Unilaw-R1: A Large Language Model for Legal Reasoning with Reinforcement Learning and Iterative Inference

Hua Cai[1] †, **Shuang Zhao**[1], **Liang Zhang**[1], **Xuli Shen**[1,2] †, **Qing Xu**[1], **Weilin Shen**[1], **Zihao Wen**[2], **Tianke Ban**[2]
[1]UniDT, [2]Fudan University

Abstract

Reasoning-focused large language models (LLMs) are rapidly evolving across various domains, yet their capabilities in handling complex legal problems remains underexplored. In this paper, we introduce Unilaw-R1, a large language model tailored for legal reasoning. With a lightweight 7-billion parameter scale, Unilaw-R1 significantly reduces deployment cost while effectively tackling three core challenges in the legal domain: insufficient legal knowledge, unreliable reasoning logic, and weak business generalization. To address these issues, we first construct Unilaw-R1-Data, a high-quality dataset containing ∼17K distilled and screened chain-of-thought (CoT) samples. Based on this, we adopt a two-stage training strategy combining Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL), which significantly boosts the model's performance on complex legal reasoning tasks and supports interpretable decision-making in legal AI applications. To assess legal reasoning ability, we also introduce Unilaw-R1-Eval, a dedicated benchmark designed to evaluate models across single- and multi-choice legal tasks. Unilaw-R1 demonstrates strong results on authoritative benchmarks, outperforming all models of similar scale and achieving performance on par with the much larger DeepSeek-R1-Distill-Qwen-32B (54.9%). Following domain-specific training, it also showed significant gains on LawBench and LexEval, exceeding Qwen-2.5-7B-Instruct (46.6%) by an average margin of 6.6%. Code is available at: https://github.com/Hanscal/Unilaw-R1.

1 Introduction

In recent years, the rapid iteration of large language models (LLMs) has significantly propelled the evolution of artificial intelligence towards artificial general intelligence (AGI). Models such as OpenAI's o1-series (OpenAI Team, 2024) have enhanced their ability for complex reasoning tasks by extending the length of the "chain-of-thought" through an "exploration-reflection-iteration" mechanism. Similar o1-like LLMs, such as QwQ (Qwen, 2025) and Marco-o1 (Zhao et al., 2024b), have demonstrated significant improvements across tasks like mathematics, programming, and logical reasoning.

Although general reasoning models exhibit considerable potential, their application in specialized domains such as legal is limited. Legal reasoning requires not only legal, economic and mathematical knowledge, but also step-by-step and verifiable logic. Existing models face three major challenges: (1) inconsistencies in legal data increase preprocessing complexity and weaken reasoning (Koenecke et al., 2025; Mishra et al., 2025; Sheik et al., 2024; Steging et al., 2023; Aumiller et al., 2021); (2) the black-box nature of LLMs lack transparency, falling short of traceability standards (Wang et al., 2023; Zhao et al., 2024a; Tong et al., 2024; Chaudhary, 2024); and (3) insufficient legal knowledge leads to unreliable or incoherent reasoning processes (Blair-Stanek and Van Durme, 2025; Dahl et al., 2024). Moreover, effective legal reasoning must adhere to both the external validity of codified law and the internal procedural consistency that ensures fairness and predictability in legal interpretation (Zou, 2021; Raz, 2009; Fuller, 1969).

To address these, we introduce Unilaw-R1, a legal reasoning LLM built upon a high-quality legal dataset and optimized through a two-stage training paradigm. Unilaw-R1 overcomes fragmentation, opacity, and generalization issues in legal AI systems. Our key contributions are as follows:

- **High-Quality Legal Reasoning and Eval Dataset**: We propose Unilaw-R1-Data and Unilaw-R1-Eval datasets that constructed from multiple-choice questions. These cover a wide range of legal topics including civil

†Corresponding authors: Hua Cai (hua.cai@unidt.com) and Xuli Shen (xlshen20@fudan.edu.cn)

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 18117–18131
November 4-9, 2025

law, criminal law, administrative law, and procedural law, providing a robust foundation for training and evaluation in legal scenarios.

- **Two-Stage Model Construction Framework**: We introduce a two-stage pipeline involving Supervised Fine-Tuning (SFT) on high-quality Chain-of-Thought (CoT) reasoning data, followed by Reinforcement Learning (RL) with a legal validity reward function integrated into GRPO. This design improves both reasoning accuracy and legal conformity.
- **Explicit Legal Iterative Inference**: Unilaw-R1 incorporates an iterative multi-agent inference strategy, enabling advanced legal decision-making and strong generalization across diverse legal domains.

2 Related Work

The capabilities of large language models have advanced rapidly through innovations in training paradigms and reasoning strategies. The o1-series models (Jaech et al., 2024) introduced iterative "exploration-reflection" mechanisms that lengthen the CoT process, thereby improving reasoning depth. Subsequent efforts such as QwQ (Qwen, 2025), Marco-o1 (Zhao et al., 2024b) and Fin-R1 (Liu et al., 2025) extended this approach across domains including logic, mathematics, and finance. In the legal domain, compliant adaptations of o1-class models, such as HK-O1aw and PatientSeek (HKAIR, 2024; whyhow ai, 2025), have shown the potential of LLMs in simulating human-like legal reasoning. Yu et al. (Yu et al., 2025) further pushed this frontier by employing test-time scaling techniques to enhance performance on legal tasks.

Distinct from the above, DeepSeek-R1 (Guo et al., 2025) takes an efficient reinforcement learning (RL) approach, training LLMs via thousands of steps of unsupervised RL combined with a cold-start corpus and multi-stage curriculum learning. This strategy results in emergent reasoning capabilities and improved readability, highlighting the promise of RL in scaling inference power.

Despite significant advancements, applying LLMs to the legal domain introduces unique challenges due to domain-specific constraints. Previous research has emphasized the necessity for structured legal datasets, transparency, and reliable performance across scenarios, areas where current models still fall short. Unilaw-R1 addresses these gaps through a domain-tailored, multi-stage training framework and an iterative inference strategy, enhancing its capability to navigate the complexities of legal reasoning.

3 Approach

3.1 Overview

We propose a two-stage framework for legal reasoning model construction, as illustrated in Figure 1. In the data generation stage, we construct a high-quality legal reasoning dataset, Unilaw-R1-Data, by leveraging a data distillation approach grounded in DeepSeek-R1 and incorporating an LLM-as-judge filtering mechanism (Xu et al., 2023) to ensure annotation consistency and reasoning rigor. In the model training stage, we develop the Unilaw-R1 model based on Qwen2.5-7B-Instruct (Yang et al., 2024), utilizing Supervised Fine-Tuning (SFT) in combination with the Group Relative Policy Optimization (GRPO) algorithm (Shao et al., 2024). To further enhance reasoning performance, we introduce an iterative inference mechanism with a collaborative Assessor-Reviser agent setup, enabling the model to refine its reasoning trajectory for more accurate, coherent, and legally sound outputs. The overall process ensures that the model delivers structured, standardized outputs aligned with professional requirements.

3.2 Data Construction

We aim to develop Unilaw-R1-Data, a high-quality supervised fine-tuning dataset tailored for the legal domain. To this end, we designed a comprehensive data construction pipeline that filters and refines data for accuracy and reliability. We also rewrite samples to align with the syllogistic reasoning framework common in legal analysis. As shown in Figure 2, the pipeline includes Answer Check, Chain Rewriting, Explanation Generation and Chain Selection, where an LLM evaluates DeepSeek-R1 outputs for correctness and scores the reasoning paths to ensure logical coherence.

3.2.1 Data Source

Unilaw-R1-Data consists of objective question answering entries in the legal domain, drawn from two primary sources: the open-source JEC-QA dataset (Yue et al., 2023) and proprietary data. JEC-QA includes 26, 365 multiple-choice questions, each with a question and four options. The proprietary portion includes 1, 700 multiple-choice ques-

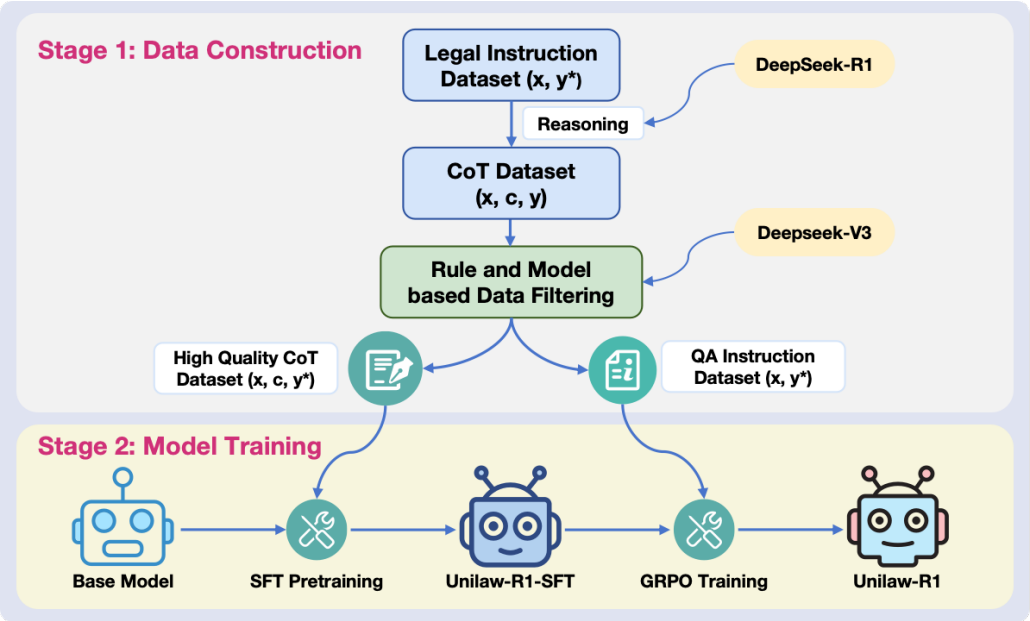

Figure 1: The pipeline for constructing Unilaw-R1. The diagram depicts the two-stage construction framework of Unilaw-R1: Data Generation (using DeepSeek-R1 for reasoning to generate CoT data, followed by quality filtering with the DeepSeek-V3) and Model Training (including SFT pretraining and GRPO optimization for Unilaw-R1).

tion answering entries from National Judicial Examination of China from year 2015 to 2021. These were collected as PDFs, converted to markdown using Mineru (Wang et al., 2024), and structured into question-answer pairs via regex-based extraction. All entries were manually reviewed for accuracy. From our proprietary data, 800 high-quality samples were retained to form the Unilaw-R1-Eval set for model evaluation.

3.2.2 Data Processing

Unilaw-R1-Data was constructed through a rigorous, multi-stage process involving data distillation and filtering. The dataset do not contain any answer explanations. To collect SFT examples, we first distilled multiple-choice questions into a question-thinking-answering format using the reasoning model DeepSeek-R1, following its official parameter configurations during distillation.

Data filtering comprises four key components: **Answer Check**, **Chain Rewriting**, **Explanation Generation** and **Reasoning Selection**. In the Answer Check stage, we retain only those responses that strictly align with the reference answers. Specifically, any response generated by DeepSeek-R1 that diverges from the ground truth in the dataset is immediately excluded. We apply exact match to ensure correctness.

For the exactly matched responses, we sampling 10% of it for Chain Rewriting. This component focuses on restructuring intermediate reasoning chains to ensure they conform to domain-specific logic and legal standards. For the unmatched responses, we sampling 10% of it for Explanation Generation to keep the diverse style. We input both the question and corresponding answer into DeepSeek-V3 and ask it to output the explanation only. We integrate legal rules and definitions as rewriting and explanating constraints to ensure the reasoning paths remain consistent with normative legal interpretations.

All generated chains from Chain Rewriting and Explanation Generation modules, along with those filtered by Answer Check module, are passed into the Reasoning Selection phase to evaluate the plausibility and legal soundness of multiple reasoning trajectories using the instruction model DeepSeek-V3 (Liu et al., 2024). Responses are scored based on their adherence to legal reasoning principles, such as the correct application of rules, consistency with precedent, and logical coherence. These dimensions were employed to comprehensively evaluate the model's reasoning trajectory data. The model's reasoning path must not only lead to the correct answer but also demonstrate a valid and interpretable argumentative structure. When multiple valid paths exist, we prioritize those that align more closely with recognized legal standards and practices. Further details on the experimental setup and findings are provided in Appendix A.

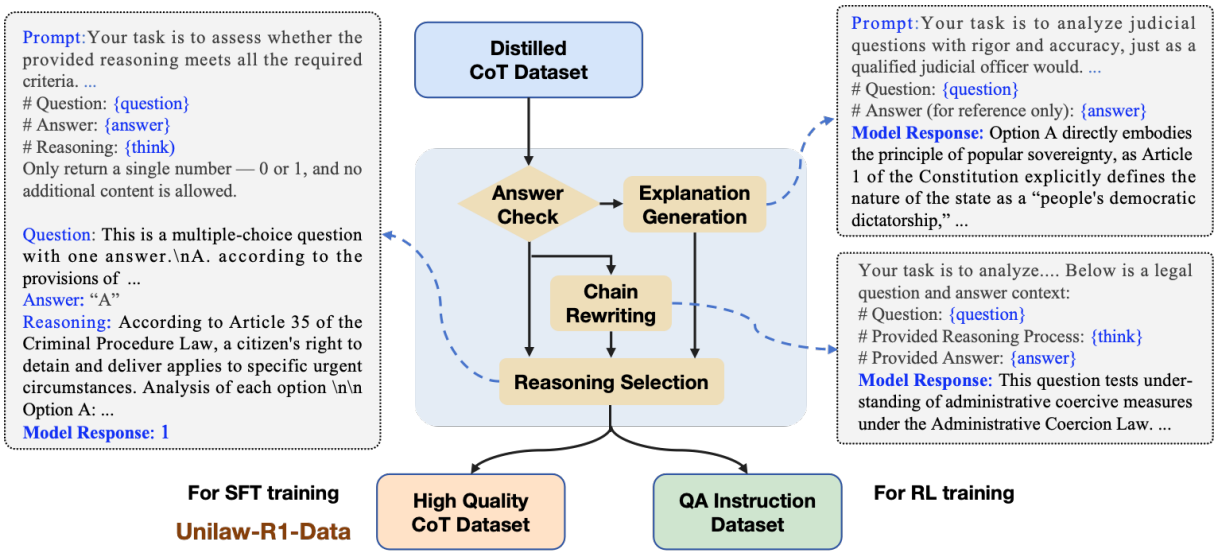

Figure 2: The pipeline of Data Construction (Stage 1): (1) Data Distillation, (2) Data Filtering, including Answer Check and Reasoning Selection, Chain Rewriting, and Explanation Generation. "Reasoning" represents the reasoning output, while "Model Response" refers to the evaluation process of the judgment model.

3.2.3 Data Statistics

After the data processing, we scored and filtered the reasoning paths, retaining only high-quality trajectories to construct the Unilaw-R1-Data for supervised fine-tuning, then randomly selected 8,000 QA entries — half from the unselected pool and half from Unilaw-R1-Data — for reinforcement learning. The Unilaw-R1-Data and Unilaw-R1-Eval datasets is presented in Table 1. The table systematically details the descriptions of these datasets, including the data used stage, the question type, and average token length distribution of prompt, chain of thought reasoning and answer.

The datasets include both knowledge-driven questions and case-based questions. Knowledge-driven questions assess the understanding of legal concepts, while case-based questions focus on the logical analysis of real-world legal scenarios. These two categories comprehensively cover a wide range of legal business scenarios. For evaluation, Unilaw-R1-Eval is categorized into knowledge and case-based subsets, and each question is also labeled with its specific legal domain, further details are provided in Appendix A.5.

Stage	Data Number		Token Length		
	SC	MC	PRM	THT	ANS
Unilaw-R1-Data	9534	7001	332	723	228
Unilaw-R1-Eval	426	374	176	-	2

Table 1: The data statistics for Unilaw-R1-Data and Unilaw-R1-Eval, including the number of single-choice (SC) and multi-choice (MC) questions, as well as the average token lengths for prompts (PRM), chain-of-thought reasoning (THT), and answers (ANS).

3.3 Training Method

Unilaw-R1 is first trained via Supervised Fine-Tuning (SFT) using a high-quality legal reasoning dataset to enhance its reasoning ability. Building on this, we implement Group Relative Policy Optimization (GRPO) reinforcement learning to leverage legal question-answer data, incorporating a triple reward mechanism to improve both the accuracy of response formatting and content. The Stage 2 in Figure 1 intuitively summarizes the comprehensive training framework, illustrating the synergistic integration of the supervised learning and reinforcement learning components. Additional details about the training setup can be found in Appendix C.

3.3.1 Training Data Template

This section outlines the data training format and its role in the subsequent training process.

SFT Training Data During the Supervised Fine-Tuning (SFT) phase, each sample s in the training dataset S comprises three components, i.e., $s = (x, c, y^*)$, where x denotes the question, c represents the reasoning trace formatted as `<think>...</think>`, and y^* corresponds to the answer, formatted as `<answer>...</answer>`. During the SFT stage, x is used as the input of the training set, c and y^* are used as the output of the training set. This phase enables the model to

learn structured legal reasoning patterns, refining its parameters to generate well-formed reasoning traces and accurate answers.

RL Training Data During the reinforcement learning (RL) phase, each sample s in the training dataset S consists of two components, i.e., $s = (x, y^*)$, where x denotes the question and y^* represents the model's output, which includes only the answer without reasoning traces. Reinforcement learning further enhances output quality by improving answer accuracy and ensuring compliance with the expected format.

3.3.2 Supervised Fine-Tuning

We initially performed Supervised Fine-Tuning on Qwen2.5-7B-Instruct using the LoRA efficient parameter tuning method to optimize key aspects of legal reasoning. The fine-tuning was conducted on the Unilaw-R1-Data dataset, incorporating a high-quality CoT reasoning process. This fine-tuning process effectively mitigated the reasoning failures observed when applying the general-purpose model to legal reasoning tasks. Following SFT, the model not only exhibited improved performance in legal reasoning but also learned to generate reasoning trajectories in the <think>...</think> format.

3.3.3 Group Relative Policy Optimization

During the reinforcement learning phase, we employ the Group Relative Policy Optimization (GRPO) algorithm. In each training iteration, G candidate outputs $\{o_i\}_{i=1}^{G}$ are sampled from the old policy π_{old}, each assigned a reward r_i. The group-relative advantage A_i then computed as:

$$A_i = \frac{r_i - \mu_{\{r\}}}{\sigma_{\{r\}}}, \tag{1}$$

where $\mu_{\{r\}}$ and $\sigma_{\{r\}}$ denote the mean and standard deviation of reward values within the group. Outputs exceeding group averages receive higher advantage values for prioritized optimization. The policy update now maximizes the following objective function:

$$\begin{aligned}\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\mathbf{s}\sim P(\mathbf{S}),\, \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(O|\mathbf{s})} \\ \frac{1}{G}\sum_{i=1}^{G}\Bigg\{ \min\left[r_i A_i,\ \text{clip}\left(r_i, 1-\epsilon, 1+\epsilon\right) A_i\right] \\ - \beta\, D_{\text{KL}}\left(\pi_\theta \,\|\, \pi_{\text{ref}}\right)\Bigg\},\end{aligned} \tag{2}$$

where $r_i = \frac{\pi_\theta(o_i|\mathbf{v})}{\pi_{\theta_{\text{old}}}(o_i|\mathbf{v})}$ represents the importance sampling ratio that quantifies the relative likelihood of generating output o_i under the new policy π_θ compared to the old policy $\pi_{\theta_{\text{old}}}$; A_i denotes the group-relative advantage, calculated by normalizing each reward with respect to the group's mean and standard deviation to emphasize outputs that surpass the group average; the clipping operator $\text{clip}(\cdot)$ restricts the update magnitude within the trust region $[1-\epsilon, 1+\epsilon]$ to avoid destabilizing large parameter changes; the minimum operation between the unclipped term $r_i A_i$ and its clipped counterpart ensures a conservative update that balances aggressive improvements with training stability; and finally, $D_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}})$ is the KL divergence and β is the hyper-parameter.

3.3.4 Reward Function Design

In the process of training the reward model based on GRPO, we employs three reward mechanisms: accuracy reward, format reward and legal validity reward.

Accuracy Reward We use the rule-based regular expressions methods to extract the content within the <answer>...</answer> tags from the model's output. This extracted answer is then compared against a reference solution. If the output within the <answer> tags is semantically consistent with the reference answer, a reward score of 1 is assigned; otherwise, it receives a score of 0. The accuracy reward function is defined as follows:

$$R_{\text{Acc}}(y, y^*) = \begin{cases} 1, & \text{if } y = y^* \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where y is model's output (from <answer>... </answer> tags). y^* is the standard answer.

Format Reward We encourage outputs that include a sequence of reasoning steps enclosed within <think>...</think> tags and a concise final answer enclosed within <answer>...</answer> tags. A format incentive score of 1 is awarded if all four tags appear exactly once with no extraneous content outside these tags; otherwise, a score of 0 is assigned. The format reward function is defined as follows:

$$R_{\text{Fmt}}(y) = \begin{cases} 1, & \text{if the format matches} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where y denotes the model's output. Format matching indicates that the output strictly adheres to the specified format by containing exactly one pair of <think> tags and one pair of <answer>

tags, with no additional content outside these tags.

Legal Validity Reward The precise and contextually accurate answers are essential in legal scenarios. To ensure this, we employ an instruct model to evaluate whether the reasoning model's output aligns with the intended legal solution. This approach offers a more robust assessment compared to traditional rule-based methods. The model-based verifier plays a crucial role in ensuring the correctness of responses, particularly in complex and nuanced legal contexts.

The evaluation criteria in the prompt provided to the LLM are largely aligned with those used in the chain rewriting task, and more details are outlined in Appendix B. These instructions include the application of key legal principles, such as **syllogism**, which follows a structure of a major premise (general legal rule), a minor premise (specific case fact), and a conclusion (legal inference). Additionally, the model is required to adhere to formal legal citation standards. Based on the model's output and its adherence to the rules specified in the prompt, the output score can be one of the following values: 2, 1, or 0. The score is determined by the extent to which the model's output aligns with the expected answer. The legal validity reward function is thus defined as follows:

$$R_{\text{Legal}}(y, y^*) = \begin{cases} 2, & \text{if } y \text{ consistent with } y^* \\ 1, & \text{if } y \text{ partially consistent with } y^* \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where y represents the model's output (extracted from the <think>... </think> tags), and y^* is the standard legal answer.

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

$$\mathcal{R} = \alpha R_{\text{Acc}} + \beta R_{\text{Fmt}} + \gamma R_{\text{Legal}}, \quad (6)$$

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

3.4 Iterative Inference

To enhance response quality in legal language generation, we propose an *iterative inference* framework, as shown in Figure 3. It consists of four main stages: sampling, reviewing, refinement, and final answer selection. The reviewing and refinement stages involve a multi-agent setup, with separate Assessor and Revisor agents. These two stages are applied over n iterations to progressively refine candidate responses.

3.4.1 Sampling Chains

Given an input prompt x, we first generate a set of k diverse candidate responses using the post-trained legal reasoning language model $\mathcal{M}_{\text{Unilaw}}$. These candidates are generated by sampling with different parameters to ensure diversity among the outputs:

$$\{y_i^{(0)}\}_{i=1}^{k} \sim \text{Sampling}(\mathcal{M}_{\text{Unilaw}}(x), k) \quad (7)$$

Here, $y_i^{(0)}$ denotes the i-th candidate in the initial generation batch ($Iter = 0$).

3.4.2 Assessing Candidate Responses

Each candidate $y_i^{(t)}$ at iteration t is evaluated using a Assessor agent $\mathcal{K}$, which produces a step-wise quality score and an actionable feedback:

$$fb_i^{(t)} = \mathcal{K}(x, y_i^{(t)}) \quad (8)$$

The agent takes a chain as input, scores each step, and then identifies problematic steps based on these scores. Responses that fall below a predefined threshold score are flagged for refinement, with potential solutions for improvement in the next stage. A one-shot prompt is provided to guide the reviewer on how to score each step in the chain and generate targeted feedback. The prompt for the Assessor can be found in Appendix D.2.

3.4.3 Revising Problematic Responses

A Revisor agent $\mathcal{F}$ is then applied to the selected low-quality responses to improve their relevance, coherence, or correctness. For each low-scoring candidate:

$$y_i^{(t+1)} = \begin{cases} \mathcal{F}(x, fb_i^t, y_i^{(t)}), & \text{if } s_i^{(t)} < \tau \\ y_i^{(t)}, & \text{otherwise} \end{cases} \quad (9)$$

By highlighting specific errors in the reasoning chain, the targeted feedback enables the Revisor to address mistakes more effectively, as it clearly identifies which step are incorrect; We use 1-shot prompt to teach the Revisor how to fix the error and improve a reasoning chain based on targeted feedback. The refined candidate are then passed into the next review iteration. The prompt for the Reviser is shown in Appendix D.2.

3.4.4 Final Answer Selection

This review-refine loop continues for n total iterations. At the end of each iteration, we evaluate whether the refined solutions represent an improvement useing the outcome reward model (ORM).

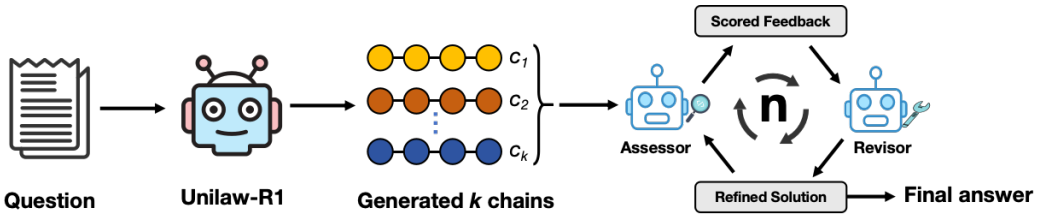

Figure 3: Iterative inference pipeline, consisting of four main stages: sampling, reviewing, refinement, and final answer selection. The reviewing and refinement stages involve a multi-agent setup, with separate Assessor and Revisor agents.

Specifically, we compare the $2k$ reasoning chains - k initial and k refined - and retain the top k based on their global ORM scores. After the final iteration $t = n$, the final answer is selected via self-consistency over the retained top k chains. This iterative inference process effectively combines generation diversity with feedback-driven refinement to produce high-quality legal responses.

4 Experiment

4.1 Datasets

We evaluate our model on Unilaw-R1-Eval dataset and two additional Chinese legal domain multi-task benchmarks: LawBench (Fei et al., 2024) dataset and LexEval (Li et al., 2024). 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.

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.

4.2 Baselines

To comprehensively evaluate the reasoning capabilities of Unilaw-R1 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-SFT and Unilaw-R1-RL. 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 Unilaw-R1 within legal applications.

4.3 Main Results

Table 2 presents the results of our comprehensive benchmarking evaluation across multiple legal business scenarios. Unilaw-R1 demonstrated notable performance advantages despite its lightweight 7B parameter size. Leveraging a two-stage training framework, it achieved an average score of 53.2%. Remarkably, Unilaw-R1 outperformed all participating models of similar scale and even achieved performance comparable to the much larger DeepSeek-R1-Distill-Qwen-32B (54.9%). Following domain-specific training, Unilaw-R1 exhibited significant performance improvements in other legal benchmarks such as LawBench, LexEval, surpassing Qwen-2.5-7B-Instruct (46.6%) by an average margin of 6.6%.

Fine-tuning Qwen-2.5-7B-Instruct on Unilaw-R1-Data and RL data resulted in the Unilaw-R1-SFT and Unilaw-R1-RL models, with average performance improvements of 1.4% and 3.8%, respectively. These results demonstrate strong cross-task generalization and effectiveness in legal applications.

4.4 Ablation Study

We conducted an ablation study to assess the performance impact of different inference strategies for

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-SFT	7B	52.2	58.6	33.3	48.0
Unilaw-R1-RL	7B	54.2	60.6	35.6	50.4
Unilaw-R1	7B	56.6	63.5	39.5	53.2

Table 2: Accuracy evaluation of Unilaw-R1-SFT and Unilaw-R1 on different legal benchmarks.

Method	SC	MC		Avg.
	Acc.(%)	Acc.(%)	F1	Acc.(%)
Zero-shot CoT	53.8	23.2	67.4	39.5
Best-of-k ($k = 10$)	62.3	25.7	67.9	45.2
Majority Vote	56.8	33.1	66.6	45.7
Iterative Infer. ($Iter = 1$)	65.5	33.4	71.9	50.5
Iterative Infer. ($Iter = 2$)	**66.3**	33.8	**72.2**	50.6
Iterative Infer. ($Iter = 3$)	65.7	**34.3**	71.3	**51.0**

Table 3: Performance comparison of Unilaw-R1 with different inference methods on the Unilaw-R1-Eval benchmark.

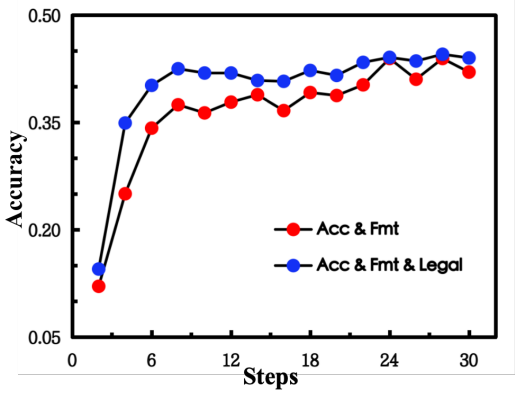

Figure 4: Comparison of convergence behavior of Unilaw-R1-SFT under different combinations of reinforcement learning reward functions on the Unilaw-R1-Eval benchmark.

the Unilaw-R1 model, as well as the convergence behavior of various combinations of reinforcement learning reward functions for the Unilaw-R1-SFT model, using the Unilaw-R1-Eval benchmark.

As shown in Table 3, we compared zero-shot CoT (Wei et al., 2022), best-of-k sampling, majority vote (Wang et al.) and iterative inference methods across single-choice (SC) and multi-choice (MC) tasks. The zero-shot CoT baseline achieved 53.8% accuracy on SC tasks and 23.2% on MC tasks. Implementing best-of-k ($k = 10$) sampling and majority vote led to improvements, raising the average accuracy from 39.5% to 45.2% and 45.7%, respectively. The iterative inference approach demonstrated more substantial gains. With a single iteration, SC accuracy increased to 65.5% and MC to 33.4%. The performance gains form further iterations were limited: the second iteration achieved 66.3% accuracy on SC and 33.8% on MC, while the third iteration reached 65.7% (SC) and 34.3% (MC), respectively. These results indicate that iterative inference significantly enhances model performance, particularly in the first iteration. However, additional iterations offer marginal improvements, suggesting a trade-off between computational cost and performance gains. Therefore, a single iteration of refinement may provide an optimal balance for practical applications.

As shown in Figure 4, we also compared two variants of Unilaw-R1-SFT: one with accuracy and format rewards (Acc & Fmt), and one with an additional legal reward (Acc & Fmt & Legal). The latter showed faster convergence and higher accuracy, highlighting the effectiveness of the legal reward function.

5 Conclusion

We introduce Unilaw-R1, a legal-domain reasoning LLM that combines distilled chain-of-thought data, a two-stage Supervised Fine-Tuning (SFT) followed by Reinforcement Learning (RL) training pipeline, and iterative inference multi-agent setup. This approach addresses data fragmentation, opaque reasoning, and poor generalization, achieving strong performance on legal benchmarks. Additionally, we propose a legal benchmark Unilaw-R1-Eval, which plays a critical role in assessing the model's performance in real-world legal scenarios.

Limitations

Despite notable advancements, our model faces several limitations:

Limited Training Data Coverage: Currently, training data is confined to objective legal multi-choice questions, and it has not yet reached the satisfactory target. Future training will be expanded to a broader range of legal datasets.

Single-Modality Architecture: The model text-only architecture hinders its ability to process legal documents containing visual elements such as charts and tables. We plan to consider multimodal extension to address this limitation.

Insufficient Evaluation of CoT Reasoning: Our current evaluation compares model outputs against referenced answers but lacks analysis of the model's step-by-step legal reasoning. Future evaluations will focus on assessing the model's ability to perform structured legal reasoning, such as syllogistic reasoning, to align with legal standards.

We are committed to addressing the aforementioned limitations, expanding our model's application to emerging domains, and promoting broader adoption to strengthen legal risk management and compliance, ultimately increasing real-world impact and applicability.

References

Dennis Aumiller, Satya Almasian, Sebastian Lackner, and Michael Gertz. 2021. Structural text segmentation of legal documents. pages 2–11.

Andrew Blair-Stanek and Benjamin Van Durme. 2025. Llms provide unstable answers to legal questions. *arXiv preprint arXiv:2502.05196*.

Gyandeep Chaudhary. 2024. Unveiling the black box: Bringing algorithmic transparency to ai. *Masaryk University Journal of Law and Technology*, 18(1):93–122.

Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E Ho. 2024. Large legal fictions: Profiling legal hallucinations in large language models. *Journal of Legal Analysis*, 16(1):64–93.

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.

Lon Luvois Fuller. 1969. The morality of law. pages 46–90.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

HKAIR. 2024. Hk-o1aw models: Leveraging o1 slow thinking in the development of hong kong legal large language models. *GitHub repository*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Allison Koenecke, Jed Stiglitz, David Mimno, and Matthew Wilkens. 2025. Tasks and roles in legal ai: Data curation, annotation, and verification. *arXiv preprint arXiv:2504.01349*.

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*.

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*.

Zhaowei Liu, Xin Guo, Fangqi Lou, Lingfeng Zeng, Jinyi Niu, Zixuan Wang, Jiajie Xu, Weige Cai, Ziwei Yang, Xueqian Zhao, and 1 others. 2025. Fin-r1: A large language model for financial reasoning through reinforcement learning. *arXiv preprint arXiv:2503.16252*.

Venkatesh Mishra, Bimsara Pathiraja, Mihir Parmar, Sat Chidananda, Jayanth Srinivasa, Gaowen Liu, Ali Payani, and Chitta Baral. 2025. Investigating the shortcomings of llms in step-by-step legal reasoning. *arXiv preprint arXiv:2502.05675*.

OpenAI Team. 2024. Learning to reason with llms.

Qwen. 2025. Qwq-32b: Embracing the power of reinforcement learning.

Joseph Raz. 2009. The authority of law: essays on law and morality. pages 224–226.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Reshma Sheik, KP Siva Sundara, and S Jaya Nirmala. 2024. Neural data augmentation for legal overruling task: Small deep learning models vs. large language models. *Neural Processing Letters*, 56(2):121.

Cor Steging, Silja Renooij, and Bart Verheij. 2023. Improving rationales with small, inconsistent and incomplete data. pages 53–62.

Hanshuang Tong, Jun Li, Ning Wu, Ming Gong, Dongmei Zhang, and Qi Zhang. 2024. Ploutos: Towards interpretable stock movement prediction with financial large language model. *arXiv preprint arXiv:2403.00782*.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, Bo Zhang, Liqun Wei, Zhihao Sui, Wei Li, Botian Shi, Yu Qiao, Dahua Lin, and Conghui He. 2024. Mineru: An open-source solution for precise document content extraction.

Saizhuo Wang, Hang Yuan, Leon Zhou, Lionel M Ni, Heung-Yeung Shum, and Jian Guo. 2023. Alpha-gpt: Human-ai interactive alpha mining for quantitative investment. *arXiv preprint arXiv:2308.00016*.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

whyhow ai. 2025. Patientseek. *HuggingFace Models*.

Cheng Xu, Xiaofeng Hou, Jiacheng Liu, Chao Li, Tianhao Huang, and Xiaozhi et al. Zhu. 2023. MMBench: Benchmarking end-to-end multimodal dnns and understanding their hardware-software implications. In *2023 IEEE International Symposium on Workload Characterization (IISWC)*, pages 154–166, Reno, NV, USA. IEEE.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Yaoyao Yu, Leilei Gan, Yinghao Hu, Bin Wei, Kun Kuang, and Fei Wu. 2025. Evaluating test-time scaling llms for legal reasoning: Openai o1, deepseek-r1, and beyond. *arXiv preprint arXiv:2503.16040*.

Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Xuanjing Huang, and 1 others. 2023. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *arXiv preprint arXiv:2309.11325*.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024a. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38.

Yu Zhao, Huifeng Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2024b. Marco-o1: Towards open reasoning models for open-ended solutions. *arXiv preprint arXiv:2411.14405*.

Bihua Zou. 2021. Nine-step method on constitutive elements of the civil trial. pages 30–40. Originally published in Chinese as 要件审判九步法.

A Prompts of Data Construction

Throughout the data construction pipeline, we designed prompts tailored to four critical stages: data distillation, explanation generation, chain-of-thought rewriting, and reasoning selection. These prompts were carefully crafted to guide the model in producing high-quality, logically consistent, and legally grounded outputs at each stage.

A.1 The prompt of data distillation

In the data distillation phase, we drew inspiration from the official prompt design of DeepSeek-R1 and adapted it to the legal domain. Our prompt, illustrated in Figure 5, was designed to elicit clear, structured reasoning traces from the base model. It ensured that the distilled responses were both informative and aligned with the expected chain-of-thought (CoT) format, serving as foundational supervision data for subsequent training stages.

The Prompt for Data Distilling

Your task is to answer legal multiple-choice questions through rigorous deductive analysis. You must clearly present your reasoning process before giving the final answer.

Please carefully analyze the following legal multiple-choice question:
Question: **{question}**

Your response must follow this format:
1. Begin with a rigorous, step-by-step deductive reasoning process enclosed within <think>...</think> tags.
2. Then, clearly state the final answer enclosed within \\boxed{{}}.

Figure 5: The prompt of data distillation that we used for DeepSeek-R1.

A.2 The prompt of explanation generation

During the initial stage of data screening, we applied a regex-based answer check to filter the responses. For those that failed this check, we utilized the instruction-following model DeepSeek-V3 to regenerate explanations, providing it with the original question and answer as context. The specific prompting strategy used for explanation generation is illustrated in Figure 6.

A.3 The prompt of chain rewriting

To preserve reasoning diversity, we randomly sampled 10% of the examples that passed the answer check stage for chain rewriting. These samples were then used to generate alternative reasoning chains by leveraging the instruction-following capabilities of the DeepSeek-V3 model. Specifically, we provided the model with the original question,

The Prompt for Explanation Generation

Your task is to analyze judicial multiple-choice questions with rigor and accuracy, just as a qualified judicial officer would. Your reasoning should be logical, comprehensive, and supported by relevant legal provisions where applicable.

Question: **{question}**
Provided Answer (for reference only): **{answer}**

Note:
1. Analyze each option one by one, stating clearly why it is correct or incorrect.
2. Where appropriate, cite relevant legal provisions to support your analysis.
3. Use the provided answer only as a reference—do not simply derive the answer from it.
4. Return only the reasoning content. Do not include any phrases like "reasoning process," and do not repeat the original question or answer.

Figure 6: The prompt of explanation generation that we used for DeepSeek-V3.

reference answer, and existing reasoning as context, prompting it to reconstruct the reasoning process. This approach introduces variation in logical pathways while maintaining answer consistency. The detailed prompting strategy used for this reasoning chain rewriting is illustrated in Figure 7.

A.4 The prompt of reasoning selection

Finally, to ensure the generation of high-quality reasoning trajectories, we introduced a reasoning selection data screening process. In this stage, we proposed five specific evaluation criteria to assess the model's reasoning performance. These criteria were carefully crafted to align with the core elements of effective legal reasoning. Furthermore, we designed and refined the prompt shown in Figure 8 to guide the model toward generating accurate and interpretable responses.

In the initial preprocessing step, we conducted a detailed evaluation of the model-generated reasoning using the DeepSeek-V3 instruction model. This evaluation followed five predefined judgment criteria. For each reasoning output, a binary score of 1 was assigned if it met the criterion, and 0 otherwise. This binary scoring scheme (0/1) was applied systematically to ensure the consistency, reliability, and stability of the evaluation process.

A.5 The statistics of Unilaw-R1-Eval

The Unilaw-R1-Eval comprises 800 curated comparative question-answer pairs, and we further constructed in a fine-grained and domain-relevant manner. These samples are categorized to reflect the diverse challenges encountered in real-world legal reasoning. More detailed statistics of question types are summarized in Table 4.

The Prompt for Chain Rewriting

Your task is to rewrite the reasoning process and final answer based on the key points provided in my analysis and answer. You do not need to evaluate the correctness of the provided analysis or the answer. Your rewritten response must follow these requirements:

1. Write out the full name of laws, such as \"The Civil Code of the People's Republic of China,\" instead of abbreviations.
2. Accurate Article Numbers and Content: Ensure that cited articles are accurate and consistent with official legal texts.
3. Standardized Citation Format: Use a format like \"According to Article 54 of the Civil Code of the People's Republic of China, ...\"
4. Completeness of Options: All options must be mentioned and analyzed at least once.
5. Sequential Analysis: Either analyze options in sequence (A→B→C→D) or groups them logically (e.g., A→C, B→D).
6. Rewriting the Analysis and Reasoning: Retain the knowledge points and logical relationships from the original analysis as much as possible. If some case-specific details do not apply, they may still be kept, with appropriate adjustments.

Below is a legal multiple-choice question and answer context:
Question: **{question}**
Provided Reasoning Process: **{think}**
Provided Answer: **{answer}**

Please rewrite the reasoning process and final answer. The rewritten reasoning process must be enclosed within <think>...</think> tags, and the final answer must be placed within \\boxed{{}}.

Figure 7: The prompt of chain rewriting that we used for DeepSeek-V3.

The Prompt for Reasoning Selection

Your task is to assess whether the provided reasoning meets all the required criteria. I will provide a question, a standard answer, and a reasoning process. Please evaluate whether the reasoning meets the following rules. If it satisfies the rules, return 1. Otherwise, return 0.

Rules:
1. Content Consistency: The thinking reasoning process is logically coherent and capable of deriving the correct answer.
2. Logical Consistency: There are no contradictions in the logical steps of the reasoning.
3. Option-by-Option Analysis: Each answer option must be analyzed.
4. Legal Citation: Any cited legal provisions must be clearly stated.
5. Task Relevance: The reasoning process must align with the instructions in the original question.

Question: **{question}**
Answer: **{answer}**
Reasoning: **{think)**

Only return a single number — 0 or 1, and no additional content is allowed.

Figure 8: The prompt for reasoning selection that we used for DeepSeek-V3.

We provide a categorical statistical analysis of the dataset through two concentric pie charts. Each chart corresponds to one of the two question formats included in the benchmark: single-choice (SC) and multi-choice (MC).

As illustrated in Figure 9(a), the chart visualizes the distribution of question types for the single-choice tasks, divided into two main categories:

- **Case-driven** questions, which focus on logical reasoning and judgment over real or hypothetical legal scenarios.
- **Knowledge-driven** questions, which test the model's mastery of legal definitions, statutes, and normative concepts.

These above two categories represent complementary dimensions of legal AI: foundational legal knowledge and applied legal reasoning. Together, they cover a broad spectrum of legal domains allowing for domain-specific performance insights, as the outer ring shows. The legal subdomains include *"Criminal Law", "Criminal Procedure", "Labor Law", "Commercial Law", "International Law", "Constitutional Law", "Civil Law", "Civil Procedure", "Legal History", "Jurisprudence", "Intellectual Property", "Economic Law", "Administrative Law"*, and *"Administrative Procedure"*. This layered categorization enables granular evaluation of a model's capabilities in both conceptual understanding and real-world legal problem-solving.

As shown in Figure 9(b), the chart reflects the distribution of multi-choice questions, which require models to evaluate multiple legal options simultaneously. These tasks often demand more comprehensive reasoning chains and sensitivity to nuanced distinctions between legal provisions. Similar to the single-choice chart, the inner ring categorizes questions into knowledge-driven and case-driven types, while the outer ring provides a domain-level breakdown. The multi-choice questions particularly emphasize complex decision-making scenarios, such as those involving overlapping legal principles or multiple liable parties.

By providing detailed categorization for both question types and domain coverage, Unilaw-R1-Eval offers a rigorous, fine-grained benchmark for assessing legal-domain LLMs across knowledge comprehension, reasoning reliability, and generalization capacity. This dual-structured evaluation framework is instrumental for identifying both model strengths and performance bottlenecks across varied legal tasks.

	Knowledge	**Case**	Total
Single-Choice	99	327	426
Multi-Choice	70	304	374
All	169	631	800

Table 4: The statistics of question types in Unilaw-R1-Eval.

(a) Single-choice question distribution in Unilaw-R1-Eval.

(b) Multi-choice question distribution in Unilaw-R1-Eval.

Figure 9: Distribution of question types and legal subdomains in Unilaw-R1-Eval. The figure presents categorical statistics for both single-choice (SC) and multi-choice (MC) legal questions. The inner rings distinguish between knowledge-driven and case-driven types, while the outer rings represent their distribution across legal subdomains.

B Prompt of Legal Validity Reward

To enhance the alignment of the model's outputs with legal correctness during reinforcement learning, we incorporate a model-based feedback mechanism. Specifically, we utilize an instruction language model Qwen2.5-7B-Instruct as a verifier to assess the quality of the reasoning trajectories generated by the policy model. This verifier evaluates each response against predefined legal reasoning criteria, including logical consistency, legal validity, and alignment with the expected legal outcome.

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

The evaluation criteria used in the Legal Validity prompt are largely consistent with those in chain-of-thought rewriting, with an added emphasis on syllogistic reasoning in legal contexts, applying legal rules to case facts to derive conclusions.

- **Choice Analysis:** This emphasizes completeness by systematically analyzing each option

either sequentially or in groups, ensuring that all answer choices are 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 align with official legal texts. Additionally, it requires writing out the full names of laws rather than abbreviations.

C Details of Training Setup

We provide detailed training configurations used in both the Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL) phases of Unilaw-R1. In the SFT phase, we utilize LoRA to learn the `<think>...</think>\n\n<answer>...</answer>` format, with a LoRA rank of 8. In the RL phase, 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. 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 8×96GB NVIDIA H20 GPUs. Key hyperparameters for both stages are summarized in Table 5.

Parameter	SFT	RL(GRPO)
Batch Size	16	128
Epochs	5	1
Learning Rate	1.0e-4	1.0e-6
Warmup Ratio	0.1	0.03
Max Sequence Length	4096	4096
Gradient Accumulation	8	4
Optimizer	AdamW	AdamW
Weight Decay	0.01	0.01
LR Scheduler	Cosine	Cosine
Evaluation Interval	500 steps	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 SFT and GRPO stages.

D Details of Iterative Inference Setup

To enhance the model's legal reasoning performance through iterative refinement, we adopt a multi-agent setup comprising two collaborative components: an Assessor agent and a Reviser agent. These agents operate in tandem to identify and correct reasoning flaws, enabling a more robust and interpretable inference process.

D.1 Implementation details

We employ the Qwen2.5-7B-Instruct model to serve as both the Assessor and Reviser in our iterative inference framework. During the process, we need to evaluate the outcome quality, the InternLM-7B was selected as the outcome reward model (ORM) to computing the chain-level scores. By default, we sample $k = 10$ reasoning chains in each iteration, with the decoding temperature parampter fixed at 0.9. The maximum number of iterations is set to 3. We conducted comparative analyses against three distinct methodological approaches:

- **Zero-shot Chain-of-Thought (CoT):** Generates a single reasoning chain per question without subsequent aggregation.

- **Best-of-k Sampling:** Produces multiple candidate chains for each question and selects the optimal output base on maximal ORM score.

- **Majority Vote:** Employs Self-Consistency mechanisms to determine final answers through consensus voting across multiple generated chains.

D.2 The prompt of iterative inference

Assessor Prompt: The Assessor is tasked with critically evaluating the initial reasoning output from the Unilaw-R1 model. Its prompt is designed to identify potential flaws in logic, incompleteness in option analysis, and inconsistencies with legal principles or cited laws. As illustrated in Figure 10, the Assessor highlights specific errors or weaknesses and provides structured feedback based on the provided in-context learning question, solution and feedback.

Reviser Prompt: The Reviser then utilizes both the original reasoning and the Assessor's critique to produce an improved version. As shown in Figure 11, the prompt guides the model to incorporate the Assessor's feedback while preserving alignment with the legal context and the original question intent. The Reviser ensures that the revised output is not only more accurate but also logically coherent and legally compliant, using the provided one-shot

Assessor Prompt

You are an expert legal assessor. Your task is to provide item-by-item feedback on a given legal question and its corresponding answer. Be accurate, fair, and constructive in your evaluations. You will receive a legal-related question along with an answer. Your task is as follows:

1. Analyze each option within the answer individually.
2. Assign a score at the end of each option in the format (Score: n/10), where 10 indicates the option is entirely correct (100% accurate), and 0 indicates the option is entirely incorrect (0% accurate).
3. If any option receives a score below 6, carefully analyze the issues with that option and provide detailed feedback, including suggestions for how to improve it.
4. Do not propose new answers - only explain the mistakes and offer suggestions for improvement.

Question: **{icl_question}**
Current Answer: **{icl_solution}**

Let's review the lower-scoring options, identify the issues, and provide detailed feedback:
{icl_feedback}

Figure 10: The prompt for assessor the model answer that we used.

Reviser Prompt

You are a legal expert reviser. Your task is to revise a student's multiple-choice answer based on the teacher's feedback. Your goal is to improve the answer by correcting any legal or logical errors identified in the feedback and ensuring that the revised response aligns with sound legal reasoning.

1. Carefully review the original answer and the teacher's feedback.
2. Provide a corrected and improved version of the answer.
3. The final answer must be one or more of the options A, B, C, and D.
4. Clearly indicate the final choice by enclosing it in \\boxed{{}}.

Question: **{icl_question}**
Current Answer: **{icl_solution}**
Teacher's Feedback: **{icl_feedback}**

Now, let's revise the answer based on the feedback provided. Below is the corrected response to the legal question:
{icl_refined_solution}

Figure 11: The prompt for revising the model answer that the one-shot in-context learning refined content comes at the end.

in-context learning example - including question, solution feedback, and the refined solution.

Through multiple rounds of Assessor–Reviser interaction, the system progressively refines its output, achieving higher-quality legal reasoning. This multi-agent collaboration mimics peer-review processes and enhances both the correctness and explainability in legal decision-making.

D.3 Strategy in instruction model

We prompt the Qwen2.5-7B-Instruct model to generate explicit reasoning traces enclosed in `<think>...</think>` tags. Table 6 summarizes the model's performance under different inference strategies, including zero-shot Chain-of-Thought (CoT), Best-of-k sampling, Majority Vote, and our proposed Iterative Inference method with varying iteration steps ($Iter = 1$ to $Iter = 3$). Results on the Unilaw-R1-Eval benchmark demonstrate that Iterative Inference consistently improves performance, achieving the highest overall accuracy (35.9%) with three iterations.

Method	SC	MC		Avg.
	Acc.(%)	Acc.(%)	F1	Acc.(%)
Zero-shot CoT	43.2	14.2	52.4	29.9
Best-of-k ($k = 10$)	52.1	10.2	56.0	32.5
Majority Vote	51.9	15.0	60.2	34.6
Iterative Infer. ($Iter = 1$)	53.1	15.8	61.2	35.6
Iterative Infer. ($Iter = 2$)	52.2	**17.3**	**63.1**	35.8
Iterative Infer. ($Iter = 3$)	**53.3**	16.1	61.8	**35.9**

Table 6: Performance comparison of Qwen2.5-7B-Instruct with different inference methods on the Unilaw-R1-Eval benchmark.