# Towards Better Legal Reasoning LLMs: Signal Balancing and Reward Scheduling

Anonymous ACL submission

## Abstract

In the legal domain, large language models (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 (GRPO) are not without drawbacks, facing issues including costly reward design, performance plateaus, and reward hacking. To address these challenges, we 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 activates and deactivates auxiliary reward signals based on their impact on primary performance metrics, enabling efficient reward utilization without manual intervention. Trained on 2.7 million real-world judicial decision triplets and evaluated on a professionally annotated benchmark of 3,462 cases, our method is fine-tuned on Qwen3-8B and achieves state-of-the-art performance on average across evaluation metrics, outperforming both specialized legal LLMs and larger general-purpose models. *The related resources will be released once the paper is accepted.*

## 1 Introduction

Recent advances in large language models (LLMs) have emphasized the significance of explicit reasoning (OpenAI, 2025; Qwen Team, 2025; DeepSeek-AI, 2025), where intermediate thought processes are generated before arriving at final answers, to enhance reliability and interpretability across complex tasks. This form of structured reasoning is particularly crucial in the legal domain, where decisions must be grounded in statutory provisions, logically coherent, and procedurally defensible (Levi, 2022). Due to the high demands for domain knowledge and logical reasoning, as well as the low tolerance for hallucinations (Dahl et al., 2024), achieving this in the legal field is far more challenging than in general domains.

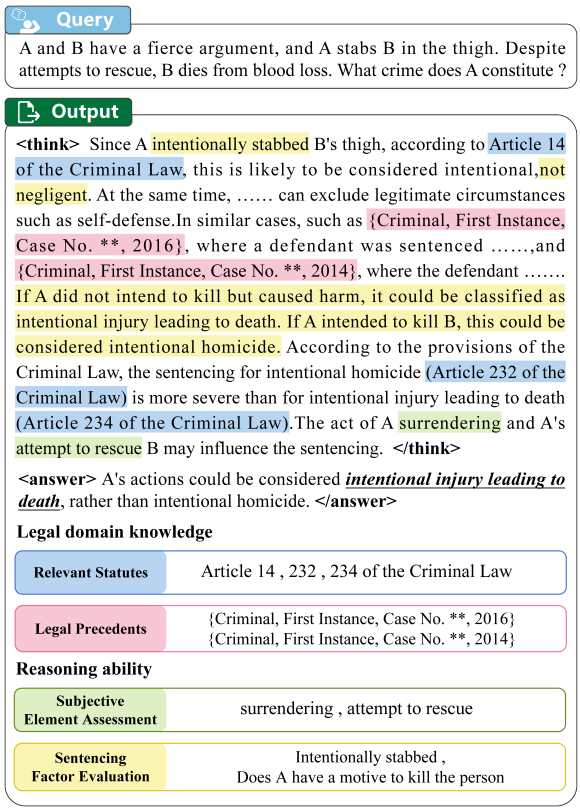


Figure 1: A diagram of a legal reasoning trace.

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. Regarding knowledge, this involves 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, 2011; Li et al., 2025). On the other hand, legal reasoning must adhere to several critical logical constraints, which include ensuring consistency between statutes and conclusions, maintaining internal logical coherence throughout the analysis, and comprehensively identifying all material legal issues (Desai et al., 2016).

Post-training plays a crucial role in enhancing the reasoning capabilities of LLMs. However, training LLMs to generate high-quality reasoning traces is far from straightforward (Chu et al., 2025). During the Supervised Fine-Tuning (SFT) stage, traditional legal-domain LLMs such as LawGPT (Zhou et al., 2024), HanFei (He et al., 2023), Wisdom-Interrogatory (Wu et al., 2023), and InternLM-Law (Fei et al., 2024) focus on injecting legal domain knowledge, but do not train the models to explicitly reason. DISC-LawLLM (Yue et al., 2023), which incorporates the legal syllogism framework into the training data design, helps LLMs acquire a thought structure. However, since reasoning traces are typically three to four times longer than final answers, **token-level loss accumulation implicitly biases the optimization objective toward the reasoning traces**, thereby diluting the learning signal for the final answers. Another approach is to use reinforcement learning, particularly Group Relative Policy Optimization (GRPO). LexPam (Zhang et al., 2025a), SyLeR (Zhang et al., 2025b), and Unilaw-R1 (Cai et al., 2025) all employ GRPO to explore areas such as numerical computation and reasoning path diversity. However, these methods require multiple handcrafted reward functions to evaluate various aspects of the reasoning process, such as format compliance and citation accuracy. **In practice, reward design is time-consuming, and reward signals that are initially effective may lose their effectiveness over time.**

We address these challenges with a two-stage training framework. In Stage I, we introduce **JurisCoT-SFT**, a variant of standard SFT that incorporates a length-normalized dual objective to address signal imbalance. In Stage II, we present **LABPO** (Lifecycle-Aware Backtrackable Policy Optimization), which dynamically manages the activation of auxiliary reward signals based on their impact on primary evaluation metrics. Each auxiliary signal is activated only when it measurably improves performance. Once its contribution plateaus or degrades overall model performance, it is automatically retired, and the policy reverts to a prior stable checkpoint, eliminating the need for continuous monitoring.

Our training corpus is derived from real judicial decisions, comprising over 2.7 million high-quality {*Query*, *Reasoning*, *Answer*} triplets. We evaluate our approach on a professionally annotated testbed of 3,462 real-world judicial cases. Fine-tuned on the Qwen3-8B model, our approach achieves state-of-the-art performance on average, surpassing specialized legal LLMs (Wu et al., 2023; Yue et al., 2024) and outperforming larger general-purpose models, including GPT-5.1 (OpenAI, 2025), Gemini-3-Pro-Preview (Google Cloud, 2025), Qwen3-235B-A22B-Instruct (Team, 2025), and Grok4 (xAI, 2025). Direct evaluation of reasoning traces shows substantial improvement. Compared with the strong open-source model DeepSeek-R1 (DeepSeek-AI, 2025), our method's win rate rises from 18.37% to 70.19% on Helpfulness and from 16.46% to 64.47% on Factual Correctness. Ablation studies also confirm the necessity of both stages: JurisCoT-SFT alone increases the average score from 0.6116 to 0.7580 by balancing learning signals between reasoning and answers, while LABPO further refines performance to 0.7702.

Our contributions can be summarized as follows:

1. We introduce **JurisCoT-SFT**, a novel variant of supervised fine-tuning, which effectively balances the learning signals between reasoning traces and final answers. This approach improves the model's ability to internalize critical legal knowledge, enhancing both reasoning quality and answer generation.

2. We present **LABPO**, a dynamic reward scheduling technique that adapts to the model's learning progress. LABPO ensures efficient use of auxiliary reward signals, eliminating the need for manual reward selection and minimizing performance plateaus.

3. Our approach, fine-tuned on a large-scale legal corpus, achieves state-of-the-art performance across multiple legal tasks. The model's improvements in reasoning trace quality and final answer accuracy represent a meaningful advance in legal reasoning performance for LLMs.

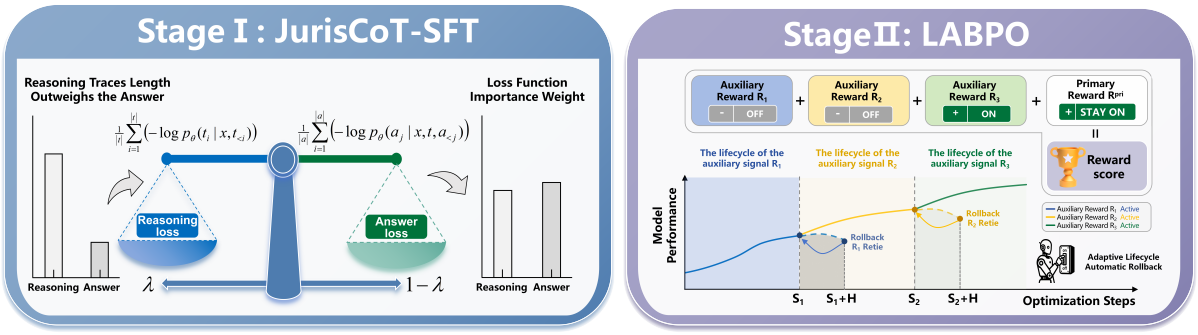


Figure 2: The left diagram depicts Stage I, JurisCoT-SFT, where the coefficient $\lambda$ balances learning between reasoning traces and final answers. The right diagram illustrates Stage II, LABPO, in which auxiliary reward signals are dynamically activated and deactivated. Solid lines indicate periods when an auxiliary reward signal is active, while dashed lines denote a decline in primary evaluation metrics that triggers a rollback.

## 2 Related Work

Traditional legal domain LLMs generally adopt a two-stage paradigm, consisting of continued pre-training and domain-specific fine-tuning, to achieve improvements in their legal capabilities. Examples include LawGPT (Zhou et al., 2024), HanFei (He et al., 2023), LawLLM (Shu et al., 2024), InternLM-Law (Fei et al., 2024), and Wisdom-Interrogatory (Wu et al., 2023). Prior studies have shown that explicitly generating intermediate reasoning traces before producing final answers can improve performance on certain legal reasoning tasks (Hu et al., 2025b). In the legal domain, where conclusions must be grounded in legally valid premises and structured argumentation, a growing body of work has therefore focused on training LLMs to generate high-quality reasoning traces. DISC-LawLLM (Yue et al., 2023) integrates a legal syllogism framework into the training data design. LexPam (Zhang et al., 2025a) targets legal numerical reasoning with a two-stage GRPO approach, optimizing numerical precision and incorporating legal elements into the reward function. SyLeR (Zhang et al., 2025b) enables syllogistic legal reasoning by combining SFT pre-training with PPO to explore multiple reasoning paths, though it primarily relies on general metrics like ROUGE, which may not fully capture the nuances of legal reasoning. LexPro (Chen et al., 2025) integrates SFT and GRPO for training but lacks a reward function tailored to legal contexts. Unilaw-R1 (Cai et al., 2025) introduces a reward function considering alignment with expected legal solutions but may transfer biases from untrained adjudication models. Despite these advancements and some new insights, existing methods still face challenges in how to better enable training LLMs to learn more from long reasoning traces, as well as how to more efficiently and effectively use reward functions to optimize the model's reasoning components. Based on 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, developing reliable LLMs requires ensuring both procedural legitimacy and conclusion correctness. This necessitates "think-then-answer" reasoning, where models first, in the "reasoning" stage, identify key issues and cite relevant legal provisions, and then, in the "answering" stage, provide succinct legal conclusions. To equip the model with explicit reasoning capabilities and to inject more comprehensive legal knowledge, we apply SFT to the large-scale dataset. 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, $t = (t_1, \ldots, t_{|t|})$ represents the provided reasoning sequence, and $a = (a_1, \ldots, a_{|a|})$ denotes the final answer. Given model parameters $\theta$, we optimize the next-token cross-entropy loss, with the reasoning and answer segments weighted separately and normalized by their respective lengths:

$$\mathcal{L}_{\text{JurisCoT}}(\theta) = \lambda \cdot \frac{1}{|t|} \sum_{i=1}^{|t|} \big( -\log p_\theta(t_i \mid x, t_{<i}) \big) + (1-\lambda) \cdot \frac{1}{|a|} \sum_{j=1}^{|a|} \big( -\log p_\theta(a_j \mid x, t, a_{<j}) \big) \tag{1}$$

where $\lambda \in [0, 1]$ is a balancing coefficient that trades off the importance between the reasoning traces ($t$) and the final answer ($a$). A smaller $\lambda$ prioritizes answer correctness, ensuring the model learns to produce correct final answers while still being guided by the structured reasoning process.

### 3.2 Stage II: Lifecycle-Aware Backtrackable Policy Optimization

Although models fine-tuned with JurisCoT-SFT demonstrate a competent grasp of legal knowledge and structured output, SFT fundamentally optimizes the token-level maximum likelihood objective on the given labeled data. Its performance ceiling, however, is often limited by the quality of the training data. More importantly, SFT lacks the ability to provide fine-grained supervision of intermediate reasoning steps. When reasoning traces contain various flaws, such as missing legal requirements, misapplication of precedent, or logical breaks, this method cannot offer differentiated feedback signals to guide the model in identifying and correcting specific types of errors. Therefore, a subsequent phase of alignment training is necessary. GRPO is currently the mainstream method for alignment training, but in standard GRPO, the reward function scores only at the outcome level. The result-based reward fails to reflect whether the intermediate reasoning steps are legally sound. A straightforward improvement is to design multiple experience-based rewards for different dimensions of the reasoning process (Cai et al., 2025; Wu, 2025; Wang et al., 2025). However, we have found that although these rewards initially constrain and guide the thinking process during the early stages of training, **the reward signals gradually lose effectiveness over time and may be vulnerable to reward hacking** (Khalaf et al., 2025; Laidlaw et al., 2025).

To address these challenges, we propose **LABPO** (**L**ifecycle-**A**ware **B**acktrackable **P**olicy **O**ptimization). In this framework, we define primary rewards aligned with the final evaluation objectives, which remain active throughout training to guide core optimization. Concurrently, we introduce auxiliary process rewards to provide intermediate constraints and guidance. The key innovation is that each auxiliary reward is governed by an independent lifecycle: it remains active only as long as it contributes to measurable improvements in the primary objectives. Once its contribution saturates or begins to degrade performance, it is automatically retired. Let the task type be $t \in \mathcal{T} = \{\text{CRIMINAL}, \text{CIVIL}, \text{ADMINISTRATIVE}\}$. For an input $x$ and model output $y$, we define the training reward at stage $\tau$ as:

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

where $R_t^{\text{pri}}$ is the fixed primary reward aligned with downstream evaluation metrics (Section 4.1), $\{R_i^{\text{aux}}\}_{i=1}^m$ are auxiliary reward components instantiated from auxiliary process signals (Table 1), and $\alpha_i(\tau) \in \{0, 1\}$ is a stage-dependent gating variable that controls the lifecycle of the $i$-th auxiliary reward. In the following sections, we provide a detailed explanation of how the dynamic lifecycle control mechanism updates $\alpha_i(\tau)$ during training, and describe the design and implementation of the auxiliary reward components $R^{\text{aux}}$.

#### 3.2.1 Adaptive Auxiliary Signal Scheduling

We propose an Adaptive Auxiliary Signal Scheduling mechanism that automatically adjusts the lifecycle of auxiliary signals during the training process, thereby alleviating plateauing and suppressing reward hacking. The pseudo-code of the algorithm can be found in Appendix A.3.

**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.** Every $n$ optimization steps, we evaluate the current policy $\pi_{\theta_\tau}$ on a hold-out development set $\mathcal{D}_{\text{dev}}$, which is independent of the validation and test sets. The performance score $S(\theta_\tau)$ obtained from this evaluation serves as the sole criterion for determining the lifecycle of the auxiliary signal.

**Dynamic Update of Auxiliary Signal Weights.** Based on the continuous evaluation results, we update the weight $\alpha_a$ of the currently active auxiliary signal $R_a^{\text{aux}}$. The update rule is as follows:

$$\alpha_a(\tau+1) = \begin{cases} 0, & \max\limits_{k\in\{1,\dots,H\}} S(\theta_\tau) - S(\theta_{\tau-k}) < \epsilon, \\ 1, & S(\theta_\tau) - S(\theta_{\tau-1}) > 0, \\ \alpha_a(\tau), & \text{otherwise.} \end{cases} \tag{3}$$

If the latest evaluation score shows improvement over the previous one, the current auxiliary signal is considered beneficial, and its activation state is maintained. Otherwise, we monitor performance over a sliding evaluation window of the most recent $H$ evaluations. If no significant improvement is observed within this window compared to the best historical checkpoint, we conclude that the auxiliary signal's marginal benefit has diminished or that it has induced optimization divergence.

**Checkpoint Rollback and Mechanism Iteration.** When the weight of the auxiliary signal is reset to 0, the policy parameters are rolled back to the checkpoint $\theta_{\tau-H}$, which was recorded $H \times n$ steps earlier. This rollback operation undoes any negative effects caused by ineffective auxiliary signals, returning the training to a more stable checkpoint. When an auxiliary signal is deactivated, the mechanism selects and activates the next auxiliary signal in a predetermined order and repeats the evaluation and dynamic weight update cycle.

#### 3.2.2 Auxiliary Process Signals for Legal Reasoning

Traditional reward functions typically require carefully crafted signals to ensure their effectiveness. In contrast, our method reduces reliance on precise signal design. Even if certain signals contribute little to the final task objective or cause interference, our rollback mechanism (Section 3.2.1) **can automatically identify and eliminate ineffective signals**. Specifically, if a signal does not result in significant improvements in performance over consecutive evaluations, the method retires the signal and reverts to the previous checkpoint. It then automatically switches to the next signal for testing. This mechanism ensures that the training process is not misled by ineffective signals as training progresses, thereby improving the method's robustness to noisy or irrelevant signals.

| Signal | Description |
|---|---|
| S1 | Tag-format Validation |
| S2 | JSON Schema Validation |
| S3 | Statute Citation Format Verification |
| S4 | Language Style Validation |
| S5 | Statute Authenticity Verification |
| S6 | Statute-conclusion Consistency Verification |

Table 1: List of auxiliary process signals used as rewards during training to guide legal reasoning. For details, please refer to Appendix A.4.

As shown in Table 1, we provide a set of simple and intuitive reward functions to score the model's thinking traces and final answers.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets. Training Data.** We construct a large-scale legal training corpus consisting of **{*Query*, *Reasoning*, *Answer*}** triplets. The *Query* and *Answer* components are extracted from judicial documents, while the *Reasoning Traces* are generated by a strong large language model. To control generation quality, we apply a variant of rejection sampling to filter the generated reasoning traces (Appendix A.1). To address the long-tail distribution of the training data, we further adopt a Tempered Sampling Approach (Appendix A.2). The resulting training dataset contains 2,734,100 instances, covering 35,010 distinct statutes at the clause level and over 400 causes of action. Detailed dataset statistics are provided in Appendix A.6.

**Evaluation Data.** We evaluate all models on a legal test set constructed in collaboration with a judicial institution. All the data in this dataset are real data, not synthetic, and they reflect real-world court decision-making scenarios. The dataset contains 3,462 cases from judicial proceedings and adjudication records between 2019 and 2024, spanning criminal, civil, and administrative domains.

**Baselines.** We evaluate our model against a series of strong baselines on the test set. These baselines include open-source general-purpose LLMs, such as Llama3.1-405B (Meta, 2024), DeepSeek-R1 (DeepSeek-AI, 2025), Qwen3-235B-A22B-

| Models | Avg. | Law | Reason | Focus | Criminal Case | | | | | Civil Case | | Administrative |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Recid. | Mitig. | Charge | Sentence | Fine | Support | Payment | Judge |
| *Specialized Legal LLMs* | | | | | | | | | | | | |
| WisdomInterrogatory | 0.3732 | 0.159 | 0.384 | 0.582 | 0.108 | 0.408 | 0.436 | 0.608 | 0.135 | 0.661 | 0.215 | 0.326 |
| LawLLM | 0.4641 | 0.181 | 0.547 | 0.729 | 0.370 | 0.727 | 0.414 | 0.466 | 0.080 | 0.682 | 0.052 | 0.420 |
| *Open-source LLMs* | | | | | | | | | | | | |
| Llama-3.1-405B | 0.5745 | 0.342 | 0.575 | 0.713 | 0.500 | 0.885 | 0.957 | 0.602 | 0.476 | 0.711 | 0.505 | 0.712 |
| DeepSeek-R1 | 0.6838 | 0.422 | 0.718 | 0.819 | 0.715 | 0.961 | 0.969 | 0.740 | 0.563 | 0.801 | 0.706 | 0.787 |
| Qwen3-235B-A22B-Instruct | 0.6897 | 0.461 | 0.748 | 0.812 | 0.765 | 0.974 | 0.967 | 0.780 | 0.603 | 0.799 | 0.687 | 0.653 |
| Grok-4 | 0.7011 | 0.499 | 0.728 | 0.818 | 0.668 | 0.973 | 0.976 | 0.724 | 0.501 | 0.781 | 0.645 | 0.799 |
| *Proprietary LLMs* | | | | | | | | | | | | |
| GPT-5.1 | 0.6549 | 0.422 | 0.707 | 0.763 | 0.665 | 0.965 | 0.975 | 0.691 | 0.556 | 0.787 | 0.665 | 0.688 |
| Claude-Sonnet-4.5 | 0.7114 | 0.540 | 0.729 | 0.811 | 0.705 | 0.968 | 0.970 | 0.751 | 0.578 | 0.793 | 0.685 | 0.764 |
| Doubao-Seed-1.6-Thinking | 0.7299 | 0.560 | 0.751 | 0.825 | 0.715 | 0.974 | **0.977** | 0.773 | 0.595 | 0.812 | **0.714** | 0.783 |
| Gemini-3-Pro-Preview | 0.7650 | 0.644 | **0.767** | 0.843 | 0.766 | 0.974 | 0.971 | **0.786** | **0.612** | 0.839 | 0.700 | **0.826** |
| *Our Method* | | | | | | | | | | | | |
| Qwen3-8B | 0.6116 | 0.315 | 0.652 | 0.791 | 0.659 | 0.921 | 0.960 | 0.727 | 0.457 | 0.742 | 0.653 | 0.624 |
| with JurisCoT-SFT | 0.7580 | 0.667 | 0.748 | 0.830 | 0.747 | **0.977** | 0.976 | 0.770 | 0.574 | 0.823 | 0.677 | 0.802 |
| with JurisCoT-SFT + LABPO | **0.7702** | **0.689** | 0.749 | **0.847** | **0.786** | 0.972 | 0.974 | 0.766 | 0.578 | **0.841** | 0.681 | 0.811 |

Table 2: The model evaluation results table. In this table, the best performance and the second-best performance among all models are indicated in **bold** and underlined, respectively.

Instruct (Team, 2025) and Grok-4 (xAI, 2025); closed-source general-purpose LLMs, including GPT-5.1 (OpenAI, 2025), Claude-Sonnet-4.5 (Anthropic, 2025), Doubao-Seed-1.6-Thinking and Gemini-3-Pro-Preview (Google DeepMind, 2025); and legal-domain-specific LLMs, including WisdomInterrogatory (Wu et al., 2023) and LawLLM-7B (Yue et al., 2024).

**Metrics.** For all test cases, we define three common tasks: determining the correct legal applicability (**Law**), producing legally grounded judicial reasoning (**Reason**), and identifying the central focus of dispute in a case (**Focus**). Because different types of litigation, namely criminal, civil, and administrative cases, involve distinct evaluation dimensions, we further design case-type-specific tasks. For **criminal cases**, we assess a series of decision-oriented tasks related to criminal liability and sentencing, including recidivism determination (**Recid.**), mitigation or sentence reduction decisions (**Mitig.**), charge determination (**Charge**), sentence prediction (**Sentence**), and fine amount determination (**Fine**). For **civil cases**, we evaluate whether the court should support the litigation request (**Support**), as well as the judicial determination of payment obligations (**Payment**). For **administrative cases**, in addition to the shared legal reasoning tasks, we evaluate the prediction of the final administrative judgment outcome (**Judge**). All classification-based tasks are evaluated using the F1 score, while amount-related judicial decisions are assessed with NAD_recall, following prior work (Feng et al., 2022; Xiao et al., 2018). Further details are provided in Appendix A.7.

**Implementation Details.** We adopt a two-stage training strategy. In the first stage, we perform SFT on Qwen3-8B with a learning rate of $1 \times 10^{-5}$ and a sequence length cutoff of 16,384. In the second stage, reinforcement learning is applied with a learning rate of $1 \times 10^{-6}$. Training is carried out on 32 NVIDIA A100 GPUs (80GB).

### 4.2 Main Results

We report the main experimental results, as shown in Table 2. Overall, specialized legal-domain LLMs (e.g., WisdomInterrogatory and LawLLM), due to their smaller base model sizes and outdated architectures, perform poorly on most tasks and significantly lag behind current state-of-the-art large models. This gap is especially evident in numerical decision-making tasks (e.g., Fine and Payment). These results indicate that if a model lacks sufficient inherent capability, possessing extensive legal knowledge alone is insufficient to ensure strong performance in legal scenarios.

In contrast, both open-source and closed-source general-purpose models demonstrate stronger overall performance across multiple tasks. Among open-source models, Qwen3-235B-A22B-Instruct ranks second, achieving scores of 0.780 on the Sentence task and 0.603 on the Fine task. Among closed-source models, Gemini-3-Pro-Preview ranks second overall with an average score of 0.7650, and performs particularly well on complex reasoning tasks such as Reason and Focus. However, these general-purpose models still fall short on tasks requiring high factual accuracy, such as Law, where they achieve a score of 0.644, trail-

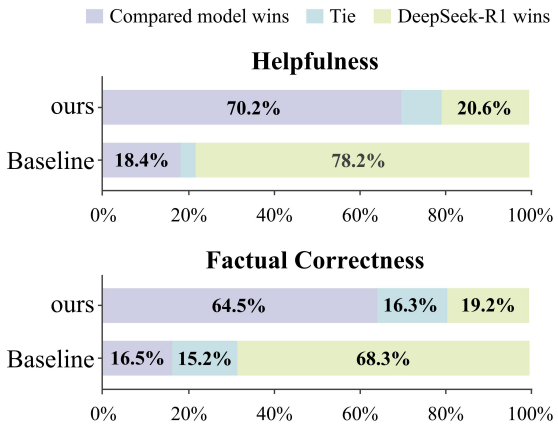


Figure 3: Comparison of Reasoning Traces: Helpfulness and Factual Correctness. Purple indicates cases where DeepSeek-R1 loses, blue indicates ties with the evaluated model, and green indicates cases where DeepSeek-R1 wins.

ing behind our specialized lightweight model.

Our proposed two-stage training approach achieves significant improvements on the lightweight base model Qwen3-8B. Using only the first stage, JurisCoT-SFT, the model's average score reaches 0.7580, surpassing all comparison models, including Gemini, on the Law task (0.667), and excels in criminal tasks such as Mitig.. Introducing the second stage of training, LABPO, further enhances the model's overall performance to 0.7702, representing a 25.9% improvement over the original base model, and achieves the best or second-best results across multiple tasks.

### 4.3 Reasoning Performance Comparison

To further investigate improvements in model reasoning ability, we conduct a direct evaluation of reasoning traces across different models. We adopt a win-rate based comparison framework to perform pairwise evaluations among the baseline model Qwen3-8B, our proposed method (JurisCoT-SFT + LABPO), and the strong open-source model DeepSeek-R1. Since the process-level signals S1–S6 introduced earlier involve substantial format-specific alignment requirements, Qwen3-8B and DeepSeek-R1 are not able to reliably follow such formats during the reasoning process; using these signals for evaluation would therefore introduce unfairness. We instead evaluate reasoning traces along two more general dimensions: **Helpfulness** and **Factual Correctness**. A strong LLM evaluator, Gemini-3-Pro-Preview, is employed as the judge. The detailed evaluation procedure is provided in Appendix A.5.

As shown in Figure 3, our method demonstrates notable improvements in reasoning quality compared 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, a much larger 671B-scale model.

Table 3: Ablation study of different model configurations. Symbols denote: ✓ (our proposed method), ○ (traditional method), and × (not used).

| Experiment Name | Stage I | Stage II | Average |
|---|---|---|---|
| JurisCoT-SFT + LABPO (Ours) | ✓ | ✓ | 0.7702 |
| JurisCoT-SFT + GRPO (all) | ✓ | ○ | 0.7513 |
| JurisCoT-SFT Only | ✓ | × | 0.7580 |
| Naive SFT Only | ○ | × | 0.7313 |
| **Baseline (no training)** | × | × | 0.6116 |

### 4.4 Ablation Studies

In our ablation study, we evaluate the individual impact of each component on the proposed framework, as shown in Table 3.

The first row demonstrates the highest performance (0.7702), achieved by using our proposed method (JurisCoT-SFT + LABPO) in both stages. The second row corresponds to a two-stage setting in which the first stage applies JurisCoT-SFT, while the second stage adopts the conventional GRPO and aggregates all primary and auxiliary signals to compute the reward score for each sample. We observe that this configuration yields lower performance than the third row. This result suggests that directly combining multiple auxiliary signals may introduce conflicts, thereby negatively affecting overall model performance. These findings further underscore the necessity of LABPO in the second stage. The third row represents a model trained solely using JurisCoT-SFT, yielding a performance of 0.7580. The fourth row presents a model trained only with traditional naive SFT, which achieves a performance of 0.7313. These results suggest that JurisCoT-SFT performs better on legal tasks than traditional SFT. The final row represents the untrained base model, with a performance of 0.6116.

| Query | | Defendant **, from June to September 2016, illegally sold gasoline without the required retail and hazardous chemicals licenses. He procured the gasoline, and co-conspirators sold it at 4.5 CNY per liter by truck to drivers on the roadside, totaling 83,523 CNY. In September 2016, they were caught during a sale with 2.98 tons of gasoline (worth about 18,000 CNY). ** voluntarily surrendered in June 2017, confessed to the crime, and agreed to a guilty plea. What should the judgment be, and which laws should be referenced? | | | |
|---|---|---|---|---|---|
| **Responser** | | **Ground truth** | **Qwen3-8B** | **DeepSeek-R1** | **Ours** |
| **Key Points of Reasoning Traces** | **Charge and Sentence** | Illegal business operations<br>8 months prison<br>1-year probation | **Illegal business operations** ✓<br>**1 year prison** ✗ | **Illegal business operations** ✓<br>**1 year 6 months prison** ✗ | **Illegal business operations** ✓<br>**6 months prison** ⚠ **(reasonable error)**<br>**1-year probation** ✓ |
| | **Law articles** | Articles 225, 67(1), 72, 73, and 76 of the Criminal Law. | Articles **225**, **67**, ✓ and **64** ✗ of the Criminal Law.<br>⚠ **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.<br>⚠ **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** 🏆 **(Supplemented the legal provisions regarding fines)** 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.<br>⚠ **Omission of Articles 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 4: Case Study: Legal Reasoning Accuracy and Completeness in Illegal Business Operations Judgment.

## 5 Case study

To analyze whether the model maintains high accuracy in reasoning traces and final answers when handling complex reasoning tasks, we selected representative examples from Figure 4 for qualitative analysis. This case originates from a typical criminal case of illegal business operations, involving multiple sub-tasks such as charge identification, sentencing discretion, application of legal provisions, and fine calculation. We will compare the reasoning traces of the model trained using the algorithm proposed in this paper with those of two open-source models, Qwen3-8B and DeepSeek-R1, across three dimensions: "Charge and Sentence," "Law Articles," and "Fine."

The case shows that our model accurately identified key facts, such as the defendant's voluntary surrender, low social harm, and eligibility for probation, with only a minor error of a 2-month discrepancy in the predicted prison sentence (6 months vs. the reference of 8 months). However, it missed one legal provision on the probation supervision mechanism (Article 76 of the Criminal Law). In contrast, both Qwen3-8B and DeepSeek-R1 failed to recognize the possibility of probation, directly predicting prison sentences of 12 and 18 months, respectively. Additionally, both models missed legal provisions related to probation (such as Articles 72, 73, and 76). This incomplete coverage of legal knowledge led to an incomplete sentencing reasoning chain, with the models defaulting to a prison sentence due to the absence of probation judgment. This demonstrates that the completeness of legal provision recall is essential for ensuring the reasonableness of sentencing.

Regarding fine generation, Qwen3-8B predicts a fine of 10,000 CNY and DeepSeek-R1 predicts 50,000 CNY, both deviating substantially from the reference fine of 20,000 CNY. In contrast, our model produces a more accurate amount. Notably, before generating the final answer, our model explicitly cites relevant legal provisions governing fine determination, although these provisions are not included in the reference answer. This suggests that reliable legal numerical decision-making depends not only on strong multi-step reasoning, but also on the integration of domain-specific legal knowledge or structural constraints.

In summary, this case illustrates common limitations of general-purpose large language models in judicial reasoning, such as inadequate sensitivity to procedural and conditional legal rules and weak modeling of legal numerical logic. The proposed training method mitigates these issues, improving judgment element accuracy while maintaining interpretable reasoning traces.

## 6 Conclusion

In this paper, we introduce a two-stage training framework for enhancing the legal reasoning capabilities of LLMs. We propose **JurisCoT-SFT**, a novel variant of supervised fine-tuning that balances learning signals between reasoning traces and final answers, and **LABPO**, a dynamic reward scheduling technique that optimizes auxiliary reward signals. Our approach achieves state-of-the-art performance on a large-scale legal corpus, surpassing specialized legal LLMs and general-purpose models. Extensive evaluations confirm the effectiveness of our method in improving both reasoning trace quality and final answer accuracy.

## Limitations

Despite the strong empirical performance and methodological innovations of our approach, several limitations should be acknowledged. First, while our training data is large-scale and sourced from real judicial decisions, it is primarily drawn from a single jurisdiction, predominantly focusing on Chinese civil, criminal, and administrative cases. This may limit the generalizability of our trained model to other legal systems. Second, although the auxiliary signals accompanying the LABPO framework no longer require meticulous selection due to the presence of a backtracking mechanism, the model still relies on a predefined reward function. Fully automating the design of reward functions (e.g., allowing LLMs to autonomously design suitable reward signals) remains a challenge, and this is an area we intend to explore further in future work. Finally, due to resource constraints, we were unable to validate the effectiveness of our proposed two-stage approach on models of larger scale.

## 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 anonymize or replace sensitive data with neutral third-person references when constructing both training datasets and evaluation benchmarks. This ensures that our research upholds the principles of privacy protection and responsible AI development.

## References


Anthropic. 2025. Introducing claude sonnet 4.5.

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.

Haotian Chen, Yanyu Xu, Boyan Wang, Chaoyue Zhao, Xiaoyu Han, Fang Wang, Lizhen Cui, and Yonghui Xu. 2025. Lexpro-1.0 technical report.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. 2025. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*.

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.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Saoirse Desai, Stian Reimers, and David Lagnado. 2016. Consistency and credibility in legal reasoning: A bayesian network approach. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 38.

Zhiwei Fei, Songyang Zhang, Xiaoyu Shen, Dawei Zhu, Xiao Wang, Maosong Cao, Fengzhe Zhou, Yining Li, Wenwei Zhang, Dahua Lin, et al. 2024. Internlm-law: An open source chinese legal large language model. *arXiv preprint arXiv:2406.14887*.

Yi Feng, Chuanyi Li, and Vincent Ng. 2022. Legal judgment prediction: A survey of the state of the art. In *IJCAI*, pages 5461–5469.

Michael J Gerhardt. 2011. *The power of precedent*. Oxford University Press.

Google Cloud. 2025. Gemini 3 pro preview model card. https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/3-pro. Accessed: 2025-11-18.

Google DeepMind. 2025. Gemini 3 pro model card and technical overview. https://en.wikipedia.org/wiki/Gemini_(language_model). Accessed: 2025-12.

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.

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*, page 13759–13781. Association for Computational Linguistics.

Hadi Khalaf, Claudio Mayrink Verdun, Alex Oesterling, Himabindu Lakkaraju, and Flavio du Pin Calmon. 2025. Inference-time reward hacking in large language models. *arXiv preprint arXiv:2506.19248*.

Cassidy Laidlaw, Shivam Singhal, and Anca Dragan. 2025. Correlated proxies: A new definition and improved mitigation for reward hacking.

Edward H Levi. 2022. *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.

Meta. 2024. Meta-llama-3.1-405b model card. https://huggingface.co/meta-llama/Llama-3.1-405B. Accessed: 2025-12.

OpenAI. 2025. Gpt-5.1 instant and gpt-5.1 thinking system card addendum. Technical report, OpenAI.

Qwen Team. 2025. Qwen3-235b-a22b-thinking-2507.

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*, pages 4882–4889.

Qwen Team. 2025. Qwen3 technical report.

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. *arXiv preprint arXiv:2510.11457*.

Xiaobao Wu. 2025. A comprehensive survey on learning from rewards for large language models: Reward models and learning strategies. *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 17847–17875.

Yiquan Wu, Yuhang Liu, Yifei Liu, Ang Li, Siying Zhou, and Kun Kuang. 2023. wisdominterrogatory. Available at GitHub.

xAI. 2025. Grok 4.

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, et al. 2018. Cail2018: A large-scale legal dataset for judgment prediction. *arXiv preprint arXiv:1807.02478*.

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

Shengbin Yue, Shujun Liu, Yuxuan Zhou, Chenchen Shen, Siyuan Wang, Yao Xiao, Bingxuan Li, Yun Song, Xiaoyu Shen, Wei Chen, et al. 2024. Lawllm: Intelligent legal system with legal reasoning and verifiable retrieval. In *International Conference on Database Systems for Advanced Applications*, pages 304–321. Springer.

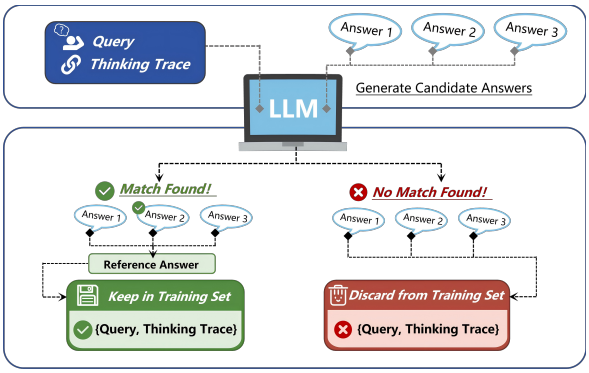


Figure 5: An overview of the Attempt-Bounded Self-Verification framework, in which a query and its reasoning trace are fed into the target model, and the pair is retained only if the model can infer the correct answer within a bounded number of attempts.

Wang ZeJun. 2022. simbert-base-chinese. https://huggingface.co/WangZeJun/simbert-base-chinese.

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.

Kepu Zhang, Weijie Yu, Zhongxiang Sun, and Jun Xu. 2025b. An explicit syllogistic legal reasoning framework for large language models.

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.

# A Appendix

## A.1 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 large language model 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 generated candidates matches the reference answer, the training triple is discarded.

$$\mathcal{D}' = \Big\{ (x,t,a) \in \mathcal{D} \;\Big|\; \exists\, k \le K,\ \hat{a}^{(k)} \sim p_\theta(\cdot \mid x,t),\ v(\hat{a}^{(k)}, a) = 1 \Big\}. \tag{4}$$

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 $\theta$, and $v(\hat{a}^{(k)}, a) \in \{0,1\}$ is a verification function indicating whether the candidate answer agrees with the ground-truth answer $a$.

### A.2 A Tempered Sampling Approach for Judicial Data

In real-world judicial practice, LLMs often generate factual errors (i.e., 'hallucinations') and produce responses that are of limited use when addressing rare causes of action or infrequently cited legal provisions. These issues result from the **long-tail distribution** of training data. The long-tail distribution refers to the severe class imbalance inherent in real judicial data, with a small number of causes of action accounting for the majority of cases, while many legally valid categories appear only sporadically. If training samples are drawn strictly according to the natural distribution, the resulting dataset contains too few long-tail instances for the model to adequately learn these underrepresented categories. On the other hand, equal sampling across all categories substantially over-amplifies rare causes of action while under-representing high-frequency ones. This trade-off poses particular challenges in the judicial domain. High-frequency causes of action dominate real-world judicial workloads, significantly impacting the practical utility of legal large language models. However, long-tail categories, though rare, often correspond to specialized regulations or exceptional legal circumstances that require precise handling (e.g., treason).

To mitigate this issue, we employ a tempered rebalancing strategy to enhance the sampling of long-tail causes of action while preserving the overall structure of the real-world distribution. Specifically, we apply an additional resampling step based on a power-temperature transformation of the true cause-of-action distribution $P(c)$, where $c$ denotes the cause-of-action category associated with each training sample, as shown below:

$$\tilde{P}_\tau(c) := \frac{(P(c))^\tau}{\sum_k (P(k))^\tau}, \quad \tau \in (0,1]. \tag{5}$$

When $\tau < 1$, this transformation moderately increases the sampling probability of long-tail categories while correspondingly reducing that of head categories.

### A.3 Pseudo-code for Adaptive Auxiliary Signal Scheduling

This section provides the pseudo-code for Lifecycle-Aware Backtrackable Policy Optimization, as shown in the pseudo-code Figure 1.

### A.4 Reward function for auxiliary signals

In Section 3.2.2 and Table 1, we provide a set of simple and intuitive reward functions, and the specific calculation formulas are as follows.

#### A.4.1 Tag-format Validation

Model outputs are expected to follow a strict structure, with the reasoning process enclosed in `<think>...</think>`. Specifically, if the tags are correct and no extraneous content exists outside them, the model receives a reward of 1. If these conditions are not met, the reward is 0. The reward function is defined as:

$$R_{\text{Fmt}}(y) = \begin{cases} 1, & \text{if the format is correct,} \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

#### A.4.2 JSON Schema Validation

Model outputs must be valid JSON and contain all required keys. The reward depends on the JSON structure's completeness and validity. If the output is valid JSON with all required keys and correct data types, the model receives a reward of 1. If the JSON is valid but some optional keys are missing, the reward is 0.5. If the output is invalid JSON, the reward is 0. The reward function is defined as:

$$R_{\text{JSON}}(y) = \begin{cases} 1, & \text{valid, all required keys} \\ 0.5, & \text{valid, missing optional keys} \\ 0, & \text{invalid JSON} \end{cases} \tag{7}$$

#### A.4.3 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.

**Algorithm 1** Adaptive Auxiliary Signal Scheduling for Legal Reasoning

**Require:** training dataset $\mathcal{D}_t$; policy $\pi_\theta$ with parameters $\theta$; reference $\pi_{\text{ref}}$ (KL); fixed held-out development set $\mathcal{D}_{\text{dev}}$; primary metric score $S(\theta)$ (score of $\pi_\theta$ on $\mathcal{D}_{\text{dev}}$); auxiliary rewards $\{R_i^{\text{aux}}\}_{i=1}^m$ (at most one active at a time); evaluation interval $n$; gain threshold $\epsilon$; backtrack window $H$=3.

Select an active auxiliary index $a$; set $\alpha_a \leftarrow 1$ and $\alpha_{j\neq a} \leftarrow 0$; save checkpoint $\theta_0$
**for** optimization step $s = 1, 2, \ldots$ **do**
  Sample $x \sim \mathcal{D}_t$, roll out $y \sim \pi_\theta(\cdot|x)$, and update $\pi_\theta$ by GRPO using Eq. 2
  **if** $s \bmod n = 0$ **then**
    $\tau \leftarrow s/n$; save checkpoint $\theta_\tau \leftarrow \theta$; compute $S(\theta_\tau)$ on $\mathcal{D}_{\text{dev}}$
    **if** $S(\theta_\tau) - S(\theta_{\tau-1}) > 0$ **then**
      **continue** ▷ keep $R_a^{\text{aux}}$ active
    **else if** $\max_{k\in\{1,\ldots,\min(H,\tau)\}} \big(S(\theta_\tau) - S(\theta_{\tau-k})\big) < \epsilon$ **then**
      Retire $R_a^{\text{aux}}$: set $\alpha_a \leftarrow 0$; backtrack parameters $\theta \leftarrow \theta_{\tau-H}$
      Switch to a new auxiliary reward $a$ and set $\alpha_a \leftarrow 1$
    **end if**
  **end if**
**end for**

If the citations are missing or incorrect, the reward is 0. The reward function is defined as:

$$R_{\text{CFmt}}(y) = \begin{cases} 1, & \text{correct citation format} \\ 0, & \text{incorrect format or missing} \end{cases} \tag{8}$$

#### A.4.4 Language Style Validation

The model's output needs to ensure a formal, neutral, and objective writing style. It is difficult to judge the language style using rules alone, so we use a strong LLM, such as Qwen-max, to evaluate the text's writing style. We provide the LLM with a set of instructions detailing the criteria for style judgment. Based on these instructions, the LLM outputs a score.

$$R_{\text{Style}}(y) = \text{LLM}(\text{prompt}, \text{text}) \tag{9}$$

The meanings of the different scores are as follows:

- A score of 1 is assigned when the style is formal and neutral.
- A score of 0.5 is given when the style is mostly formal and objective, with minor subjective emotions that slightly affect the user's experience.
- A score of 0 is assigned when the style is informal or when the response is filled with a large amount of subjective emotions.

#### A.4.5 Statute Authenticity Verification

In our process, we perform a truthfulness check on all the statutes cited in the reasoning traces of the generated answer $y$. We require that the referenced statutes be real and not fabricated. The design of this reward score is inspired by the procedure outlined in (Hu et al., 2025a).

1. Extract the generated statutes using regular expressions or LLMs, denoted as $L_{\text{gen}} = \{S_{\text{name}}, S_{\text{number}}, S_{\text{content}}\}$, where $S_{\text{name}}$ is the name of the statute, $S_{\text{number}}$ is the statute number, and $S_{\text{content}}$ is the content of the statute.
2. Embed the extracted content using the SimBERT (ZeJun, 2022) model, alongside all contents from a real statute database.
3. Calculate the semantic similarity between the generated content and the real statutes. Select the real statute with the highest similarity as the response, denoted as $L_{\text{best}} = \{S'_{\text{name}}, S'_{\text{number}}, S'_{\text{content}}\}$. 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, assess whether $L_{\text{gen}}$ and $L_{\text{best}}$ indicate hallucinations in the model-generated statutes. Specifically, we consider the generated statute $L_{\text{gen}}$ to be non-hallucinated if: $S'_{\text{content}} \subseteq S_{\text{content}}, S'_{\text{number}} = S_{\text{number}}, S'_{\text{name}} \in$ valid appellations of $S_{\text{name}}$.

We consider the rate of non-hallucinated statutes as the reward score:

$$R_{\text{StatReal}}(y) = \frac{\text{non-hallucinated statutes}}{\text{Total number of statutes}} \tag{10}$$

where $R_{\text{StatReal}}$ is a value between 0 and 1. A higher value of $R$ indicates that a larger proportion of the

generated statutes are accurate and do not contain hallucinations.

#### A.4.6 Statute-conclusion Consistency Verification

Model outputs must ensure that the cited statutes are consistent with the final conclusion. Due to the difficulty of judging the consistency between the cited statutes and the final conclusion through rules alone, we utilize a strong language model, such as Qwen-max, to assess this consistency.

If the statutes directly support the conclusion, the model receives a reward of 1. If the statutes potentially support the conclusion, the reward is 0.5. If the statutes contradict or are irrelevant to the conclusion, the reward is 0. The reward function is defined as follows:

$$R_{\text{Consistency}}(y) = \begin{cases} 1, & \text{direct support} \\ 0.5, & \text{potential support} \\ 0, & \text{contradiction/irrelevance} \end{cases} \tag{11}$$

### A.5 Reasoning Performance Comparison Details

1. We use the test set mentioned in 4.1 to generate complete reasoning traces for three different models: Qwen3-8B, 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. We then provide the input, formatted as {query, ground truth answer, golden answer reasoning traces, modelA reasoning traces, modelB reasoning traces}, to Gemini-3-Pro-Preview. The model is asked to output the win, tie, and lose results for the reasoning traces of modelA and modelB across two dimensions (Helpfulness and Factual Correctness). To ensure fairness, the temperature setting for Gemini-3-Pro-Preview is set 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, the evaluator model, Gemini-3-Pro-Preview, refers to the ground truth answer and golden answer reasoning traces to compare which model's reasoning traces contain factual elements that align more closely with the reference answers, or have smaller discrepancies.

4. For any pairwise model comparison, the win rate for a specific dimension is defined as:

$$\text{Win Rate} = \frac{\text{Number of wins in that dimension}}{\text{Total number of samples}} \tag{12}$$

Table 4: Training dataset Statistics

| Item | Count |
|---|---|
| Training instances ($\{Q, T, A\}$) | 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 in Reasoning Traces | 3922.4 |
| Average token count in Answer | 902.2 |

### A.6 Training dataset Statistics

Table 4 provides detailed information about our training data.

### A.7 Detailed Definitions of Evaluation Metrics

We list our specific evaluation tasks and their corresponding evaluation metrics in Table 5. Depending on the nature of the task, we employ different evaluation criteria, which are detailed below.

For classification-based tasks, we adopt the **F1 score**, defined as the harmonic mean of precision and recall:

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{13}$$

For amount-related judicial decision tasks, we instead employ **NAD-recall** (Normalized Absolute Deviation Recall) to evaluate the accuracy of numerical predictions. Given a tolerance threshold $\tau$, a prediction $\hat{y}$ is considered correct if its normalized absolute deviation from the ground-truth value $y$ satisfies:

$$\frac{|\hat{y} - y|}{y} \leq \tau. \tag{14}$$

NAD-recall is then computed as the proportion of samples whose predictions fall within this accept-

able deviation range:

$$\text{NAD-recall} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\left(\frac{|\hat{y}_i - y_i|}{y_i} \leq \tau\right), \quad (15)$$

where $N$ denotes the total number of samples and $\mathbb{I}(\cdot)$ is the indicator function.

In the main text, the **AVERAGE** metric reported in Tables 2 and 3 is computed as a *sample-size weighted* average rather than a simple arithmetic mean. For each metric (or metric group) indexed by $k$, let $n_k$ be the number of samples it covers and $N$ be the total number of samples. We define the weight as

$$w_k = \frac{n_k}{N}. \quad (16)$$

We further define the domain set

$$\mathcal{D} = \{\text{Criminal},\ \text{Civil},\ \text{Administrative}\}, \quad (17)$$

where for each domain $d \in \mathcal{D}$, $\bar{s}_d$ denotes the mean score over the sub-metrics within that domain. The overall AVERAGE is then computed as

$$\text{AVG} = \frac{\sum_{m\in\{\text{Law, Reason, Focus}\}} w_m s_m \ + \ \sum_{d\in\mathcal{D}} w_d \bar{s}_d}{\sum_{m\in\{\text{Law, Reason, Focus}\}} w_m \ + \ \sum_{d\in\mathcal{D}} w_d}. \quad (18)$$

| Task | Task Full Name | Litigation Type | Metric |
|---|---|---|---|
| Law | Legal Applicability Determination | General | F1 |
| Reason | Judicial Reasoning Generation | General | F1 |
| Focus | Dispute Focus Identification | General | F1 |
| Recid. | Recidivism Determination | Criminal | F1 |
| Mitig. | Mitigation or Sentence Reduction Determination | Criminal | F1 |
| Charge | Charge Determination | Criminal | F1 |
| Sentence | Sentence Length Prediction | Criminal | NAD_recall |
| Fine | Fine Amount Determination | Criminal | NAD_recall |
| Support | Litigation Request Support Determination | Civil | F1 |
| Payment | Payment Obligation Determination | Civil | NAD_recall |
| Judge | Administrative Judgment Outcome Prediction | Administrative | F1 |

Table 5: Summary of evaluation tasks with an explicit litigation type column. Classification tasks are evaluated using the F1 score, while amount-related decisions are evaluated using NAD_recall.