Legal Mathematical Reasoning with LLMs: Procedural Alignment through Two-Stage Reinforcement Learning

Anonymous ACL submission

Abstract

Legal mathematical reasoning is essential for applying large language models (LLMs) in high-stakes legal contexts, where outputs must be both mathematically accurate and procedurally compliant. However, existing legal LLMs lack structured numerical reasoning, and open-domain models, though capable of calculations, often overlook mandatory legal steps. To address this, we present **LexNum**, the first Chinese legal mathematical reasoning benchmark, covering three representative scenarios where each instance reflects legally grounded procedural flows. We further propose **LexPam**, a two-stage reinforcement learning framework for efficient legal reasoning training. Leveraging curriculum learning, we use a stronger teacher model to partition data into basic and challenging subsets. A lightweight 1.5B student model is then fine-tuned with Group Relative Policy Optimization, which avoids costly value networks and enables stable training from sparse, end-of-sequence rewards. The first stage improves accuracy and format; the second introduces a novel reward to guide procedural alignment via task-specific legal elements. Experiments show that existing models perform poorly on LexNum, while LexPam enhances both mathematical accuracy and legal coherence, and generalizes effectively across tasks and domains.

1 Introduction

Legal mathematical reasoning is a critical capability for applying large language models (LLMs) in real-world legal systems. This domain-specific task—spanning scenarios such as compensation calculation and liability apportionment—requires not only the precise application of formulas and parameters, but also adherence to legally prescribed procedures, such as confirming responsibility and determining applicable insurance. Failing either component—computation or procedure—can result in legally invalid or non-compliant outcomes.

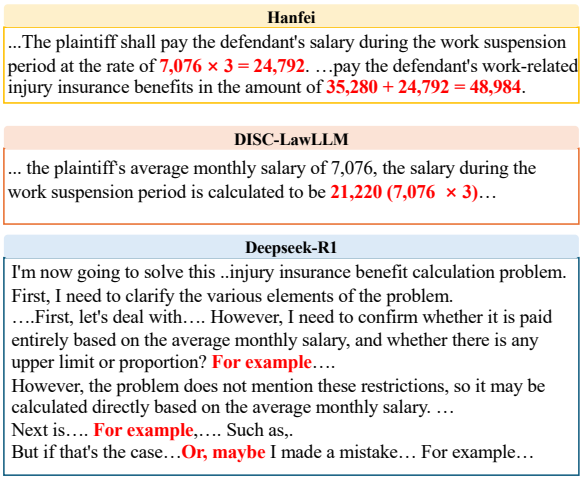


Figure 1: Responses to a work injury compensation case. Hanfei (He et al., 2023) and DiscLaw-LLM (Yue et al., 2023) miscalculates; DeepSeek-R1 (Guo et al., 2025) is uncertain and lacks procedural alignment.

This dual requirement makes legal mathematical reasoning fundamentally more complex than general mathematical (Cobbe et al., 2021) or legal language tasks (Yao et al., 2022).

However, existing approaches fall short in meeting this requirement from two directions. Legal-domain LLMs, such as fuzi.mingcha (Wu et al., 2023) and DISC-LawLLM (Yue et al., 2023), are fine-tuned for legal knowledge retention and textual generation, but lack the ability to perform accurate mathematical inference. As shown in Figure 1, these models may apply fixed formulas but miscalculate even simple operations, leading to numerically incorrect results. In contrast, open-domain reasoning models like DeepSeek-R1 (Guo et al., 2025) and OpenAI-o1 (Jaech et al., 2024) demonstrate strong step-by-step calculation abilities, yet fail to incorporate domain-specific legal logic. Their responses tend to over-explain or remain uncertain, often omitting essential statutory steps or legal constraints. As the bottom of Figure 1 illustrates, DeepSeek-R1 may recognize the need

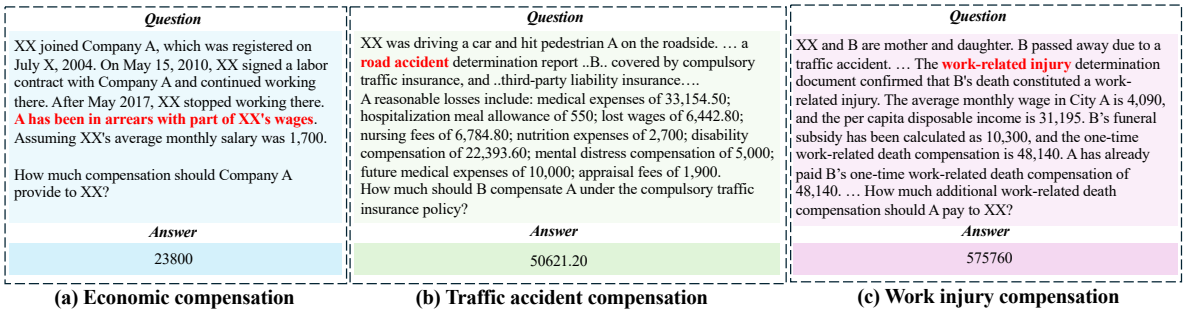


Figure 2: Examples from LexNum, our proposed benchmark for legal mathematical reasoning. Each scenario—(a) economic compensation, (b) traffic accident compensation, and (c) work injury compensation—requires multi-step reasoning that combines procedural logic with numerical calculation, reflecting real-world legal complexities.

for legal context but are unable to operationalize it, resulting in answers that are verbose, ambiguous, or procedurally invalid. These limitations underscore a critical gap: the need for models that can integrate both precise mathematical reasoning and legally grounded procedural logic.

Bridging this gap requires overcoming three key challenges. **First**, Most prior benchmarks either focus on general legal question answering (Zhong et al., 2020; Chen et al., 2023) or abstract arithmetic reasoning (Cobbe et al., 2021; Hendrycks et al., 2021), making it difficult to assess how well a model integrates legal procedure with numerical computation. While recent datasets such as LexEval (Li et al., 2024) and LawBench (Fei et al., 2023) include criminal sentencing tasks, these typically involve single-step reasoning with discrete outputs (e.g., prison terms), which differ significantly from the multi-step, interdependent calculations required in civil disputes. For instance, traffic accident compensation involves liability assessment, insurance coverage, and layered monetary calculations. Without benchmarks reflecting such real-world complexity, model predictions are difficult to validate or deploy in practice. **Second**, legal mathematical reasoning typically requires multi-step inference aligned with statutory procedures, yet fine-grained supervision over intermediate steps is rarely available. Annotating such reasoning paths is not only costly and time-consuming, but also prone to inconsistency due to the ambiguity and variability in legal interpretation. This lack of step-level supervision poses a significant obstacle for model training, requiring methods that can learn effectively and stably from sparse, end-of-sequence reward signals without relying on explicitly annotated intermediate steps. **Third**, aligning model behavior with legal reasoning requires training objectives that account for both numerical accuracy and procedural validity. However, most reward designs in existing reinforcement learning setups, such as the accuracy- and format-based rewards used in DeepSeek-R1, primarily target surface-level correctness and struggle to capture the structured logic mandated by legal norms. These reward functions provide insufficient guidance for producing responses that are not only correct in value, but also coherent with the legally required reasoning path as shown in Figure 1.

To address these challenges, we propose **LexNum**, the first Chinese benchmark explicitly designed for legal mathematical reasoning, and **LexPam**, a two-stage reinforcement learning framework for training LLMs under supervision constraints. LexNum covers three high-impact civil law scenarios—economic compensation, work injury, and traffic accident cases—each requiring multi-step reasoning that combines legal procedural logic with arithmetic precision. All instances are constructed to reflect real-world procedural dependencies, enabling rigorous evaluation of a model's ability to perform lawful and interpretable computation.

Building on this benchmark, LexPam introduces an efficient training strategy that aligns model outputs with legal reasoning goals while avoiding costly supervision. Inspired by curriculum learning, we use a stronger teacher model to partition the training data into basic and challenging subsets. A lightweight 1.5B student model is then fine-tuned using Group Relative Policy Optimization (GRPO) (Guo et al., 2025) that eliminates the need for value functions and learns directly from sparse, end-of-sequence rewards. The first stage improves numerical accuracy and response formatting, while the second introduces a novel reward that encourages procedural alignment by promoting the inclusion of scenario-specific legal elements.

This design enables training of LLMs for legally valid mathematical reasoning—without relying on manually annotated reasoning steps. Extensive experiments on LexNum show that existing legal LLMs perform poorly, with average accuracy below 15%. Our proposed LexPam achieves 60.65% accuracy, surpassing DeepSeek-R1-7B (31.52%) and GPT-4o-mini (56.93%), and approaching the performance of much larger models such as QwQ-32B (70.94%) and DeepSeek-R1-671B (73.42%). These results highlight the effectiveness of our efficient and procedurally aligned training framework.

To summarize, our contributions are as follows:

- We present LexNum, the first dataset explicitly designed for legal mathematical reasoning in Chinese, spanning three representative civil law scenarios: economic compensation, work injury, and traffic accident cases.
- We introduce LexPam, a reinforcement learning framework that incorporates legal procedural awareness into reward design, enabling LLMs to generate answers that are both mathematically accurate and procedurally compliant.
- We conduct comprehensive evaluations across legal- and reasoning-specific LLMs. Results show that existing models perform poorly on LexNum, while LexPam significantly improves legal mathematical reasoning, even outperforming models several times larger in scale.

2 Related Work

2.1 Legal LLM and Benchmark Dataset

Legal LLMs are typically adapted from open-domain models through domain-specific pretraining or fine-tuning using legal statutes, case documents, synthetic legal QA pairs, and annotated task-specific datasets (Yue et al., 2023; Wu et al.; Liu et al., 2023; He et al., 2023). These adaptations improve factual grounding and enable more relevant responses in legal applications.

Several benchmarks evaluate legal LLMs from different perspectives, including LexEval (Li et al., 2024), LawBench (Fei et al., 2023), and CitaLaw (Zhang et al., 2024b), which assess capabilities across judgment prediction, legal comprehension, citation extraction, and general legal QA. Recent studies have also begun to explore legal reasoning (Yu et al., 2025; Deng et al., 2024; Zhang et al., 2024a), but most focus on classification-oriented tasks such as judgment prediction or reading comprehension, rather than structured numerical reasoning.

While some prior benchmarks include legally motivated calculations—e.g., the "Penalty Prediction" task in LexEval and "Prison Term Prediction" tasks in LawBench, derived from CAIL-2018 (Xiao et al., 2018)—they are limited in scope and reasoning depth. These tasks typically involve discrete outputs and can often be solved via single-step application of criminal statutes.

In contrast, our proposed dataset LexNum differs in three key ways:

- **Domain scope**: LexNum targets civil law scenarios, including economic compensation, workplace injury, and traffic accident cases—practical domains central to everyday legal consultation.
- **Reasoning depth**: Unlike single-step prison term prediction, LexNum requires multi-step computation guided by statutory rules, such as liability ratio determination and conditional insurance application.
- **Answer format**: LexNum involves continuous-valued monetary outcomes with arithmetic operations (e.g., multiplication, division, percentage), moving beyond discrete classification.

By addressing legal mathematical reasoning in high-frequency civil contexts, LexNum fills a critical gap in current benchmarks and provides a foundation for developing legal LLMs that can support more realistic, procedurally grounded legal service

2.2 Reasoning

The emergence of reasoning-focused LLMs, such as DeepSeek-R1 (Guo et al., 2025) and OpenAI-o1 (Jaech et al., 2024), has led to rapid progress in open-domain reasoning. Recent work has explored reward shaping to constrain unnecessarily long reasoning chains (Aggarwal and Welleck, 2025; Luo et al., 2025), efficient token-level reasoning compression (Han et al., 2024), and boosting mathematical performance using limited high-quality data (Muennighoff et al., 2025; Ye et al., 2025). Other efforts have focused on knowledge distillation (Li et al., 2023; Zhu et al., 2024), transferring reasoning capabilities from large to smaller models for efficiency. However, this body of work focuses almost exclusively on open-domain tasks, such as math problems or code-based reasoning. In

contrast, legal mathematical reasoning introduces domain-specific procedural constraints that open-domain models fail to capture. To our knowledge, this work is the first to explore LLM reasoning under legal constraints, bridging open-domain reasoning and real-world legal applications.

3 The Proposed Dataset: LexNum

To construct LexNum, we collected real-world legal documents from pkulaw[1], covering three high-frequency litigation domains in Chinese civil law: economic compensation, workplace injury, and traffic accident cases. Each document includes structured metadata—case background, legal process, and final compensation result—and is pre-anonymized. We developed a two-stage pipeline involving LLM-assisted extraction and selective human verification to efficiently construct high-quality legal mathematical reasoning examples. Details of each subtask are provided in Appendix A.

3.1 LLM-Based Content Extraction

Raw legal documents contain substantial information irrelevant to mathematical reasoning. Rather than rely on fully manual annotation—accurate but prohibitively expensive—we adopt a lightweight LLM-based extraction approach. While deep reasoning models could potentially identify scattered reasoning chains, they are computationally inefficient. Instead, we use GPT-4o, which provides sufficient comprehension without heavy inference cost. Given a case document and relevant statutory provisions, GPT-4o is prompted to extract structured question-answer pairs related to legal mathematical reasoning. This includes identifying the numerical query, applicable rules, and the final outcome. Prompting strategies and examples are included in Appendix B.

3.2 Efficient Quality Assurance

To ensure quality while minimizing human labor, we adopt an LLM-then-human filtering strategy. First, GPT-4o is used to screen generated examples for completeness and internal consistency—i.e., whether the provided information suffices to derive the answer. Examples flagged as incomplete are passed to human reviewers. In addition, a small proportion of LLM-approved examples are also sampled for verification, as LLM outputs may contain subtle legal or numerical errors.

[1]https://www.pkulaw.com

We employ three reviewers with legal training for manual review. Initially, each annotator reviewed one full dataset, followed by a cross-checking phase where each reviewed 50% of the other two. A final audit was conducted by randomly sampling 50 examples per dataset. If any errors were found during the audit, a full re-check of the corresponding dataset was triggered. During annotation, reviewers corrected mismatches and refined unclear reasoning steps when needed. Full details are in Appendix C.

4 The Proposed Method: LexPam

In this section, we present LexPam, a two-stage reinforcement learning framework comprising curriculum-based data selection (§4.1), foundational training for legal mathematical reasoning (§4.2), and procedural alignment through targeted reward optimization (§4.3).

4.1 Curriculum-Based Data Selection

To optimize training efficiency under limited computational resources, we focus on fine-tuning a 1.5B reasoning LLM. Since the model exhibits varying performance across samples of different difficulty, we adopt a curriculum-based data partitioning strategy guided by a stronger 7B LLM (DeepSeek-R1-Distill-Qwen-7B). Specifically, we use the 7B model to perform inference over the training set. Samples it answers correctly are assigned to the first-stage training set $\mathcal{D}_1$, which emphasizes basic numerical computation and output format standardization. The remaining samples, which the 7B model fails to solve, form the second-stage training set $\mathcal{D}_2$, aimed at improving the model's ability for procedural alignment. This design provides a difficulty-aligned learning trajectory for the 1.5B model.

Discussion: We choose a 7B model rather than a larger or smaller alternative for two reasons: (1) Larger models introduce a significant performance gap, making their solvable samples too difficult for the 1.5B model to learn from effectively. (2) Smaller models, including the target 1.5B itself, provide insufficient guidance and cannot meaningfully differentiate between easy and hard samples for curriculum construction.

4.2 Stage-1: Foundation Training

The first stage focuses on low-complexity legal mathematical tasks to build the model's capability

in structured computation and standardized legal output formatting. We train on $\mathcal{D}_1$, the subset of samples solved by the teacher model, which serve as reliable examples for foundational learning.

Each input consists of a user query describing a compensation calculation, along with relevant facts. The model generates reasoning steps and a final result, where the intermediate reasoning chain is enclosed in "<think></think>" and the final answer in boxed{}. Training is performed using the GRPO algorithm (Shao et al., 2024). We define the corresponding reward as:

$$r_1 = r_{\text{correct}} + \alpha \cdot r_{\text{format}}, \tag{1}$$

where r_{correct} indicates whether the final boxed result matches the reference answer, and r_{format} evaluates whether the output adheres to the required structure. The hyperparameter α balances the focus between correctness and formatting.

4.3 Stage-2: Procedural Grounding

In this stage, we aim to instill procedural legal reasoning by training on $\mathcal{D}_2$, the subset of examples unsolved by the teacher. These samples typically require deeper legal understanding, domain-specific terminology, and adherence to statutory steps.

Building on the first stage's foundation of numerical correctness and structured output, we introduce an additional reward component, r_{law}, to promote the use of legally appropriate terminology and reasoning patterns. To operationalize this, r_{law} measures the presence of key legal elements in model outputs. These elements vary by task type, based on real-world legal procedures:

- Economic: `Compensation Type`, `Monthly Calculation`, `Compensation Calculation`
- Work Injury: `Injury Recognition`, `Liability`, `Benefit Calculation`, `Insurance`, `Compensation Calculation`
- Traffic Accident: `Liability`, `Insurance`, `Compensation Calculation`

To quantify this, we define a task-specific set with size N_j of legal keywords $\{k_{j,1}, \ldots, k_{j,N_j}\}$ for each task type j (e.g., economic compensation, work injury, or traffic accident). The procedural reward r_{law} is computed as the weighted coverage of these keywords in the model's output R:

$$r_{law} = \sum_{i=1}^{N_j} \omega_{j,i} \cdot \mathbb{I}(k_{j,i}, R), \tag{2}$$

Dataset	#Train	#Test	Avg_Train_Len	Avg_Test_Len
EC	1796	450	184.65	183.63
WC	774	194	168.79	170.16
TC	395	99	194.29	192.59

Table 1: Statistics of our LexNum. **#Train**, **#Test** denote the number of the train and test datasets. **Avg_Train_Len** and **Avg_Test_Sent** represent the average length of the train and the test datasets queries.

where $\omega_{j,i} \in [0, 1]$ is the importance weight assigned to the i-th keyword of task j, and $\mathbb{I}(\cdot)$ is the indicator function that evaluates to 1 if the keyword appears in R, and 0 otherwise. For simplicity, we use uniform weighting in our experiments. That is, we set $\omega_{j,i} = \frac{1}{N_j}$.

When the three datasets are merged into a unified training corpus (see §5.5), we first identify the target scenario from the input and compute r_{law} using the corresponding keyword set.

The full reward function for this stage combines correctness, formatting, and procedural alignment:

$$r_2 = r_{\text{correct}} + \alpha \cdot r_{\text{format}} + \beta \cdot r_{\text{law}}, \tag{3}$$

where β is the strength of procedural alignment.

Discussion: Because annotating high-quality legal reasoning paths is expensive and error-prone, we do not apply supervised fine-tuning (SFT) before RL. Instead, we directly train a distilled model using GRPO on the above reward function, improving efficiency and avoiding dependence on step-level annotations.

5 Experiments

5.1 Experimental Settings

5.1.1 Datasets and Metrics

We evaluate models on our proposed legal mathematical reasoning benchmark, LexNum, which consists of three task-specific subsets: Economic Compensation (EC), Work Injury Compensation (WC), and Traffic Accident Compensation (TC). Following standard practice in open-domain mathematical reasoning (Cobbe et al., 2021; Muennighoff et al., 2025), we report accuracy as the primary metric, measuring whether the final computed result exactly matches the ground truth. Dataset statistics are summarized in Table 1.

5.1.2 Evaluation Models

We selected both legal domain-specific LLMs and reasoning LLMs to investigate their performance on legal mathematical reasoning tasks.

For legal LLMs we chose seven models: **Zhihai** (Wu et al.), **fuzi.mingcha** (Wu et al., 2023), **DISC-LawLLM** (Yue et al., 2023), **LawGPT_zh** (Liu et al., 2023), **Tailing**[2], **LexiLaw**[3], and **HanFei**(He et al., 2023). These models have been extensively trained on legal datasets and possess substantial knowledge of legal scenarios.

We fine-tuned our models based on DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025) and compared them with **GRPO-Base** (Shao et al., 2024), which performs direct GRPO training using rewards that consider only response format and calculation results. Our proposed methods include **GRPO-Law**, which directly incorporates rewards based on legal procedures, and **LexPam**, which uses a two-stage RL training approach. For larger-scale LLMs, we evaluated models distilled by DeepSeek, specifically DeepSeek-R1-Distill-Qwen-7B (**DeepSeek-R1-7B**), DeepSeek-R1-Distill-Qwen-32B (**DeepSeek-R1-32B**), and **DeepSeek-R1** itself. In addition, we included **QwQ-32B-Preview** (Team, 2024) and **GPT-4o-mini** in our evaluation. Appendix D provides more details (URLs and licenses).

5.1.3 Implementation Details

We conducted training and testing on dual A6000 GPUs. For GRPO training, we set the learning rate to 1e-6, the number of generations (num_generations) to 4, and the maximum completion length to 768. We used LoRA for efficient fine-tuning of the LLM, with LoRA parameters set to r is 16 and alpha is 16. The hyperparameters α and β were both set to 0.1. For testing, we set the temperature to 0 to ensure reproducibility. We trained the model using DeepSpeed ZeRO-2 (Rajbhandari et al., 2020) and accelerated inference with vLLM (Kwon et al., 2023). More details can be found in `https://anonymous.4open.science/status/LexPam-50AB`.

5.2 Main Results

Table 2 presents accuracy scores on the LexNum benchmark, categorized into (1) legal-domain LLMs, (2) reasoning LLMs below 10B, and (3) larger-scale reasoning LLMs. We highlight the following findings:

LexPam outperforms all legal LLMs and small-scale reasoning models. LexPam achieves an average accuracy of 60.65%, substantially outperforming all legal-domain models as well as all reasoning models 10B, such as DeepSeek-R1-1.5B (18.78%). This demonstrates the effectiveness of LexPam's two-stage training strategy in teaching legal procedural reasoning even under tight model size constraints.

Model	Scale	EC	WC	TC	Avg
Legal LLM					
DISC-LawLLM	13B	12.44	2.58	18.18	11.07
fuzi.mingcha	6B	14.00	4.12	10.10	9.41
LexiLaw	6B	4.67	2.06	5.05	3.93
Tailing	7B	18.89	10.31	15.15	14.78
zhihai	7B	6.00	3.09	5.05	4.71
LawGPT_zh	6B	5.11	3.61	5.05	4.59
HanFei	7B	7.56	1.55	8.08	5.73
Small-Scale LLM (<10B)					
Deepseek-R1-1.5B	1.5B	16.00	22.16	18.18	18.78
Deepseek-R1-7B	7B	32.22	36.08	26.26	31.52
GRPO-Base	1.5B	52.00	61.86	38.38	50.75
GRPO-Law	1.5B	59.11	66.49	46.46	57.36
LexPam	1.5B	**64.89**	**69.59**	**47.47**	**60.65**
Larger-Scale LLM (≥10B)					
Deepseek-R1-32B	32B	70.22	68.56	48.48	62.42
GPT-4o-mini	-	60.00	59.28	51.52	56.93
QwQ-32B-Preview	32B	73.33	79.90	59.60	70.94
Deepseek-R1	671B	74.67	81.96	63.64	73.42

Table 2: Performance Comparison. LexPam (1.5B) aoutperforms all legal LLMs and small-scale reasoning models, and rivals larger-scale models. Bold indicates the best score within the <10B and the legal LLM groups.

LexPam rivals or exceeds larger-scale models. Despite being 1.5B in size, LexPam surpasses GPT-4o-mini (56.93%) and comes close to 32B-scale models like DeepSeek-R1-32B (62.42%) and QwQ-32B (70.94%). This confirms the value of our curriculum-based data selection and procedural-aware reward design in improving sample efficiency and generalization.

Legal LLMs struggle with numerical reasoning. Most legal-domain LLMs perform poorly, often below 15% accuracy. These models are typically fine-tuned on legal corpora for knowledge retention but lack explicit training for multi-step numerical inference. As shown in Figure 1, even basic arithmetic steps are frequently incorrect, suggesting that legal pretraining alone is insufficient for reasoning-intensive tasks.

5.3 Ablation Study

To assess the contribution of each design component in LexPam, we conduct an ablation study using the DeepSeek-R1-Distill-Qwen-1.5B model

[2]https://github.com/DUTIR-LegalIntelligence/Tailing
[3]https://github.com/CSHaitao/LexiLaw

Model	EC	WC	TC	**Avg**
GRPO-Base	52.00	61.86	38.38	50.75
GRPO-Law	59.11	66.49	46.46	57.36
$\mathcal{D}_1$-Only	56.00	57.22	37.37	50.20
$\mathcal{D}_2$-Only	57.33	65.46	44.44	55.75
LexPam	**64.89**	**69.59**	**47.47**	**60.65**

Table 3: Ablation results. LexPam outperforms all variants, demonstrating the effectiveness of curriculum staging and procedural rewards.

Model	EC	WC	TC	**Avg**
LexPam	64.89	69.59	47.47	60.65
All-GRPO	64.67	68.04	47.47	60.06
All-GRPO-$\mathcal{D}_1$	60.22	59.28	42.42	53.97
All-GRPO-$\mathcal{D}_2$	61.78	69.59	47.47	59.61
All-LexPam	**70.89**	**71.13**	**52.53**	**64.85**

Table 4: Results of task-merging experiments. All-LexPam achieves the best generalization across domains, demonstrating its scalability.

and the GRPO training framework. All models are evaluated on the full LexNum benchmark.

We compare the following variants:

- **GRPO-Base**: Trained on $\mathcal{D}_1 \cup \mathcal{D}_2$ using only the base reward from Eq.1 (correctness + format).
- **GRPO-Law**: Same data as GRPO-Base, but uses the full reward from Eq.3, including legal procedural alignment.
- $\mathcal{D}_1$**-Only** and $\mathcal{D}_2$**-Only**: Trained on $\mathcal{D}_1$ and $\mathcal{D}_2$ respectively , using the base reward.

From Table5.3, we observe that LexPam outperforms all variants, achieving 60.65% average accuracy. Compared to GRPO-Base (50.75%), LexPam gains +9.9 points, confirming the value of both curriculum structuring and legal-aware reward design. GRPO-Law (+6.6 over GRPO-Base) shows that procedural rewards alone provide a strong signal even without curriculum scheduling.

Interestingly, $\mathcal{D}_2$-Only outperforms $\mathcal{D}_1$-Only by a wide margin (+5.5), aligning with findings from prior work (Ye et al., 2025) that training on harder samples promotes more transferable reasoning skills. However, LexPam still surpasses $\mathcal{D}_2$-Only by +4.9, demonstrating that gradual learning from easier samples (via $\mathcal{D}_1$) further stabilizes and strengthens model performance.

These results validate LexPam's two-stage RL design: combining curriculum progression with procedural reward shaping yields consistently stronger legal reasoning.

5.4 Cross-Domain Generalization

To evaluate the generalization ability of our method beyond in-domain scenarios, we assess whether models trained on one legal domain can transfer effectively to others. Specifically, we conduct cross-domain experiments by training on one of the three LexNum subsets—Economic Compensation (EC), Work Injury Compensation (WC), or Traffic Accident Compensation (TC)—and testing on the remaining tasks.

As shown in Figure 3, LexPam consistently outperforms GRPO-Base and performs on par with or better than larger-scale models such as DeepSeek-R1-7B, despite having only 1.5B parameters. This pattern holds across all training–testing configurations (EC→TC, WC→TC, TC→EC, TC→WC), demonstrating that the procedural-aware training strategy of LexPam enables strong cross-task transfer.

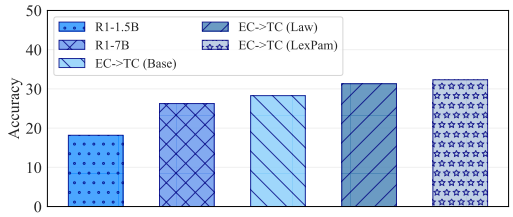
Notably, GRPO-Law also improves over GRPO-Base in most cases, further validating the effectiveness of the legal reward design. The ability of LexPam and GRPO-Law to maintain high accuracy across varying domains—despite differences in data distributions and legal formulations—highlights their robustness.

5.5 Generalization via Task Merging

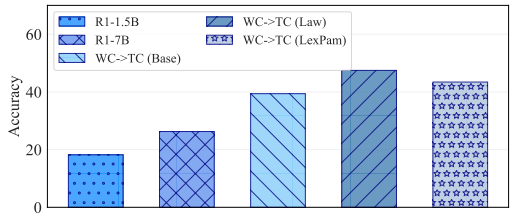
Although the three LexNum tasks differ in legal provisions and calculation logic, they all center on compensation reasoning. We investigate whether merging these tasks can yield synergistic effects and enhance model generalization. We evaluate the following multi-task training strategies:

- **All-GRPO**: GRPO with base reward, trained on the full merged dataset.
- **All-GRPO-**$\mathcal{D}_1$ **/** $\mathcal{D}_2$: Trained only on merged easy or hard samples, respectively.
- **All-LexPam**: LexPam applied to merged data—trained on $\mathcal{D}_1$ with base reward, then on $\mathcal{D}2$ with scenario-specific legal rewards ($rlaw$ determined by task type).

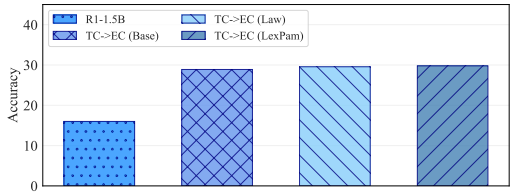
As shown in Table 4, All-GRPO already surpasses the original LexPam, suggesting that training on diverse but structurally related legal tasks



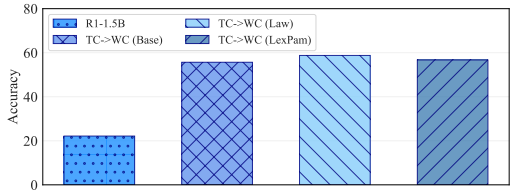
(a) Train on EC, test on TC.



(b) Train on WC, test on TC.



(c) Train TC, test on EC



(d) Train on TC, test on WC

Figure 3: Cross-domain Results. TC, EC, WC are short for traffic compensation, economic compensation, work injury compensation. R1-1.5B (7B) corresponds to DeepSeek-R1-Distill-Qwen-1.5B (7B). LexPam shows strong generalization, matching or surpassing larger models.

provides transferable inductive signals. All-LexPam achieves the best overall performance (64.85%), confirming that LexPam's two-stage structure scales effectively under task merging.

These results highlight the generalization potential of LexPam: its curriculum and procedural alignment mechanisms not only improve in-domain reasoning but also extend to heterogeneous legal settings through unified training.

5.6 Human Evaluation

To assess the procedural quality and user-facing suitability of model outputs, we conduct a human evaluation on responses generated for legal mathematical reasoning. We randomly sampled 40 queries from the traffic accident compensation dataset. For each query, we collected responses from four models: DISC-LawLLM, Tailing, DeepSeek-R1, and LexPam. The responses were anonymized and randomly ordered. Each query and its four associated responses were presented together to four independent annotators—Chinese law students with domain knowledge (distinct from those involved in §3.2).

Model	Completeness	Coverage	Conciseness	**AVG**
DISC-LawLLM	3.89	3.92	4.58	4.13
Tailing	3.19	2.94	**4.74**	3.62
Deepseek-R1	4.62	4.75	3.63	4.33
LexPam	**4.69**	**4.76**	4.66	**4.70**

Table 5: Human evaluation results.

Annotators rated each response on a 5-point Likert scale (1–5, higher is better) along three criteria:

- **Completeness**: Are all key steps—liability, insurance, and compensation—present?
- **Coverage**: Are the relevant legal elements from the question appropriately addressed?
- **Conciseness**: Is the response clear and easy to read without unnecessary verbosity?

Table 5 shows the average scores across all raters. LexPam achieves the highest overall rating (4.70), with strong performance across all dimensions. Compared to DeepSeek-R1, which tends to overexplain, LexPam provides legally grounded yet concise outputs. These results suggest that LexPam not only improves procedural fidelity but also enhances user-oriented response quality.

Additional examples and evaluation details are available in Appendix E.

6 Conclution

We present LexNum, the first benchmark for legal mathematical reasoning in Chinese, and introduce LexPam, a two-stage reinforcement learning framework that improves both computational accuracy and procedural compliance. LexPam leverages curriculum-based data selection and a novel legal-aware reward, enabling a lightweight 1.5B model to outperform a range of legal-domain and reasoning-focused LLMs. Extensive experiments demonstrate LexPam's effectiveness across in-domain and cross-domain settings, as well as in multi-task training scenarios. Human evaluation further confirms its strength in producing procedurally sound and user-readable outputs.

7 Limitations

In this paper, due to computational resource limitations, we primarily conduct experiments on a 1.5B-parameter reasoning model to explore how to enhance its legal mathematical reasoning capabilities. In future work, when more computational resources become available, we plan to test LexPam on larger-scale LLMs to investigate whether its effectiveness holds. Moreover, we have not yet annotated the legal mathematical reasoning process with Chain-of-Thought (CoT) in this paper, as producing high-quality CoT annotations for legal reasoning involves significant human effort. Due to the domain-specific nature of the task, such annotations require careful work by legal professionals. In the future, we aim to enrich the LexNum dataset by annotating the CoT reasoning processes that lead to the final answers, thereby further advancing research in the field of legal AI.

8 Ethical Considerations

The LexNum dataset we provide is intended to facilitate research on reasoning LLMs in the legal NLP domain. The data was collected from publicly accessible websites[4], and all cases have been anonymized. During the dataset construction process, we also engaged legal professionals to review the dataset to ensure its compliance with legal and ethical standards.

References

Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*.

Andong Chen, Feng Yao, Xinyan Zhao, Yating Zhang, Changlong Sun, Yun Liu, and Weixing Shen. 2023. Equals: A real-world dataset for legal question answering via reading chinese laws. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages 71–80.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Chenlong Deng, Kelong Mao, Yuyao Zhang, and Zhicheng Dou. 2024. Enabling discriminative reasoning in llms for legal judgment prediction. *arXiv preprint arXiv:2407.01964*.

[4]https://www.pkulaw.com

Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Songyang Zhang, Kai Chen, Zongwen Shen, and Jidong Ge. 2023. Lawbench: Benchmarking legal knowledge of large language models. *arXiv preprint arXiv:2309.16289*.

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

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2024. Token-budget-aware llm reasoning. *arXiv preprint arXiv:2412.18547*.

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.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

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

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Chenglin Li, Qianglong Chen, Liangyue Li, Caiyu Wang, Yicheng Li, Zulong Chen, and Yin Zhang. 2023. Mixed distillation helps smaller language model better reasoning. *arXiv preprint arXiv:2312.10730*.

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

Hongcheng Liu, Yusheng Liao, Yutong Meng, and Yuhao Wang. 2023. Xiezhi: Chinese law large language model. https://github.com/LiuHC0428/LAW_GPT.

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. 2025. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and

Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

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

Qwen Team. 2024. Qwq: Reflect deeply on the boundaries of the unknown.

Shiguang Wu, Zhongkun Liu, Zhen Zhang, Zheng Chen, Wentao Deng, Wenhao Zhang, Jiyuan Yang, Zhitao Yao, Yougang Lyu, Xin Xin, Shen Gao, Pengjie Ren, Zhaochun Ren, and Zhumin Chen. 2023. fuzi.mingcha. https://github.com/irlab-sdu/fuzi.mingcha.

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

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*.

Feng Yao, Chaojun Xiao, Xiaozhi Wang, Zhiyuan Liu, Lei Hou, Cunchao Tu, Juanzi Li, Yun Liu, Weixing Shen, and Maosong Sun. 2022. Leven: A large-scale chinese legal event detection dataset. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 183–201.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*.

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

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

Kepu Zhang, Haoyue Yang, Xu Tang, Weijie Yu, and Jun Xu. 2024a. Beyond guilt: Legal judgment prediction with trichotomous reasoning. *arXiv preprint arXiv:2412.14588*.

Kepu Zhang, Weijie Yu, Sunhao Dai, and Jun Xu. 2024b. Citalaw: Enhancing llm with citations in legal domain. *arXiv preprint arXiv:2412.14556*.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. Jec-qa: a legal-domain question answering dataset. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9701–9708.

Xunyu Zhu, Jian Li, Can Ma, and Weiping Wang. 2024. Improving mathematical reasoning capabilities of small language models via feedback-driven distillation. *arXiv preprint arXiv:2411.14698*.

A Task Setup

LexNum consists of three tasks: economic compensation, work injury compensation, and traffic accident compensation.

Economic Compensation: Based on the relevant provisions of the "Labor Contract Law of the People's Republic of China", this task involves determining eligibility for compensation, calculating the length of service and average monthly wage, and computing the economic compensation amount to ensure that employees receive lawful and reasonable compensation upon contract termination.

Work Injury Compensation: This task identifies the responsible party and, in accordance with the "Work Injury Insurance Regulations" and other relevant laws, calculates various compensation amounts to ensure that injured employees receive lawful and reasonable compensation.

Traffic Accident Compensation: This task determines liability, calculates reasonable damages based on relevant legal provisions, and determines the final compensation amount according to insurance coverage, ensuring that the injured party receives lawful and fair compensation.

Examples of these three tasks are shown in Figure 2. Economic compensation requires accurately determining the number of months for which compensation is due based on legal provisions, as well as the maximum amount that can be compensated. Traffic accident compensation also requires distinguishing the contents included in different types of compensation. For example, the question in Figure 2(b) asks about compensation related to work-related death benefits, so funeral allowances should not be considered. Work injury compensation requires identifying which costs in the problem are related to the specified compensation, while also understanding legal provisions. For example, the question in Figure 2 (c) asks about compensation related to compulsory insurance, which has a limit. Considering medical expenses and allowances, the maximum compensation is 10,000.

B Prompt Used in the Data Costruction

Figure 4 shows the specific prompt for us to construct the dataset using GPT-4o, which was written as suggested by the legal professional mentioned in § C.

C Specific Information of the Manual Annotator

We hired three legal annotators from the law program of a university in China. They are highly familiar with the legal provisions and calculation methods involved in the LexNum dataset provided in this paper. All three have undergone years of legal education and possess practical experience in the legal field. Among them, there are two females and one male, aged between 20 and 30. We explained to them that the reviewed dataset would be used for research on Chinese legal AI and provided reasonable compensation based on local standards. We gave them detailed instructions on which legal provisions and relevant regulations should be used to review and revise the questions and answers. We also informed them that the dataset should be refined following the procedures described in § 3.2.

D More Details of the Evaluation LLM

Table 6 are the website URLs and corresponding licenses of the evaluated models.

E Qualitative Analysis

The four students who evaluated the model responses each had no less than three years of legal education in Chinese law and were familiar with the relevant statutes and legal procedures related to compensation calculation in Chinese legal contexts. Among them, there were two females and two males, aged between 20 and 30. We explained to them that the evaluation was conducted to support research in Chinese legal AI and provided reasonable compensation based on local standards. We gave them a detailed explanation of each evaluation criterion, including what specific aspects to consider, and used several examples to ensure that their evaluation standards were aligned.

Figure 5 presents example responses from four models on the traffic compensation dataset. We can observe that the legal LLM, having undergone SFT on legal QA tasks, is able to generate relatively concise responses to some extent. However, its calculation results are often incorrect, as the injection of extensive legal knowledge tends to impair its general mathematical reasoning ability. DeepSeek-R1, on the other hand, performs well on mathematical computation tasks, but its reasoning process is often overly verbose, making it less suitable for user-friendly reading. LexPam, by contrast, is able to produce relatively concise responses while also demonstrating strong mathematical computation capabilities.

Extract the Q-A pairs for calculating the compensation amount from the document below in accordance with the relevant laws and regulations.

Answer in the format {"question": , "answer":}. The "question" should include the case details and the requirement for calculating the compensation amount, and it must contain all the necessary information for the calculation. Do not directly include the compensation amount. The "answer" should be the calculated compensation amount.

Relevant laws and regulations: **{law articles}**

Document: {document}

Output (Note the comprehensiveness of the question):

(a) Prompt for Data Extraction

Check the "question" in the following original data to determine whether it contains all the information required to calculate the "answer." If the information in the "question" is incomplete and prevents the calculation of the "answer," output "no." If the information is complete, output "yes."

Requirements:

1.The "question" must contain complete calculation information, allowing the "answer" to be directly calculated based on the "question."

2.Note that the "question" should only ask about the amount of compensation and not include any other demands. If it does, output "no."

Original data: {original_data}

Is the question complete (output yes/no):

(b) Prompt for Data Inspection

Figure 4: The prompts of the data costruction. The red parts are replaced according to the specific situation. In (a), law articles is replaced with different "Relevant laws and regulations" depending on the dataset, and document refers to the specific legal document. In (b), original_data is the extracted JSON file, which contains the keywords "question" and "answer."

Type	LLM	URL	Licence
Legal Domain	fuzi.mingcha	https://github.com/irlab-sdu/fuzi.mingcha	Apache-2.0 license
	DISC-LawLLM	https://github.com/FudanDISC/DISC-LawLLM	Apache-2.0 license
	LawGPT_zh	https://github.com/LiuHC0428/LAW-GPT	
	Hanfei	https://github.com/siat-nlp/HanFei	Apache-2.0 license
	Tailing	https://github.com/DUTIR-LegalIntelligence/Tailing	
	LexiLaw	https://github.com/CSHaitao/LexiLaw	MIT license
	zhihai	https://github.com/zhihaiLLM/wisdomInterrogatory	Apache-2.0 license
Open domain	QwQ-32B-Preview	https://huggingface.co/Qwen/QwQ-32B-Preview	Apache-2.0 license
	DeepSeek-R1	https://github.com/deepseek-ai/DeepSeek-R1	MIT license

Table 6: The LLM source URLs and licenses used by CitaLaw. The parts where the license is listed as empty indicate that the author has not provided a License.

Based on the provided information, … bears **primary responsibility** for the accident, and … bears **secondary responsibility**. … has purchased … liability **insurance**. … **As the insured party**, … should cover this loss. …, they should bear 70% of the liability. **Calculation**: 5275.6 × 70% = 3692.92. … should compensate … a total of **3,692.92 yuan**.

(a) LexPam

…Based on the information provided in the question, ….
According to the information given in the question, … incurred an operational loss of 5275.6 yuan. Based on the above analysis, …, as the primary responsible party, should bear 3621.4 yuan of that amount.
In conclusion, the specific compensation amount that … should pay … **3621.4 yuan**.

(b) DISC-LawLLM

… After the accident, … incurred a loss of 5275.6 yuan..
The specific compensation amount that … should pay … is calculated as:
5275.6 yuan × 70% + 5275.6 yuan × 30% = 3702.92 yuan + 1582.78 yuan = **5285.7 yuan**.

(c) Tailing

Hmm, I need to take a closer look at this question and work it out step by step. First, the question states that…
First, I need to recall the relevant regulations on traffic accident compensation.
Generally speaking,…. However, for example, ……
But it's also possible that the correct approach is: … In that case,
…Therefore, the compensation amount that … should pay is calculated as:
5275.6 × 70% = 3692.92…

(d) Deepseek-R1

Figure 5: Responses from four models on the traffic compensation dataset.