

Towards Better Legal Reasoning LLMs: Signal Balancing and Reward Scheduling

Yinghao Hu¹, Qian Xiao², Yuting Huang¹, Kaisong Song^{2,4}, Yiquan Wu^{1,*}
Ang Li³, Junjie Cao², Jun Lin², Kun Kuang^{1,*}, Fei Wu¹

¹Zhejiang University, Hangzhou, China

²Alibaba Group, Hangzhou, China

³University of Macau, Macao, China

⁴Northeastern University, Shenyang, China

Emails: {huyinghao, wuyiquan, kunkuang}@zju.edu.cn

Abstract

In the legal domain, LLMs can improve reliability and interpretability by generating explicit reasoning traces. However, training LLMs to produce high-quality reasoning traces remains challenging. Existing supervised fine-tuning methods struggle when reasoning traces are significantly longer than final answers, as the learning signal for the answer becomes diluted. Meanwhile, reinforcement learning methods such as Group Relative Policy Optimization also have several limitations, including costly reward design, performance plateaus, and reward hacking. We therefore propose a two-stage training framework. In Stage I, JurisCoT-SFT employs a length-normalized dual objective to balance learning signals between reasoning traces and final answers. In Stage II, Lifecycle-Aware Backtrackable Policy Optimization dynamically evaluates and retires auxiliary reward signals based on their impact on final evaluation objectives, enabling efficient reward utilization. Our framework substantially improves legal reasoning performance on relevant tasks selected from multiple legal benchmarks, including ChJGBench, LawBench, and LegalBench. Experimental results demonstrate consistent improvements in both reasoning trace quality and final answer accuracy, while our models remain competitive with larger models.

1 Introduction

Recent advances in large language models (LLMs) have highlighted the importance of explicit reasoning, in which models generate intermediate thought processes before producing final answers, for enhancing reliability and interpretability across complex tasks. This form of structured reasoning is particularly crucial in the legal domain, where decisions must be grounded in legal principles and

*Corresponding authors.

Query
During an argument, A stabs B in the thigh and then voluntarily calls the police and requests an ambulance. Despite rescue efforts, B dies from blood loss. What crime has A committed?

Output
<think> Because A initiated the attack and intentionally stabbed B, Article 14 applies, while self-defense under Article 20 is unsupported. The key issue is whether A intended or knowingly allowed B's death. Courts consider the weapon, target area, number of blows, and subsequent conduct. A single thigh stab and A's call to the police and request for an ambulance favor an intent to injure rather than kill, ... Similar cases include {Criminal, First Instance, Case No. **, 2016}, in which a defendant was sentenced ... , and {Criminal, First Instance, Case No. **, 2014}, in which If A intended to injure B but was negligent as to B's death, Article 234(2) applies; Article 232 applies if A intended or knowingly allowed B's death. The voluntary report and medical assistance may support leniency; Article 67 applies only if A </think>
<answer> A is more likely guilty of intentional injury resulting in death rather than intentional homicide. </answer>

Legal domain knowledge

Relevant Statutes	Articles 14, 20, 67, 232, and 234(2) of the PRC Criminal Law.
Legal Precedents	{Criminal, First Instance, Case No. **, 2016} {Criminal, First Instance, Case No. **, 2014}
Reasoning ability	
Subjective Element Assessment	Intentional stabbing; intent to injure rather than kill
Sentencing Factor Evaluation	Possible voluntary surrender; medical assistance

Figure 1: An anonymized and abridged example of a legal reasoning trace.

statutory provisions and must be logically coherent (Levi, 2013).

We observe that high-quality legal reasoning traces are generally quite long, as they must encompass both necessary legal knowledge and a logically structured analysis, as shown in Figure 1. This requires a deep understanding of legal principles, including identifying relevant statutes and legal precedents, as well as the ability to accurately interpret and apply them (Gerhardt, 2008; Li et al., 2025). At the same time, legal reasoning must adhere to several critical logical constraints, including consistency between statutes and con-

clusions and internal logical coherence throughout the analysis (Desai et al., 2016). Given the extensive domain-specific knowledge and rigorous logical reasoning required, together with the low tolerance for hallucinations in legal contexts (Dahl et al., 2024), generating reliable legal reasoning traces remains significantly more challenging than in general domains.

Directly post-training a foundation model on legal-domain data is insufficient to address these challenges. During Supervised Fine-Tuning (SFT), reasoning traces are often several times as long as the final answers, causing token-level loss to bias optimization toward the traces and weaken the learning signal for final answers. Moreover, when reasoning traces contain flaws, such as misapplied precedents, SFT cannot provide targeted feedback to correct such errors. In reinforcement learning, methods such as Group Relative Policy Optimization (GRPO) have become mainstream alignment approaches. However, GRPO relies on outcome-level rewards that evaluate only final answers and ignore the legal soundness of intermediate reasoning. A straightforward solution is to introduce multiple rewards for different aspects of reasoning (Cai et al., 2025; Wu, 2025; Wang et al., 2025). However, this requires multiple manually designed reward functions, which are costly to construct, may introduce conflicting optimization objectives, gradually lose effectiveness during training, or even lead to reward hacking (Wang et al., 2024; Laidlaw et al., 2025).

These challenges motivate our two-stage training framework for legal reasoning models. In Stage I, we introduce **JurisCoT-SFT**, which not only injects domain knowledge into the model but also adopts a length-normalized dual objective to mitigate signal imbalance. In Stage II, we propose **LABPO** (Lifecycle-Aware Backtrackable Policy Optimization), a self-regulating reinforcement learning framework for automatically managing reward signals during policy optimization.

The central idea of LABPO is to explicitly separate reward signals into two roles: *primary rewards*, which directly correspond to the final evaluation objectives (e.g., F1 score and accuracy), and *auxiliary rewards*, which serve as stage-dependent training signals (e.g., output format and language style). Unlike conventional approaches that keep all reward components active throughout training, LABPO treats auxiliary rewards as adaptive optimization instruments whose usefulness evolves

over the course of learning. Each auxiliary signal remains active only while it remains beneficial to training; otherwise, it is retired, and the complete training state is rolled back to the best checkpoint observed during its active period.

We evaluate LLMs on selected tasks from three legal benchmarks: the statutory-law benchmarks ChJGBench (Hu et al., 2026) and LawBench (Fei et al., 2024), and the case-law benchmark LegalBench (Guha et al., 2023). Using Qwen3-8B as the base model, our two-stage framework substantially improves legal reasoning performance across these benchmarks. On ChJGBench, it increases the average score from 0.6406 to 0.7786, achieving the highest score among all evaluated models and outperforming strong general-purpose LLMs such as Gemini-3-Pro-Preview. On LawBench, the resulting model achieves 0.848 on *AP*, ranking second overall, and 0.666 on *CP*, ranking first. It also performs strongly on case-law reasoning tasks in LegalBench, attaining the highest *LRC* accuracy of 0.964 and competitive performance on the numerical reasoning task *Sara-N*. Despite having only 8 billion parameters, the resulting Qwen3-8B-based model remains competitive with much larger models and even outperforms them on some tasks. Applying the same framework to Llama-3.1-8B likewise yields substantial gains, increasing its ChJGBench average score from 0.3374 to 0.6452 and improving its performance across LawBench and LegalBench. Ablation studies further demonstrate that both stages of the framework contribute to the observed performance improvements.

Our contributions can be summarized as follows:

1. We propose **JurisCoT-SFT**, which balances learning signals across reasoning traces and final answers, improving the model’s ability to internalize legal knowledge and produce higher-quality reasoning outputs.
2. We introduce **LABPO**, which automatically manages the lifecycle of reward signals during policy optimization by dynamically evaluating and retiring auxiliary rewards based on their impact on final evaluation objectives.
3. We train a lightweight LLM that performs competitively across multiple benchmarks, establishing a strong lightweight baseline for future research on legal LLMs.

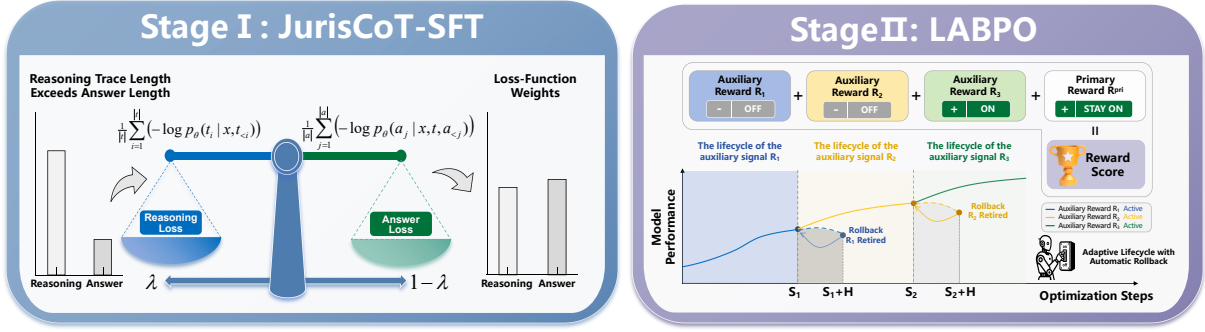


Figure 2: The framework has two stages. In Stage I, λ balances learning between reasoning traces and final answers. In Stage II, auxiliary reward signals are activated, evaluated, and retired, with checkpoint rollback when needed.

2 Related Work

Traditional legal-domain LLMs typically follow a two-stage paradigm consisting of continued pre-training and domain-specific fine-tuning to enhance legal capabilities. Representative models include LawGPT (Zhou et al., 2024), HanFei (He et al., 2023), LawLLM (US) (Shu et al., 2024), InternLM-Law (Fei et al., 2025), and Wisdom-Interrogatory (Wu et al., 2026). Previous studies have also shown that explicitly generating intermediate reasoning traces before producing final answers can improve performance on legal reasoning tasks (Hu et al., 2025b). DISC-LawLLM (Yue et al., 2024) incorporates a legal syllogism framework into training data design. Recent work has further introduced reinforcement learning to improve reasoning quality in legal LLMs. LexPam (Zhang et al., 2025a) focuses on legal numerical reasoning using a two-stage GRPO approach, optimizing numerical precision while incorporating legal constraints into the reward function. SyLeR (Zhang et al., 2025b) uses supervised fine-tuning (SFT) as a warm-up stage, followed by Proximal Policy Optimization (PPO) to explore multiple reasoning paths, though its reward function mainly relies on general similarity metrics such as ROUGE, which may not fully capture the nuances of legal reasoning. LexPro (Chen et al., 2025) integrates SFT and GRPO with rule-based format and process rewards, but these rewards primarily target output format and reasoning structure rather than domain-specific legal correctness. Legal Δ (Dai et al., 2026) improves reasoning through chain-of-thought-guided information gain with a multidimensional reward mechanism. Despite recent advances, challenges remain in learning from long reasoning traces and in designing reward functions that effectively optimize reasoning performance. To address these

challenges, we propose a two-stage training framework for legal reasoning models.

3 Method

We introduce a two-stage training framework for signal rebalancing and adaptive reward scheduling, as illustrated in Figure 2.

3.1 Stage I: JurisCoT-SFT

In the legal domain, reliable LLMs must ensure both procedural legitimacy and conclusion correctness. This requires a think-then-answer paradigm. However, using standard SFT presents a critical challenge: **the severe length imbalance between the typically lengthy reasoning traces and the concise final answers**. During training, this imbalance leads to undue emphasis on the "reasoning" part, weakening the learning signal for the shorter but crucial "answering" part. Given that even minor inaccuracies in the final answer can have high-stakes consequences, addressing this training imbalance is particularly important.

To address this issue, we propose **JurisCoT-SFT**, a variant of standard SFT that introduces a length-normalized dual objective. Formally, let a training instance be (x, t, a) , where x denotes the input query, $t = (t_1, \dots, t_{|t|})$ represents the provided reasoning sequence, and $a = (a_1, \dots, a_{|a|})$ denotes the final answer. Given model parameters θ , we optimize the next-token cross-entropy loss, with the reasoning and answer segments weighted separately and normalized by their respective lengths:

$$\begin{aligned} \mathcal{L}_{\text{JurisCoT-SFT}}(\theta) = & \lambda \cdot \frac{1}{|t|} \sum_{i=1}^{|t|} (-\log p_{\theta}(t_i | x, t_{<i})) \\ & + (1 - \lambda) \cdot \frac{1}{|a|} \sum_{j=1}^{|a|} (-\log p_{\theta}(a_j | x, t, a_{<j})), \end{aligned} \quad (1)$$

Signal	Description
S_1	Tag Format Validation
S_2	JSON Schema Validation
S_3	Statute Citation Format Verification
S_4	Language Style Validation
S_5	Statute Authenticity Verification
S_6	Statute-Conclusion Consistency Verification

Table 1: Auxiliary reward signals used during training. Detailed definitions are provided in Appendix D.

where λ is a balancing coefficient that controls the relative contributions of the reasoning trace (t) and the final answer (a) to the training objective. A smaller λ places greater emphasis on the final-answer loss while retaining supervision from the reasoning trace.

3.2 Stage II: Lifecycle-Aware Backtrackable Policy Optimization

Although models fine-tuned with JurisCoT-SFT demonstrate strong legal knowledge and produce structured outputs, supervised fine-tuning cannot provide fine-grained supervision over intermediate reasoning steps or deliver differentiated feedback when reasoning traces contain errors.

To address these challenges, we propose **LABPO**, a reinforcement learning framework that **automatically manages reward signals** during policy optimization. The central idea of LABPO is to decompose reward signals into two functional roles. **Primary rewards** are aligned with the final evaluation objectives and remain active throughout training. **Auxiliary rewards** serve as stage-dependent training signals that may facilitate optimization during certain phases of learning but are not necessarily beneficial throughout the entire training trajectory. Instead of treating auxiliary rewards as fixed components of the reward function, LABPO assigns each auxiliary reward an independent lifecycle: after an initial monitoring period, it remains active as long as the running-best development score improves by at least ϵ over the most recent H evaluation intervals and is retired otherwise.

Let τ denote the development-set evaluation index, with reward gates fixed between evaluations. For input x and output y , the training reward is defined as:

$$R(y; \tau) = R^{\text{pri}}(y) + \sum_{i=1}^m \alpha_i(\tau) R_i^{\text{aux}}(y), \quad (2)$$

where R^{pri} denotes the primary reward,

$\{R_i^{\text{aux}}\}_{i=1}^m$ denotes the set of auxiliary rewards, and $\alpha_i(\tau) \in \{0, 1\}$ is a gating variable indicating whether the i -th auxiliary reward is active.

3.2.1 Adaptive Auxiliary Signal Scheduling

To operationalize the lifecycle management of auxiliary rewards, we propose an Adaptive Auxiliary Signal Scheduling mechanism that automatically regulates the activation and retirement of auxiliary signals during training, thereby alleviating optimization plateaus and suppressing reward hacking. Pseudocode for this mechanism is provided in Appendix B.

Design Principles. Instead of keeping all auxiliary signals active throughout the entire training process, we allow for the dynamic activation of auxiliary signals. To avoid interference between multiple auxiliary signals, only one auxiliary signal is active at any given time.

Periodic Evaluation. Whenever the retained-step counter reaches a multiple of n , we evaluate the current policy π_{θ_τ} on a held-out development set $\mathcal{D}_{\text{dev}}$, which is independent of the test sets. The resulting performance score $S(\theta_\tau)$ serves as the sole criterion for determining the lifecycle of the auxiliary signal.

Dynamic Update of Auxiliary Signal Weights. Let τ_a denote the evaluation index at which the current auxiliary signal R_a^{aux} is activated. At evaluation step τ , we first define the running-best development score over the active interval:

$$B_\tau = \max_{j \in \{\tau_a, \dots, \tau\}} S(\theta_j). \quad (3)$$

If multiple checkpoints attain the same best score B_τ , we select the latest one:

$$\tau^* = \max\{j \in \{\tau_a, \dots, \tau\} \mid S(\theta_j) = B_\tau\}. \quad (4)$$

During the first H evaluations after activation, the auxiliary signal remains active, defining a minimum active period before it becomes eligible for deactivation. Once at least H evaluations have been collected, we measure the improvement in the running-best score over the most recent H evaluation intervals:

$$G_\tau = B_\tau - B_{\tau-H}, \quad \tau - \tau_a \geq H. \quad (5)$$

The weight of the active auxiliary signal is then updated as follows:

$$\alpha_a(\tau + 1) = \begin{cases} 0, & \tau - \tau_a \geq H \wedge G_\tau < \epsilon, \\ 1, & \text{otherwise.} \end{cases} \quad (6)$$

The auxiliary signal is therefore retired only when the cumulative improvement in the running-best development score over the preceding H evaluation intervals is less than ϵ . Using the running-best score rather than the latest score makes the retirement decision less sensitive to short-term fluctuations during reinforcement learning.

Checkpoint Rollback and Mechanism Iteration.

Let ω , ζ , and s denote the optimizer state, learning-rate scheduler state, and retained-step counter, respectively. Each checkpoint stores $(\theta, \omega, \zeta, s)$ together with its associated development score. When the current auxiliary signal is retired, the complete training state is restored to the highest-scoring checkpoint observed during that signal’s lifecycle:

$$(\theta, \omega, \zeta, s) \leftarrow (\theta_{\tau_\tau^*}, \omega_{\tau_\tau^*}, \zeta_{\tau_\tau^*}, s_{\tau_\tau^*}). \quad (7)$$

The stored development score is reused to initialize the next phase, while the auxiliary-reward gates and phase variable are updated separately and are not restored from the checkpoint. The mechanism then activates the next auxiliary signal in the predetermined order and starts a new evaluation-and-scheduling cycle. After all auxiliary signals have been retired, all auxiliary rewards are disabled, and training continues using only the primary reward. Training ends when the retained-step counter reaches the target budget N . If the running-best development score improves by less than ϵ over H consecutive evaluation intervals before the budget is reached, the best checkpoint observed during this phase is restored and training terminates immediately.

3.2.2 Auxiliary Reward Signal Integration

The goal of LABPO is not to eliminate reward design, but to make reward signals more reusable and easier to deploy. Unlike conventional reinforcement learning approaches that require carefully crafted reward functions to remain consistently effective throughout training, LABPO allows researchers to explore multiple auxiliary reward signals more freely, including potentially noisy or unstable heuristic signals, without pre-designing, tuning, or manually combining them in advance. Through dynamic evaluation during training, LABPO automatically identifies useful signals,

retires those that cease to provide sufficient improvement, and rolls back the training state when necessary.

As shown in Table 1, we introduce six auxiliary reward signals based on prior work and common evaluation criteria for legal reasoning. The rationale for selecting these signals and their detailed definitions are provided in Appendix D. In all main experiments, auxiliary signals are activated sequentially from S_1 to S_6 . To assess whether this ordering affects final performance, we conduct an auxiliary-signal ordering experiment with the six proposed signals, as detailed in Appendix C. The results show that, for the auxiliary signals considered in our framework, the optimization order has only a limited impact on final performance.

4 Experiments

In this section, we describe the experimental setup, report the main results, analyze the training dynamics of auxiliary signal scheduling, compare reasoning performance, and conduct ablation studies. We additionally present a controlled diagnostic experiment on reward-hacking behavior in Appendix E.

4.1 Experimental Setup

Training Data. We prepare training data for the LLMs in both statutory law and case law domains. For **statutory law**, we construct a large-scale legal training corpus consisting of {Query, Reasoning Trace, Answer} triplets. The query and answer are extracted from judicial documents, while the reasoning traces are generated by DeepSeek-R1. To ensure generation quality, we apply a variant of rejection sampling to filter generated traces (Appendix A). The dataset contains 2.73M instances (Table 12). For **case law**, we construct a corresponding dataset of triplets. The same rejection sampling procedure is applied to ensure the quality of the generated reasoning traces. The dataset contains 35,965 instances (Table 13).

Evaluation Data. We evaluate LLMs on selected tasks from three legal benchmarks: ChJGBench (Hu et al., 2026) and LawBench (Fei et al., 2024) for statutory law, and LegalBench (Guha et al., 2023) for case law. Task details are summarized in Table 14, and metric definitions are provided in Appendix K.

ChJGBench is a benchmark comprising 3,462 real-world cases spanning criminal, civil, and administrative law. It evaluates three general di-

Models	Avg.	Law	ISS	JUS	Criminal Case				Civil Case		Admin.	
					Fine	MIT	Charge	Pen_M	REC	SUP	PAY	DIS
Specialized Legal LLMs												
WisdomInterrogatory	0.3728	0.158	0.582	0.384	0.135	0.408	0.436	0.608	0.108	0.649	0.244	0.326
DISC-LawLLM-7B	0.4482	0.181	0.729	0.546	0.080	0.727	0.414	0.466	0.370	0.670	0.056	0.420
Open-source LLMs												
Llama-3.1-405B	0.6006	0.342	0.712	0.575	0.476	0.885	0.957	0.602	0.500	0.682	0.616	0.712
DeepSeek-R1	0.7080	0.422	0.819	0.718	0.563	0.961	0.969	0.740	0.715	0.783	0.753	0.787
Qwen3-235B-A22B-Instruct	0.7136	0.461	0.812	0.748	0.603	0.974	0.967	0.780	0.765	0.778	0.744	0.653
Proprietary LLMs												
Grok-4	0.7166	0.499	0.818	0.728	0.501	0.973	0.976	0.724	0.668	0.761	0.709	0.799
GPT-5.1	0.6795	0.422	0.763	0.707	0.556	0.965	0.975	0.691	0.665	0.764	0.722	0.687
Claude-Sonnet-4.5	0.7293	0.540	0.811	0.729	0.577	0.968	0.970	0.751	0.705	0.779	0.743	0.764
Gemini-3-Pro-Preview	0.7470	0.644	0.810	0.683	0.611	0.973	0.971	0.786	0.753	0.831	0.757	0.731
Our Methods												
Llama-3.1-8B	0.3374	0.099	0.592	0.357	0.102	0.507	0.221	0.513	0.318	0.575	0.117	0.254
with JurisCoT-SFT	0.6212	0.414	0.722	0.594	0.493	0.877	0.862	0.718	0.656	0.708	0.648	0.587
with JurisCoT-SFT + LABPO	0.6452	0.437	0.743	0.621	0.518	0.891	0.893	0.734	0.672	0.726	0.671	0.628
Qwen3-8B	0.6406	0.315	0.791	0.652	0.459	0.925	0.960	0.727	0.673	0.726	0.708	0.624
with JurisCoT-SFT	0.7667	0.667	0.830	0.748	0.574	0.976	0.975	0.770	0.747	0.801	0.717	0.802
with JurisCoT-SFT + LABPO	0.7786	0.689	0.846	0.751	0.579	0.971	0.974	0.766	0.784	0.822	0.724	0.812

Table 2: Evaluation results of different models on ChJGBench. The best results among all models are shown in **bold**, and the second-best results are underlined.

mensions shared by all cases: statutory provision application (*Law*), disputed issues identification (*ISS*), and judicial justification (*JUS*). Domain-specific tasks include criminal charge prediction (*Charge*), monetary fine (*Fine*), recidivism assessment (*REC*), mitigating circumstances identification (*MIT*), penalty duration prediction (*Pen_M*), civil claim support and payment obligation (*SUP*, *PAY*), and administrative disposition prediction (*DIS*).

LawBench is a public benchmark dataset for evaluating Chinese legal capabilities. We select three tasks: Article Prediction (*AP*), Charge Prediction (*CP*), and Sentence Prediction (*SP*).

LegalBench is a benchmark for evaluating the legal reasoning abilities of LLMs using tasks derived from real legal datasets and expert-designed evaluations. We select three representative reasoning tasks for evaluation: Legal Reasoning Causality (*LRC*), New York State Judicial Ethics (*NYSJE*), and SARA Numeric Reasoning (*Sara-N*).

Baselines. We evaluate our model against a series of strong baselines across the benchmark test sets. These baselines include open-source general-purpose LLMs, such as Llama-3.1-405B (Grattafiori et al., 2024), DeepSeek-R1 (Guo et al., 2025) and Qwen3-235B-A22B-Instruct (Yang et al., 2025); closed-source general-purpose LLMs, including Grok-4 (xAI, 2025),

GPT-5.1 (OpenAI, 2025), Claude-Sonnet-4.5 (Anthropic, 2025) and Gemini-3-Pro-Preview (Google DeepMind, 2026); and legal-domain-specific LLMs, including WisdomInterrogatory (Wu et al., 2026) and DISC-LawLLM-7B (Yue et al., 2024).

Implementation Details. We use Llama-3.1-8B and Qwen3-8B as the base models for subsequent fine-tuning. Since Qwen3-8B achieves better overall performance, we conduct all ablation studies using Qwen3-8B as the backbone. Detailed training hyperparameters are provided in Appendix L, and the computational overhead of LABPO is reported in Appendix M.

4.2 Main Results

4.2.1 Results on Statutory-Law Benchmarks

We report the main experimental results on the statutory-law benchmarks ChJGBench and LawBench in Tables 2 and 3.

Specialized legal LLMs built on outdated base models show limited capability. Models designed specifically for the legal domain perform relatively poorly overall. On ChJGBench, WisdomInterrogatory and DISC-LawLLM-7B achieve average scores of only 0.3728 and 0.4482, respectively, far below most modern general-purpose LLMs and our trained models. Their weaknesses are particularly evident in decision-oriented tasks such as *Fine* and *PAY*, where both models obtain substantially

Models	AP $\uparrow$	CP $\uparrow$	SP $\uparrow$
<i>Specialized Legal LLMs</i>			
WisdomInterrogatory	0.328	0.351	0.804
DISC-LawLLM-7B	0.054	0.127	0.670
<i>Open-source LLMs</i>			
Llama-3.1-405B	0.306	0.352	0.807
DeepSeek-R1	0.831	0.606	0.891
Qwen3-235B-A22B-Instruct	0.865	0.607	0.857
<i>Proprietary LLMs</i>			
Grok-4	0.844	0.553	<u>0.885</u>
GPT-5.1	0.721	0.561	0.883
Claude-Sonnet-4.5	0.818	0.550	0.775
Gemini-3-Pro-Preview	0.779	<u>0.637</u>	0.822
<i>Our Methods</i>			
Llama-3.1-8B	0.057	0.223	0.707
with JurisCoT-SFT	0.756	0.254	0.849
with JurisCoT-SFT + LABPO	0.771	0.288	0.862
Qwen3-8B	0.792	0.537	0.837
with JurisCoT-SFT	0.822	0.613	0.843
with JurisCoT-SFT + LABPO	<u>0.848</u>	0.666	0.854

Table 3: Evaluation results of different models on Law-Bench. **AP**, **CP**, and **SP** denote Article Prediction, Charge Prediction, and Sentence Prediction, respectively. The best results among all models are shown in **bold**, and the second-best results are underlined.

lower scores than recent open-source and proprietary LLMs. A similar pattern is observed on Law-Bench, where both models substantially underperform most recent general-purpose LLMs on Article Prediction and Charge Prediction. These results suggest that legal-domain training alone may be insufficient when the underlying base models lack strong general reasoning and instruction-following capabilities.

General-purpose LLMs perform reasonably well overall but still exhibit limitations in legal knowledge. Modern general-purpose LLMs achieve relatively strong overall performance across both benchmarks, indicating that large-scale pretraining provides a useful foundation for legal reasoning. On ChJGBench, Gemini-3-Pro-Preview obtains the highest average score among general-purpose models at 0.7470. However, these models may still show weaker performance on tasks requiring precise grounding in legal knowledge. For example, Gemini-3-Pro-Preview achieves the highest *Law* score among general-purpose models (0.644), but remains below our best-performing model (0.689).

Our two-stage training framework consistently improves lightweight base models. Applying our proposed two-stage framework consistently

Models	LRC $\uparrow$	NYSJE $\uparrow$	Sara-N $\downarrow$
<i>Specialized Legal LLMs</i>			
WisdomInterrogatory	0.491	0.387	–
DISC-LawLLM-7B	0.746	0.630	19.330
<i>Open-source LLMs</i>			
Llama-3.1-405B	0.909	0.709	7.720
DeepSeek-R1	0.927	0.771	0.250
Qwen3-235B-A22B-Instruct	0.891	0.791	9.618
<i>Proprietary LLMs</i>			
Grok-4	0.909	<u>0.805</u>	5.802
GPT-5.1	<u>0.946</u>	0.836	7.051
Claude-Sonnet-4.5	0.891	0.747	<u>0.994</u>
Gemini-3-Pro-Preview	0.873	0.764	7.677
<i>Our Methods</i>			
Llama-3.1-8B	0.764	0.493	21.088
with JurisCoT-SFT	0.855	0.575	6.125
with JurisCoT-SFT + LABPO	0.909	0.623	4.846
Qwen3-8B	0.873	0.596	8.093
with JurisCoT-SFT	0.927	0.627	3.949
with JurisCoT-SFT + LABPO	0.964	0.688	1.753

Table 4: Evaluation results of different models on Legal-Bench. The best results among all models are shown in **bold**, and the second-best results are underlined.

improves both lightweight backbones. On ChJG-Bench, Llama-3.1-8B improves from 0.3374 to 0.6452 after JurisCoT-SFT and LABPO, while Qwen3-8B improves from 0.6406 to 0.7786. The final Qwen3-8B-based model achieves the best overall score among all evaluated models and obtains top performance on key metrics such as *Law* (0.689), *ISS* (0.846), *JUS* (0.751), *REC* (0.784), and *DIS* (0.812). A similar trend is observed on LawBench.

4.2.2 Results on Case Law Benchmark

We report the main experimental results for three case-law tasks from LegalBench in Table 4. The results show that our two-stage framework generalizes beyond statutory-law tasks and remains effective across the evaluated case-law tasks. In particular, Qwen3-8B with JurisCoT-SFT + LABPO achieves the best performance on *LRC*, improving from 0.873 to 0.964 and outperforming all evaluated models. It also substantially reduces the numerical error on *Sara-N*, from 8.093 to 1.753. For the *NYSJE* task, performance is limited by the model size, resulting in lower scores than those of larger proprietary models. However, it still demonstrates a significant improvement over the base model. Overall, these results demonstrate the effectiveness and cross-domain applicability of our training framework.

4.3 Training Dynamics of Auxiliary Signal Scheduling

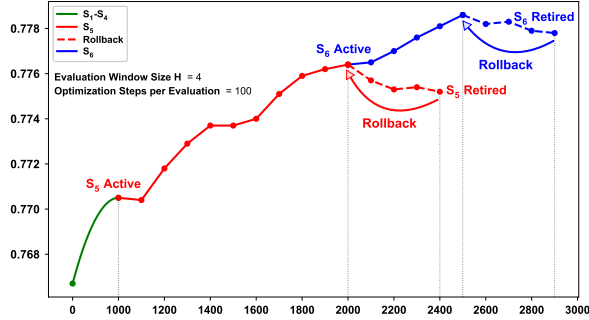


Figure 3: Training dynamics of auxiliary-signal scheduling in LABPO. Detailed trajectories for S_1 – S_4 are omitted for clarity.

As shown in Figure 3, the auxiliary signal S_5 initially improves the development score but later reaches a plateau. When the running-best development score improves by less than ϵ over $H = 4$ consecutive evaluation intervals, S_5 is retired and the policy is restored to the highest-scoring checkpoint observed since S_5 was activated. The next auxiliary signal, S_6 , is then activated, after which development performance resumes an upward trend and eventually exceeds the previous peak obtained under S_5 . This trajectory indicates that an auxiliary reward may be useful during only part of the optimization process and may become ineffective as training progresses. Restoring the best checkpoint limits the performance degradation accumulated after the auxiliary signal ceases to provide sufficient improvement.

4.4 Reasoning Performance Comparison

To further investigate improvements in reasoning ability, we conduct a direct evaluation of reasoning traces. Using a win-rate framework, we separately compare the model trained with our method and the Qwen3-8B baseline against DeepSeek-R1. We evaluate reasoning traces along two general dimensions: **Helpfulness** and **Factual Correctness**. Gemini-3-Pro-Preview is employed as the judge.

The details of this experiment can be found in Appendix F. The specifics of the alignment evaluation between LLMs and human annotators are available in Appendix H. The LLM evaluator shows strong alignment with aggregated human judgments, achieving an accuracy of 0.84 and a Cohen’s κ of 0.756.

As shown in Figure 4, our method demonstrates notable improvements in reasoning quality com-

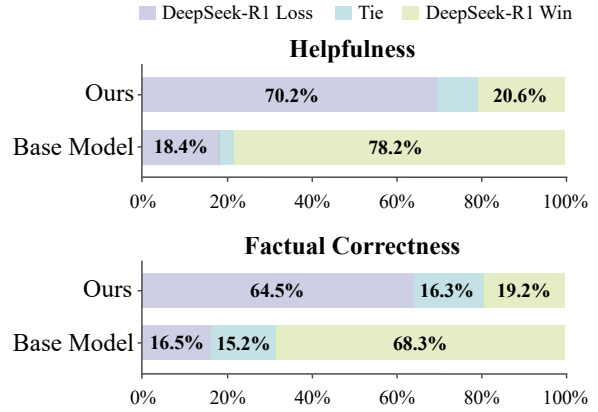


Figure 4: Pairwise reasoning-trace win rates against DeepSeek-R1 on ChJGBench. Ours: our two-stage model; Baseline: Qwen3-8B. Purple, blue, and green denote a DeepSeek-R1 loss, tie, and win, respectively.

pared to the baseline Qwen3-8B. In direct comparisons with the strong open-source model DeepSeek-R1, the win rate of our approach increases from 18.37% to 70.19% on Helpfulness and from 16.46% to 64.47% on Factual Correctness. Overall, the proposed training strategy enables an 8B-scale model to achieve reasoning performance on legal tasks that is competitive with or even superior to that of the 671B-parameter MoE model DeepSeek-R1.

4.5 Ablation Studies

Our ablation study on ChJGBench (Table 5) evaluates each component’s contribution; the trends on LegalBench and LawBench are consistent (see Appendix I).

(1) LABPO Mechanisms Improve Final RL Performance. Rows 1-4 analyze the effect of the Retire and Rollback mechanisms during reinforcement learning (RL). The full configuration (Row 1) achieves the best performance of 0.7786. Removing Rollback (Row 2) decreases the score to 0.7718, and removing both mechanisms (Row 3) further decreases it to 0.7671, indicating that both core mechanisms contribute to LABPO’s final performance. Moreover, Row 3 performs slightly worse than Row 4, which optimizes using only the primary reward. A possible reason is that different rewards may not share fully aligned optimization objectives, and aggregating multiple rewards can introduce noise or optimization conflicts.

(2) JurisCoT-SFT Improves the Effectiveness of Supervised Training. When JurisCoT-SFT is adopted (Row 5), the model achieves a score of

No.	SFT		RL			Avg. $\uparrow$
	Used	Variant	Used	Retire	Rollback	
1	✓	✓	✓	✓	✓	0.7786
2	✓	✓	✓	✓	×	0.7718
3	✓	✓	●	×	×	0.7671
4	✓	✓	○	×	×	0.7684
5	✓	✓	×	×	×	0.7667
6	✓	×	✓	✓	✓	0.7687
7	✓	×	×	×	×	0.7399
8	×	×	✓	✓	✓	0.6735
9	×	×	×	×	×	0.6406

Table 5: Ablation study of different training configurations. ✓ indicates the component is enabled and × indicates it is disabled. Row 3 reduces to GRPO with aggregated primary and auxiliary rewards, while Row 4 reduces to GRPO using only the primary reward.

0.7667, outperforming naive SFT (Row 7, 0.7399). A further ablation study of the SFT objectives, together with a sensitivity analysis of λ , is provided in Appendix G.

(3) LABPO Improves Performance Across Initialization Settings. Comparisons of Rows 1/5, 6/7, and 8/9 show that LABPO consistently improves performance under different initializations, boosting JurisCoT-SFT, naive SFT, and base models from 0.7667 to 0.7786, 0.7399 to 0.7687, and 0.6406 to 0.6735, respectively. This indicates that the gains stem not only from JurisCoT-SFT initialization but also from the proposed RL strategy.

5 Conclusion

In this paper, we introduce a two-stage training framework to enhance the legal reasoning capabilities of LLMs. We propose **JurisCoT-SFT**, which balances learning signals between reasoning traces and final answers, and **LABPO**, which dynamically schedules auxiliary reward signals. Our approach achieves strong performance on statutory and case law benchmarks. The ablation results further confirm the contributions of both training stages.

Limitations

Despite the strong empirical performance and methodological innovations of our approach, several limitations should be acknowledged. First, although the auxiliary signals in the LABPO framework no longer require meticulous selection thanks to the backtracking mechanism, the model still depends on a predefined reward function. Fully automating the design of reward functions (e.g., enabling LLMs to autonomously design appropri-

ate reward signals) remains an open challenge, and we leave this direction for future work. Second, due to resource constraints, we were unable to validate the effectiveness of our proposed two-stage approach on larger-scale models. Third, compared with simpler reinforcement learning methods such as GRPO, LABPO introduces additional computational overhead. The lifecycle management of auxiliary reward signals, periodic evaluation on the development set, and potential rollback operations require extra training iterations and evaluation costs. Although these mechanisms contribute to final performance, they may increase the overall computational budget. Designing more efficient reward scheduling or lightweight rollback strategies to reduce this overhead is an important direction for future research.

Ethics Statement

Given the sensitive nature of the legal domain, the application of artificial intelligence in this field necessitates rigorous ethical management. To address potential ethical concerns, we implement several safeguards. In particular, to prevent the leakage of private information (e.g., personal names), we replace sensitive information in the training datasets with neutral third-person references. This ensures that our research upholds the principles of privacy protection and responsible AI development.

Acknowledgements

This work was supported by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang (2025C02037), Research Project of the Philosophy and Social Sciences Planning Research Platform (Laboratory) of Zhejiang Province, and Key R&D Program of Hangzhou (2025SZDA0254). It was also supported by the National Natural Science Foundation of China (62106039) and Alibaba Research Intern Program. Any opinions expressed in this paper are those of the authors and do not necessarily reflect the views of the funding organizations.

We declare that AI-based writing tools were used only for limited language polishing and grammatical refinement in preparing this paper. The authors have carefully verified all AI-assisted text and take full responsibility for the content of the paper.

References

- Anthropic. 2025. [Claude Sonnet 4.5 System Card](#).
- 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 18117–18131, Suzhou, China. Association for Computational Linguistics.
- Haotian Chen, Yanyu Xu, Boyan Wang, Chaoyue Zhao, Xiaoyu Han, Fang Wang, Lizhen Cui, and Yonghui Xu. 2025. [Lexpro-1.0 technical report](#). *Preprint*, arXiv:2503.06949.
- 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.
- Xin Dai, Buqiang Xu, Zhenghao Liu, Yukun Yan, Huiyuan Xie, Xiaoyuan Yi, Shuo Wang, and Ge Yu. 2026. [LegalΔ: Enhancing legal reasoning in llms via reinforcement learning with chain-of-thought guided information gain](#). In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 16912–16916.
- Saoirse Connor Desai, Stian Reimers, and David A. Lagnado. 2016. [Consistency and credibility in legal reasoning: A bayesian network approach](#). In *Proceedings of the 38th Annual Meeting of the Cognitive Science Society, Recognizing and Representing Events, CogSci 2016, Philadelphia, PA, USA, August 10-13, 2016*. cognitivesciencesociety.org.
- Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Alan Huang, Songyang Zhang, Kai Chen, Zhixin Yin, Zongwen Shen, Jidong Ge, and Vincent Ng. 2024. [LawBench: Benchmarking legal knowledge of large language models](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7933–7962, Miami, Florida, USA. Association for Computational Linguistics.
- Zhiwei Fei, Songyang Zhang, Xiaoyu Shen, Dawei Zhu, Xiao Wang, Jidong Ge, and Vincent Ng. 2025. [InternLM-law: An open-sourced Chinese legal large language model](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9376–9392, Abu Dhabi, UAE. Association for Computational Linguistics.
- Michael J. Gerhardt. 2008. *The Power of Precedent*. Oxford University Press.
- Google DeepMind. 2026. [Gemini 3 Pro Model Card](#). Model release: November 2025; last updated: May 2026.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Aditya K, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory Dickinson, Haggai Porat, Jason Hegland, and 21 others. 2023. [Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 44123–44279. Curran Associates, Inc.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. [Deepseek-r1 incentivizes reasoning in llms through reinforcement learning](#). *Nature*, 645(8081):633–638.
- Wanwei He, Jiabao Wen, Lei Zhang, Hao Cheng, Bowen Qin, Yunshui Li, Feng Jiang, Junying Chen, Benyou Wang, and Min Yang. 2023. [Hanfei-1.0](#). <https://github.com/siat-nlp/HanFei>.
- Yinghao Hu, Leilei Gan, Wenyi Xiao, Kun Kuang, and Fei Wu. 2025a. [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, Abu Dhabi, UAE. Association for Computational Linguistics.
- Yinghao Hu, Linggang Kong, Qian Xiao, Yuting Huang, Yunlong Zhang, Yujie Zhang, Fei Duan, Yiquan Wu, Kaisong Song, Jun Lin, Kun Kuang, and Fei Wu. 2026. [Chjgbench: A comprehensive benchmark for chinese legal judgment generation with diagnostic evaluation](#).
- Yinghao Hu, Yaoyao Yu, Leilei Gan, Bin Wei, Kun Kuang, and Fei Wu. 2025b. [Evaluating test-time scaling LLMs for legal reasoning: OpenAI o1, DeepSeek-r1, and beyond](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 13759–13781, Suzhou, China. Association for Computational Linguistics.
- Cassidy Laidlaw, Shivam Singhal, and Anca Dragan. 2025. [Correlated proxies: A new definition and improved mitigation for reward hacking](#). In *International Conference on Learning Representations*, volume 2025, pages 88379–88421.
- Edward H Levi. 2013. *An introduction to legal reasoning*. University of Chicago Press.

- Ang Li, Yiquan Wu, Yifei Liu, Ming Cai, Lizhi Qing, Shihang Wang, Yangyang Kang, Chengyuan Liu, Fei Wu, and Kun Kuang. 2025. [UniLR: Unleashing the power of LLMs on multiple legal tasks with a unified legal retriever](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11953–11967, Vienna, Austria. Association for Computational Linguistics.
- OpenAI. 2025. [Gpt-5.1 instant and gpt-5.1 thinking system card addendum](#). Technical report, OpenAI.
- Dong Shu, Haoran Zhao, Xukun Liu, David Demeter, Mengnan Du, and Yongfeng Zhang. 2024. [Lawllm: Law large language model for the us legal system](#). In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM '24*, pages 4882–4889, New York, NY, USA. Association for Computing Machinery.
- Beining Wang, Weihang Su, Hongtao Tian, Tao Yang, Yujia Zhou, Ting Yao, Qingyao Ai, and Yiqun Liu. 2025. [From <answer> to <think>: Multidimensional supervision of reasoning process for llm optimization](#). Preprint, arXiv:2510.11457.
- ZeJun Wang. 2022. [simbert-base-chinese](https://huggingface.co/WangZeJun/simbert-base-chinese). <https://huggingface.co/WangZeJun/simbert-base-chinese>.
- Zihao Wang, Chirag Nagpal, Jonathan Berant, Jacob Eisenstein, Alexander Nicholas D’Amour, Sanmi Koyejo, and Victor Veitch. 2024. [Transforming and combining rewards for aligning large language models](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 51161–51176. PMLR.
- Xiaobao Wu. 2025. [A comprehensive survey on learning from rewards for large language models: Reward models and learning strategies](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 17847–17875, Suzhou, China. Association for Computational Linguistics.
- Yiquan Wu, Yuhang Liu, Yifei Liu, Ang Li, Siying Zhou, Kun Kuang, and Fei Wu. 2026. [Wisdominterrogatory \(luwen\): An open-source legal large language model technical report](#). Preprint, arXiv:2604.06737.
- xAI. 2025. [Grok 4 Model Card](#). Last updated: August 20, 2025.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). Preprint, arXiv:2505.09388.
- Shengbin Yue, Shujun Liu, Yuxuan Zhou, Chenchen Shen, Siyuan Wang, Yao Xiao, Bingxuan Li, Yun Song, Xiaoyu Shen, Wei Chen, Xuanjing Huang, and Zhongyu Wei. 2024. [Lawllm: Intelligent legal system with legal reasoning and verifiable retrieval](#). In *Database Systems for Advanced Applications*, pages 304–321, Singapore. Springer Nature Singapore.
- Kepu Zhang, Guofu Xie, Weijie Yu, Mingyue Xu, Xu Tang, Yaxin Li, and Jun Xu. 2025a. [Legal mathematical reasoning with LLMs: Procedural alignment through two-stage reinforcement learning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1586–1598, Suzhou, China. Association for Computational Linguistics.
- Kepu Zhang, Weijie Yu, Zhongxiang Sun, and Jun Xu. 2025b. [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, CIKM '25*, pages 4117–4127, New York, NY, USA. Association for Computing Machinery.
- Zhi Zhou, Jiang-Xin Shi, Peng-Xiao Song, Xiao-Wen Yang, Yi-Xuan Jin, Lan-Zhe Guo, and Yu-Feng Li. 2024. [Lawgpt: A chinese legal knowledge-enhanced large language model](#). Preprint, arXiv:2406.04614.

A Attempt-Bounded Self-Verification

This section presents a method for improving the quality of training data by filtering reasoning traces, as illustrated in Figure 5. This method is called **Attempt-Bounded Self-Verification**. Specifically, given a question–answer pair (x, a) , we prompt a strong LLM to generate a reasoning trace t . We then feed the pair (x, t) into the model to be trained, which produces multiple candidate answers $\hat{a}^{(k)}$. If any candidate answer matches the ground-truth answer a , we regard the corresponding training triple (x, t, a) as valid and retain it in the training set. We allow up to K attempts; if none of the

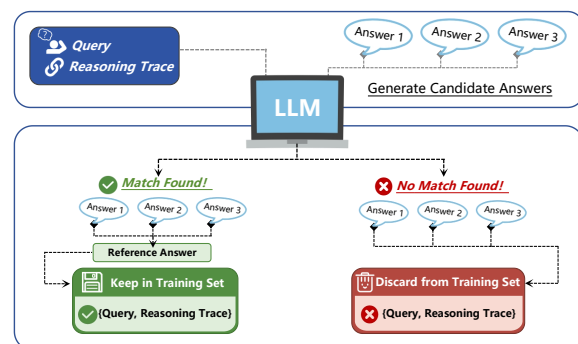


Figure 5: An overview of the Attempt-Bounded Self-Verification framework, in which a query and its reasoning trace are fed into the target model, and the pair is retained only if the model can infer the correct answer within a bounded number of attempts.

Signal Order	Avg. $\uparrow$
$S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4 \rightarrow S_5 \rightarrow S_6$	0.6735
$S_6 \rightarrow S_5 \rightarrow S_4 \rightarrow S_3 \rightarrow S_2 \rightarrow S_1$	0.6682
$S_4 \rightarrow S_3 \rightarrow S_6 \rightarrow S_1 \rightarrow S_5 \rightarrow S_2$	0.6671
$S_5 \rightarrow S_1 \rightarrow S_2 \rightarrow S_4 \rightarrow S_3 \rightarrow S_6$	0.6723
$S_3 \rightarrow S_4 \rightarrow S_1 \rightarrow S_6 \rightarrow S_5 \rightarrow S_2$	0.6704

Table 6: Ablation study on the effect of auxiliary signal ordering using Qwen3-8B. Reported values are average performance scores computed using Eq. (22).

generated candidates matches the reference answer, the training triple is discarded.

$$\mathcal{D}' = \left\{ (x, t, a) \in \mathcal{D} \mid \exists k \leq K, \hat{a}^{(k)} \sim p_\theta(\cdot \mid x, t), \right. \\ \left. v(\hat{a}^{(k)}, a) = 1 \right\}. \quad (8)$$

Here, $\hat{a}^{(k)}$ denotes the k -th candidate answer sampled from the model conditioned on (x, t) , $p_\theta(\cdot \mid x, t)$ represents the conditional distribution defined by model parameters θ , and $v(\hat{a}^{(k)}, a) \in \{0, 1\}$ is a verification function indicating whether the candidate answer agrees with the ground-truth answer a .

B Pseudocode for Adaptive Auxiliary Signal Scheduling

Algorithm 1 summarizes the scheduling procedure. Each checkpoint stores the policy parameters, optimizer state, learning-rate scheduler state, retained-step counter, and associated development score; $\text{score}(\mathcal{C})$ returns the score stored in checkpoint $\mathcal{C}$.

C Effect of Auxiliary Signal Ordering

Because LABPO activates auxiliary signals sequentially, we examine its sensitivity to signal ordering. To isolate this factor, all runs start from the Qwen3-8B base checkpoint, bypass the JurisCoT-SFT stage, and are trained on the Chinese statutory-law dataset using LABPO. The six auxiliary signals defined in Table 1 are kept fixed, with only their order varied across the following five settings:

1. Forward order: $S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4 \rightarrow S_5 \rightarrow S_6$
2. Reverse order: $S_6 \rightarrow S_5 \rightarrow S_4 \rightarrow S_3 \rightarrow S_2 \rightarrow S_1$
3. Random order 1: $S_4 \rightarrow S_3 \rightarrow S_6 \rightarrow S_1 \rightarrow S_5 \rightarrow S_2$
4. Random order 2: $S_5 \rightarrow S_1 \rightarrow S_2 \rightarrow S_4 \rightarrow S_3 \rightarrow S_6$
5. Random order 3: $S_3 \rightarrow S_4 \rightarrow S_1 \rightarrow S_6 \rightarrow S_5 \rightarrow S_2$

Table 6 summarizes the results for the five tested signal orders. The average scores range from

0.6671 to 0.6735, with a mean of 0.6703 and a sample standard deviation of 0.0027. The difference between the highest and lowest scores is 0.0064. **These findings suggest limited performance variation across the tested signal orders in our experimental setting.** This conclusion is specific to the auxiliary signals evaluated here and should not be interpreted as evidence that auxiliary signals beyond this set are similarly insensitive to ordering.

D Reward Functions for Auxiliary Signals

Section 3.2.2 and Table 1 introduce six auxiliary reward signals for legal reasoning. This appendix provides their detailed definitions and calculation formulas.

These signals broadly cover three categories of errors. Tag Format Validation and JSON Schema Validation address output structure and parsability. Statute Citation Format Verification and Language Style Validation address legal expression and presentation. Statute Authenticity Verification and Statute-Conclusion Consistency Verification address substantive legal reliability.

D.1 Tag Format Validation

Model outputs are expected to follow a strict structure consisting of exactly one `<think>...</think>` block followed by exactly one `<answer>...</answer>` block. The former contains the reasoning process, whereas the latter contains the final answer. Whitespace is permitted before, between, and after the two blocks, but no other content is allowed. Let $\mathcal{Y}_{\text{Fmt}}$ denote the set of outputs that satisfy this structure. The reward function is defined as:

$$R_{\text{Fmt}}(y) = \begin{cases} 1, & y \in \mathcal{Y}_{\text{Fmt}}, \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

D.2 JSON Schema Validation

The JSON Schema Validation reward is applied only to the payload extracted from the unique `<answer>...</answer>` block. Let $a(y)$ denote this payload. We parse $a(y)$ as a JSON object and validate it against the schema specified for the corresponding task. A reward of 1 is assigned if the payload is valid JSON and contains all required keys with the expected data types. A reward of 0.5 is assigned if it is valid JSON but fails to fully satisfy the schema because one or more required keys are missing or have incorrect data types. If no unique answer payload can be extracted, or if the

Algorithm 1 Adaptive Auxiliary Signal Scheduling for Legal Reasoning

Require: Training set $\mathcal{D}_t$; policy π_θ ; optimizer state ω ; learning-rate scheduler state ζ

Require: Development set $\mathcal{D}_{\text{dev}}$; score $S(\theta)$

Require: Auxiliary rewards $\{R_i^{\text{aux}}\}_{i=1}^m$; activation order $\sigma = (\sigma_1, \dots, \sigma_m)$

Require: Target retained-step budget N ; evaluation interval n ; minimum gain ϵ ; patience window H

Ensure: Final policy parameters θ

```
1:  $\alpha \leftarrow \mathbf{0}$ ,  $k \leftarrow 1$ ,  $a \leftarrow \sigma_k$ ,  $\alpha_a \leftarrow 1$ , and phase  $\leftarrow$  auxiliary
2: Set the retained-step counter  $s \leftarrow 0$  and the lifecycle-local evaluation index  $r \leftarrow 0$ 
3:  $B_0 \leftarrow S(\theta)$ ; save  $\mathcal{C}^{\text{best}} \leftarrow \text{Checkpoint}(\theta, \omega, \zeta, s, B_0)$ 
4: while  $s < N$  do
5:   Sample  $x \sim \mathcal{D}_t$ , roll out  $y \sim \pi_\theta(\cdot | x)$ , and update  $\pi_\theta$  by GRPO using Eq. 2
6:    $s \leftarrow s + 1$ 
7:   if  $s \bmod n = 0$  then
8:      $r \leftarrow r + 1$ ; compute  $q_r \leftarrow S(\theta)$ 
9:     if  $q_r \geq B_{r-1}$  then
10:       $\mathcal{C}^{\text{best}} \leftarrow \text{Checkpoint}(\theta, \omega, \zeta, s, q_r)$ 
11:     end if
12:      $B_r \leftarrow \max(B_{r-1}, q_r)$ 
13:     if  $r \geq H$  and  $B_r - B_{r-H} < \epsilon$  then
14:       if phase = auxiliary then
15:         Retire the active reward:  $\alpha_a \leftarrow 0$ 
16:         Restore  $(\theta, \omega, \zeta, s)$  from  $\mathcal{C}^{\text{best}}$ ; keep  $\alpha$  and phase unchanged
17:         if  $k < m$  then
18:            $k \leftarrow k + 1$ ,  $a \leftarrow \sigma_k$ , and  $\alpha_a \leftarrow 1$ 
19:         else
20:            $\alpha \leftarrow \mathbf{0}$  and phase  $\leftarrow$  primary
21:         end if
22:         Reset the phase-local history:  $r \leftarrow 0$  and  $B_0 \leftarrow \text{score}(\mathcal{C}^{\text{best}})$ 
23:       else
24:         Restore  $(\theta, \omega, \zeta, s)$  from  $\mathcal{C}^{\text{best}}$ ; keep  $\alpha$  and phase unchanged
25:       return  $\theta$   $\triangleright$  Terminate after restoring the best checkpoint from the phase using only the primary reward
26:     end if
27:   end if
28: end if
29: end while
30: return  $\theta$ 
```

$\triangleright$ The retained-step budget N has been reached

extracted payload is not valid JSON, the reward is 0. The reward function is defined as:

$$R_{\text{JSON}}(y) = \begin{cases} 1, & \text{valid and schema-complete } a(y), \\ 0.5, & \text{valid but schema-incomplete } a(y), \\ 0, & \text{missing or invalid } a(y). \end{cases} \quad (10)$$

D.3 Statute Citation Format Verification

Model outputs should include statute citations in a canonical format. If all statute citations are correctly formatted, the model receives a reward of 1. If the citations are missing or incorrect, the reward is 0. The reward function is defined as:

$$R_{\text{CFmt}}(y) = \begin{cases} 1, & \text{correct citation format,} \\ 0, & \text{incorrect format or missing.} \end{cases} \quad (11)$$

D.4 Language Style Validation

Model outputs are expected to adhere to a formal, neutral, and objective writing style. Because these stylistic attributes are difficult to assess reliably using rule-based methods alone, we employ Gemini-3-Pro-Preview as an LLM-based evaluator. Given

a predefined evaluation prompt P_{Style} specifying the assessment criteria, the evaluator assigns each response y a style reward:

$$R_{\text{Style}}(y) = E_{\text{Style}}(y; P_{\text{Style}}) \in \{0, 0.5, 1\}, \quad (12)$$

where E_{Style} denotes the LLM-based style evaluator. The reward is assigned according to the following criteria:

- A score of 1 is assigned when the response is formal, neutral, and objective.
- A score of 0.5 is assigned when the response is generally formal, neutral, and objective, with only minor subjective or emotional expressions.
- A score of 0 is assigned when the response is informal or contains substantial subjective or emotional language.

D.5 Statute Authenticity Verification

We verify the authenticity of every statute citation instance in the reasoning trace included in output

y , requiring each cited statute to correspond to an actual legal provision. This reward builds on the Non-Hallucinated Statute Rate introduced in prior work (Hu et al., 2025a).

1. For each statute citation instance, extract the generated statute using regular expressions or LLMs, denoted as $L_{\text{gen}} = \{S_{\text{name}}, S_{\text{number}}, S_{\text{content}}\}$, where S_{name} is the name of the statute, S_{number} is the statute number, and S_{content} is the content of the statute.
2. Embed the extracted content together with all statute texts in the reference database using SimBERT (Wang, 2022).
3. Calculate the semantic similarity between the generated content and each real statute. Select the real statute with the highest similarity as the best match and denote it by $L_{\text{best}} = \{S'_{\text{name}}, S'_{\text{number}}, S'_{\text{content}}\}$. The rationale behind this process is that although large models may not generate entirely hallucination-free statutes, they generally maintain high semantic consistency with real statutes. The closer the model-generated statute is to a real statute, the higher its semantic similarity.
4. Using rule-based comparisons, we determine whether the generated statute L_{gen} is consistent with the matched real statute L_{best} . Specifically, L_{gen} is considered non-hallucinated if the generated statute content occurs as a substring of the matched official statute text, $S_{\text{number}} = S'_{\text{number}}$, and $S_{\text{name}} \in \mathcal{A}(S'_{\text{name}})$, where $\mathcal{A}(S'_{\text{name}})$ denotes the predefined set of accepted names and aliases for the matched statute.

Let $N_{\text{cite}}(y)$ denote the total number of statute citation instances in the reasoning trace of output y , and let $N_{\text{nh}}(y)$ denote the number of non-hallucinated statute citation instances among them. We use the proportion of non-hallucinated statute citation instances as the reward score:

$$R_{\text{StatReal}}(y) = \frac{N_{\text{nh}}(y)}{N_{\text{cite}}(y)}. \quad (13)$$

If no statute citation instance is extracted, i.e., $N_{\text{cite}}(y) = 0$, we set $R_{\text{StatReal}}(y) = 0$. Otherwise, $R_{\text{StatReal}}(y)$ takes a value between 0 and 1. A higher value indicates that a larger proportion of the generated statute citation instances correspond to authentic legal provisions.

D.6 Statute-Conclusion Consistency Verification

We evaluate whether the statutes cited in each model response support its final conclusion. Because this relationship is difficult to assess reliably using rule-based methods alone, we employ Gemini-3-Pro-Preview as an LLM-based evaluator. Given a predefined evaluation prompt specifying the assessment criteria, the evaluator assigns each response a consistency reward according to the relationship between the cited statutes and the final conclusion:

$$R_{\text{consistency}}(y) = \begin{cases} 1, & \text{direct support,} \\ 0.5, & \text{potential support,} \\ 0, & \text{contradiction or irrelevance.} \end{cases} \quad (14)$$

A reward of 1 is assigned when the cited statutes directly support the conclusion, 0.5 when they provide potential support, and 0 when they contradict or are irrelevant to the conclusion.

E Identifying Reward-Hacking Behavior

To examine whether continuously optimizing against a fixed auxiliary reward can induce reward-hacking behavior, we conduct a controlled diagnostic experiment.

Experimental Setup. We initialize both reinforcement-learning variants from the same JurisCoT-SFT checkpoint, which is also directly evaluated as the SFT baseline. During reinforcement learning, we retain the complete set of primary rewards used in the main experiments and introduce S_5 , Statute Authenticity Verification, as the only auxiliary reward.

We evaluate all three models on the same fixed subset of ChJGBench and report four metrics. In addition to the Law and S_5 reward scores, we report Avg. Statute Citations per Response and Avg. Mentions per Distinct Correct and Relevant Statute to determine whether an increase in the S_5 reward score is driven by redundant citations or genuine improvements in legal-task performance.

The two reinforcement-learning variants use the same training data, sampling parameters, and primary rewards; their key distinction is how S_5 is scheduled. In conventional GRPO, S_5 remains active throughout training, whereas LABPO manages its lifecycle and retires it when its marginal contribution diminishes.

We compare three settings: (1) direct evaluation of JurisCoT-SFT without reinforcement learning; (2) JurisCoT-SFT followed by conventional GRPO using the primary rewards and S_5 ; and (3) JurisCoT-SFT followed by LABPO using the same rewards.

Results. As shown in Table 7, continuously optimizing against S_5 under conventional GRPO increases the S_5 reward from 0.542 to 0.934, while the task-level *Law* score changes only marginally, from 0.667 to 0.669. Meanwhile, the average number of statute citations per response increases from 3.18 to 12.46, and the average number of mentions per distinct correct and relevant statute increases from 1.21 to 4.87.

This pattern reveals a clear reward-hacking strategy. Because S_5 is computed over individual statute citation instances, repeatedly citing an authentic statute adds to both the number of non-hallucinated citation instances and the total number of citation instances while leaving the number of hallucinated citation instances unchanged. Consequently, the model can increase the ratio-based S_5 reward by diluting the contribution of hallucinated citations, without meaningfully improving its performance on statutory provision application.

In comparison, LABPO substantially limits this behavior: the average number of statute citations per response decreases to 4.27, and the average number of mentions per distinct correct and relevant statute decreases to 1.48, while the *Law* score increases to 0.681. Although its S_5 reward is lower than that achieved by conventional GRPO, LABPO performs better on the underlying legal task.

F Reasoning Performance Comparison Details

1. We use the test set described in Section 4.1 to generate complete reasoning traces for three different models: Qwen3-8B, Qwen3-8B with JurisCoT-SFT + LABPO, and DeepSeek-R1, all under the same prompt template. The generation settings are as follows: temperature = 0.7, top_p = 0.95, and max_new_tokens = 8192.
2. For each comparison, we provide Gemini-3-Pro-Preview with an input formatted as {query, ground-truth answer, reference reasoning trace, Model A reasoning trace, Model B reasoning trace}. The evaluator is asked to produce a win, tie, or loss judgment for Model

A relative to Model B along two dimensions: Helpfulness and Factual Correctness. To reduce sampling variability, we set the evaluator temperature to 0.

3. The definitions for these two evaluation dimensions are as follows: Helpfulness primarily evaluates whether the reasoning process is clear, logical, and contributes to solving the problem. Factual Correctness assesses whether the facts in the reasoning process are accurate and error-free. For Factual Correctness, Gemini-3-Pro-Preview uses the ground-truth answer and reference reasoning trace to determine which model’s reasoning trace better aligns with the reference information or contains fewer factual discrepancies.
4. For any pairwise model comparison, the win rate for a specific dimension is defined as:

$$\text{Win Rate} = \frac{\text{Number of wins in that dimension}}{\text{Total number of samples}}. \quad (15)$$

G Ablation Study of Signal-Rebalancing Objectives for SFT

Experimental Setup. To better understand the source of the performance gains from JurisCoT-SFT at the SFT stage, we compare standard SFT, a signal-rebalancing baseline based on Answer-Token Upsampling, and JurisCoT-SFT under different values of λ . All experiments in this section exclude LABPO, allowing us to isolate the effects of the SFT objective. Across all variants, we use the same base model (Qwen3-8B), training data, batch size, learning-rate schedule, maximum sequence length, and number of training steps.

Standard SFT. For each training instance, let x denote the input query, $t = (t_1, \dots, t_{|t|})$ the reasoning trace, and $a = (a_1, \dots, a_{|a|})$ the final answer. We define the unnormalized negative log-likelihood losses over the reasoning and answer tokens as follows:

$$\mathcal{L}_t = \sum_{i=1}^{|t|} -\log p_{\theta}(t_i | x, t_{<i}). \quad (16)$$

$$\mathcal{L}_a = \sum_{j=1}^{|a|} -\log p_{\theta}(a_j | x, t, a_{<j}). \quad (17)$$

The standard SFT objective minimizes the average token-level negative log-likelihood over the concatenated reasoning and answer sequence:

Model	Law	S_5 Reward Score	Avg. Statute Citations per Response	Avg. Mentions per Distinct Correct and Relevant Statute
JurisCoT-SFT	0.667	0.542	3.18	1.21
JurisCoT-SFT + GRPO	0.669	0.934	12.46	4.87
JurisCoT-SFT + LABPO	0.681	0.761	4.27	1.48

Table 7: Results of the controlled diagnostic experiment on reward hacking using a fixed subset of ChJGBench.

$$\mathcal{L}_{\text{SFT}} = \frac{\mathcal{L}_t + \mathcal{L}_a}{|t| + |a|}. \quad (18)$$

Answer-Token Upsampling. Because legal reasoning traces are often substantially longer than final answers, reasoning tokens may dominate the standard SFT objective, thereby attenuating the supervision provided by the final answer. We therefore consider an Answer-Token Upsampling baseline that increases the contribution of answer tokens by a fixed factor k :

$$\mathcal{L}_{\text{AnsUp}} = \frac{\mathcal{L}_t + k\mathcal{L}_a}{|t| + k|a|}. \quad (19)$$

We set $k = 4$ because, on average, the reasoning traces in our data contain approximately four times as many tokens as the final answers. This baseline provides a simple heuristic for compensating for the attenuated answer-token signal. However, it requires k to be selected a priori and cannot adapt to instance-specific variation in the relative lengths of the two segments.

JurisCoT-SFT. JurisCoT-SFT instead normalizes the reasoning and answer losses separately and combines them using a weighting coefficient λ :

$$\mathcal{L}_{\text{JurisCoT-SFT}} = \lambda \frac{\mathcal{L}_t}{|t|} + (1 - \lambda) \frac{\mathcal{L}_a}{|a|}. \quad (20)$$

We evaluate $\lambda \in \{0.25, 0.50, 0.75\}$. The setting $\lambda = 0.50$ assigns equal aggregate weight to the reasoning trace and the final answer. Smaller values place greater emphasis on the final answer, whereas larger values place greater emphasis on the reasoning trace.

Results. Standard SFT achieves an average score of 0.7399. Answer-Token Upsampling with $k = 4$ increases the score to 0.7428, an absolute improvement of 0.0029. This modest gain indicates that simply increasing the global weight of answer tokens provides only limited benefit.

By contrast, all three JurisCoT-SFT variants outperform both standard SFT and Answer-Token Upsampling. The best performance is achieved with

Objective	Setting	Avg. $\uparrow$
Standard SFT	–	0.7399
Answer-Token Upsampling	$k = 4$	0.7428
JurisCoT-SFT	$\lambda = 0.25$	0.7629
JurisCoT-SFT	$\lambda = 0.50$	0.7667
JurisCoT-SFT	$\lambda = 0.75$	0.7612

Table 8: Ablation study of SFT objectives without LABPO. All variants are trained with Qwen3-8B under the same experimental configuration. Avg. denotes the average score on ChJGBench.

$\lambda = 0.50$, yielding a score of 0.7667 and absolute improvements of 0.0268 over standard SFT and 0.0239 over Answer-Token Upsampling. These results suggest that separately normalizing the two segment-level losses and explicitly controlling their relative contributions is more effective in this setting than either standard token-level averaging or fixed answer-token upsampling.

One possible explanation for why $\lambda = 0.50$ performs best is the quality of the reasoning supervision. In our main experiments, reasoning traces are retained for training only after undergoing quality-control procedures such as Attempt-Bounded Self-Verification. Consequently, these traces provide relatively reliable supervision, making it reasonable to assign comparable aggregate weights to the reasoning and answer losses in the present setting. However, this result should not be interpreted as evidence that equal weighting is universally optimal. When reasoning traces are less reliable, smaller values of λ may be preferable because they reduce the influence of noisy intermediate reasoning while placing greater emphasis on the final answer. This ability to adjust the balance according to the quality of the reasoning supervision illustrates the flexibility of the parameterized JurisCoT-SFT objective.

H Human Evaluation and Inter-Annotator Agreement

Human Evaluation Setup. We conduct a human evaluation to assess the reliability of the proposed reasoning evaluation protocol and to analyze

inter-annotator consistency. Three annotators with formal legal training independently evaluate 500 queries sampled from the test set using a stratified sampling strategy. To ensure representativeness, the queries are stratified by (i) legal domain, including criminal, civil, and administrative cases, and (ii) case complexity, measured by the length of the factual description. Each query corresponds to one pairwise comparison instance. The 500 instances are evenly divided into two comparison settings: our proposed method versus DeepSeek-R1 and our proposed method versus Qwen3-8B, ensuring a balanced number of pairs for both baselines.

Annotation Protocol. For each query, annotators perform pairwise comparisons between two anonymized reasoning traces along two dimensions: **Helpfulness** and **Factual Correctness**. The order of the two candidates is randomized for each instance to avoid positional bias. All annotations are conducted with annotators blinded to the model identities associated with the reasoning traces.

Aggregation Rule. For each query and each evaluation dimension, annotators assign one of three labels: $\{A \text{ wins}, \text{Tie}, B \text{ wins}\}$. Final human labels are determined using a conservative majority rule: a model is assigned a *win* label only if at least two out of three annotators explicitly prefer its reasoning trace. If neither model receives at least two win votes, the final label is determined as *Tie*. This aggregation strategy reduces the impact of subjective or ambiguous judgments and yields more reliable human evaluation results.

Agreement Metrics. We report a set of standard metrics to evaluate both human annotator consistency and human-LLM alignment.

For human inter-annotator agreement, we use Krippendorff’s α and Majority Agreement. Krippendorff’s α measures chance-corrected agreement among multiple annotators and is suitable for nominal labels. In our setting, the three comparison outcomes (*A wins*, *Tie*, *B wins*) are treated as nominal categories. Majority agreement measures the percentage of queries for which at least two of the three annotators assign the same label, indicating whether a clear dominant human preference exists and supporting the use of majority voting for label aggregation.

To assess the reliability of the LLM evaluator (Gemini-3-Pro-Preview), we compare its judgments with the aggregated human labels using Ac-

α	Majority	LLM Acc.	LLM κ
0.615	96.0%	0.84	0.756

Table 9: Human inter-annotator agreement and human-LLM alignment results on 500 queries.

No.	SFT		RL			Avg. $\uparrow$
	Used	Variant	Used	Retire	Rollback	
1	✓	✓	✓	✓	✓	0.7893
2	✓	✓	×	×	×	0.7593
3	✓	×	✓	✓	✓	0.7651
4	✓	×	×	×	×	0.7484
5	×	×	×	×	×	0.7220

Table 10: Ablation study on LawBench. A ✓ indicates that the component is enabled, while × indicates that it is disabled. Average denotes the mean of all metrics in the range [0, 1].

curacy and Cohen’s κ . Accuracy measures the proportion of exact matches between the LLM prediction and the human consensus, while Cohen’s κ provides a chance-corrected estimate of agreement between the two.

Results. Table 9 summarizes human inter-annotator agreement and human-LLM alignment. Krippendorff’s $\alpha = 0.615$ indicates a reasonably good level of agreement among the annotators, while the 96.0% majority agreement shows that at least two annotators selected the same label for most queries. The LLM evaluator achieves an accuracy of 0.84 against the aggregated human labels, with Cohen’s $\kappa = 0.756$ indicating substantial agreement under the same evaluation protocol. These results support the use of the LLM evaluator as an automatic judge for this task.

I Ablation Studies on LawBench and LegalBench

Table 10 and Table 11 report the ablation results on LawBench and LegalBench, respectively, using Qwen3-8B as the backbone model.

J Training Dataset Statistics

J.1 Statutory Law Dataset

Table 12 provides detailed statistics of the statutory law training dataset.

J.2 Case Law Dataset

Table 13 provides detailed statistics of the case law training dataset.

No.	SFT		RL			Avg. $\uparrow$	Sara-N $\downarrow$
	Used	Var	Used	Ret	Roll		
1	✓	✓	✓	✓	✓	0.826	1.753
2	✓	✓	×	×	×	0.777	3.949
3	✓	×	✓	✓	✓	0.787	5.906
4	✓	×	×	×	×	0.759	7.231
5	×	×	×	×	×	0.735	8.093

Table 11: Ablation study on LegalBench. A ✓ indicates that the component is enabled, and × indicates that it is disabled. Abbreviations: Var = Variant, Ret = Retire, Roll = Rollback. Average denotes the mean of all accuracy metrics. Sara-N corresponds to the Log MSE metric and is therefore reported separately.

Item	Count
Training instances (x, t, a)	2,734,100
Distinct statutes (clause level)	35,010
Causes of action	> 400
Criminal cases	581,579
Civil cases	1,625,584
Administrative cases	526,937
Average token count per reasoning trace	3,922.4
Average token count per answer	902.2

Table 12: Statistics of the statutory-law training dataset, including the total number of training instances, the number of distinct statutes at the clause level, the distribution of legal cases, and the average token counts per reasoning trace and per answer.

K Detailed Definitions of Evaluation Metrics

K.1 ChJGBench Evaluation Metrics

All diagnostic dimensions used in this work, together with their corresponding evaluation metrics, are adopted from ChJGBench (Hu et al., 2026). We refer readers to the original benchmark for their detailed definitions.

To account for the unequal numbers of diagnostic dimensions across litigation domains, we introduce **AVERAGE**, a domain-balanced hierarchical macro-averaging metric. Directly averaging all dimension-level scores would give greater weight to domains evaluated with more dimensions. We therefore first average the dimension-level scores within each domain and then assign equal weight to the resulting three domain-level scores. The resulting **AVERAGE** scores are reported in Tables 2 and 5.

Let $M_{\text{shared}} = \{\text{Law}, \text{ISS}, \text{JUS}\}$, and let M_d denote the domain-specific metrics. For the criminal, civil, and administrative domains, these

Item	Count
Training instances (x, t, a)	35,965
Average token count per reasoning trace	2,369.2
Average token count per answer	3.5

Table 13: Statistics of the case-law training dataset, including the total number of training instances and the average token counts in reasoning traces and answers.

metrics are $\{\text{Fine}, \text{MIT}, \text{Charge}, \text{Pen_M}, \text{REC}\}$, $\{\text{SUP}, \text{PAY}\}$, and $\{\text{DIS}\}$, respectively. Each domain-level score averages its shared and domain-specific dimensions:

$$S_d = \frac{1}{|M_{\text{shared}} \cup M_d|} \sum_{m \in M_{\text{shared}} \cup M_d} s_m. \quad (21)$$

The final **AVERAGE** score assigns equal weight to the three domains:

$$\text{AVERAGE} = \frac{S_{\text{Crim}} + S_{\text{Civ}} + S_{\text{Adm}}}{3}. \quad (22)$$

K.2 LawBench Evaluation Metrics

We select three representative Legal Judgment Prediction tasks from **LawBench** for evaluation. For classification-based tasks, namely Article Prediction and Charge Prediction, we use **F1** as the evaluation metric.

For Sentence Prediction, we use the official LawBench **Normalized Log-Distance (NLD)** metric (Fei et al., 2024). Let N_{eval} be the number of fixed-term-imprisonment cases after excluding references involving the death penalty or life imprisonment, and let y_i be the nonnegative reference term in months. After converting Chinese numerals, the evaluator parses the first duration in months, or the first duration in years multiplied by 12 if no month duration is found, as the nonnegative integer $\hat{y}_i$. It then computes

$$d_i = \begin{cases} |\log(\hat{y}_i + 1) - \log(y_i + 1)|, & \hat{y}_i \text{ is parsed,} \\ \log 216, & \text{otherwise,} \end{cases} \quad (23)$$

$$\text{NLD} = 1 - \frac{1}{N_{\text{eval}} \log 216} \sum_{i=1}^{N_{\text{eval}}} d_i. \quad (24)$$

If no duration is parsed, the response is an abstention and contributes an instance-level score of zero.

K.3 LegalBench Evaluation Metrics

For classification-based tasks in **LegalBench**, we adopt the **Accuracy (Acc)** metric to measure prediction correctness. Accuracy is defined as the

proportion of correctly predicted samples among all evaluated instances:

$$\text{Acc} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(\hat{y}_i = y_i), \quad (25)$$

where N denotes the total number of samples, $\hat{y}_i$ is the predicted label, y_i is the ground-truth label, and $\mathbb{I}(\cdot)$ is the indicator function that returns 1 when the prediction matches the ground truth and 0 otherwise.

For numerical reasoning tasks such as **Sara-N**, we employ **Log Mean Squared Error (Log MSE)** to evaluate prediction accuracy on continuous numerical outputs. Log MSE measures the squared difference between the logarithms of predicted and ground-truth values, which helps stabilize variance when values span multiple orders of magnitude:

$$\text{LogMSE} = \frac{1}{N} \sum_{i=1}^N (\log(\hat{y}_i) - \log(y_i))^2, \quad (26)$$

where N denotes the number of samples, $\hat{y}_i$ is the predicted numerical value, and y_i is the corresponding ground-truth value.

L Implementation Details

We adopt a two-stage training strategy. Training is conducted on 32 NVIDIA A100 GPUs with 80 GB of memory each.

In the first stage, we perform SFT on the Qwen3-8B and Llama-3.1-8B base models with a learning rate of 1×10^{-5} and a sequence length cutoff of 16,384. The per-device training batch size is set to 1, with 2 gradient accumulation steps. We use a cosine learning-rate scheduler with a warmup ratio of 0.1. The balancing coefficient is set to $\lambda = 0.5$.

In the second stage, we apply reinforcement learning with a learning rate of 1×10^{-6} , a global training batch size of 256, and a micro batch size of 128. In all main experiments, we set the target number of retained optimization steps to $N = 3000$, the minimum gain threshold to $\epsilon = 0.0005$, the patience window to $H = 4$, and the evaluation interval to $n = 100$.

LABPO distinguishes retained steps from total executed steps. The retained-step counter s determines progress toward N and is restored with the complete lifecycle-best checkpoint, including the policy parameters, optimizer state, and learning-rate scheduler state. The total executed-step count is not restored and is used only for overhead accounting. Consequently, steps discarded by an

auxiliary-signal rollback are re-executed during the subsequent training phase.

The primary reward remains active throughout reinforcement learning, while at most one auxiliary reward is active at any time. Development-set evaluation is performed after every $n = 100$ retained-step increments. Each phase maintains a running-best development score and its corresponding checkpoint. Once $r \geq H$ and $B_r - B_{r-H} < \epsilon$, the active auxiliary signal is retired, the lifecycle-best checkpoint is restored, and the next auxiliary signal is activated. The phase-local evaluation history is then reset. The stored score of the restored checkpoint initializes the new phase, avoiding an additional development-set evaluation.

After the last auxiliary signal has been retired, all auxiliary rewards are disabled and training continues using only the primary reward. The same monitoring criterion is applied during this final phase. If the criterion is satisfied, the phase-best checkpoint is restored and training terminates immediately without re-executing the discarded steps; otherwise, training continues until the retained-step counter reaches N .

The development set is used strictly for model selection, including auxiliary-signal retirement, checkpoint rollback, and early-termination decisions. It is never used for gradient updates or for optimizing the policy or reward models. The training, development, and test sets are strictly disjoint, and all final results are evaluated on a completely held-out test set that is never accessed during training or model selection.

M Computational Overhead Analysis

Compared with standard GRPO, LABPO introduces additional overhead from re-executing optimization steps discarded by auxiliary-signal rollback, periodic development-set evaluation, and checkpoint I/O.

Let N denote the target number of retained optimization steps, U the total number of executed optimization steps, t_{step} the average time per optimization step, and t_{eval} the time per development-set evaluation. For auxiliary-signal rollback a , let r_a and r_a^* denote the retirement index and the lifecycle-best checkpoint index, respectively. Since retirement is checked after every evaluation, the restored checkpoint can be at most H evaluation intervals before the retirement point. Therefore,

Task	Task Full Name	Litigation Type	Benchmark	Metric
<i>Statutory Law Tasks</i>				
Law	Statutory Provision Application	General	ChJGBench	Case-level set F1
ISS	Disputed Issues Identification	General	ChJGBench	LLM-as-a-judge
JUS	Judicial Justification	General	ChJGBench	LLM-as-a-judge
Charge	Criminal Charge Prediction	Criminal	ChJGBench	Defendant-level set F1
Fine	Monetary Fine	Criminal	ChJGBench	NVS
REC	Recidivism Assessment	Criminal	ChJGBench	Binary macro-F1
MIT	Mitigating Circumstances Identification	Criminal	ChJGBench	Binary macro-F1
Pen_M	Penalty Duration Prediction	Criminal	ChJGBench	NVS
SUP	Civil Claim Support Determination	Civil	ChJGBench	Three-class macro-F1
PAY	Payment Obligation	Civil	ChJGBench	NVS
DIS	Administrative Disposition Prediction	Administrative	ChJGBench	LLM-as-a-judge
AP	Article Prediction	Criminal	LawBench	F1
CP	Charge Prediction	Criminal	LawBench	F1
SP	Sentence Prediction	Criminal	LawBench	NLD
<i>Case Law Tasks</i>				
LRC	Legal Reasoning Causality	-	LegalBench	Accuracy
NYSJE	New York State Judicial Ethics	-	LegalBench	Accuracy
Sara-N	SARA Numeric Reasoning	-	LegalBench	Log MSE

Table 14: Summary of evaluation tasks, including statutory law tasks and case law tasks, with their corresponding litigation types, benchmark sources, and evaluation metrics. In ChJGBench (Hu et al., 2026), NVS stands for Normalized Value Similarity.

$$d_a = n(r_a - r_a^*) \leq Hn,$$

$$D_{\text{aux}} = \sum_{a=1}^{A_{\text{aux}}} d_a \leq A_{\text{aux}}Hn \leq mHn, \quad (27)$$

where $A_{\text{aux}} \leq m$ is the number of auxiliary-signal rollbacks and D_{aux} is the total number of re-executed steps. A rollback during the final phase in which only the primary reward is used is not included in D_{aux} , because it is followed immediately by termination.

If training reaches N retained steps, then $U = N + D_{\text{aux}}$. Early termination during the final phase can only reduce U . The development set is evaluated once before training and whenever the retained-step counter reaches a multiple of n , including repeated evaluations when discarded intervals are re-executed. Because the stored score of a restored checkpoint is reused at phase transitions,

$$U \leq N + D_{\text{aux}} \leq N + mHn,$$

$$E(U) = 1 + \left\lfloor \frac{U}{n} \right\rfloor \leq 1 + \left\lfloor \frac{N + mHn}{n} \right\rfloor, \quad (28)$$

where $E(U)$ is the total number of development-set evaluations.

The computational costs of standard GRPO and LABPO are

$$C_{\text{GRPO}} = Nt_{\text{step}},$$

$$C_{\text{LABPO}} = Ut_{\text{step}} + E(U)t_{\text{eval}} + C_{\text{io}}, \quad (29)$$

where C_{io} denotes the scheduling-related cost of checkpoint saving, loading, and restoration.

LABPO performs at most m auxiliary-signal restorations and one additional restoration during the final phase in which only the primary reward is used.

Under our configuration of $N = 3000$, $m = 6$, $H = 4$, and $n = 100$, the worst-case bounds are

$$\begin{aligned} D_{\text{aux}} &\leq 2400, \\ U &\leq 5400, \\ E(U) &\leq 55, \\ C_{\text{LABPO}} &\leq 5400t_{\text{step}} + 55t_{\text{eval}} + C_{\text{io}}. \end{aligned} \quad (30)$$

The corresponding relative overhead over standard GRPO is bounded by

$$\begin{aligned} \text{Overhead} &= \frac{C_{\text{LABPO}} - C_{\text{GRPO}}}{C_{\text{GRPO}}} \\ &\leq 0.8 + \frac{55}{3000} \frac{t_{\text{eval}}}{t_{\text{step}}} + \frac{C_{\text{io}}}{3000t_{\text{step}}}. \end{aligned} \quad (31)$$

Thus, auxiliary-signal rollback increases the optimization-step computation by at most 80%, corresponding to at most $1.8\times$ the optimization-step cost of standard GRPO before evaluation and checkpoint I/O are included. This is a theoretical worst-case bound; the actual re-execution overhead D_{aux}/N is smaller when the lifecycle-best checkpoint is close to the retirement point.

N Case Study

To assess the model’s accuracy in both reasoning traces and final answers on complex legal reasoning tasks, we examine the representative case shown

Query		From June to September 2016, Defendant [ANONYMIZED] illegally sold gasoline without the required retail and hazardous chemicals licenses. He procured the gasoline, and his co-conspirators sold it from a truck to roadside drivers at 4.5 CNY per liter, generating 83,523 CNY in sales. In September 2016, they were caught during a sale involving 2.98 tons of gasoline worth approximately 18,000 CNY. The defendant voluntarily surrendered in June 2017, confessed to the crime, and entered a guilty plea. What judgment should be rendered, and which legal provisions apply?			
Model Responses		Ground Truth	Qwen3-8B	DeepSeek-R1	Ours
Key Points of Reasoning Traces	Charge and Sentence	Illegal business operations 8 months in prison 1-year probation	Illegal business operations ✓ 1 year in prison ✗	Illegal business operations ✓ 1 year and 6 months in prison ✗	Illegal business operations ✓ 6 months in prison ⚠ (minor error) 1-year probation ✓
	Legal Provisions	Articles 225, 67(1), 72, 73, and 76 of the Criminal Law.	Articles 225, 67 ✓, and 64 ✗ of the Criminal Law. ⚠ Omission of Articles 72, 73, and 76 of the Criminal Law	The provisions for illegal business operations are in Article 225 ✓ of the Criminal Law... The provisions regarding voluntary surrender are in Article 67(1) ✓ of the Criminal Law. ⚠ Omission of Articles 72, 73, and 76 of the Criminal Law	According to Article 225 ✓ of the Criminal Law... The provisions on voluntary surrender are found in Article 67(1) ✓ of the Criminal Law... The basis for imposing fines is Article 52 ♡ (Correct addition beyond the ground truth) of the Criminal Law... Probation may be applicable, so the conditions for applying probation are governed by Articles 72 ✓ and 73 ✓ of the Criminal Law. ⚠ Omission of Article 76 of the Criminal Law
	Fine	A fine of 20,000 CNY.	A fine of 50,000 CNY ✗. recovery of illegal gains amounting to 83,523 CNY.	Illegal business operations usually involve a fine, which may be based on the business volume, such as one or multiple times that amount. However, the specific amount is uncertain and needs to be reasonably set, estimated at 10,000 CNY ✗.	The basis for imposing fines is Article 52 of the Criminal Law of the People's Republic of China ... Therefore, the defendant is fined 20,000 CNY ✓.

Figure 6: Case study: legal reasoning accuracy and completeness in a judgment concerning illegal business operations.

in Figure 6. This example is drawn from a typical criminal case involving illegal business operations and comprises multiple subtasks, including charge identification, sentencing determination, application of legal provisions, and fine calculation. We compare the outputs produced by the model trained using our proposed algorithm with those generated by two open-source models, Qwen3-8B and DeepSeek-R1, across three dimensions: Charge and Sentence, Legal Provisions, and Fine.

The case shows that our model correctly identifies the defendant’s voluntary surrender and eligibility for probation, with only a minor two-month discrepancy between the predicted prison sentence (six months) and the reference sentence (eight months). However, it omits one legal provision governing probation supervision (Article 76 of the Criminal Law). In contrast, both Qwen3-8B and DeepSeek-R1 fail to recognize the possibility of probation and instead predict prison sentences of 12 and 18 months, respectively. Additionally, both models omit legal provisions related to probation, including Articles 72, 73, and 76. This incomplete coverage of applicable legal provisions is reflected in incomplete sentencing rationales: both models predict imprisonment without considering probation. This example illustrates the importance of comprehensively recalling applicable legal provisions for reasonable sentencing.

Regarding fine determination, Qwen3-8B predicts a fine of 50,000 CNY, and DeepSeek-R1 predicts a fine of 10,000 CNY; both differ substantially from the reference fine of 20,000 CNY. In contrast, our model matches the reference amount. Notably, before generating the final answer, our

model correctly identifies and applies Article 52 of the Criminal Law, providing a more comprehensive legal basis for fine determination than the reference answer. This suggests that reliable numerical decision-making in legal tasks depends not only on strong multi-step reasoning but also on the integration of domain-specific legal knowledge or structural constraints.

In summary, this example illustrates how general-purpose LLMs may exhibit inadequate sensitivity to procedural and conditional legal rules and weak numerical reasoning in legal contexts. In this case, the proposed training method mitigates these issues, improving accuracy across key judgment elements while maintaining interpretable reasoning traces.