Decompose-and-Refine: Structured Legal Question Answering with Parametric Retrieval

Anonymous ACL submission

Abstract

Large language models (LLMs) have shown strong performance in the legal domain, demonstrating notable potential in Legal Question Answering (LQA). However, unlike general QA, LQA requires answers that are not only accurate but also rigorously grounded in explicit legal authority. In statutory LQA, many questions require multi-hop reasoning across multiple legal issues, substantially increasing the risk of hallucination, thereby making accurate retrieval of supporting statutory provisions a critical prerequisite. Despite recent progress in multi-hop QA, existing approaches often rely on reasoning in natural language or retrieval without explicit query reformulation, leaving the vocabulary gap between user questions and statutory text largely unaddressed. To address this challenge, we propose **Decompose-and-Refine (DAR)**, a statute-grounded LQA framework that tightly integrates step-wise question decomposition with parametric knowledge–based query refinement. DAR progressively decomposes a complex legal question into atomic sub-questions and generates statute-aligned parametric queries for each sub-question, enabling the selection of a single most central statutory provision corresponding to each legal issue. We evaluate DAR on KoBLEX, a Korean multi-hop LQA benchmark grounded in statutory law, using Qwen3-32B and Gemma3-27B. Experimental results demonstrate that DAR consistently improves both retrieval accuracy and final answer quality over existing approaches. Moreover, by explicitly separating sub-questions and their corresponding statutory provisions, DAR facilitates transparent, issue-level verification of complex legal reasoning processes.

1 Introduction

Large language models (LLMs) have demonstrated strong performance across a wide range of tasks in the legal domain, showing particular promise in Legal Question Answering (LQA) (Lai et al., 2024). However, unlike general knowledge-based QA (Yang et al., 2018; Khashabi et al., 2018; Talmor et al., 2019), LQA demands not only substantial domain expertise but also answers that are rigorously grounded in explicit legal authority. In practice, many legal questions cannot be resolved by a single statutory provision; instead, they require multi-hop reasoning that sequentially integrates multiple legal issues and statutory requirements (Zhou et al., 2023). Such complexity substantially increases the risk of unfaithful or hallucinated reasoning, making it crucial to ensure that generated answers are firmly supported by verifiable legal evi-

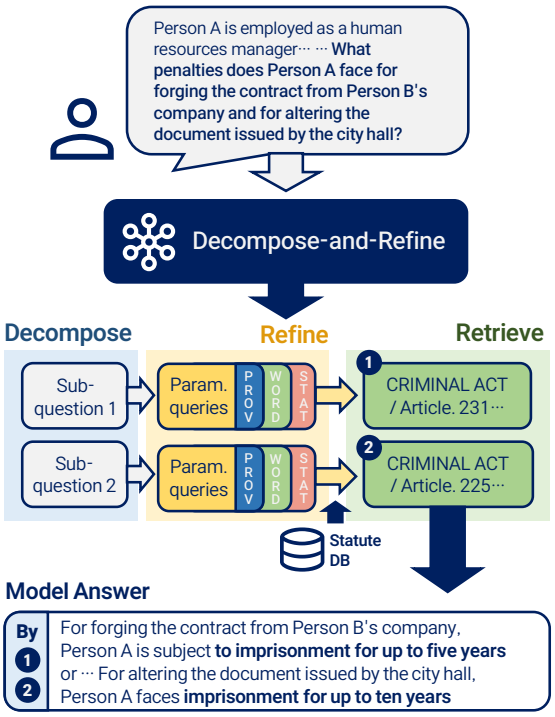

Figure 1: Overview of Decompose-and-Refine (DAR). DAR decomposes a complex legal question into atomic sub-questions, generates three types of parametric queries for each sub-question, and retrieves exactly one core statutory provision per issue, enabling controlled and evidence-centric legal reasoning.

dence (Magesh et al., 2025). Accordingly, accurate retrieval of relevant legal evidence has emerged as a fundamental prerequisite for building reliable LQA systems.

In statutory LQA, the primary supporting evidence is derived from statutory law. However, statutory provisions are typically written in abstract and normative language and are often organized within complex hierarchical structures. By contrast, user queries are usually expressed in natural, fact-oriented language, leading to a pronounced vocabulary gap between questions and statutes (Furnas et al., 1987). As a result, directly applying conventional retrieval methods such as BM25 or general-purpose dense retrievers often fails to reliably identify the appropriate legal provisions, particularly for complex questions that involve multiple interdependent legal issues (Zheng et al., 2025).

To address the reasoning complexity of multi-hop questions in general QA settings, prior multi-hop QA approaches have proposed generating step-wise reasoning processes using the implicit knowledge of LLMs (Wei et al., 2022), or decomposing questions into a sequence of sub-questions that are answered iteratively (Press et al., 2023; Trivedi et al., 2023). These methods represent meaningful progress by breaking complex questions into explicit intermediate reasoning steps, instead of attempting to solve them through a single-step inference. However, in the context of LQA, when decomposed sub-questions remain in natural language form, the vocabulary gap between user questions and statutory text persists, making it difficult to reliably retrieve the statutory provisions corresponding to each sub-question.

More importantly, we observe that the vocabulary gap in statutory LQA is multifaceted, arising not only from surface-level lexical mismatch but also from differences in normative phrasing and statute-level organization. Relying on a single query reformulation is therefore insufficient to robustly retrieve relevant statutory provisions. Motivated by this insight, we adopt a multi-channel parametric query refinement strategy, in which each sub-question is reformulated into multiple complementary representations that target different retrieval failure modes.

To this end, we propose **Decompose-and-Refine (DAR)**, a framework that tightly integrates question decomposition with parametric knowledge-–based query refinement. DAR progressively decomposes a complex legal question into atomic sub-questions and generates multiple complementary parametric queries for each sub-question based on legal knowledge (Figure 1). This design enables issue-level alignment between decomposed legal questions and their corresponding statutory evidence, allowing the selection of a single core statutory provision for each sub-question. By incrementally accumulating issue-aligned evidence, DAR reduces unnecessary context expansion while supporting evidence-centric legal reasoning.

Experimental results on KoBLEX (Lee et al., 2025), a Korean multi-hop LQA benchmark grounded in statutory law, demonstrate that DAR consistently improves both retrieval accuracy and final answer quality. By explicitly separating and accumulating statutes on a per-issue basis, DAR maintains interpretability of the reasoning process while achieving substantial performance gains.

Our contributions are summarized as follows:

- We propose DAR, a novel statute-grounded LQA framework that integrates step-wise question decomposition with multi-channel parametric query refinement.
- We align each atomic sub-question with exactly one supporting statutory provision, enabling interpretable, issue-level legal reasoning and transparent verification.
- We demonstrate on the KoBLEX benchmark that DAR consistently outperforms existing baselines in both retrieval and answer accuracy, while relying solely on a standard BM25 retriever without domain-specific fine-tuning.

2 Related Works

2.1 Query Refinement

A vocabulary gap often exists between user queries and target documents (Furnas et al., 1987). To mitigate this gap, prior work has explored various query refinement and expansion techniques, ranging from lexical expansion to semantic reformulation (Carpineto and Romano, 2012).

With the advent of LLM, query refinement has shifted from surface-level transformations toward semantic reformulation. For example, Gao et al. (2023) generate hypothetical documents corresponding to a given query, while Wang et al. (2023a) leverage LLMs to produce pseudo-documents for retrieval expansion.

In the legal domain, query rewriting research has predominantly focused on case law retrieval.

Askari and Verberne (2021) propose summarizing lengthy case law queries or extracting key terms, while Zhou et al. (2023) improve retrieval performance by using LLMs to select or summarize salient content that is decisive for judicial outcomes. More recently, Kim et al. (2025) propose a generative query rewriting approach that leverages LLMs to alleviate vocabulary mismatch between queries and legal documents.

However, these approaches primarily target case law retrieval, where the objective is to identify precedents with similar factual circumstances, and thus differ substantially from statute-based LQA, which requires precise alignment with normative legal provisions. Moreover, the applicability of multi-channel query refinement – in which a single query is transformed into multiple complementary variants for retrieval—has received limited attention in the context of LQA.

2.2 Legal Question Answering

Several existing LQA benchmarks (Guha et al., 2023; Fei et al., 2024; Fan et al., 2025) focus on settings in which LLMs directly generate answers without an explicit retrieval step. While such benchmarks are useful for evaluating the legal knowledge encoded in LLMs, they do not adequately reflect the evidence retrieval and verification processes required in real-world LQA. Louis et al. (2024) introduce an LQA benchmark and baselines based on a retrieve-and-read paradigm and evaluate multiple LLMs; however, they do not explicitly address issue-level retrieval strategies, limiting their applicability to complex legal questions involving multiple intertwined issues.

Prior multi-hop QA approaches (Press et al., 2023; Trivedi et al., 2023; Jiang et al., 2023; Cao et al., 2023; Chu et al., 2024) improve complex reasoning through step-wise decomposition and iterative retrieval. However, they generally use decomposed sub-questions directly as retrieval queries, paying limited attention to retrieval-oriented query reformulation. More recently, DualRAG (Cheng et al., 2025) introduces a dual-process retrieval framework that iteratively combines retrieval and reasoning to improve multi-hop question answering performance, while L-MARS (Wang and Yuan, 2025) studies multi-step legal reasoning with retrieval-augmented generation in the legal domain. While these approaches improve reasoning quality through tighter retrieval-reasoning integration, they primarily focus on iterative evidence acquisition and reasoning generation, rather than explicitly transforming decomposed legal issues into retrieval-effective statute-oriented queries. As a result, evidence corresponding to multiple legal issues may still be retrieved and utilized in a mixed manner.

In the legal domain, vocabulary mismatch between user queries and statutory text occurs frequently, and a single legal issue often spans multiple requirements and conditions distributed across statutory structures (Zhou et al., 2023). As a result, question decomposition alone is insufficient to reliably identify the correct statutory provision corresponding to each sub-question. This highlights that, in LQA, not only how to decompose a question, but also how to transform decomposed sub-questions into retrieval-effective queries, is an equally important challenge.

Motivated by this observation, Lee et al. (2025) propose ParSeR that leverages parametric provisions to bridge the representational gap between questions and statutes. However, ParSeR maintains a single-query formulation for the entire question, which limits its ability to separately handle multiple legal issues within a single query. Consequently, statutory provisions corresponding to different legal issues may be retrieved and used in a mixed manner, potentially reducing the transparency of evidence usage in downstream reasoning.

Recent studies further report that, in legal reasoning tasks, providing more contextual information does not necessarily lead to better performance (Kim and Lee, 2025). Instead, selectively identifying and utilizing only the core statutory provisions relevant to each issue is crucial for improving both accuracy and interpretability.

The proposed DAR integrates question decomposition with parametric query refinement to construct issue-level statute-aligned evidence. By selectively accumulating only core statutory provisions for each legal issue, DAR enables precise and interpretable legal reasoning.

3 Method: Decompose-and-Refine

DAR aims to simultaneously improve retrieval effectiveness and reasoning accuracy by progressively decomposing a complex legal question into atomic sub-questions and generating multiple law-informed query representations for each sub-question.

The overall pipeline consists of four stages:

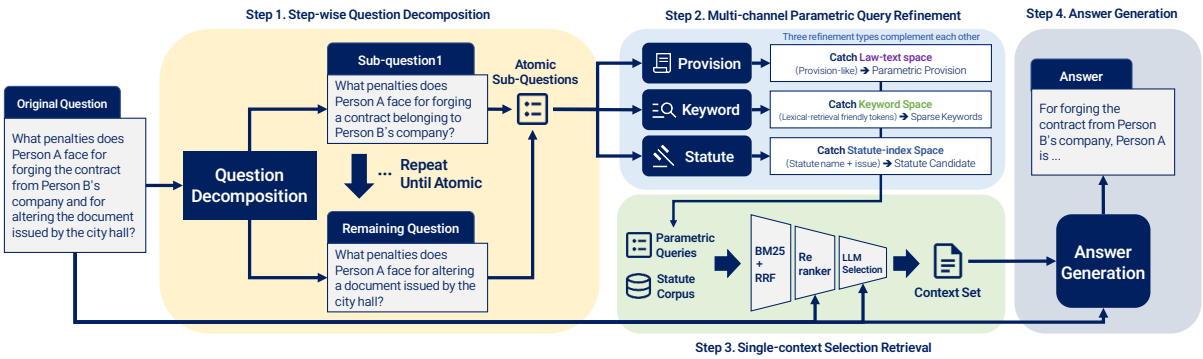

Figure 2: Illustration of Decompose-and-Refine (DAR) framework. Given a legal question, the model iteratively decomposes it into atomic sub-questions. For each sub-question, DAR generates multiple complementary parametric queries, including provision-style queries, sparse keywords, and candidate statute queries. Each query is used for BM25-based lexical retrieval, and the resulting candidate sets are fused via RRF and reranked, and a single most relevant statute is selected. Selected statutes are accumulated across reasoning steps and used to generate a final context-grounded answer.

(1) step-wise question decomposition, (2) multi-channel parametric query refinement, (3) single-context selection retrieval, and (4) answer generation (Figure 2). Detailed prompt templates used at each stage are provided in Appendix B.

3.1 Step-wise Question Decomposition

Given an input question Q, DAR processes it sequentially using a step-wise decomposition strategy. At each step, the model determines whether the current question is an *atomic question* that cannot be further decomposed, or a *non-atomic question* that requires additional decomposition. If the question is classified as non-atomic, the model extracts a single atomic sub-question q_i to be resolved at the current step and reconstructs the remaining unresolved part as the next question q_{i+1}. By iteratively applying this procedure, the original question is decomposed into a sequence of sub-questions of up to T steps. The decomposition process terminates when the remaining question q_{i+1} is classified as atomic and treated as the final reasoning target.

3.2 Multi-channel Parametric Query Refinement

For each atomic sub-question q_i, DAR generates a **set of complementary parametric queries** rather than relying on a single query representation. Each refinement type targets a distinct retrieval failure mode and plays a complementary role in bridging the gap between natural-language questions and statutory text.

Specifically, we employ the following three types of query refinement strategies.

(1) Parametric Provision. Given a sub-question q_i, the LLM generates a set of sentences $\{p_{i,1}, \ldots, p_{i,k}\}$ formulated to mirror statutory language and structure. By projecting the query into a law-centric representation space, this channel addresses linguistic mismatch by aligning the query with the normative phrasing of statutes.

(2) Parametric Sparse Keyword. For the same sub-question, the LLM extracts core legal concepts and procedural elements to produce a concise set of 5–10 sparse, noun-centric keywords. By emphasizing legally salient terms while discarding syntactic structure, this channel is well suited for lexical-based retrieval methods, which are particularly effective in the legal domain due to the term-sensitive nature of statutory text.

(3) Parametric Statute Candidate. Finally, the LLM predicts a set of statute names that are likely to be relevant to the sub-question and generates statute-aware retrieval queries by combining each statute name with the corresponding legal issue. By restricting retrieval to these predicted statutes, this channel reduces corpus-level ambiguity and suppresses spurious matches.

To make the distinctions among the three refinement strategies more concrete, we provide a brief illustrative example below.

Illustrative Example. Given the sub-question: *"When managing Person A's property, whose interests should Person B prioritize?"*

- **Parametric Provision:** "A trust is established for the purpose of managing trust property for the benefit of the beneficiary and using and

deriving profits from it in accordance with its intended purpose." (...)

- **Sparse Keyword:** trust, beneficiary priority, trustee duty, breach of duty, damages, trust property management, (...)
- **Statute Candidate:** (1) Civil Act, beneficiary's interests in a trust (2) Trust Act, trustee's fiduciary duties (...)

3.3 Single-context Selection Retrieval

For each sub-question, we design the retrieval process to yield exactly one supporting statutory provision. First, we perform BM25-based lexical retrieval (Robertson and Zaragoza, 2009) for each parametric query generated in Section 3.2. We observe that statutory provisions appearing repeatedly across retrieval results from multiple parametric queries are more likely to be semantically relevant to the underlying sub-question. Based on this intuition, we aggregate the retrieval results using Reciprocal Rank Fusion (RRF) (Cormack et al., 2009), which assigns higher ranks to provisions consistently retrieved by different queries. Following Lee et al. (2025), we then apply a cross-encoder–based reranker to the top N candidate provisions obtained from RRF, re-ranking them according to their semantic relevance to the sub-question.

Finally, given the re-ranked candidate set, an LLM selection module chooses a single statutory provision that most directly answers the corresponding sub-question. Through this process, DAR retains only the minimal core evidence required for each legal issue. We further analyze the effect of enforcing single-statute selection for each decomposed sub-issue in Appendix C.

3.4 Answer Generation

The statutory provisions selected at each step are accumulated iteratively, forming a deduplicated evidence set. In the final stage, the original question Q and the accumulated statutory provisions are provided as input to a context-grounded QA model, which generates the final answer based solely on the given legal evidence. We adopt a standard prompting strategy, in which the model is given the retrieved statutory context together with a small number of few-shot examples and is directly prompted to generate the answer.

Overall, by integrating step-wise question decomposition with parametric query refinement, DAR suppresses unnecessary retrieval expansion while systematically constructing an interpretable, issue-aligned set of statutory evidence. Given that DAR already yields tightly grounded, issue-specific contexts, we hypothesize that direct answer generation is preferable to explicit reasoning narration, a claim we empirically validate in Appendix D.

Method	Decomp.	QR.
Standard / CoT	×	×
Self-Ask / IRCoT / FLARE / ProbTree / BeamAggr	✓	×
ParSeR	×	✓
DAR (Ours)	✓	✓

Table 1: Comparison of baseline methods along two key design dimensions. Decomp. and QR indicate *step-wise question decomposition* and *query refinement*.

4 Experiments

This section describes the experimental setup used to evaluate the effectiveness of DAR.

All experiments are conducted in a few-shot setting, where pre-trained large language models are used directly without any additional task-specific training. We employ Qwen3-32B (Yang et al., 2025) and Gemma3-27B (Team et al., 2025), both of which support multilingual QA, as the backbone LLMs. For retrieval, we use a standard BM25 retriever (Robertson and Zaragoza, 2009), and a BGE-based reranker (Chen et al., 2024) is applied for candidate re-ranking. Further experimental details are provided in Appendix A.

4.1 Dataset and Evaluation Metrics

We conduct experiments on KoBLEX (Lee et al., 2025), a statutory law–grounded multi-hop legal question answering benchmark. KoBLEX consists of 226 question–answer pairs, each of which requires sequential reasoning over multiple legal issues and their corresponding statutory provisions. Each sample includes a *background scenario*, a *question*, an *answer*, a set of gold statutory provision contexts required to derive the answer, and a corpus of statutory provisions serving as the retrieval collection.

We evaluate both answer quality and retrieval effectiveness.

- **Answer Quality.** We measure answer quality using token-level F-1 score and LF-Eval, an LLM-as-a-judge metric designed to assess legal faithfulness.

	Retrieval				Answer			
	F-1		EM		Token F-1		LF-Eval	
	Qwen	Gemma	Qwen	Gemma	Qwen	Gemma	Qwen	Gemma
SP (Brown et al., 2020)	-	-	-	-	31.77	31.29	42.89	45.54
CoT (Wei et al., 2022)	-	-	-	-	26.78	28.51	41.99	48.01
Self-Ask (Press et al., 2023)	31.62	33.64	13.27	8.41	33.03	19.22	50.52	36.96
IRCoT (Trivedi et al., 2023)	29.65	17.55	4.42	2.65	29.86	22.98	49.36	33.95
FLARE (Jiang et al., 2023)	14.94	15.93	3.1	2.65	30.7	27.45	46.04	44.27
ProbTree (Cao et al., 2023)	23.03	21.74	6.64	4.87	27.25	28.06	50.67	49.96
BeamAggr (Chu et al., 2024)	21.08	21.79	5.75	1.77	31.32	27.64	44.05	45.70
ParSeR (Lee et al., 2025)	46.24	45.27	17.70	13.27	40.65	42.29	56.00	61.16
DAR (ours)	**58.63**	**55.08**	**34.51**	**28.76**	**45.53**	**43.97**	**68.13**	**65.35**

Table 2: Experimental Results on KoBLEX benchmark. The table compares retrieval performance (F-1 / EM) and answer quality (Token F-1 / LF-Eval) under Qwen3-32B and Gemma-27B. DAR (ours) consistently outperforms prior reasoning and retrieval-based baselines, demonstrating the effectiveness of query refinement and structured retrieval.

- **Retrieval Quality.** To evaluate retrieval performance, we compute provision-level Exact Match (EM) and F-1 scores between the set of provisions selected by the model and the gold provision set.

4.2 Baselines

To assess the performance of DAR, we compare it against a diverse set of baseline methods, which are categorized along two design axes: whether question decomposition is employed and whether parametric query refinement is used (Table 1).

- **Direct QA Methods.** Methods that process the entire question as a single input and generate answers end-to-end without explicit question decomposition or parametric query refinement (Standard Prompting (Brown et al., 2020), Chain-of-Thought (Wei et al., 2022) (CoT)).

- **Question Decomposition Methods.** Methods that decompose a question into sub-questions or intermediate reasoning steps to enable stepwise reasoning and retrieval, but do not explicitly incorporate parametric query refinement (Self-Ask (Press et al., 2023), IRCoT (Trivedi et al., 2023), FLARE (Jiang et al., 2023), ProbTree (Cao et al., 2023), BeamAggr (Chu et al., 2024)).

- **Parametric Query Reformulation Methods.** Methods that rewrite the question into parametric representations aligned with statutory language to improve retrieval performance, but do not perform explicit question decomposition (ParSeR (Lee et al., 2025)).

All baseline methods are implemented using the same statutory corpus and an identical retrieval pipeline to ensure a fair comparison. Specifically, for all retrieval-based methods, we apply BM25-based lexical retrieval followed by cross-encoder reranking and an LLM-based selection module, using the same hyperparameter settings as in DAR. Direct QA methods, including Standard Prompting and CoT, do not involve any retrieval step and generate answers solely based on the input question. To further control for model capacity, the same backbone LLMs are used across all methods, including DAR. Each baseline follows the original experimental settings proposed in the corresponding literature, with only minimal adaptations to accommodate the legal domain.

5 Results and Analysis

5.1 Overall Results

Table 2 presents the overall experimental results on the KoBLEX benchmark. DAR substantially outperforms all baseline methods in both retrieval effectiveness and final answer quality across both backbone LLMs (Qwen3-32B and Gemma3-27B).

In particular, in terms of retrieval F-1, DAR achieves substantial gains, approaching a twofold improvement relative to existing question decomposition–based methods (Self-Ask, IRCoT, FLARE, ProbTree, and BeamAggr), and also records the highest accuracy under the EM metric. These results indicate that question decomposition alone

Decomp.	QR.	F-1 / EM
✓	Par. + Spar. + Stat.	**58.63 / 34.51**
✓	Par. + Spar.	57.73 / 33.19
✓	Par.	55.52 / 30.09
✓	×	45.44 / 26.55
×	Par.	46.74 / 19.03

Table 3: Ablation study of DAR on retrieval performance (Qwen3-32B). Par., Spar., and Stat. denote *parametric provision* queries, *parametric sparse keyword* queries, and *parametric statute candidate* queries, respectively, while Decomp. and QR indicate *step-wise question decomposition* and *query refinement*.

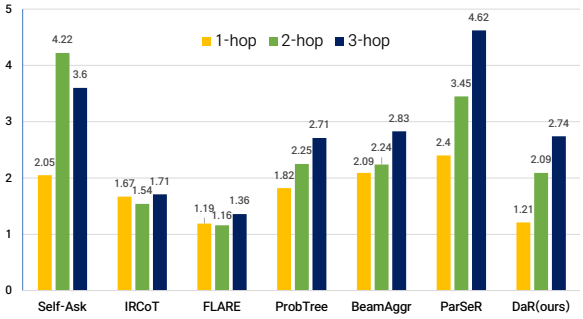

Figure 3: Average number of retrieved statutes per question across different number of gold contexts. DAR exhibits a controlled and near-linear increase in retrieved evidence, closely aligned with the underlying evidence requirements of the questions.

is insufficient for effective statutory retrieval, and that reconstructing decomposed questions into law-aligned representations is essential for identifying the correct legal evidence.

DAR also consistently outperforms ParSeR, demonstrating the advantage of explicitly handling multiple legal issues in a step-wise manner while preserving the benefits of parametric query refinement. This suggests that combining issue-level decomposition with parametric reformulation is more effective than applying either strategy in isolation.

From a design perspective (Table 1), Question decomposition–based methods provide a step-wise reasoning structure, but fail to sufficiently address the vocabulary gap because decomposed sub-questions are used directly as retrieval queries. ParSeR improves retrieval through parametric provisions, but its single-query formulation leads to the mixing of evidence when multiple legal issues are present within a single question.

By integrating these two design axes, DAR enforces the selection of exactly one core statutory provision per sub-question. This design choice leads to consistent improvements in both retrieval accuracy and reasoning transparency.

5.2 Ablation Study: Components

Table 3 reports the results of an ablation study examining the contributions of the main components of DAR.

We first analyze the effect of parametric query refinement. When all three types of parametric queries (provision, sparse keyword, and statute candidate) are employed, DAR achieves the highest retrieval performance, with F-1/EM scores of 58.63/34.51. In contrast, removing query refinement entirely (i.e., using question decomposition alone) results in a substantial performance drop to 45.44/26.55. This confirms that parametric query refinement is a core component supporting retrieval effectiveness.

Moreover, a gradual performance degradation is observed as individual query types are removed: removing statute-aware queries yields 57.73/33.19, while removing both statute-aware and sparse keyword queries further reduces performance to 55.52/30.09. These trends indicate that the different query types contribute complementarily to retrieval performance.

The importance of question decomposition is even more pronounced. Compared to the configuration that retains decomposition while applying parametric query refinement (55.52/30.09), removing decomposition and relying solely on parametric queries leads to a sharp drop in retrieval performance, with F-1 decreasing to 46.74 and EM to 19.03. This result demonstrates that query refinement alone is insufficient for accurately organizing and aligning multiple legal issues within a complex question.

Overall, these findings suggest that in statutory LQA, query refinement by itself is not enough; explicit question decomposition is essential for structurally isolating and ordering legal issues. By integrating decomposition with parametric query refinement, DAR reliably identifies the core statutory provision for each issue, leading to simultaneous improvements in both retrieval effectiveness and final answer quality.

5.3 Controlled Evidence Accumulation

Figure 3 reports the average number of retrieved statutes per question, grouped by the number of gold contexts and method. This analysis highlights

Retriever Setting	F-1	EM
Sparse (with Query Refinement)	58.63	34.51
Dense (with Query Refinement)	58.20	34.07
Sparse (without Query Refinement)	45.44	26.55
Dense (without Query Refinement)	46.46	27.43

Table 4: Retrieval performance comparison between sparse (BM25) and dense (BGE) retrievers under a fixed question decomposition setting, with and without query refinement.

how different methods scale their use of evidence as the number of required supporting statutes increases.

Across existing baselines, the number of retrieved statutes does not exhibit a consistent relationship with the number of gold contexts. For question decomposition–based methods such as Self-Ask, IRCoT, FLARE, ProbTree, and Beam-Aggr, the amount of retrieved evidence fluctuates across different levels of gold contexts and does not increase monotonically as more supporting statutes are required. ParSeR retrieves a relatively large number of statutes across all settings, but the growth is not well aligned with the underlying multi-context structure of the questions.

In contrast, DAR shows a clear and gradual increase in the number of retrieved statutes as the required gold contexts increases (1.21 for questions with 1-hop, 2.08 for 2-hop, and 2.74 for 3-hop). This pattern closely reflects the actual evidence requirements of the questions, indicating that DAR retrieves statutes in a structured manner. By aligning evidence accumulation with the number of required gold contexts, DAR avoids both under-retrieval and unnecessary context expansion, supporting precise and interpretable multi-hop legal reasoning.

5.4 Ablation Study: Retriever Type

Table 4 compares the retrieval performance of sparse and dense retrievers under a fixed question decomposition setting, with and without query refinement. A common intuition is that dense retrievers, by capturing semantic similarity in embedding space, may reduce the need for explicit query refinement. To examine this hypothesis, we performed additional experiments using a Korean fine-tuned dense retriever based on the BGE embedding model (Chen et al., 2024)[1] and compared it with a standard BM25-based sparse retriever, while keeping the other procedure identical in all settings.

[1] https://huggingface.co/dragonkue/BGE-m3-ko

The results show that when query refinement is applied, the performance difference between sparse and dense retrievers becomes relatively small. Sparse retrieval with query refinement achieves an F-1/EM of 58.63/34.51, while dense retrieval with the same refinement yields 58.20/34.07. This suggests that, under structured question decomposition and law-aligned query refinement, a sparse lexical retriever can achieve retrieval performance comparable to that of the evaluated dense retriever.

In contrast, when query refinement is removed, retrieval performance drops substantially for both retriever types. Dense retrieval without query refinement achieves an F-1/EM of 46.46/27.43, showing a slight improvement over sparse retrieval without refinement (45.44/26.55). However, the improvement remains limited, suggesting that retriever architecture alone may not adequately resolve complex legal questions involving multiple legal issues and fine-grained statutory conditions.

Overall, these findings suggest that the performance gains of DAR are driven more by explicit query refinement and issue-level query construction than by retriever architecture alone. Notably, DAR achieves strong retrieval performance using a standard BM25 retriever without additional dense retriever fine-tuning, highlighting the practicality of the proposed framework.

6 Conclusion

In this paper, we presented **Decompose-and-Refine (DAR)**, a statute-grounded framework for legal question answering that targets the challenges of multi-hop statutory reasoning. By integrating step-wise question decomposition with parametric query refinement, DAR enables effective alignment between natural-language questions and normative statutory text and demonstrates strong performance on the KoBLEX benchmark, even without retriever fine-tuning. We further demonstrate that DAR supports controlled and interpretable evidence accumulation by selecting a minimal, issue-aligned set of statutory provisions, thereby facilitating transparent and reliable legal reasoning.

Limitations

Despite its effectiveness on statute-grounded multi-hop legal question answering, the proposed DAR has several limitations.

First, DAR is specifically designed for statutory law–grounded legal question answering, and its ap-

plicability to case law–centric reasoning remains unexplored. Statute-based LQA requires precise alignment with normative provisions, whereas case law retrieval focuses on identifying precedents with similar factual patterns and judicial reasoning. As a result, the query structures, evidence granularity, and relevance criteria differ substantially between the two settings. Extending DAR to case law would therefore require alternative query reformulation strategies and retrieval objectives tailored to precedent-based reasoning.

Second, DAR relies heavily on the accuracy of step-wise question decomposition. When the decomposition stage fails to correctly isolate the core legal issues, errors can propagate to subsequent parametric query generation, retrieval, and answer generation stages. As illustrated in Appendix E, an incorrect or misaligned sub-question can lead to systematically irrelevant statute retrieval, even when later components function as intended. This error propagation highlights the importance of robust decomposition mechanisms and suggests that incorporating decomposition verification or correction strategies may be a promising direction for future work.

Overall, these limitations point to future research directions involving broader legal domains, more resilient decomposition techniques, and adaptive evidence modeling beyond strictly statute-grounded settings.

Ethical Considerations

The KoBLEX dataset is released under the Creative Commons Attribution–NonCommercial 4.0 International (CC BY-NC 4.0) license.[2] All question–answer pairs and supporting statutory contexts were constructed based solely on publicly available statutory law and were further reviewed and manually revised by legal experts. As a result, the dataset does not contain personal data, private information, or records of real individuals, and thus does not pose risks related to privacy infringement or data misuse.

All experiments in this study were conducted using LLMs with permissive or explicitly stated licenses, including Qwen3, which is released under the Apache 2.0 license,[3] and Gemma 3 with its own license.[4]

[2] https://github.com/daehuikim/KoBLEX
[3] https://github.com/QwenLM/Qwen3
[4] https://ai.google.dev/gemma/terms

In preparing this manuscript, we also made limited use of AI-assisted writing tools (GPT-5) to translate Korean examples into English. These translations were reviewed by the authors to ensure faithfulness.

The intended use of this research is to support academic investigation into LQA methods and explainable multi-hop reasoning over statutory law. The proposed framework are not designed for deployment in real-world legal practice, nor should they be interpreted as providing legal advice or guidance for resolving actual disputes or incidents.

References

Arian Askari and Suzan Verberne. 2021. Combining lexical and neural retrieval with longformer-based summarization for effective case law retrieval. In *Biennial Conference on Design of Experimental Search & Information Retrieval Systems*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Shulin Cao, Jiajie Zhang, Jiaxin Shi, Xin Lv, Zijun Yao, Qi Tian, Lei Hou, and Juanzi Li. 2023. Probabilistic tree-of-thought reasoning for answering knowledge-intensive complex questions. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12541–12560, Singapore. Association for Computational Linguistics.

Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1).

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Rong Cheng, Jinyi Liu, Yan Zheng, Fei Ni, Jiazhen Du, Hangyu Mao, Fuzheng Zhang, Bo Wang, and Jianye Hao. 2025. DualRAG: A dual-process approach to integrate reasoning and retrieval for multi-hop question answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31877–31899, Vienna, Austria. Association for Computational Linguistics.

Zheng Chu, Jingchang Chen, Qianglong Chen, Haotian Wang, Kun Zhu, Xiyuan Du, Weijiang Yu, Ming Liu, and Bing Qin. 2024. BeamAggR: Beam aggregation reasoning over multi-source knowledge for multi-hop question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1229–1248, Bangkok, Thailand. Association for Computational Linguistics.

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 758–759, New York, NY, USA. Association for Computing Machinery.

Yu Fan, Jingwei Ni, Jakob Merane, Yang Tian, Yoan Hermstrüwer, Yinya Huang, Mubashara Akhtar, Etienne Salimbeni, Florian Geering, Oliver Dreyer, Daniel Brunner, Markus Leippold, Mrinmaya Sachan, Alexander Stremitzer, Christoph Engel, Elliott Ash, and Joel Niklaus. 2025. Lexam: Benchmarking legal reasoning on 340 law exams. *arXiv preprint arXiv:2505.12864*.

Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Alan Huang, Songyang Zhang, Kai Chen, Zhixin Yin, Zongwen Shen, Jidong Ge, and Vincent Ng. 2024. LawBench: Benchmarking legal knowledge of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7933–7962, Miami, Florida, USA. Association for Computational Linguistics.

G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. 1987. The vocabulary problem in human-system communication. *Commun. ACM*, 30(11):964–971.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, and 21 others. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Preprint*, arXiv:2308.11462.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface:a challenge set for reading comprehension over multiple sentences. In *Proceedings of North American Chapter of the Association for Computational Linguistics (NAACL)*.

Daehui Kim, Deokhyung Kang, Jonghwi Kim, Sangwon Ryu, and Gary Lee. 2025. GuRE:generative query REwriter for legal passage retrieval. In *Proceedings of the Natural Legal Language Processing Workshop 2025*, pages 424–438, Suzhou, China. Association for Computational Linguistics.

Yongjae Kim and Wonjae Lee. 2025. Where does legal ai fail? evaluating rag pipelines. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, page 1396–1405, New York, NY, USA. Association for Computing Machinery.

Jinqi Lai, Wensheng Gan, Jiayang Wu, Zhenlian Qi, and Philip S. Yu. 2024. Large language models in law: A survey. *AI Open*, 5:181–196.

Jihyung Lee, Daehui Kim, Seonjeong Hwang, Hyounghun Kim, and Gary Lee. 2025. KoBLEX: Open legal question answering with multi-hop reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4019–4053, Suzhou, China. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2024. Interpretable long-form legal question answering with retrieval-augmented large language models. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D. Manning, and Daniel E. Ho. 2025. Hallucination-free? assessing the reliability of leading ai legal research tools. *Journal of Empirical Legal Studies*, 22(2):216–242.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational*

Linguistics: EMNLP 2023, pages 5687–5711, Singapore. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.

Liang Wang, Nan Yang, and Furu Wei. 2023a. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, Singapore. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Ziqi Wang and Boqin Yuan. 2025. L-mars: Legal multi-agent workflow with orchestrated reasoning and agentic search. *arXiv preprint arXiv:2509.00761*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Lucia Zheng, Neel Guha, Javokhir Arifov, Sarah Zhang, Michal Skreta, Christopher D. Manning, Peter Henderson, and Daniel E. Ho. 2025. A reasoning-focused legal retrieval benchmark. In *Proceedings of the 2025 Symposium on Computer Science and Law*, CSLAW '25, page 169–193, New York, NY, USA. Association for Computing Machinery.

Youchao Zhou, Heyan Huang, and Zhijing Wu. 2023. Boosting legal case retrieval by query content selection with large language models. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 176–184.

A Experimental Details

This section describes the evaluation metrics and the implementation details of the baselines used in our experiments.

A.1 Implementation Details

In the overall pipeline of DAR, all major components—including question decomposition, parametric provision generation, provision selection, and final answer generation —are implemented using the same LLM. This design choice eliminates model-induced variance across different stages and allows us to more clearly analyze the effectiveness of the proposed method itself.

For each atomic sub-question, all generated parametric queries are used as inputs to a BM25-based lexical retriever (Robertson and Zaragoza, 2009), retrieving the top 100 statutory provisions per query as candidates. Retrieval results obtained from different parametric queries are then aggregated using Reciprocal Rank Fusion (RRF) (Cormack et al., 2009), with the RRF constant set to 60. Based on the aggregated RRF scores, the top 300 candidate provisions are selected and passed to a cross-encoder–based reranker. After reranking, only the top 10 provisions are forwarded to the selection module. For reranking, we use a Korean fine-tuned BGE reranker (Chen et al., 2024)[5].

All experiments are conducted on a single NVIDIA A100 GPU. Inference over the full

[5] https://huggingface.co/dragonkue/bge-reranker-v2-m3-ko

KoBLEX benchmark takes approximately three hours.

A.2 Evaluation Metrics

To evaluate the performance of DAR from both answer accuracy and legal grounding perspectives, we employ multiple complementary evaluation metrics.

Retrieval F-1. Retrieval F-1 measures the degree of partial overlap between the set of statutory provisions selected by the model and the gold-standard provision set. For each question, precision and recall are computed based on the selected and gold provisions, and their harmonic mean is reported as the F-1 score.

Retrieval Exact Match (EM). Retrieval EM evaluates whether the provision set selected by the model exactly matches the gold-standard provision set. For each question, the score is set to 1 if the two sets are identical, and 0 otherwise.

Answer Token-level F-1. Answer Token-level F-1 measures the token-level overlap between the generated answer and the gold answer. Both answers are tokenized, and precision and recall are computed based on overlapping tokens, from which the F-1 score is derived. This metric captures both content accuracy and coverage of key information.

LF-Eval. LF-Eval is a legal fidelity–oriented evaluation metric proposed by Lee et al. (2025), built upon the G-Eval framework (Liu et al., 2023). Beyond surface-level answer matching, LF-Eval assesses the legal validity and faithfulness of an answer with respect to the provided statutory provisions. Specifically, an evaluation LLM receives both the generated answer and the cited provisions as input and judges whether the answer can be legally justified in context. This allows the metric to distinguish between answers that are lexically similar but legally incorrect, and those that are phrased differently yet legally sound. LF-Eval achieves a strong correlation with human judgments, reporting a Pearson correlation of 84.90 in a human evaluation study, which demonstrates that LF-Eval reliably reflects expert assessments of legal correctness.

A.3 Baseline Implementations

This section describes the implementation details of the baseline methods used in our experiments. Each baseline follows the original configuration proposed in prior work as closely as possible, with only minimal modifications to ensure applicability to the legal domain and the KoBLEX dataset.

Direct QA Methods.

- **Standard Prompting** (Brown et al., 2020) generates answers directly from the LLM without using any retrieval module, by providing the question together with a small number of few-shot examples.
- **Chain-of-Thought (CoT)** (Wei et al., 2022) uses few-shot prompts that explicitly demonstrate step-by-step reasoning, encouraging the model to generate intermediate reasoning chains before producing the final answer, without external retrieval.

Question Decomposition Methods.

- **Self-Ask** (Press et al., 2023) decomposes a complex question into a sequence of follow-up questions and generates intermediate answers sequentially. In our implementation, each follow-up question is processed using the same three-step retrieval pipeline (BM25, reranking, and selection) to select a single supporting statute before generating the intermediate answer.
- **IRCoT** (Trivedi et al., 2023) alternates between CoT reasoning and retrieval. Each retrieval step is implemented using the unified three-step retrieval pipeline, and intermediate reasoning outputs are used as queries for subsequent iterations.
- **FLARE** (Jiang et al., 2023) monitors token-level generation confidence and triggers retrieval when low-confidence segments are detected. Whenever retrieval is triggered, the same three-step retrieval pipeline is applied using the partial generation as a query.
- **ProbTree** (Cao et al., 2023) decomposes a question into a hierarchical tree and evaluates multiple reasoning strategies at each node. For open-book strategies, retrieval is performed using the unified three-step retrieval pipeline before computing log-probability scores.
- **BeamAggr** (Chu et al., 2024) aggregates multiple candidate answers generated at lower-level nodes using beam search. Retrieval at

each node is implemented via the same three-step retrieval pipeline, ensuring consistency across methods.

Parametric Retrieval Methods.

- **ParSeR** (Lee et al., 2025) reformulates a question into statute-like parametric queries to improve retrieval effectiveness. Since an official implementation of ParSeR tailored to the KoBLEX benchmark is publicly available, we directly use the authors' official code[6] without modification. The same three-step retrieval pipeline is applied to ParSeR.

B Prompt Design

This section presents the prompts used at each stage of DAR, as described in Section 3. Several prompts are adopted directly from prior work without modification. Specifically, the Parametric Provision Generation Prompt, the Selection Prompt, and the Answer Generation Prompt are reused from ParSeR (Lee et al., 2025).

For each prompt, we describe its intended purpose and usage context. Slots where actual inputs are injected are indicated in *italic*, and illustrative examples are omitted for brevity.

Question Decomposition Prompt

System: You are a legal reasoning assistant for Korean law. Your task is to decompose a complex legal question into smaller, atomic sub-questions one step at a time, and update the remaining question.
Definitions:
- Atomic sub-question: a question that can be answered by referring to a small number of specific legal provisions, without further decomposition.
- Non-atomic question: a question that still consists of multiple legal issues, steps, or conditions.

At each step:
1. Read the Background and the CURRENT Question.
2. Decide whether the CURRENT Question is atomic or not.
3. If NON-ATOMIC:
- Extract EXACTLY ONE atomic sub-question that should be solved at this step.
- Rewrite the remaining unresolved part as the Remaining Question.
4. If ATOMIC:
- Treat the CURRENT Question itself as the final Sub-question.
- Set Remaining Question to NONE.

Guidelines:
- Work in Korean when writing questions.
- Keep the sub-question as narrow and concrete as possible.
- The Remaining Question + the solved Sub-question should still be logically equivalent to the original question.
- If the CURRENT Question is already focused on a single legal issue, mark it as ATOMIC and reuse it as Sub-question.

Output format (must strictly follow):

Decomposition
Type: NON-ATOMIC or ATOMIC
Sub-question: <ONE atomic sub-question in Korean>
Remaining Question: <Updated remaining question in Korean, or 'NONE' if nothing remains>

User: <EXAMPLE> ... ...
<Query>
Background: *background*
Question: *question*

Decomposition

Parametric Provision Generation Prompt

System: You are an expert legal assistant whose task is to identify and return all relevant statutory provisions that support the answer to a given legal question.
Your role is not to provide interpretations, summaries, or conclusions, but to retrieve and list the exact legal clauses that serve as a legal basis for the scenario described.
- Answer must be a list of clauses in the following format: ["Name of Law (Title or Clause Summary) Exact clause text or its key portion.","Name of Law (Title or Clause Summary) Exact clause text or its key portion.",...] without any other explanations.
- Do not provide legal analysis or paraphrased answers—your response must consist solely of statutory provisions.
- Do not include article or clause numbers (e.g., "Article 123", "제116조의2")—summarize the clause title briefly in parentheses instead.
- If multiple laws are involved, list all clauses together in a single list.
- All internal quotation marks in the output must be escaped (i.e., use ‘ ’).
- If no directly applicable statutory provision exists, generate the most plausible clause in the same format, as if it were part of the relevant law.
- Do not explain or consider actual application, just answer in format.
User: <EXAMPLE> ... ...
<Query>
Question: *background + question*
Answer:

Parametric Sparse Keyword Generation Prompt

System: You are an expert legal assistant whose task is to rewrite queries for Korean statutory retrieval. Your goal is to convert a user's legal question into a high-performance BM25 sparse keyword query.

[6] https://github.com/daehuikim/KoBLEX

Maximize the recall of relevant statutory provisions by extracting core legal concepts and converting them into precise keywords.

Rules:
- Output MUST be a single line of 5 to 10 Korean keywords (space-separated).
- Prefer nouns, legal terms, and statute-style terminology.
- Strictly REMOVE filler words, particles (조사), verb endings (어미), politeness, and full sentences.
- Do NOT include quotes, bullets, numbering, or extra explanations. - Avoid duplicates; keep keywords diverse and informative.
- If the question implies a legal effect (e.g., punishment, cancellation, damages), include the specific legal remedy/effect keyword (e.g., 처벌, 무효, 취소, 해제, 손해배상, 기산점).
- If parties/roles matter, include role keywords (e.g., 피고인, 피해자, 임차인, 임대인).
- If procedure matters, include procedure keywords (e.g., 항소심, 소송, 재결, 기소).

User: <EXAMPLE> ... ...
<Query>
Background: *background*
Question: *question*
Answer:

Parametric Statute Candidate Generation Prompt

System: You are an expert legal assistant whose task is to predict the most relevant Korean statute names for the given sub-question, and generate short retrieval queries by combining each statute name with key legal issues.

Rules:
- Answer must be a Python list of strings.
- Format: ["Statute Name Keyword1 Keyword2", "Statute Name Keyword3 Keyword4", ...]
- Do not provide any explanations, just the list.
- Do not include specific article numbers (e.g., NO "Article 123", NO "제116조의2").
- If multiple laws are potentially relevant, generate queries for all of them.
- If no directly applicable statute is obvious, infer the most plausible area of law (e.g., Civil Act, Criminal Act).
- Keywords should be concise (2 5 words), noun-heavy, and strictly relevant to the statute.
- All internal quotation marks must be escaped.
- Output MUST be in Korean.

User: <EXAMPLE> ... ...
<Query>
Background: *background*
Question: *question*
Answer:

Selection Prompt

System: You are given a question, and a list of candidate passages with associated passage IDs.
Your task is to identify the most proper passage that directly support the answer to the question.
Please select only one ID among given candidate passages.

Important instructions:
- Do not include other information except the passage ID in your answer.
- Even if multiple passages seem relevant or none seem perfectly appropriate, you must select exactly one passage ID.
- your answer should be following after "Answer:" without other explanations.
User: <EXAMPLE> ... ...
<Query>
Background: *background*
Question: *question*
Candidate: *canidates*
Answer:

Answer Generation Prompt

System: You are a legal assistant AI. Given a user Background and a Question, generate a concise and accurate answer by grounding your response only in the provided legal passages (Context).
-Do not draw on any external knowledge, statutes, precedent, or opinion.
-Keep your answers concise and strictly grounded in the quoted text -Do not refuse or say insufficient context. Instead, use the provided Context to the fullest and deliver a complete, concise answer grounded solely in those passages. **User:** <EXAMPLE> ... ...
<Query>
Question: *background + question*
Context: *context*
Answer:

C Issue-Level Statute Selection Ablation

We assume that each atomic sub-issue obtained through decomposition can be resolved with a single statutory provision. To examine the validity of this assumption, we conduct an additional ablation study in which each sub-issue is allowed to reference multiple statutes simultaneously. Specifically, we relax the one-statute-per-issue constraint and allow the model to select up to $k \leq 3$ and $k \leq 5$ statutes for each issue decomposition step, while keeping all other experimental settings identical (Qwen3-32B backbone). The results are reported in Table 5.

We observe a consistent degradation across all evaluation metrics as k increases. In particular, Retrieval F1 decreases from 58.63 to 52.32, while Retrieval EM drops substantially from 34.51 to 19.47 when allowing up to five statutes. Similar trends are also observed in answer generation metrics (Token-F1 and LF-Eval).

These findings suggest that, once a legal question is decomposed into fine-grained atomic sub-

	Retrieval		Answer	
Setting	F-1	EM	Token F-1	LF-Eval
k = 1 (Default)	58.63	34.51	45.53	68.13
$k \leq 3$	53.32	21.68	44.57	67.76
$k \leq 5$	52.32	19.47	43.94	67.26

Table 5: Ablation study on the maximum number of statutes selected per sub-issue. Increasing the number of selected statutes consistently degrades both retrieval and downstream answer generation performance.

issues, precise alignment between each sub-issue and its corresponding statutory provision becomes more critical than broad evidence aggregation. Expanding the evidence set with multiple statutes introduces substantial contextual noise and weakens issue-level grounding, ultimately reducing retrieval and answer precision. Overall, the results empirically support our design choice of enforcing single-statute selection per sub-issue as an effective form of controlled evidence regularization.

D Answer Prompting Ablation

Prompting Strategy	Token F-1	LF-Eval
Naive (Standard Prompting)	**45.86**	**67.85**
Chain-of-Thought (CoT)	39.76	60.88
CoT Self-Consistency (n=5)	40.42	63.73

Table 6: Ablation of answer-generation prompting strategies on KoBLEX (Qwen3-32B). All settings use the same retrieved contexts; only the final QA prompting differs.

Table 6 reports the results of an ablation study examining the effect of different prompting strategies in the final answer generation stage, where the retrieved context is provided as input. We compare three prompting strategies: naive prompting (Brown et al., 2020), Chain-of-Thought (CoT) prompting (Wei et al., 2022), and CoT self-consistency (Wang et al., 2023b). Across all settings, the retrieval and selection results (i.e., the input context provided to the QA model) are kept identical, and only the answer generation prompt is varied.

The results show that the naive prompting strategy achieves the highest performance in both Token F-1 and LF-Eval. In contrast, applying CoT leads to a consistent degradation in both metrics. While CoT self-consistency (n = 5) partially recovers performance compared to CoT, it still underperforms the naive prompting setting.

These trends can be attributed to the characteristics of legal QA and to the constraints imposed by our QA prompting design, which enforces answers to be strictly grounded in the provided statutory context. First, answers in KoBLEX are often expressed as concise legal conclusions. When CoT prompting increases the length of generated outputs, token-level overlap with the gold answer may decrease, leading to lower Token F-1 scores. Second, during the explicit narration of reasoning steps, CoT may encourage the model to rely on its general legal knowledge or commonsense reasoning rather than directly reflecting the provided statutory text, which can be detrimental under context-grounded constraints and negatively affect legal faithfulness as measured by LF-Eval. Third, although self-consistency aggregates multiple reasoning paths via majority voting, when the retrieved context is already issue-aligned and unambiguous, additional CoT sampling may introduce unnecessary variance and weaken evidence consistency.

Overall, these findings suggest that when the retrieval pipeline of DAR provides precise, issue-level statutory evidence, concise, evidence-grounded answer generation via naive prompting can be more effective than reasoning-oriented prompting strategies such as CoT, both in terms of answer accuracy and legal faithfulness.

E Case Study

This section presents a qualitative analysis of representative success and failure cases to provide an intuitive understanding of the behavior and limitations of DAR. The cases are categorized into three types according to the success or failure of *question decomposition* and *statutory context retrieval*. All examples presented in the tables are originally written in Korean, and those not provided in English in the original dataset have been machine-translated for clarity and presentation.

(1) Successful Case: Correct Decomposition and Retrieval. Table 7 illustrates a case in which both question decomposition and context retrieval are successfully performed. DAR accurately decomposes a complex legal question into a set of independent atomic sub-questions and generates appropriate parametric queries for each sub-question, thereby identifying the core statutory provisions required for answering the question in a step-by-step manner. Because each sub-question is explicitly paired with its corresponding statute, the multi-

hop legal reasoning process becomes transparent, enabling issue-level verification of the reasoning steps.

(2) Error Analysis: Failure due to Question Decomposition. Table 8 presents a failure case caused by an error in the question decomposition stage. The initial sub-question fails to accurately capture the core legal issue of the original question, which subsequently leads to poorly aligned parametric queries. As a result, errors propagate through the retrieval stage, ultimately degrading the quality of the final answer. This case highlights that the performance of DAR is highly dependent on the accuracy of question decomposition, and that errors at this stage can have cascading effects on downstream retrieval and answer generation.

(3) Error Analysis: Correct Decomposition but Erroneous Retrieval. Table 9 shows a case where question decomposition is successfully performed, but an inappropriate statute is retrieved for a particular sub-question. Although the sub-question itself is clearly defined, the generated parametric queries and retrieval results fail to align precisely with the most relevant statutory provision. Consequently, incorrect legal evidence is accumulated during the reasoning process. This case demonstrates that, in legal question answering, question decomposition alone is insufficient, and that accurate issue-level statute alignment and selection remain critical challenges.

Overall, these case studies demonstrate that when DAR operates as intended, it can decompose complex legal questions into issue-level sub-questions and systematically accumulate the corresponding core statutes, enabling interpretable and evidence-grounded legal reasoning. At the same time, the analysis clearly reveals how errors in either the decomposition or retrieval stage can affect the entire reasoning pipeline, highlighting the need for improved decomposition robustness and more precise issue-level retrieval in future work.

Gold	
Background	Person A has filed a petition with the Constitutional Court against Person B's political party, alleging actions in violation of the national constitution. The Constitutional Court has accepted the case, and Person A is curious about the timeline for the final decision regarding Person B's party and the consequences if a dissolution order is issued.
Question	When the Constitutional Court accepts a case regarding the constitutionality of Person B's political party, within how many days must the final decision be rendered, and what happens to Person B's party if a decision ordering its dissolution is issued?
Gold_context	[1] CONSTITUTIONAL COURT ACT / Article. 38 / Time Limit of Adjudication Article 38 (Time Limit of Adjudication) The Constitutional Court shall pronounce the final decision within 180 days after it receives the case for adjudication: Provided, That if the attendance of seven Justices is impossible due to vacancies of Justices, the period of vacancy shall not be counted in the period of adjudication. [2] CONSTITUTIONAL COURT ACT / Article. 59 / Effect of Decision Article 59 (Effect of Decision) When a decision ordering the dissolution of a political party is pronounced, the political party shall be dissolved.
Model Response	
Sub-Question 1	When the Constitutional Court accepts a case concerning the constitutionality of Party B, within how many days must the final decision be rendered?
Parametric Query	[1] Constitutional Court Act (Conclusion of Adjudication and Decision) — The Constitutional Court shall pronounce a final decision within 180 days from the date it receives the petition. [2] Constitutional Court Act (Effect of a Decision Ordering Party Dissolution) — A decision ordering the dissolution of a political party results in the party's assets and facilities being vested in the State and the prohibition of the party's activities. [3] Constitutional Court unconstitutionality review, political party dissolution, decision period, final decision, adjudication procedure, constitutional complaint [4] Constitutional Court Act, unconstitutional political party, dissolution adjudication [5] Constitutional Court Act, time limit for pronouncing a final decision [6] Constitutional Court Act, political party dissolution procedure
Retrieved Context	CONSTITUTIONAL COURT ACT / Article. 38 / Time Limit of Adjudication Article 38 (Time Limit of Adjudication) The Constitutional Court shall (...)
Sub-Question 2	If a decision ordering the dissolution of Party B is pronounced, what happens to Party B?
Parametric Query	[1] Constitutional Court Act (Effect of a Political Party Dissolution Decision) — A decision ordering the dissolution of a political party has the effect of terminating the existence and activities of the party concerned. [2] Constitutional Court Act (Pronouncement of a Political Party Dissolution Decision) — When the Constitutional Court decides to dissolve a political party in a party dissolution adjudication, it shall clearly state the reasons for the decision. [3] Constitutional Court Act (Filing of Petition and Adjudication Procedure) — The Constitutional Court shall conclude oral arguments and pronounce its decision within 180 days from the date it receives the petition for adjudication. [4] Effects of a political party dissolution decision, Constitutional Court, party dissolution procedure, handling of property rights, obligations, disadvantages of a dissolution order, starting point of legal effects [5] Constitutional Court Act, effect of a political party dissolution decision [6] Constitutional Court Act, legal consequences following political party dissolution [7] Political Parties Act, party dissolution decision procedure
Retrieved Context	CONSTITUTIONAL COURT ACT / Article. 59 / Effect of Decision Article 59 (Effect of Decision) When a decision ordering the dissolution of a (...)

Table 7: Successful case study of DAR on a multi-hop statute-based legal question. For brevity, repeated statutory text in the retrieved context that overlaps with the gold context is abbreviated with ellipses.

Gold	
Background	Five years ago, Person A lent money to Person B, but Person B has not repaid the debt. Concerned that the statute of limitations on the debt is approaching, Person A wishes to take legal action to interrupt the prescription period. Person A has decided to first apply for a provisional attachment.
Question	What actions can Person A take to interrupt the statute of limitations on the debt owed by Person B?
Gold_context	[1] CIVIL ACT / Article. 168 / Causes Interrupting Extinctive Prescription Article 168 (Causes Interrupting Extinctive Prescription) Extinctive prescription shall be interrupted in any of the following cases:1. Demand;2. (...) [2] CIVIL EXECUTION ACT / Article. 279 / Request for Provisional Seizure / Para. 1 The following matters shall be entered in a request for provisional seizure:1. Indication of a requested claim, and when such requested claim is not a specific sum of money, the amount converted into the money; 2. Indication of the facts which are to become the reasons for a provisional seizure under Article 277.
Model Response	
Sub-Question 1	Is it possible for Person A to apply for a provisional attachment in order to interrupt the statute of limitations on the debt owed by Person B?
Parametric Query	[1] Civil Act (Interruption of Extinctive Prescription) — Extinctive prescription is interrupted upon the occurrence of any of the following grounds: transfer or inheritance of a claim; (...) [2] Debt, provisional attachment, interruption of extinctive prescription, creditor, debtor, Civil Execution Act [3] Civil Procedure Act, provisional attachment, interruption of extinctive prescription [4] Civil Act, claim, extinctive prescription, provisional attachment [5] Debtor Rehabilitation and Bankruptcy Act, effect of provisional attachment
Retrieved Context	Civil Act / Article 168(2) (Causes Interrupting Extinctive Prescription) Attachment, provisional attachment, or provisional disposition shall interrupt extinctive prescription.
Sub-Question 2	Apart from applying for a provisional attachment, what other measures can Person A take to interrupt the statute of limitations on the debt owed by Person B?
Parametric Query	[1] Civil Act (Interruption of Extinctive Prescription) — Extinctive prescription shall be interrupted in any of the following cases: demand; attachment, provisional attachment, or provisional disposition; acknowledgment of the obligation by the debtor. [2] Civil Act (Demand) — Where a creditor makes a demand against the debtor, extinctive prescription shall be interrupted; however, if the creditor fails to take further action within six months after becoming aware that the demand is groundless, the prescription shall resume running. [3] Civil Act (Acknowledgment of Obligation) — Where the debtor acknowledges the existence of the obligation or expresses an intention to perform it, extinctive prescription shall be interrupted. [4] Debt, interruption of extinctive prescription, provisional attachment, payment order, lawsuit, creditor, debtor [5] Civil Procedure Act, interruption of extinctive prescription [6] Civil Procedure Act, claim, provisional attachment [7] Civil Act, claim, interruption of extinctive prescription [8] Civil Act, non-performance of obligation
Retrieved Context	Civil Act / Article 168(1) (Causes Interrupting Extinctive Prescription) A demand shall interrupt extinctive prescription.

Table 8: Failure case of DAR caused by error propagation from an incorrect question decomposition to subsequent retrieval steps.

Gold	
Background	Person A and Person B entered into a contract for the sale of goods. Person A agreed to supply a specific product to Person B within one month. However, Person A failed to deliver the goods by the contractual deadline. As a result, Person B intends to rescind the contract. The contract included a clause stating, In the event of a delay in performance, Person B does not require additional written notice to rescind the contract.
Question	Under what conditions can Person B rescind the contract if Person A fails to fulfill their contractual obligations, and how does the presence of other clauses in the contract affect this?
Gold_context	[1] CIVIL ACT / Article. 544 / Delay of Performance and Rescission Article 544 (Delay of Performance and Rescission) If one of the parties does not perform his obligation, the other party may fix a reasonable period and give peremptory notice demanding its performance, and may rescind the contract, if no performance is effected within such period: Provided, That if the obligor declares in advance his/her intention that he will not effect such performance, no peremptory notice shall be required. [2] CIVIL ACT / Article. 105 / Optional Provisions Article 105 (Optional Provisions) If the parties to a juristic act have declared an intention which differs from any provisions of statutes, which are not concerned with good morals or other social order, such intention shall prevail.
Model Response	
Sub-Question 1	Under what conditions can Person B rescind the contract if Person A fails to perform their contractual obligations?
Parametric Query	[1] Civil Act (Rescission of Contract) — Where the obligor delays performance of the obligation, the obligee may fix a reasonable period and urge performance; (...) [2] Civil Act (Rescission of Contract) — Where there is a special agreement (...) [3] Sale contract, delay of performance, rescission, conditions for exercising (...) [4] Civil Act, rescission for delay [5] Civil Act, non-performance, rescission [6] Civil Act, delay of performance, right of rescission [7] Commercial Act, rescission of sale contract
Retrieved Context	CIVIL ACT / Article 545 (Fixed-Time Acts and Rescission) Article 545 provides that where, due to the nature of the contract or the parties' declared intentions, the purpose of the contract cannot be achieved unless performance is rendered by a specified date or within a specified period, and one party fails to perform at (...).
Sub-Question 2	How does the situation change if the contract contains other clauses related to rescission?
Parametric Query	[1] Civil Act (Rescission of Contract) — Where the obligor fails to make a genuine (...) [2] Civil Act (Exercise of the Right of Rescission) — The right of rescission shall (...) [3] Civil Act (Principle of Priority of Special Agreements) — Where the contracting (...) [4] Contract rescission, delay of performance, conflict of contractual clauses, (...) [5] Civil Act, rescission for delay of performance [6] Civil Act, special agreement clauses on rescission [7] Civil Act, declaration of intention for rescission [8] Civil Act, non-performance, right of rescission
Retrieved Context	ACT ON THE REGULATION OF TERMS AND CONDITIONS / Article 5 Article 5 governs the interpretation of standardized contract terms and provides guiding principles for resolving ambiguities or conflicts in contractual clauses.

Table 9: Failure case of DAR where correct question decomposition is followed by erroneous statute retrieval. Lengthy parametric queries and retrieved contexts are partially omitted and indicated with (...).