.00039.

Biao Dong, Haoze Yu, and Haisheng Li. 2021. A knowledge graph construction approach for legal domain. *Tehnicki vjesnik - Technical Gazette*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Erwin Filtz, Sabrina Kirrane, and Axel Polleres. 2021. The linked legal data landscape: Linking legal data across different countries. *Artificial Intelligence and Law*, 29:485–539.

Supachoke Hanwiboonwat, Chaichana Thavornthaveekul, Prachya Boonkwan, Apivadee Piyatumrong, and Peerapon Vateekul. 2025. A comparative study on the development of a Thai legal QA framework using large language models and mixed legal datasets. In *Proceedings of the 30th International Conference on Applications of Natural Language to Information Systems (NLDB 2025)*, pages 201–215. Springer.

Rinke Hoekstra, Joost Breuker, Marcello Di Bello, and Alexander Boer. 2007. The LKIF core ontology of basic legal concepts. In *Proceedings of the Workshop on Legal Ontologies and Artificial Intelligence Techniques (LOAIT 2007)*, volume 321 of *CEUR Workshop Proceedings*, pages 43–63.

Sarika Jain, Pooja Harde, Nandana Mihindukulasooriya, Sudipto Ghosh, Ankush Bisht, and Abhinav Dubey. 2022. Constructing a knowledge graph from indian legal domain corpus. In *TEXT2KG/MK@ESWC*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Monica Palmirani and Fabio Vitali. 2011. *Akoma-Ntoso for Legal Documents*, pages 75–100. Springer Netherlands, Dordrecht.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*.

Pimchanok Promwang and Pruet Boonma. 2025. Development of a retrieval-augmented generation system for legal data in Thai language. *Data Science and Engineering (DSE) Record*, 6(1).

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2024)*.

Nattapat Tantapong, Sadanan Arsaibun, Pittipol Kantavat, Surapant Meknavin, and Boonserm Kijsirikul. 2025. Curriculum-structured fine-tuning for citation-faithful QA under the Thai land and buildings tax act. *Research Square (preprint)*.

A Appendix

A.1 GraphRAG Entity Taxonomy

Microsoft GraphRAG uses its default extraction prompt (gpt-4o-mini) over the same 5,127-section pool as Agentic RAG, producing an entity-relationship graph with generic entity types. The resulting taxonomy is: EVENT (2,156 instances, mostly statutory sections); ORGANIZATION (579, conflating laws and agencies); PERSON (522); GEO (61); together with 5,708 untyped free-text relationship edges. Statutory sections are typed as "events" and statutes as "organizations," and hierarchical containment is not represented. For example, "Section 89 of the Revenue Code" appears as an EVENT with a free-text relation "is part of" pointing to the ORGANIZATION "Revenue Code"; this is structurally indistinguishable from an unrelated clause that happens to share "Revenue Code" in its context window. Against this generic taxonomy, FourCorners's five typed edges (Table 1) encode exactly the structural semantics that legal reasoning requires.

A.2 Cross-Source Co-Citation Network

The 190,944 reference law edges reveal co-citation patterns: which laws are cited together in the same Supreme Court decision. Figure 3 visualizes the 15 strongest co-citation pairs among 12 frequently cited laws, colored by legal domain: civil/property (blue), criminal (red), labor (green), and administrative/tax (purple). The Civil & Commercial Code is the most cited law (180 citing decisions), and its thickest edge (co-citation count 52) connects to the Land Code. Cross-domain edges reveal how legal domains interact in practice: the Civil & Commercial Code connects to the Revenue Code (tax disputes involving contracts), to labor statutes (employment cases), and to the Land Code (property transactions).

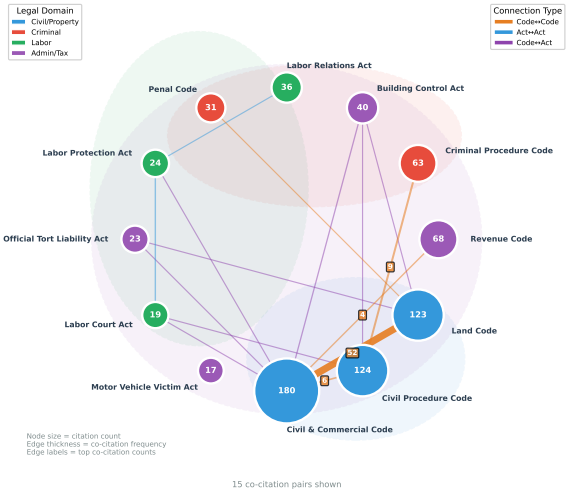

Figure 3: Law co-citation network from Supreme Court decisions. Node size and label indicate citation count; edge thickness and label indicate co-citation frequency. Node color denotes legal domain; edge color denotes connection type (Code–Code, Act–Act, or Code–Act).

A.3 Temperature Sensitivity

To assess robustness under stochastic decoding, we run the FourCorners setting at three additional temperatures (0.3, 0.7, 1.0), keeping all other variables identical to the main experiment ($n = 50$ questions per temperature, Claude 4.6 Opus, same prompt and tool configuration). Table 5 reports citation metrics with 95% bootstrap confidence intervals.

All four metrics are highly stable across temperatures: F1 ranges from 0.808 to 0.818 (SD = 0.004) and F2 from 0.881 to 0.902 (SD = 0.008). Recall ranges from 0.967 to 0.988 (SD = 0.008), confirming that the knowledge graph's structural constraints anchor retrieval regardless of sampling temperature. Precision is similarly stable (0.740–0.759, SD = 0.007), with all confidence intervals overlapping substantially. This confirms that the main results at temperature 0 are not an artifact of de-

Temp.	Prec.	Rec.	F1	F2
0.0	0.751 ±.06	0.983 ±.02	0.812 ±.06	0.886 ±.04
0.3	0.749 ±.08	0.980 ±.02	0.808 ±.07	0.881 ±.05
0.7	0.759 ±.08	0.967 ±.03	0.818 ±.07	0.902 ±.05
1.0	0.740 ±.08	0.988 ±.02	0.815 ±.07	0.895 ±.04
SD	0.007	0.008	0.004	0.008

Table 5: Temperature sensitivity of FourCorners on NitiBench-Tax. All citation metrics shown with 95% bootstrap CIs. The bottom row reports standard deviation of the four per-temperature means.

terministic decoding, and that FourCorners can be deployed at moderate temperatures to balance response diversity with citation reliability.

A.4 Data Source Timelines

The OCS corpus comprises 4,118 distinct laws across 18 legal categories: 1,395 Acts, 1,735 Ministerial Regulations, 552 Royal Decrees, 8 Codes, and 17 constitutional versions, among others. We track 6,561 distinct law versions in total. Figure 4 shows law versions published per year, with peaks in 1979 (constitutional reform) and 2021 (COVID-19 response).

The DEKA corpus includes 87,394 Supreme Court decisions spanning from 1920 to present. Court decisions cite specific statutory provisions, creating 236,911 cross-reference edges linking decisions to OCS sections. An additional 13,157 lower court case nodes are extracted from decision text as citation targets. Figure 5 shows decisions per year.

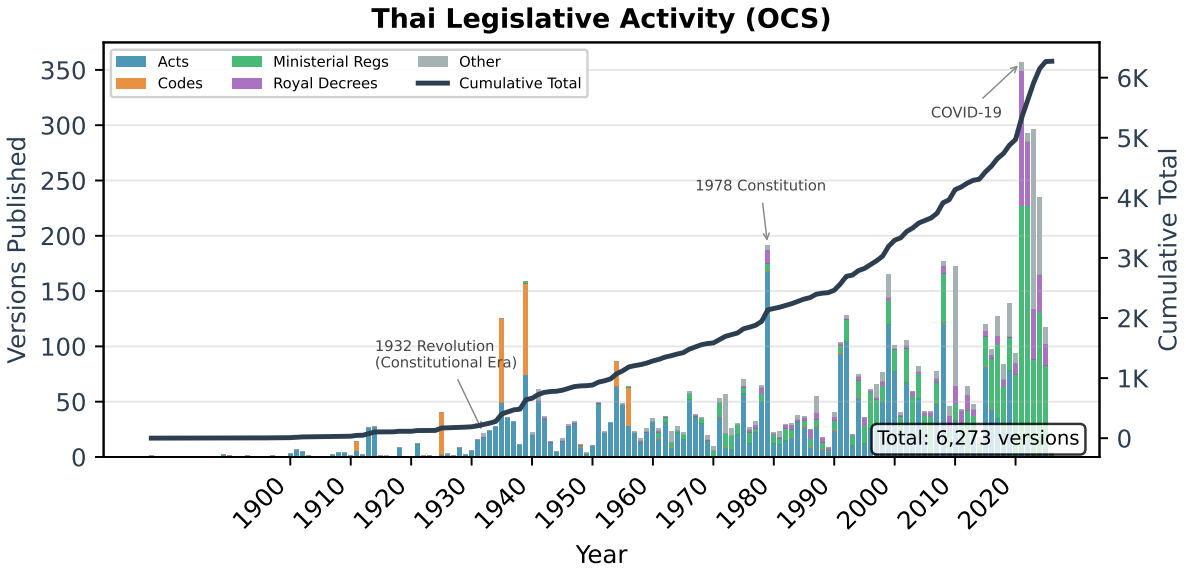

Figure 4: OCS law versions published per year by category, with cumulative total.

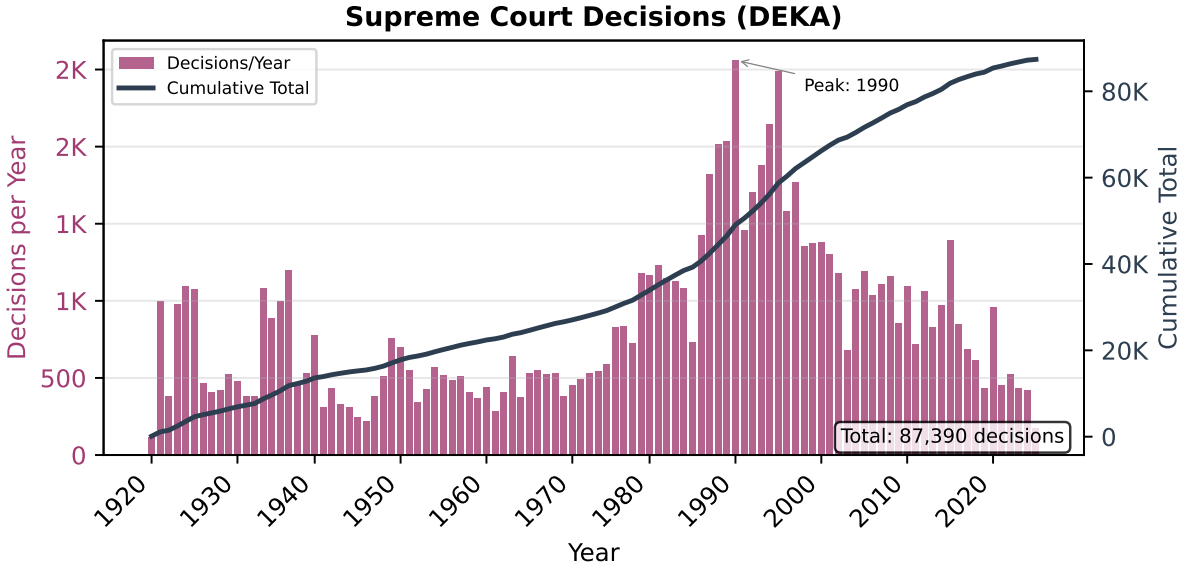

Figure 5: Supreme Court (DEKA) decisions per year, 1920–2025. Peak activity around 1990 precedes procedural reforms that reduced caseload.