IterLegal: Enhancing Legal Large Language Model Reasoning with Iterative Search via Reinforcement Learning

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have demonstrated significant potential in enhancing reasoning abilities, as evidenced by models like OpenAI's GPT series and DeepSeek-R1. However, effectively integrating legal reasoning with external search processes remains challenging, particularly for complex legal inquiries requiring multiple retrieval steps. To address this, we first construct a hierarchical legal knowledge base that organizes Chinese laws and regulations into a structured, searchable repository of legal provisions. Building upon this, we propose **IterLegal**, a novel framework that trains LLMs to actively query this knowledge base during reasoning, using reinforcement learning without requiring supervised data on intermediate steps. Our approach treats legal search operations as integral components of the reasoning process: the model autonomously decides when and how to retrieve relevant legal texts, and the retrieved results iteratively refine subsequent analysis. We train **IterLegal** on the Qwen2.5-7B-Instruct model and conduct extensive experiments across multiple legal reasoning benchmarks. Despite training on a single legal dataset, our model exhibits strong generalizability to various legal tasks. Further analysis reveals that **IterLegal** autonomously develops advanced legal reasoning abilities through its iterative search-reasoning process.

1 Introduction

In recent years, Large Language Models (LLMs) have demonstrated remarkable performance across a wide array of tasks (Yang et al., 2024; Lin et al., 2024; DeepSeek-AI et al., 2025; Anthropic, 2025). Beyond leveraging their internal knowledge acquired during pre-training, LLMs exhibit the capability to utilize external tools, particularly search engines, to retrieve factual and time-sensitive information, thereby mitigating hallucinations (Schick et al., 2023; Shen et al., 2023; Ma et al., 2024; Chen et al., 2024b). This capability, often referred to as Retrieval-Augmented Generation (RAG), has been widely investigated (Gao et al., 2023; Yan et al., 2024; Asai et al., 2024; Zhao et al., 2024). However, designing robust multi-step RAG strategies for complex real-world problems remains challenging, particularly in legal domains where many issues inherently require multi-step reasoning (Wang et al., 2024; Shao et al., 2023; Trivedi et al., 2023).

Recent advances have improved LLM reasoning through chain-like thinking prior to final output generation (Wei et al., 2022; Zelikman et al., 2022), as seen in OpenAI-o1 (OpenAI, 2024), and DeepSeek-R1 (DeepSeek-AI et al., 2025), highlighting the importance of test-time scaling in reasoning (Snell et al., 2024; Muennighoff et al., 2025). Yet, training LLMs to perform interactive reasoning with information retrieval remains an open challenge. Most multi-step RAG approaches rely on manually designed prompts or heuristics, which lack scalability (Shao et al., 2023; Trivedi et al., 2023; Press et al., 2023). Moreover, labeling reasoning steps is often impractical due to high cost and time constraints.

Reinforcement learning (RL) has emerged as a promising direction for enhancing reasoning without requiring supervised data on reasoning steps (DeepSeek-AI et al., 2025; Shao et al., 2024). RL-based training methods have demonstrated significant improvements in complex reasoning tasks, where models learn to decompose problems into manageable steps through trial and error rather than explicit instruction. However, these methods primarily focus on internal reasoning, with limited exploration of combining reasoning with external knowledge retrieval, particularly in specialized domains such as law. Legal knowledge evolves over time, and training should not be confined to internal reasoning that treats legal knowledge as fixed within the model.

In this paper, we propose **IterLegal**, a frame-

work that trains Legal LLMs to perform iterative search via reinforcement learning. We use Group Relative Policy Optimization (GRPO) to train **IterLegal** from scratch on Qwen2.5-7B-Instruct, enabling the model to learn legal reasoning with search operations autonomously. The model generates a reasoning chain that interleaves thinking (`<think>`), search queries (`<search>`), and retrieval results (`<information>`), agentically deciding when to search based on prior thinking and using retrieved results to guide subsequent reasoning. Since no supervised reasoning steps are required, reinforcement learning directly encourages effective iterative search behavior. Our contributions are as follows:

- We introduce a hierarchy legal article knowledge base that covers diverse legal domains and conforms to the legal theory of sources of law, providing a high-quality, searchable repository that enables LLMs to actively retrieve relevant articles during reasoning.
- We propose a novel framework that shifts legal LLMs from static "closed-loop thinking" to dynamic "active iterative searching," enabling models to identify knowledge boundaries and proactively seek external information to handle legal reasoning complexity.
- Through detailed analysis of the training process, we demonstrate that **IterLegal** autonomously develops advanced reasoning capabilities without relying on any predefined heuristics or supervised signals.

2 Related Work

Legal Reasoning with LLMs. Early legal LLMs were primarily applied to judgement prediction and information extraction (Hou et al., 2025). However, systematic evaluations have revealed substantial limitations in their reasoning capabilities, driving a shift toward step-by-step reasoning. Recent works have explored a variety of approaches, including formalizing legal syllogism with methods like LexChain (Xie et al., 2026), applying process supervision with LegalReasoner (Shi et al., 2025) and SyLeR (Zhang et al., 2025), and introducing agentic frameworks such as GLARE (Yang et al., 2025) that invoke dynamic modules. Despite these advances, most current approaches still operate as closed-book systems, relying on static internal parameters rather than dynamic external retrieval. This limitation prevents the transition from "internal thinking" (reasoning solely within fixed model weights) to "active search," where models proactively seek external information. To bridge this gap, we propose a unified framework that integrates agentic search with legal reasoning, enabling this paradigm shift and equipping LLMs with the capacity for dynamic interaction with external legal environments.

Legal-Aware Reinforcement Learning. Recent legal RL methods, such as SyLeR, LegalDelta, and Unilaw-R1, enforce syllogistic reasoning or factual correctness using citation-aware rewards and hard-sample mining (Dai et al., 2025; Cai et al., 2025; Zhang et al., 2025; Hu et al., 2025). However, these methods rely on static training environments, which limits their ability to adaptively navigate complex legal scenarios. Our framework addresses this gap by training an LLM end-to-end for interleaved, iterative retrieval tailored to legal reasoning, thus enabling truly legal-aware reinforcement learning.

Multi-step RAG and Tool Learning. Standard RAG performs single-step retrieval (Gao et al., 2023; Asai et al., 2024; Yan et al., 2024). Multi-step strategies include iterative retrieval (Shao et al., 2023), interleaved reasoning and retrieval (Trivedi et al., 2023; Press et al., 2023), and adaptive retrieval (Jiang et al., 2023; Jeong et al., 2024). However, they often rely on hand-crafted prompts. Tool learning extends LLMs with APIs (Schick et al., 2023; Shen et al., 2023; Qin et al., 2024), yet tool use is frequently isolated from reasoning. Our work unifies iterative search and chain-like reasoning via reinforcement learning without explicit supervision, directly enabling the dynamic interaction.

3 Preliminary

3.1 Legal Knowledge Base

Knowledge Base Construction. We construct a hierarchical legal knowledge base that organizes Chinese laws and regulations into a structured framework, enabling efficient retrieval for downstream models. The knowledge base stores essential legal texts, including *law names*, *chapters*, and *specific article contents*, each represented as a vector embedding that captures semantic information. These embeddings support advanced querying and seamless integration with search tools, facilitating accurate retrieval of relevant legal provisions. The over-

all design ensures that legal knowledge is readily accessible for model-driven analysis and decision-making across a wide range of legal applications.

Knowledge Graph Specification. The knowledge graph developed in this paper mainly comprises four types of legal documents: *the Constitution*, *laws*, *administrative regulations*, and *judicial interpretations*, while the remaining categories include departmental rules, normative documents, judicial documents, industry standards, and others. The first three follow the hierarchy of legal sources established in the *Legislation Law of the People's Republic of China*. Judicial interpretations are legally subordinate to the laws they interpret and do not constitute an independent hierarchical level; therefore, we model them as child nodes of the corresponding law nodes, linked via an "Interpretations" relationship. The Constitution, which is rarely used directly as a basis for adjudication in Chinese jurisprudence and appears only exceptionally in cited cases, is not treated as a separate level in the dataset. Instead, it is retained solely as a top-level node in the knowledge graph for structural completeness.

To quantify the distribution of legal provisions, we analyze two dimensions: by legal level of authority and by legal field (see Appendix D.1 and D.2 for details). For each dimension, we report the proportion of total articles contributed by different categories. Specifically, we identify 13 major legal fields and display the top 10 individually; for the legal level hierarchy, we present the four most article-rich levels. All remaining minor categories are grouped as "Others". These results are visualized in Appendix Figure 4.

3.2 Training Data Construction

We construct our training dataset from JEC-QA (Trivedi et al., 2022), a large-scale Chinese judicial examination question bank, by selecting questions that involve citations to legal articles. Based on the hierarchical structure of our legal knowledge base, we treat each selected question as being associated with one or more legal citations. To eliminate the inherent distribution bias where citations to laws far outnumber those to administrative regulations, we adopt a hierarchically balanced sampling strategy that respects legal authority.

Specifically, we sample candidate cases from two levels of authority according to the hierarchical position of the legal source cited: the *Law level* and the *Administrative Regulation level*. Questions that cite judicial interpretations are assigned to the Law level, following the legal nature of the underlying law. For questions citing multiple legal sources, we employ DeepSeek-V3 (Liu et al., 2024) to determine the primary level of authority based on the most relevant article referenced. Independent sampling from both levels yields an initial set with approximately 1:1 ratio: 3,950 Law-level and 4,050 Administrative-level samples, containing 5,066 single-choice and 2,934 multi-choice questions (see Table 1). This balanced sampling corrects the original imbalance in JEC-QA, where citations to laws significantly outnumber those to administrative regulations.

To focus training on scenarios that genuinely require external retrieval, we refine this initial set using DeepSeek-V3. For each question, we generate $k = 8$ candidate trajectories interleaving `<think>`, `<search>` (querying our knowledge base), `<information>` (retrieved results), and `<answer>`. We prune trajectories with incorrect answers or unsupportive search actions (verified by DeepSeek-V3), and a human expert verifies a random subset for quality. This refinement is applied independently to each level; due to comparable rejection rates, the final high-quality dataset preserves the approximate 1:1 balance. The resulting dataset, characterized by high difficulty and necessary search actions, serves as the foundation for the reinforcement learning stage described next.

Dataset	Question Type		Level of Authority	
	SC	MC	Law	Admin. Reg.
Data Number	5066	2934	3950	4050

Table 1: Statistics of the training data: number of single-choice (SC) and multi-choice (MC) questions, as well as sample counts from the Law level and Administrative Regulation level. The two levels are balanced at approximately a 1:1 ratio.

4 Methodology

4.1 Overview

We formulate legal agentic search as a sequential decision-making process. Given a legal query x, the agent (policy π_θ) interacts with an external search environment E_{search} to gather information, producing a reasoning trajectory τ and final answer y. For multi-step retrieval tasks, deciding when and how to search is critical, yet collecting labeled reasoning data is difficult. Our approach,

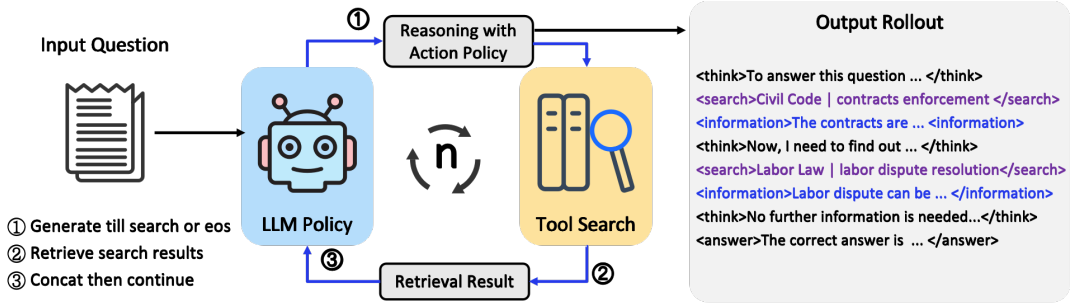


Figure 1: Reinforcement learning training overview of IterLegal: A step-by-step breakdown of the rollout generation process.

IterLegal, uses reinforcement learning to address this by sampling multiple reasoning-with-search chains (rollouts) and optimizing the policy to favor higher-reward trajectories, as illustrated in Figure 1. The figure details the iterative rollout process: the LLM policy generates a reasoning trace step by step; when it decides to retrieve, the generation pauses, the query is sent to the tool search environment, and the retrieved results are appended; the rollout then continues with further reasoning and optional additional searches until the final answer is produced. This think–retrieve–answer loop constitutes the complete trajectory τ, which is used to compute rewards and update the policy via GRPO.

4.2 Reinforcement Learning

We adopt Group Relative Policy Optimization (GRPO) as the learning algorithm, which estimate the baseline from a group of rollouts instead of training a separate critic model in Proximal Policy Optimization (PPO). Given an existing policy $\pi_{\theta_{\text{old}}}$ and a reference policy $\pi_{\theta_{\text{ref}}}$, for each input $s \sim \mathcal{S}$, we sample G rollouts $\tau = \{y_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot|x)$. Each training sample $s = (x, y^*)$ consists of a question x and the model's output y^*, which includes only the answer without reasoning traces. The objective of GRPO is to optimize the policy π_θ by maximizing the following:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\mathbf{s}\sim P(\mathbf{S}),\, \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(O|\mathbf{s})} \frac{1}{G}\sum_{i=1}^{G}\Big\{ \min\left[r_i A_i,\, \text{clip}\left(r_i, 1-\epsilon, 1+\epsilon\right) A_i\right] - \beta\, D_{\text{KL}}\left(\pi_\theta \,\|\, \pi_{\text{ref}}\right)\Big\}, \quad (1)$$

where A_i is the normalized advantage of the i-th rollout in current group, r_i is the reward from old policy $\pi_{\theta_{\text{old}}}$, ϵ is the clipping ratio, and β is the KL loss coefficient. The KL divergence penalty prevents the policy from deviating too far from the reference model.

Reasoning Chain Format The reasoning chain format primarily consists of the tags `<think>`, `<search>`, `<information>`, and `<answer>`. During the training phase, each reasoning trajectory is structured as follows: the `<think>` tag encapsulates the reasoning process, the `<search>` tag triggers the tool-based search for relevant content, and the `<information>` tag stores the retrieved, compressed information. Finally, the `<answer>` tag contains the model's conclusion. This format allows the model to improve both its reasoning abilities and adherence to structured formats, ensuring that the reasoning process is clear, traceable, and contextually relevant.

Introspective Action Space The reasoning process unfolds over T steps. At each step t, the agent generates an action z_t conditioned on input x and history $h_t = (z_1, o_1, \ldots, z_{t-1}, o_{t-1})$. The action space $\mathcal{Z}$ consists of three types: (1) **introspective reasoning** (z_t^{think}), where internal thinking (within `<think>` tags) evaluates whether parametric knowledge is sufficient and transitions accordingly; (2) **active inquiry** (z_t^{search}), triggered when knowledge is deemed insufficient, where the agent issues a search query (within `<search>` tags), receives observation $o_t = \mathcal{E}_{\text{search}}(z_t)$ (wrapped in `<information>`), and loops back to reasoning; and (3) **answer generation** (y), which terminates the process when the accumulated history h_t is sufficient, producing the final legal response. The complete trajectory is $\tau = (z_1^{\text{think}}, [z_1^{\text{search}}, o_1, z_2^{\text{think}}, \ldots], y)$, representing an introspective process of dynamic thinking and active inquiry that concludes upon reaching knowledge sufficiency.

Rollout with Search Compared with conventional rollout Unlike conventional rollouts that only contain text-based thinking, our rollouts also include search queries and retrieval results.

We enclose search queries with <search> and </search>, as described in the prompt templates in Appendix A. The rollout iterates between thinking, search, and retrieval. Specifically, when the generation process encounters </search> tag, the query between the last <search> and current </search> tags will be used as the search query to retrieve relevant factual information, and the retrieval results will be enclosed by <information> and </information> tags. Then, existing rollout concated with the retrieval results will be used as the next input to generate following response iteratively, until the generation encounters end-of-sentence (eos) tag (i.e., <endoftext> or <im_end> in Qwen-2.5 Models).

Following Search-R1, we mask retrieval results in the loss calculation to avoid biasing the policy. That is, when computing Equation 1, only tokens in text-based thinking and search queries are considered; tokens in retrieval results are ignored.

4.3 Reward Function Design

During reinforcement learning of IterLegal, there is no supervised reasoning data, and we only use a simple reward on rollouts to guide the optimization of LLMs. Experimentally, only rule-based reward function is enough to successfully elicit capabilities of reasoning with search for LLMs. Our reward function considers following three parts: accuracy reward, format reward and legal validity reward.

Accuracy Reward We extract the content within <answer>...</answer> tags using regular expressions and compare it to the reference answer a^*. A score of 1 is given if the extracted answer a is semantically consistent with a^*; otherwise 0.

$$R_{\text{Acc}}(a, a^*) = \begin{cases} 1, & \text{if } a = a^* \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Format Reward We encourage outputs that correctly use the required tags throughout the entire reasoning trajectory. A trajectory may contain multiple rounds of <think> followed by <search> (and corresponding <information>), and must end with a final <answer>. The format reward is 1 if all tag pairs are properly closed, no extraneous text appears outside the tags, and the overall structure follows the pattern described in Section 4.2; otherwise 0.

$$R_{\text{Fmt}}(y) = \begin{cases} 1, & \text{if the format matches} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Legal Validity Reward Legal answers demand precision and contextual correctness. We employ an instruct model as a verifier to assess whether the reasoning output aligns with the intended legal solution, following criteria detailed in Appendix E. The verifier considers principles such as legal syllogism (major premise, minor premise, conclusion) and formal citation standards. It assigns a score of 2 (fully consistent), 1 (partially consistent), or 0 (inconsistent). which typically compares final answers, this verifier operates on the reasoning content extracted from the <think>...</think> tags. Let r denote the extracted reasoning and r^* the reference reasoning. The reward is defined as:

$$R_{\text{Legal}}(r, r^*) = \begin{cases} 2, & \text{if } r \text{ consistent with } r^* \\ 1, & \text{if } r \text{ partially consistent with } r^* \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

The total reward is computed as the weighted sum of the above rewards, formulated as follows:

$$\mathcal{R} = \alpha R_{\text{Acc}} + \beta R_{\text{Fmt}} + \gamma R_{\text{Legal}}, \tag{5}$$

where $\alpha = 0.9$, $\beta = 0.1$, and $\gamma = 0.1$.

5 Experiments

Benchmarks To evaluate the effectiveness of IterLegal, we conduct experiments on three standard benchmarks on multi-hop question answering tasks, including LawBench (Fei et al., 2024) dataset and LexEval (Li et al., 2024), and Unilaw-R1-Eval (Press et al., 2023). LawBench assesses the legal capabilities of LLMs across three cognitive levels: memory, understanding, and application. It comprises 20 tasks with various formats, including multiple-choice, extraction, generation, and regression, simulating real-world legal scenarios such as statute prediction, case analysis, and legal consultation. LexEval, the largest and most comprehensive Chinese legal benchmarking dataset, evaluates performance of LLMs across six cognitive abilities defined by the LexCog taxonomy: memory, understanding, logical reasoning, discrimination, generation, and ethics. It consists of 14,150 entries across 23 legal tasks, providing a diverse set for evaluating LLM performance. Unilaw-R1-Eval consists of 800 samples divided into case and knowledge driven question types, also covering a broad spectrum of legal domains allowing for domain-specific performance insights.

We evaluate our model in zero-shot settings. The inputs to the LLMs are only instructions and

queries. We use Accuracy and F1 to to evaluate the Unilaw-R1-Eval data. For LawBench and LexEval datasets, we employ automated evaluation methods tailored to the diverse task types within their benchmarks, ensuring objective and consistent assessment of large language models in legal contexts.

Baselines To comprehensively evaluate the reasoning capabilities of IterLegal in legal scenarios, we conducted a thorough comparative assessment against multiple baseline models. These models include DeepSeek-R1, DeepSeek-V3, DeepSeek-R1-Distill-Qwen-32B, DeepSeek-R1-Distill-Qwen-14B, DeepSeek-R1-Distill-Qwen-7B, Qwen-2.5-32B-Instruct, Qwen-2.5-14B-Instruct, Qwen-2.5-7B-Instruct, Unilaw-R1. The selection of these models encompasses a spectrum ranging from lightweight to high-performance architectures, taking into account factors such as reasoning capability and computational resource consumption. This comprehensive comparison aims to provide a holistic evaluation the performance of IterLegal within legal applications.

Evaluation Metrics For evaluate the correctness of the final answer, we first use Exact Match (*EM*) where the prediction is correct if it matches the ground truth answer exactly. However, such exact match is too strict for our setting, since the retrieval environment is open-ended and the result is described by natural language.

Implementation Details We train our model based on Qwen2.5-7B-Instruct using a reinforcement learning framework built on verl (Sheng et al., 2024). We train on approximately 8,000 samples (constructed as described in Section 3.2) for two epochs. The retrieval environment is implemented with the hybrid mode of PyMilvus (Wang et al., 2021) using bge-m3 embedding (Chen et al., 2024a) as both the dense and sparse encoder, combining semantic vector search with keyword search via a weighted reranker. A collection of legal articles serves as the knowledge base. All corpus indexing and embedding are preprocessed before training. During rollouts (both training and evaluation), we retrieve the top-3 results for each query and limit the model to at most three search actions per trajectory. In the ablation study (Section 5.2), we relax this limit to ten to explore the effect of increased search numbers. For policy optimization, we employ Group Relative Policy Optimization (GRPO) with a group size of 4, which integrates a model-based reward signal from a verifier (Qwen2.5-7B-Instruct) with policy optimization to ensure legal accuracy and reasoning consistency. All training and testing were performed on machines equipped with 8 $\times$ 96 GB NVIDIA H20 GPUs. Key hyperparameters and additional implementation details are provided in Appendix C.

5.1 Main Results

The main results of baselines and IterLegal are demonstrated in Table 2, and we show the methods based on LLMs with different sizes respectively. From the main results, we can draw the following observations:

Effectiveness of IterLegal Compared with all the baselines, IterLegal achieves significant improvements on all the benchmarks, which demonstrates the effectiveness of our proposed framework. Despite its lightweight 7B parameter size, IterLegal achieved remarkable performance, attaining an average score of 53.8%. It outperformed all other models of similar scale and even matched the performance of the much larger DeepSeek-R1-Distill-Qwen-32B (54.9%). After reinforcement training on domain-specific data, IterLegal showed notable performance gains on legal benchmarks such as LawBench and LexEval, surpassing Qwen-2.5-7B-Instruct (46.6%) by an average margin of 7.2%. These results underscore the strong cross-task generalization and effectiveness of IterLegal in legal applications.

Generalization Ability During reinforcement learning, IterLegal learns the ability of reasoning with search, which is independent of specific knowledge or multi-hop patterns, and such ability is generalizable. Our model IterLegal is only trained on the selected JEC-QA Training dataset, but from the results, we can observe that it can generalize to other benchmarks with different question types and structures, which demonstrates the generalization ability of IterLegal.

5.2 Ablation Study

We first compare IterLegal against two naive baselines: (1) *Zero-shot CoT*, where the instruction-tuned model generates answers directly without any retrieval augmentation; and (2) *Naive RAG*, a standard retrieval-augmented setup in which retrieved passages are simply concatenated with the question before answer generation. Additionally, we

Model	Parameters	LawBench	LexEval	Unilaw-R1-Eval	Avg.(%)
DeepSeek-R1	671B	61.8	67.2	55.2	61.4
DeepSeek-V3	671B	61.3	65.7	50.6	59.2
DeepSeek-R1-Distill-Qwen-32B	32B	57.0	65.2	42.6	54.9
Qwen-2.5-32B-Instruct	32B	63.8	66.9	42.2	57.6
DeepSeek-R1-Distill-Qwen-14B	14B	51.8	54.8	24.0	43.5
Qwen-2.5-14B-Instruct	14B	58.3	64.3	29.4	50.6
DeepSeek-R1-Distill-Qwen-7B	7B	38.3	47.3	23.6	36.4
Qwen-2.5-7B-Instruct	7B	52.3	57.8	29.9	46.6
Unilaw-R1	7B	56.6	63.5	39.5	53.2
IterLegal	7B	54.3	64.8	42.3	53.8

Table 2: Accuracy evaluation of IterLegal on different legal benchmarks.

consider two representative approaches designed to enhance multi-step RAG: (3) *IRCoT* (Trivedi et al., 2023), an interleaving method where retrieval and chain-of-thought (CoT) reasoning mutually guide each other; and (4) *ReAct* (Shao et al., 2023), which iteratively interleaves reasoning traces with tool-use actions (here limited to a search operation). Since all these baselines are prompt-based methods, we implement them using the Qwen2.5-7B-Instruction model to ensure a fair comparison with our IterLegal, which shares the same backbone size. The prompt as described in Appendix F.

Model	Unilaw-R1-Eval		Avg.(%)
	SC	*MC*	
Zero-shot CoT	35.68	20.86	28.75
Naive RAG	31.93	17.38	25.13
IRCoT	33.57	22.73	28.50
ReAct	38.73	24.57	32.63
IterLegal(w/o Legal)	40.32	35.57	36.40
IterLegal	46.40	38.56	42.32

Table 3: Single Choice (SC, %) and Multi Choice (MC, %) results on Unilaw-R1-Eval. The best results are highlighted in bold, and the best results across baselines are underlined.

Training with Validation Reward The performance of IterLegal when the legal validation reward is removed is reported in Table 3. Specifically, the ablated variant *IterLegal (w/o Legal)* achieves scores of 40.32%, 35.57%, and 36.40% across the three evaluation metrics, respectively. In contrast, the full IterLegal model attains scores of 46.40%, 38.56%, and 42.32%. The observed declines across all metrics (e.g., a drop of over 6 points on the first metric and approximately 6 points on the third) underscore the critical role of the legal validation reward in guiding the model's reasoning process. Without this domain-specific supervisory signal, the model exhibits a notable degradation in its ability to ground its responses in valid legal provisions, resulting in less accurate and less coherent answer generation. This comparison provides strong empirical evidence that incorporating a specialized legal verifier as a reward component substantially enhances the model's performance on complex legal reasoning tasks.

Response Length We define response length as the total number of tokens generated by the model, excluding retrieved content, which reflects the test-time reasoning cost. As shown in the left panel of Figure 2, the response length exhibits a distinct *non-monotonic* trend during training. In the initial exploration phase (0–250 steps), the length gradually increases as the model generates more extensive reasoning traces. This is followed by a sharp decline between steps 250 and 450, indicating that reinforcement learning encourages the model to internalize reasoning and reduce verbosity. Notably, a significant *phase transition* occurs after step 500, where the response length surges abruptly (e.g., from ∼40 to ∼430 tokens). However, this increase does not translate into higher accuracy; in fact, the model's performance peaks during the mid-training compression phase (around step 450), suggesting that excessive length and search operations may lead to over-thinking or degraded reliability.

Number of Search Operations We also track the average number of search operations per rollout during training, shown in the right panel of Figure 2. Consistent with the response length dynamics, the number of search operations follows

Question: According to the *Regulations of the People's Republic of China on Anti-Dumping*, as revised on March 31, 2004, which authority initiates and determines whether an anti-dumping investigation should be conducted regarding foreign products allegedly being dumped in China?

A. General Administration of Customs **B.** State Tax Commission
C. Ministry of Commerce **D.** Ministry of Agriculture

Answer: ["C"]

<think>According to the question, we need to identify the authority responsible for conducting anti-dumping investigations and making determinations ... investigations and determinations.</think>
<search>Regulations of the People's Republic of China on Anti-Dumping | anti-dumping investigation and determination</search>
<information>[Reference materials retrieved by querying ...] Ref1: Title: Anti-Dumping ... Content: Where an imported ... | Ref2: Title: Anti-Dumping ... Content: Anti-dumping duties shall ... | Ref3: Title: Anti-Dumping ... Content: Where an export operator ...</information>
<think>Based on the reference materials, the Regulations expressly stipulate that the authority responsible for investigating and ... Therefore, only Option C, the Ministry of Commerce, is correct.</think>
<answer>\boxed{C} </answer>

Table 4: A case study of IterLegal during training.

a U-shaped trajectory. It initially decreases from ∼1.8 to a minimum of ∼1.0 around step 450, where the model achieves the relative high accuracy. However, after step 500, the search operations increase dramatically (e.g., rising to 9.0 at step 600), but this does not yield further gains. This suggests that while mid-training optimization promotes frugal and effective search behavior, later stages may lead to excessive tool invocation and reasoning redundancy, harming overall performance.

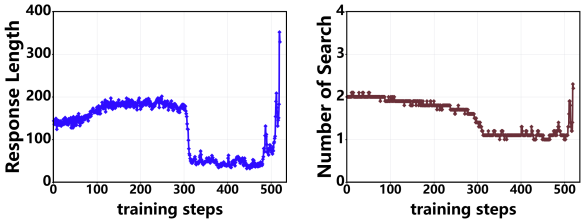


Figure 2: Response length and the number of search operations during training.

5.3 Case Study

To provide more intuitive understanding of IterLegal's behavior, we present a case study in Table 4, drawn from a reinforcement learning trajectory. For brevity, retrieval results are truncated using "...". As illustrated, the model decomposes the complex question, identifies key legal terms (e.g., "anti-dumping investigation"), and reasons within the <think> tags. This reasoning process is crucial for deciding when and what to search. In this instance, the model generates a targeted search query that combines the regulation name with the core task of investigation and determination. After retrieving the relevant legal provisions, the model directly extracts the statutory rule that designates the *Ministry of Commerce* as the responsible authority and applies it to the question, concluding with the correct answer (Option C). This demonstrates IterLegal's ability to integrate external legal knowledge into its reasoning chain, which is an iterative, on-demand retrieval capability that distinguishes it from static baselines. See the bad case study in the appendix (Section G) for failure examples.

6 Conclusion

In this paper, we presented IterLegal, a novel framework that trains large language models to reason with search via reinforcement learning, without requiring any supervised data on reasoning steps. Our approach integrates search operations as intrinsic components of the reasoning chain: the model proactively decides when and how to search, and iteratively incorporates retrieved results to guide subsequent reasoning. Extensive experiments on multiple multi-hop question answering benchmarks demonstrate that IterLegal achieves significant improvements over baseline methods. The results also highlight the framework's potential for more realistic application scenarios. Furthermore, analysis of the training process reveals that IterLegal naturally elicits advanced reasoning capabilities without relying on predefined heuristics.

Limitations

Despite notable advancements, our model has several limitations.

Limited Training Data. Current training is confined to objective multiple-choice questions and remains below the desired scale. Future work will expand to broader legal datasets.

Single-Modality Architecture. The text-only model cannot process legal documents with visual elements (e.g., charts, tables). Multimodal extensions are planned.

Insufficient CoT Evaluation. Our evaluation compares final answers to references but does not assess intermediate reasoning steps. Future work will analyze structured legal reasoning (e.g., syllogistic logic) to align with legal standards.

We are committed to addressing these limitations, expanding to emerging domains, and enhancing real-world impact in legal risk management and compliance.

References

Anthropic. 2025. Claude 3.7 sonnet and claude code.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *ICLR*. OpenReview.net.

Hua Cai, Shuang Zhao, Liang Zhang, Xuli Shen, Qing Xu, Weilin Shen, Zihao Wen, and Tianke Ban. 2025. Unilaw-r1: A large language model for legal reasoning with reinforcement learning and iterative inference. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18128–18142.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. 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.

Mingyang Chen, Haoze Sun, Tianpeng Li, Fan Yang, Hao Liang, Keer Lu, Bin Cui, Wentao Zhang, Zenan Zhou, and Weipeng Chen. 2024b. Facilitating multi-turn function calling for llms via compositional instruction tuning. *CoRR*, abs/2410.12952.

Xin Dai, Buqiang Xu, Zhenghao Liu, Yukun Yan, Huiyuan Xie, Xiaoyuan Yi, Shuo Wang, and Ge Yu. 2025. LegalΔ: Enhancing legal reasoning in llms via reinforcement learning with chain-of-thought guided information gain. *arXiv preprint arXiv:2508.12281*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 81 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Alan Huang, Songyang Zhang, Kai Chen, Zhixin Yin, Zongwen Shen, and 1 others. 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.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *CoRR*, abs/2312.10997.

Zhitian Hou, Zihan Ye, Nanli Zeng, Tianyong Hao, and Kun Zeng. 2025. Large language models meet legal artificial intelligence: A survey. *arXiv preprint arXiv:2509.09969*.

Yinghao Hu, Leilei Gan, Wenyi Xiao, Kun Kuang, and Fei Wu. 2025. Fine-tuning large language models for improving factuality in legal question answering. In *Proceedings of the 31st international conference on computational linguistics*, pages 4410–4427.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7036–7050.

Zhengbao Jiang, Frank F 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.

Haitao Li, You Chen, Qingyao Ai, Yueyue Wu, Ruizhe Zhang, and Yiqun Liu. 2024. Lexeval: A comprehensive chinese legal benchmark for evaluating large language models. *arXiv preprint arXiv:2409.20288*.

Mingan Lin, Fan Yang, Yanjun Shen, Haoze Sun, Tianpeng Li, Tao Zhang, Chenzheng Zhu, Tao Zhang, Miao Zheng, Xu Li, Yijie Zhou, Mingyang Chen, Yanzhao Qin, Youquan Li, Hao Liang, Fei Li, Yadong Li, Mang Wang, Guosheng Dong, and 6 others. 2024. Baichuan alignment technical report. *CoRR*, abs/2410.14940.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others.

2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. 2024. Agentboard: An analytical evaluation board of multi-turn LLM agents. In *NeurIPS*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel J. Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *CoRR*, abs/2501.19393.

OpenAI. 2024. Learning to reason with LLMs.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *EMNLP (Findings)*, pages 5687–5711. Association for Computational Linguistics.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, and 1 others. 2024. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *International Conference on Learning Representations*, volume 2024, pages 9695–9717.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In *NeurIPS*.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *EMNLP (Findings)*, pages 9248–9274. Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving AI tasks with chatgpt and its friends in hugging face. In *NeurIPS*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient RLHF framework. *CoRR*, abs/2409.19256.

Weijie Shi, Han Zhu, Jiaming Ji, Mengze Li, Jipeng Zhang, Ruiyuan Zhang, Jia Zhu, Jiajie Xu, Sirui Han, and Yike Guo. 2025. Legalreasoner: Step-wised verification-correction for legal judgment reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7297–7313.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *CoRR*, abs/2408.03314.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *ACL (1)*, pages 10014–10037. Association for Computational Linguistics.

Jianguo Wang, Xiaomeng Yi, Rentong Guo, Hai Jin, Peng Xu, Shengjun Li, Xiangyu Wang, Xiangzhou Guo, Chengming Li, Xiaohai Xu, and 1 others. 2021. Milvus: A purpose-built vector data management system. In *Proceedings of the 2021 international conference on management of data*, pages 2614–2627.

Junjie Wang, Mingyang Chen, Binbin Hu, Dan Yang, Ziqi Liu, Yue Shen, Peng Wei, Zhiqiang Zhang, Jinjie Gu, Jun Zhou, Jeff Z. Pan, Wen Zhang, and Huajun Chen. 2024. Learning to plan for retrieval-augmented large language models from knowledge graphs. In *EMNLP (Findings)*, pages 7813–7835. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Huiyuan Xie, Chenyang Li, Huining Zhu, Chubin Zhang, Yuxiao Ye, Zhenghao Liu, and Zhiyuan Liu. 2026. Lexchain: Modeling legal reasoning chains for chinese tort case analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 35913–35921.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *CoRR*, abs/2401.15884.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Xinyu Yang, Chenlong Deng, and Zhicheng Dou. 2025. Glare: Agentic reasoning for legal judgment prediction. *arXiv preprint arXiv:2508.16383*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. Star: Bootstrapping reasoning with reasoning. In *NeurIPS*.

Kepu Zhang, Weijie Yu, Zhongxiang Sun, and Jun Xu. 2025. Syler: A framework for explicit syllogistic legal reasoning in large language models. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 4117–4127.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. Dense text retrieval based on pretrained language models: A survey. *ACM Trans. Inf. Syst.*, 42(4):89:1–89:60.

A Training Template

Our rollout process relies on recognizing predefined special tags (e.g., pausing at `</search>` to invoke the search environment). Therefore, it is crucial that the policy LLM generates outputs strictly following the expected format. To teach the model this format (in particular, the tags indicating when the search operation is invoked), we design a prompt template for the instruction-tuned model, as illustrated in Figure 3. Inspired by DeepSeek-R1, the template is kept simple and concise, allowing the model to naturally follow the required structure during reinforcement learning. Specifically, for the instruction-tuned model, this prompt serves as the system prompt and is used together with the model's native chat template.

B Prompt for Data Construction

This section describes the three prompts used with DeepSeek-V3 to construct our training data. These prompts address the following tasks: (1) determining the primary level of authority (Law or Admin) for questions that cite multiple legal sources, (2) generating structured reasoning trajectories interleaved with search actions, and (3) verifying whether retrieved information supports a given reasoning step. Table 7 presents the exact prompt templates for each task.

C Details of Training Setup

We provide detailed training configurations used in Reinforcement Learning (RL) of IterLegal. We employ Group Relative Policy Optimization (GRPO) with a group size of 4, which combines a model-based reward signal with policy optimization to ensure legal accuracy and reasoning consistency, to learn the `<think>` ... `</think>\n\n<answer>` ... `</answer>` format. The reward signal is generated by a verifier model (Qwen2.5-7B-Instruct) based on legal principles. All our training and test results were performed on machines equipped with 8 $\times$ 96 GB NVIDIA H20 GPUs. Key hyperparameters for both stages are summarized in Table 5.

Parameter	RL(GRPO)
Batch Size	32
Epochs	2
Learning Rate	1.0e-6
Warmup Ratio	0.03
Max Sequence Length	4096
Gradient Accumulation	4
Optimizer	AdamW
Weight Decay	0.01
LR Scheduler	Cosine
Evaluation Interval	10 steps
Reward Signal	Acc & Fmt & Legal
Reward Granularity	Step-level
Rollout Temperature	1.0
Rollout Samples	5
KL Coefficient (β)	1.0e-2
Clip Parameter (ϵ)	1.0e-6

Table 5: Training hyperparameters for GRPO Reinforcement learning.

D Legal Knowledge Base Statistics

The constructed legal knowledge base comprises **1,074 legal documents** and **54,589 articles**, with an average of approximately **50.8 articles per document**. Tables 6 and 8 summarize the distribution by level of authority and by legal field, respectively.

D.1 By Level of Authority

As shown in Table 6, the category *Laws* contains the largest number of documents (455, 42.4%) and articles (32,872), with an average of 72.2 articles per law. *Administrative Regulations* ranks second (307 documents, 13,413 articles, average 43.7). *Departmental Rules* have the highest average length (104.3 articles per document), indicating detailed provisions. In contrast, *Legal Interpretations* (29 documents, average 1.5 articles) and *Major Decisions on Legal Issues* (57 documents, average 4.9 articles) are much shorter, reflecting their supplementary or amendatory nature. *Judicial Interpretations* (147 documents, 5,081 articles, average 34.6) demonstrate the refinement of judicial practice.

Level of Authority	Docs	Articles(Avg.)
Constitution	7	48.3
Laws	455	72.2
Legal Interpretations	29	1.5
Administrative Regulations	307	43.7
Judicial Interpretations	147	34.6
Normative Documents	22	37.8
Departmental Rules	6	104.3
Judicial Documents	12	12.0
International Treaties	2	82.5
Industry Standards	4	60.5
Decisions on Amendment/Repeal	26	21.2
Major Decisions on Legal Issues	57	4.9

Table 6: Statistics by Level of Authority

D.2 By Legal Field

Table 8 shows coverage across 13 legal fields. *Administrative Law* has the most documents (279, 15,459 articles), accounting for 26.0% of the total, reflecting the large and complex administrative

Training Prompt

You are a helpful assistant that solves legal questions step by step using a legal knowledge search tool (for statutes and provisions).

Given a question, first think through the reasoning process inside `<think> </think>` tags. If available information is insufficient, your thinking should include what keywords to search for, specifying the needed legal provisions or clauses. You may output up to two keywords. Each keyword must be concise and independent, using the format `law|clause` (where `|` separates the law name and the clause). Multiple keywords must be separated by semicolons (`;`). Do not output any extra content. Place your search query inside `<search> </search>` tags.

Example: `<search>Civil Code of the PRC|disputes over private lending;Countervailing Regulations|definition of subsidy</search>`

Search results will be returned inside `<information> </information>` tags. You may search multiple times until sufficient information is obtained. When no more information is needed, place your final reasoning in `<think></think>` and the final answer in `<answer> </answer>`. The final exact answer should be enclosed in `\boxed{}` (LaTeX format).

Question: **{question}**

Figure 3: Prompt templates used as the system prompt for training from instruction-tuned model. question will be replaced with the actual question.

Task	Prompt
Primary Level Classification	You are a legal expert tasked with analyzing Chinese judicial examination questions. Classify the primary level of authority for each question based on the legal sources cited. - **Law level**: laws, judicial interpretations (treated as equivalent to laws). - **Administrative Regulation level**: administrative regulations. Rules: - If a question cites a judicial interpretation, assign it to the Law level. - If a question cites multiple legal sources, identify the most relevant article referenced and determine its corresponding level as the primary level. Output only "Law" or "Admin". # Question: **{question}** **`Primary Level:`**
Reasoning Trajectory Generation	You are an AI legal assistant that uses iterative search to answer legal questions. Follow the format exactly. For each step, you may: - **`Think:`** <think> reasoning </think> - **`Search:`** <search> query </search> (The system will automatically retrieve from the legal knowledge base and return **`<information>`** retrieved text **`</information>`**) - **`Answer:`** <answer> final answer </answer> You may repeat <think> and <search> cycles as needed. Only search when you lack knowledge. Stop and output <answer> when you have sufficient information. # Question: **{question}**
Support Verification	You are evaluating a legal reasoning trace. Determine whether retrieved information supports the subsequent reasoning step. - **Search query:** {query} - **Retrieved information:** {information} - **Subsequent reasoning:** {next_reasoning} Answer "True" if the retrieved information is used or clearly beneficial; otherwise answer "False". **`Support:`**

Table 7: Prompts for data construction using DeepSeek-V3.

legal system in China. *Constitutional Law* (190 documents, 6,390 articles) ranks second, followed by *Economic Law* (137, 7,715) and *Commercial Law* (86, 4,913). In terms of average length, *Civil Procedure* (30 documents, avg. 75.5 articles), *Civil Law* (46, avg. 74.7), and *Environmental Law* (44, avg. 68.6) are the most detailed. *International Law* (60, avg. 32.3) and *Administrative Procedure* (23, avg. 30.6) have lower averages, which may be due to a higher proportion of principle-based provisions or international treaties.

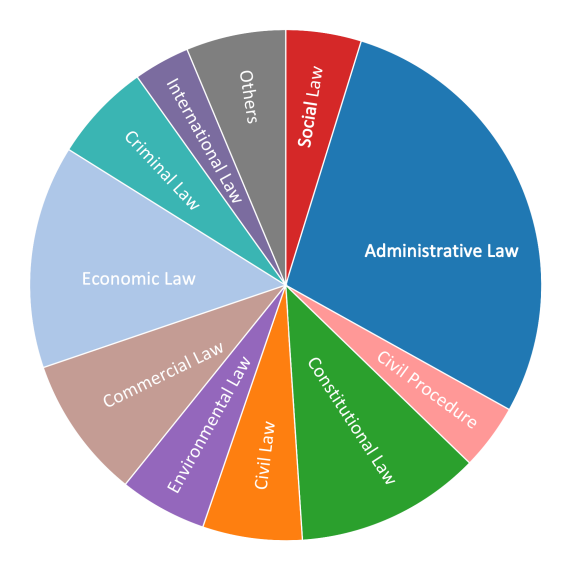
These statistics indicate that the knowledge base has good coverage and internal diversity in both authority levels and legal fields, serving as an effective external knowledge source for active retrieval in downstream legal reasoning tasks.

Legal Field	Docs	Articles(Avg.)
Criminal Law	85	40.1
Criminal Procedure	27	59.7
Social Law	43	60.6
Commercial Law	86	57.1
International Law	60	32.3
Constitutional Law	190	33.6
Civil Law	46	74.7
Civil Procedure	30	75.5
Environmental Law	44	68.6
Intellectual Property	23	48.8
Economic Law	137	56.3
Administrative Law	279	55.4
Administrative Procedure	23	30.6

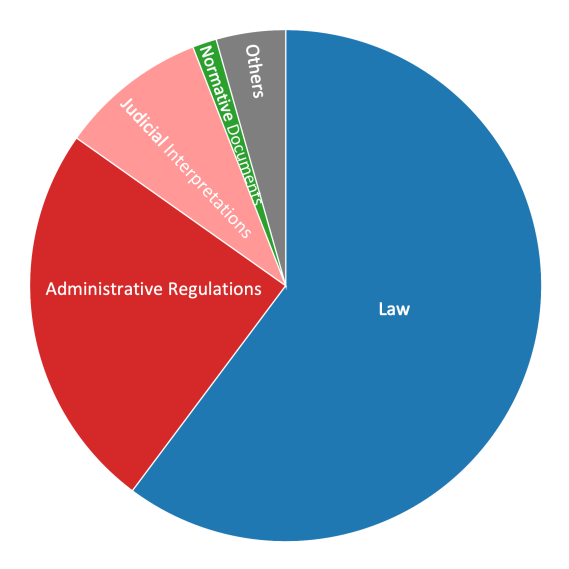
Table 8: Statistics by Legal Field

D.3 Proportional Distribution of Legal Articles

Figure 4 illustrates the distribution of total articles by legal field and by level of authority. Subfigure 4a shows that Administrative Law accounts for the largest proportion (28.3%), followed by Economic Law (14.1%) and Constitutional Law (11.7%). Civil Law and Civil Procedure together represent approximately 10.4%, while Criminal Law and Criminal Procedure contribute about 9.2%. As presented in Subfigure 4b, Laws dominate the knowledge base with 60.2% of all articles, followed by Administrative Regulations (24.6%) and Judicial Interpretations (9.3%). Other categories, including the Constitution, Legal Interpretations, and Departmental Rules, collectively account for less than 6%. These statistics demonstrate a strong emphasis on public law and economic regulation, with statutory laws and administrative regulations being the primary sources of legal provisions.



(a) Proportion of articles by legal field.



(b) Proportion of articles by level of authority.

Figure 4: Proportion of Total Articles by Level of Authority and by Legal Field in Legal Knowledge Base.

E Prompt of Legal Validity Reward

To improve the legal correctness of model outputs during reinforcement learning, we incorporate a model-based feedback mechanism. Specifically, we employ Qwen2.5-7B-Instruct as a verifier to evaluate the reasoning trajectories generated by the policy model. The verifier assesses each response based on predefined legal reasoning criteria, including logical consistency, legal validity, and alignment with expected outcomes.

The model-based feedback serves as a reward signal in the RL fine-tuning stage, either replacing or complementing traditional rule-based or reference-based rewards. This strategy enables the training process to dynamically adjust to nuanced legal judgments rather than relying solely on static ground-truth answers. By leveraging the LLM's own legal reasoning capabilities, we obtain a more flexible and context-aware reinforcement signal that supports the development of high-quality, legally sound responses.

The evaluation criteria for the Legal Validity prompt place additional emphasis on syllogistic reasoning in legal contexts, specifically, applying legal rules to case facts to derive conclusions.

- **Choice Analysis:** This emphasizes completeness by systematically analyzing each option either sequentially or in groups, with all answer choices explicitly considered. Inaccurate or incomplete analysis may indicate failures

in this structured deductive reasoning process, particularly when syllogistic reasoning is required.

- **Legal Format:** This assesses the accuracy and consistency of cited legal provisions, including article numbers and their content, which should match official legal texts. It also requires writing out the full law names instead of abbreviations.

F Prompts for Baseline Evaluation

Throughout our experiments, we designed prompts for different baseline methods to ensure fair comparison. These prompts guide the model to produce outputs in the required format for each paradigm. The following subsections present the exact prompts used for standard Chain-of-Thought (CoT), single-step RAG, and multi-step RAG.

F.1 Standard CoT Prompt

Standard CoT does not involve any retrieval. The model is instructed to generate step-by-step reasoning solely based on its internal knowledge, then output the final answer. Figure 5 shows the prompt template.

The Prompt for Standard CoT

You are a legal expert with specialized proficiency in statutory interpretation and legal analysis. Please answer the following legal question.

Question: **{question}**

Figure 5: The prompt for standard CoT.

F.2 Single-step RAG Prompt

For single-step RAG, the model first retrieves relevant information from the legal knowledge base using a search query, then produces the answer based on the retrieved content. The retrieval is performed only once before the final answer. Figure 6 illustrates the prompt.

The Prompt for Standard RAG

You are a legal expert with specialized proficiency in statutory interpretation and legal analysis. Please answer the following legal question.
An external knowledge base provides relevant legal provisions as references. Please make full use of this information for reasoning and answering.

Reference Materials: **{retrieval_information}**
Question: **{question}**

Figure 6: The prompt for single-step RAG.

F.3 Multi-step RAG Prompt

For multi-step RAG, we adopt two representative paradigms: IRCoT and ReAct. IRCoT alternates between query generation and retrieval, using the collected information to guide subsequent reasoning. ReAct follows a Thought–Action–Observation loop, where each Thought directs the next Action (`SearchLaw` or `Finish`). Both methods support multiple search steps and are designed to mimic human-like legal research. Their corresponding prompts are shown in Figure 7.

The Prompt for IRCoT

Reason Prompt:
You are a legal expert with specialized proficiency in statutory interpretation and legal analysis. Based on the user's question and the existing retrieval information, please generate an appropriate search query to locate relevant legal provisions.

Question: **{question}**
Existing Retrieval information: **{retrieval_information}**
Based on the above information, generate a concise and relevant search query to better locate the answer. If the existing information is already sufficient, output "Sufficient data accumulated."

Answer Prompt:
You are a legal expert with specialized proficiency in statutory interpretation and legal analysis. Please answer the following legal question.An external knowledge base provides relevant legal provisions as references. Please make full use of this information for reasoning and answering.
Reference materials: **{retrieval_information}**
Question: **{question}**

(a) The prompt for IRCoT.

The Prompt for ReAct

You are a legal expert with specialized proficiency in statutory interpretation and legal analysis. Follow these rules strictly:

1. You will receive a legal multiple-choice question and prior tool-use history.
2. Generate exactly **one new Thought** and **one new Action** in the format below.
3. Format:
   Thought: <your reasoning or retrieval direction>
   Action: <SearchLaw["keywords"]> or <Finish["answer(s)"]>
4. Allowed Actions:
   - SearchLaw["keywords"] e.g., Action: SearchLaw["contract void fraud"]
   - Finish["answer(s)"] e.g., Action: Finish["B"] or Action: Finish["A,B"]
5. Finish requirements: output only the answer letter(s), no extra text or line breaks.
6. Output nothing outside Thought and Action. Never repeat a previous Thought or Search.

Question: {question}
(Continue from any existing Thought/Action/Observation history below.)
Question: **{question}**
(Note: The question may be followed by existing Thought / Action / Observation history. Please continue from that point.)

(b) The prompt for ReAct.

Figure 7: The prompt for multi-step RAG.

Question: Which of the following statements regarding the court's handling of criminal cases is incorrect?
A. If, after hearing the case, the court deems the evidence insufficient for conviction, it should adhere to the principle of presumption of innocence and declare the defendant not guilty according to law.
B. If, after hearing the case, the court deems the evidence questionable for sentencing, it should make a ruling favorable to the defendant when sentencing.
C. To protect the defendant's legitimate rights, all confessions obtained by investigators outside designated investigation locations should be excluded according to law.
D. During pre-trial conferences, evidence disputed by both the prosecution and defense should be investigated thoroughly; evidence without objection can be investigated more simply.
Answer: ["A,C,D"]

<think>Based on the relevant provisions of the Criminal Procedure Law, analyze the correctness of the options ... Option A conforms to ... Option B conforms to ... Option C conforms to ... Option D conforms to ... Based on the above analysis, option C is incorrect. </think>
<search>Criminal Procedure Law of the People's Republic of China | Presumption of Innocence | Insufficient Evidence</search>
<information>[Related information obtained from searching "Criminal Procedure Law of the People's Republic of China | Presumption of Innocence | Insufficient Evidence"] Ref1: Title: Criminal Procedure Law of the People's Republic of China. Content: ... Ref2: Title: Criminal Procedure Law of the People's Republic of China. Content: ... Ref3: Title: Criminal Procedure Law of the People's Republic of China. Content: ... All cases must be judged based on ... reasonable doubt has been eliminated regarding the facts that have been determined.</information>
<search>Criminal Procedure Law of the People's Republic of China | Exclusionary Rule of Illegally Obtained Evidence | Legal Provisions</search>
<information>[Relevant information obtained by searching "Criminal Procedure Law of the People's Republic of China | Exclusionary Rules for Illegally Obtained Evidence | Legal Provisions"] Ref1: Title: Criminal Procedure Law of the People's Republic of China. Content: ... Ref2: Title: Criminal Procedure Law of the People's Republic of China. Content: ... Ref3: Title: Criminal Procedure Law of the People's Republic of China. Content: ... </information>
<search>Interpretation of the Supreme People's Court on the Application of the Criminal Procedure Law of the People's Republic of China | Pre-trial Conference | Evidence</search>
<information>[Relevant information obtained by searching "Interpretation of the Supreme People's Court on the Application of the Criminal Procedure Law of the People's Republic of China | Pre-trial Conference | Evidence"] Ref1: Title: Criminal Procedure Law of the People's Republic of China. Content: ... Ref2: Title: Criminal Procedure Law of the People's Republic of China. Content: ... Ref3: Title: The Organic Law of the People's Courts of the People's Republic of China, Content: Chapter 2, Organization and Powers of the People's Courts, Article 32, The Supreme People's Court shall interpret how to specifically apply laws and decrees in the trial process.</information>
<answer>Option C is incorrect. To protect the legitimate rights and interests of the defendant, confessions obtained by investigators outside designated investigation areas should exclude the possibility of being illegally obtained, but this is not always the case.</answer>

Table 9: A bad case study of IterLegal during training.

G Bad Case Study

During model training, we observed that the number of search rounds sometimes increased with more training steps. The model also exhibited abnormal behaviors, such as: search actions inconsistent with the thought process, searches not following the search instructions, and retrieval of irrelevant content. Table 9 shows a typical example, from which we identify the following issues:

- In the first `think` process, the model was able to produce an answer, indicating that it believed its internal knowledge was sufficient

to solve the problem, even though its answer was incorrect. However, instead of directly outputting the answer in the *answer* stage, it proceeded with multiple search rounds. This suggests that the search process did not follow the reasoning process.

- Our prompt instructed the model to provide only one keyword per search round. However, subsequent searches contained two keywords; for example, the first round used *Presumption of Innocence* and *Insufficient Evidence*. This shows that the model deviated from the given search instructions.

- The search should have focused on the *Criminal Procedure Law of the People's Republic of China* or related legal sources, such as judicial interpretations of this law by the Supreme People's Court. However, as the search rounds increased, starting from the third round the model retrieved the *Organization Law of the People's Courts of the People's Republic of China*, which is irrelevant to the question. Consequently, the final answer remained incorrect. In other words, adding more search rounds and accumulating more retrieved information did not help the model answer correctly; it might even have the opposite effect.