LexChain: Modeling Legal Reasoning Chains for Chinese Tort Case Analysis

Huiyuan Xie[1*], Chenyang Li[2,3*], Huining Zhu[4], Chubin Zhang[2,3], Yuxiao Ye[1†], Zhenghao Liu[5], Zhiyuan Liu[1†]

[1]Tsinghua University
[2]Queen Mary University of London
[3]Beijing University of Posts and Telecommunications
[4]East China University of Political Science and Law
[5]Northeastern University
{xieh, yeyuxiao, liuzy}@tsinghua.edu.cn, chenyang.li@se22.qmul.ac.uk

Abstract

Legal reasoning is a fundamental component of legal analysis and decision-making. Existing computational approaches to legal reasoning predominantly rely on generic reasoning frameworks such as syllogism, which do not comprehensively examine the nuanced process of legal reasoning. Moreover, current research has largely focused on criminal cases, with insufficient modeling for civil cases. In this work, we present a novel framework to explicitly model legal reasoning in the analysis of Chinese tort-related civil cases. We first operationalize the legal reasoning process in tort analysis into the three-module LexChain framework, with each module consisting of multiple finer-grained sub-steps. Informed by the LexChain framework, we introduce the task of tort legal reasoning and construct an evaluation benchmark to systematically assess the critical steps within analytical reasoning chains for tort analysis. Leveraging this benchmark, we evaluate existing large language models for their legal reasoning ability in civil tort contexts. Our results indicate that current models still fall short in accurately handling crucial elements of tort legal reasoning. Furthermore, we introduce several baseline approaches that explicitly incorporate LexChain-style reasoning through prompting or post-training. The proposed baselines achieve significant improvements in tort-related legal reasoning and generalize well to related legal analysis tasks, demonstrating the value of explicitly modeling legal reasoning chains to enhance the reasoning capabilities of language models.

Data and code — https://github.com/thunlp/LexChain

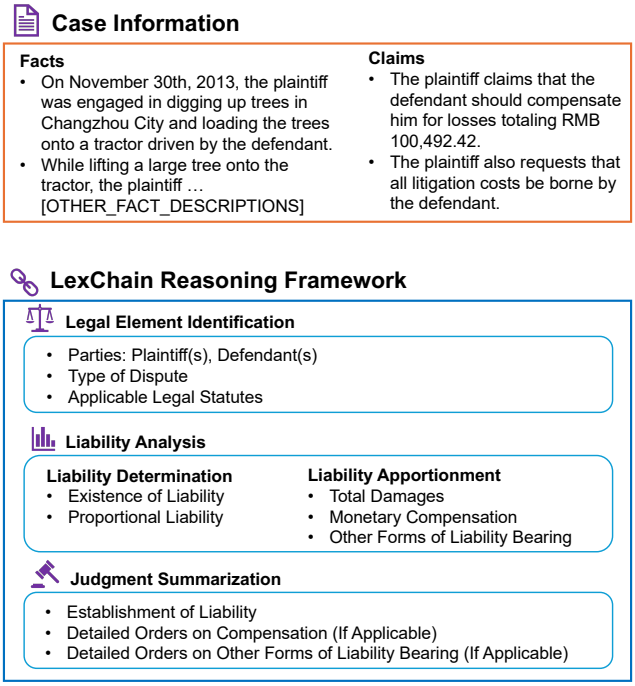

Figure 1: Illustration of the case information (English translation) in a Chinese tort case and the three-module LexChain reasoning framework.

Introduction

The rapid development of large language models (LLMs) has led to widespread adoption of these tools across diverse domains. Recent models, such as Gemini-2.5-Pro (Comanici et al. 2025) and Qwen-3 (Yang et al. 2025), now position reasoning as a default and essential component of their architecture. While notable progress has been made in advancing the reasoning abilities of LLMs in areas such as mathematics and coding (Wei et al. 2022; Huang et al. 2023; El-Kishky et al. 2025), LLMs' reasoning capabilities in the legal domain, where nuanced outputs are dependent on structured and domain-specific legal reasoning, remain underexplored.

Legal reasoning is a foundational element of legal analysis and decision-making. Legal scholars argue that there is, descriptively, something unique that can be characterized as "thinking like a lawyer" (Samuelson 1997; Schauer 2009). This encompasses distinctive forms of reasoning, such as the systematic application of codified legal rules, the decomposition of legal questions into granular elements for structured judgment, and ensuring that similar cases are treated uniformly. Better modeling of legal reasoning not only im-

*These authors contributed equally.
†Corresponding author.

proves the accuracy and efficiency of legal analysis, but also enhances the interpretability of results by providing explicit reasoning chains.

However, existing work on computational legal reasoning has primarily focused on generic legal reasoning frameworks, such as syllogism (Deng et al. 2023; Jiang and Yang 2023) and IRAC (Yu, Quartey, and Schilder 2023; Kuppa, Rasumov-Rahe, and Voses 2023; Servantez et al. 2024), often simulating reasoning through simple prompting strategies. In addition, research has concentrated predominantly on criminal law, with insufficient attention paid to civil cases, especially tort cases, which are highly relevant to everyday life. Tort law plays a fundamental role in modern legal systems, in both protecting individual rights and compensating plaintiffs for harms suffered.

In this work, we explicitly model legal reasoning chains for tort-related civil cases within the context of Chinese civil law. Drawing on established legal theory (Smith 1911; Moore 1999), we operationalize the analytical process of tort adjudication into the LexChain reasoning framework—a multi-module reasoning chain in which each module comprises a set of key steps or elements essential to tort legal reasoning (see Figure 1 for an illustration).

We introduce a novel task of tort legal reasoning and construct a benchmark designed to assess whether existing models can correctly identify the key steps and elements required for robust legal reasoning in tort disputes. We evaluate several leading LLMs on this benchmark, observing that accurately modeling the tort reasoning chain remains a challenging task for current models.

Leveraging our LexChain reasoning framework, we further implement reasoning-enhanced baselines using approaches such as legal prompting, supervised fine-tuning (SFT) and direct preference optimization (DPO). Results demonstrate that integrating explicit legal reasoning chains through these techniques consistently improves model performance on identifying key elements for tort case analysis, underscoring the value of explicit legal reasoning modeling in enhancing LLMs' legal reasoning abilities.

Finally, we evaluate our fine-tuned baselines on related legal AI tasks, including Legal Named-Entity Recognition and Criminal Damages Calculation (Fei et al. 2024). Our reasoning-enhanced models achieve comparable or superior performance to zero-shot counterparts on these tasks, demonstrating the generalizability of reasoning-enhanced approaches to other legal AI tasks.

The contributions of this work are summarized as follows:

- We operationalize legal reasoning in the domain of civil tort law by proposing a legally-informed, fine-grained reasoning framework that reflects the structured analytical processes employed in real-world tort litigation.
- We introduce the novel task of tort legal reasoning and construct a dedicated benchmark to evaluate models' performance on this task. To the best of our knowledge, the constructed benchmark is the first evaluation dataset that focuses on the assessment of legal reasoning abilities in the context of Chinese civil tort law. In addition, we curate a reasoning-enhanced training dataset to enable effective learning of legal reasoning patterns.
- Our empirical results demonstrate that current LLMs exhibit notable limitations in tort-related legal reasoning, particularly in labor dispute cases. Furthermore, we find that generic syllogistic reasoning frameworks do not yield meaningful improvements in this domain, underscoring the need to move beyond generic approaches and adopt domain-specific reasoning paradigms tailored to the complexity of civil tort law. We further show that incorporating LexChain-style reasoning through prompting, SFT and DPO leads to substantial performance gains on the tort reasoning task. These improvements also generalize effectively to other related tasks, underscoring the value of explicitly modeling structured legal reasoning in enhancing LLM performance on legal analysis tasks.

Related Work

Legal Reasoning and Tort Cases

Legal reasoning refers to the analytical process by which legal practitioners integrate legal norms with case facts to reach a legal conclusion (Samuelson 1997; Schauer 2009). Legal reasoning holds a central role in judicial practice, serving as the foundation for legal analysis and as a mechanism to ensure the consistency of judicial decisions. Distinct from general forms of logical reasoning, legal reasoning is shaped by the constraints of legal systems, precedential rules and various institutional factors (Scharffs 2004). The diversity and complexity of real-world cases also make the practical application of legal reasoning highly challenging.

Tort liability is a form of civil liability that aims to remedy infringed civil rights and safeguard the legitimate interests of civil subjects (Smith 1911). In judicial practice, tort disputes are of particular significance, as they encompass a broad spectrum of harms, such as personal injury, property damage and online harassment, permeating many aspects of everyday life (Latin 1985; Wagner 2019). In comparison to other areas of law, tort liability cases exhibit distinctive patterns in their reasoning pathways. A foundational step in tort analysis is the determination of liability, which involves a systematic examination of the constituent elements: conduct, harm, causation and fault (Smith 1911; Catala and Weir 1965; Moore 1999). Once liability is established, legal practitioners must further consider the nature of the tortious act and the severity of the harm to determine the appropriate forms of liability, with monetary compensation being the most prevalent remedy (Van Wijck and Winters 2001). In certain cases, additional complexities such as compensation for emotional distress and the application of rules for loss offset may arise (Pearson 1979; Kontorovich 2001), demanding complex legal reasoning to successfully adjudicate.

Computational Modeling of Legal Reasoning

Computational modeling of legal reasoning aims to simulate the multi-step inferential processes underpinning legal decision-making (Deng et al. 2023; Fernandes et al. 2025). In recent years, structured prompting techniques (Yu, Quartey, and Schilder 2022; Jiang and Yang 2023) have garnered increasing prominence. A significant line of

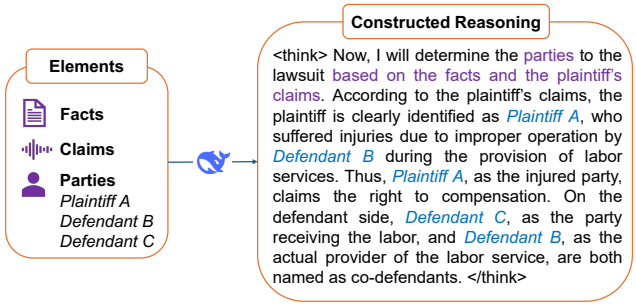

Figure 2: An example of the constructed reasoning text for identifying plaintiff(s) and defendant(s) based on legal elements extracted from the original judicial document.

work leverages syllogism-based prompting (Jiang and Yang 2023), which encourages models to organize their reasoning in a triadic structure: major premise (legal norm), minor premise (case facts) and conclusion (judgment). Other studies have incorporated schema-based reasoning frameworks such as IRAC (Yu, Quartey, and Schilder 2023; Kuppa, Rasumov-Rahe, and Voses 2023), encouraging generated outputs to resemble authentic legal argumentation. Another logic-based approach (Servantez et al. 2024) focuses on atomistic parsing of legal rules, decomposing complex provisions into atomic Boolean conditions, allowing for fine-grained alignment with statutory interpretation practices.

In addition to prompting methods, supervised fine-tuning has been used to integrate legal reasoning patterns into models. For instance, syllogism-based fine-tuning (Deng et al. 2023) encodes syllogistic legal deduction into models, reinforcing deductive reasoning during inference. Similarly, schema-based fine-tuning (Fernandes et al. 2025) trains models to transform case analyses into structured IRAC representations, effectively internalizing schema-based scaffolds within models' reasoning process.

LexChain: Benchmarking Legal Reasoning

To model legal reasoning in tort law, we first conceptualize tort-related decision-making as a structured, multi-step reasoning process, which we further operationalize as a legal reasoning chain, referred to here as LexChain. Stemming from this conceptual base, we propose a novel task of tort legal reasoning and construct an evaluation dataset (LexChain$_{eval}$). We further construct a reasoning-enhanced training dataset (LexChain$_{train}$) that incorporates explicit LexChain-style reasoning tracing, which can be used to fine-tune models for improved legal reasoning.

Operationalization of Legal Reasoning Chains

Tort analysis exhibits distinctive reasoning structures, particularly centered around the establishment and apportionment of liability. The analytical process typically begins with determining whether liability exists, and proceeds to assess the severity and consequences of the tortious act in order to determine the extent of legal responsibility and appropriate forms of compensation.

Drawing on established legal frameworks (Smith 1911; Catala and Weir 1965; Van Wijck and Winters 2001) and consultations with practicing judges and lawyers, we formalize tort analysis into a structured, three-module reasoning chain. Each module is further decomposed into a list of specific legal sub-tasks, operationalizing the abstract process of legal reasoning into concrete, verifiable components suitable for computational modeling and evaluation.

The LexChain reasoning framework consists of three modules: (1) a legal element identification module, which identifies relevant foundational information necessary for legal analysis, (2) a liability analysis module, the core module, which centers on determining and apportioning liability, and (3) a judgment summarization module, which synthesizes the outcomes of the preceding modules into a final legal decision. An illustration of the LexChain reasoning framework is presented in Figure 1.

Legal element identification module. This preparatory module focuses on identifying essential elements required for subsequent legal reasoning, including:

- **Parties**. Identification of the plaintiff(s) and defendant(s), clarifying the legal actors involved in the dispute.
- **Dispute type**. Specification of the tort category (e.g., traffic accident, defamation), which signals the nature of the civil interest allegedly infringed. This classification is critical because legal provisions in tort law are often organized according to the type of protected civil interest.
- **Applicable legal provisions**. Retrieval of the relevant statutory sources, which serve as the legal basis for the reasoning to follow.

Liability analysis module. As the central component of the LexChain, this module determines whether the defendant(s) should bear legal liability, and if so, how such liability should be apportioned. The reasoning process is divided into two sub-tasks for each defendant:

- **Liability determination**. This sub-task assesses whether the defendant has committed a legally recognized tort and thus should be held civilly liable. Following legal doctrine, we structure this analysis around four foundational elements of tort liability: conduct, harm, causation and fault (Smith 1911). The presence and interaction of these elements are evaluated to determine both the existence and extent (i.e., proportional liability) of a defendant's responsibility.
- **Liability apportionment**. Once liability is established, this sub-task determines the specific forms of legal responsibility to be imposed, in accordance with civil law. Common forms include monetary compensation, the restitution of damages or cessation of infringement (Wan et al. 2022). Given that monetary compensation is the predominant remedy in tort law (Van Wijck and Winters 2001), we explicitly add a step of damages calculation where the total loss suffered by the victim is estimated. The total of estimated damages is multiplied by the proportion of liability attributed to each defendant to obtain a precise amount of compensation. Additional remedies, such as restitution or cessation of infringement, are also considered where legally applicable.

Judgment summarization module. This final module summarizes the outputs of the previous modules to generate a complete and coherent legal decision. It synthesizes whether a tort has been legally established, the nature and extent of the liability, the remedies to be ordered, and the exact compensation to be awarded. This module mirrors the structure of real-world judicial opinions and serves to evaluate whether models can deliver legally sound and procedurally complete conclusions.

Evaluating Tort Legal Reasoning in LLMs

Facilitated by the modular and stepwise structure formalized in the LexChain reasoning framework, we introduce a novel task of tort legal reasoning (TLR) to systematically evaluate LLMs' capacity to perform structured legal reasoning in the context of tort law. We construct an evaluation benchmark, LexChain$_{eval}$, to assess models' ability to identify and reason through the critical components involved in tort case analysis. In addition, we propose an LLM-assisted scoring framework to quantify model performance at each reasoning step, enabling interpretable and modular evaluation.

The Task of Tort Legal Reasoning Tort legal reasoning (TLR) requires a model to correctly identify or predict a structured sequence of key legal elements that together form a coherent reasoning chain for tort adjudication. The LexChain framework provides a principled decomposition of the reasoning process into discrete modules, each associated with legally meaningful sub-tasks. We construct the TLR task by closely aligning the task with this formalization, allowing models to be evaluated across the entire reasoning trajectory, from case setup to judgment formulation.

More specifically, given the facts and claims of a tort case, the TLR task evaluates whether a model can correctly infer the following categories of legal reasoning elements:

- Foundational legal entities and references: including identification of the plaintiff(s), defendant(s), classification of the dispute (i.e., type of tort) and relevant legal statutes applicable to the case.
- Liability-related reasoning elements: including the structured reasoning pathway based on the four essential constituent factors of liability (i.e., conduct, harm, causation and fault), the determination of whether liability is established and, if so, the specific proportion of liability attributed to each defendant, the appropriate compensation amount and other forms of legal responsibility.
- Judgment summary: synthesizing the conclusions from prior reasoning steps into a structured summary.

By decomposing the evaluation of legal reasoning into verifiable sub-components, the TLR task enables a fine-grained evaluation of LLMs' capacity to simulate human-like legal reasoning in tort contexts.

Constructing Evaluation Dataset We construct the evaluation dataset LexChain$_{eval}$ based on real-world judicial documents. Authentic judicial decisions offer a realistic, diverse source of tort-related legal scenarios. In particular, cases that proceed to litigation typically involve greater factual and legal complexity, as they often center on issues of conceptual ambiguity or interpretive dispute (Schauer 2009). These characteristics make litigated cases especially well-suited for modeling and evaluating tort legal reasoning.

We start by sampling 1,000 real-world tort case judgments from China Judgments Online (2013). To ensure that the evaluation dataset centers on substantive legal reasoning, we exclude cases involving procedural matters, such as voluntary withdrawals, jurisdictional transfers or dismissals on procedural grounds during the sampling process. The resulting dataset comprehensively covers major types of tort disputes, encompassing 45 distinct tort subcategories.

We design an LLM-assisted extraction pipeline to systematically retrieve key legal information from court decision texts. This pipeline extracts essential elements required for tort reasoning, including: plaintiff(s), defendant(s), type of dispute, applicable legal statutes, claims of the plaintiff(s), case facts, total damages, liability determinations and judgment outcomes. For elements commonly expressed in structured formats in court decisions, such as parties, claims, citations of law and judgments, we employ rule-based extraction techniques, including keyword matching and regular expression-based heuristics. For elements requiring holistic understanding and document-wise summarization, such as fact descriptions, liability reasoning and damages calculation, we utilize DeepSeek-V3 (Liu et al. 2024) to synthesize relevant information from the court decision texts.

To ensure the quality of data present in LexChain$_{eval}$, we conduct a rigorous manual validation process on the automatically extracted case information. All extracted data are reviewed and refined by a trained legal annotator, under the supervision of a senior researcher with extensive experience in legal AI and annotation workflows. The extraction and manual validation processes ensure a high-quality benchmark for evaluating tort-related legal reasoning.

LLM-Based Scoring To systematically evaluate model outputs on the tort legal reasoning task, we adopt an "LLM-as-a-Judge" evaluation paradigm, widely used in recent literature due to its demonstrated effectiveness and scalability for automated evaluation (Zheng et al. 2023; Gu et al. 2024). We design a structured and automated scoring process using GPT-4o (Hurst et al. 2024) as the evaluation model. Importantly, instead of relying on subjective, holistic assessments, we decompose the evaluation into a series of discrete, well-defined sub-dimensions and prompt GPT-4o to make rule-based, factual judgments for each sub-dimension. This decomposition mitigates the risk of evaluation bias and enhances reliability of LLM-based evaluation, aligning with existing practices in recent studies (Liu et al. 2023; Wang et al. 2024; Gu et al. 2024).

The evaluation covers seven distinct scoring dimensions, corresponding directly to the core reasoning elements examined in the LexChain$_{eval}$ benchmark. For each test instance, GPT-4o is instructed to evaluate the model's response against gold-standard answers, i.e., reference case elements provided in the LexChain$_{eval}$ dataset. The scoring schema for each reasoning element is defined as follows:

1. Plaintiff identification: Assesses whether the model accurately identifies all plaintiffs specified in the case. A score

of 0 denotes incorrect identification; 1 indicates partially correct identification (e.g., only some plaintiffs identified in a multi-party case); 2 denotes a correct identification.

2. Defendant identification: Follows the same scoring rubric as plaintiff identification, evaluating the model's accuracy and completeness in recognizing all defendants.
3. Type of dispute: Evaluates whether the model correctly classifies the nature of the tort dispute (e.g., "motor vehicle accident liability"). Full alignment with the reference answer is required for credit.
4. Legal statutes: Assesses the accuracy of applicable legal statutes predicted by the model (e.g., specific articles from the Civil Code or judicial interpretations). Full score of 2 is awarded for correctly identifying all relevant provisions; partial score of 1 is given for incomplete citations; and a score of 0 indicates no valid references.
5. Total damages: Evaluates the correctness of the model's estimate of total damages, as compared to the reference computation, with allowances for reasonable numerical deviations or unit differences.
6. Liability apportionment: Measures the model's ability to correctly identify the liable parties, the proportion of liability, forms of civil responsibility (e.g., compensation, restitution) and the associated legal justifications.
7. Judgment summary: Compares the model-generated judgment summary to the reference list of case outcomes, as provided in the official ruling. Evaluation is based on an overlap of factual items (e.g., compensation amount, remedies awarded) and quantified using the F1 score.

To assess the robustness and reliability of the LLM-based evaluation, we conduct quality validation where we randomly sample 300 instances from GPT-4o's ratings and ask two independent annotators to review the ratings. For each of GPT-4o's ratings, annotators independently assign a binary judgment: a score of 1 if they consider GPT-4o's evaluation correct, and 0 otherwise. In cases where the two annotators disagree, a senior researcher with expertise in legal AI practices is consulted, and the annotators engage in discussion to reach a consensus. The final agreed-upon labels (0 or 1) are used as the reference for calculating the accuracy of GPT-4o's ratings. All initial disagreements are also recorded to calculate the inter-annotator agreement score to ensure a fair assessment. Based on this validation criterion, GPT-4o achieves an overall rating accuracy of 0.949 on the sampled set. Inter-annotator agreement, measured using Cohen's Kappa (Cohen 1960), yields a score of 0.619, indicating relatively strong consistency between annotators[1].

Constructing LexChain-Style Training Data

To prepare training data enriched with legal reasoning information, we sample approximately 10,600 court decision documents for tort cases. For each case, we first extract key legal elements following the same information extraction practice described in LexChain$_{eval}$ construction. We then leverage DeepSeek-V3 (Liu et al. 2024) to assist in the automatic construction of legal reasoning chains, building upon the extracted legal elements. For each step in the reasoning process, we determine the specific types of information required to infer towards the current reasoning objective and construct corresponding reasoning formulas. For example, the identification of litigation parties (i.e., plaintiff(s) and defendant(s)) requires consideration of both the case facts and the claims of the plaintiff(s), while the calculation of compensation details depends on the previously determined liability ratio and the victim's actual losses. An illustrative example is presented in Figure 2. Each constructed reasoning chain is enclosed in a <think> and </think> tag pair and prefixed to the final answer consisting of the extracted elements, resulting in the LexChain$_{train}$ training data.

We conduct human validation to assess the quality of the reasoning chains automatically generated by DeepSeek-V3. Two annotators independently evaluate 300 samples of LLM-generated reasoning chains based on two assessment criteria: consistency and coherence. Consistency measures the alignment of the generated reasoning text with the original legal elements from which it is constructed, while coherence evaluates whether the output forms a logically sound and well-structured reasoning narrative. The training data constructed by DeepSeek-V3 obtains a consistency rate of 0.973 and a coherence rate of 0.993. The inter-annotator agreement between the two annotators, measured as the Cohen's Kappa score (Cohen 1960), is 0.866, indicating a strong agreement between annotators.

Experiments and Results

Baselines

We evaluate current LLMs on LexChain$_{eval}$, including GPT-4o (Hurst et al. 2024), o3-mini (OpenAI 2025), Claude-Sonnet-4 (Anthropic 2025), DeepSeek-V3 (Liu et al. 2024), DeepSeek-R1 (Guo et al. 2025), Qwen-3-8B (Yang et al. 2025), InternLM-3-8B (InternLM 2025), and Llama-3.1-8B (Grattafiori et al. 2024), ensuring broad coverage of model architecture families. We experiment with two inference-only settings for all models, zero-shot and legal prompting, and three training-based settings for open-source models[2]: SFT with syllogism data in Deng et al. (2023), SFT with LexChain-style reasoning data and DPO with LexChain-style preference data.

Zero-shot. In the zero-shot setting, models are prompted using task descriptions only, without any examples or additional instructions.

Legal prompting (denoted as **Prompt**$_{LC}$). This prompting approach incorporates structured LexChain-style reasoning chains, guiding models to produce responses that reflect multi-step legal reasoning. This approach is applicable across LLMs without the need for model training.

SFT$_{Syll}$. This setting involves supervised fine-tuning using syllogism-style legal reasoning data provided in Deng

[1] According to Landis and Koch (1977), a Cohen's Kappa score between 0.61 and 0.80 signifies "substantial agreement".

[2] Although DeepSeek-R1 and DeepSeek-V3 are open-source models, we experimented via API calls due to computational resource considerations, and did not perform fine-tuning for them.

Model	Overall	Plaintiff	Defendant	Dispute	Statute	Liability	Damages	Judgment
GPT-4o	41.17	97.20	87.05	9.90	10.35	18.10	20.05	18.67
w/ Prompt$_{LC}$	48.83	97.40	89.15	17.60	28.70	30.30	25.75	25.75
o3-mini	40.22	97.00	86.70	6.40	11.65	18.45	17.55	13.57
w/ Prompt$_{LC}$	50.41	97.80	89.10	18.70	33.40	32.80	26.55	26.89
Claude-Sonnet-4	44.24	97.00	89.30	31.50	11.90	20.15	19.55	23.59
w/ Prompt$_{LC}$	52.90	**98.65**	91.05	38.40	30.90	34.20	28.20	30.36
DeepSeek-V3	46.91	97.80	87.35	35.20	20.85	22.45	22.60	25.60
w/ Prompt$_{LC}$	54.71	97.70	89.40	45.50	35.45	35.15	29.35	36.86
DeepSeek-R1	45.36	97.50	89.35	28.70	12.45	23.30	21.25	27.94
w/ Prompt$_{LC}$	**60.84**	98.10	**92.50**	58.40	**44.30**	**41.70**	**37.80**	**42.92**
Qwen-3-8B	48.79	97.50	88.60	29.50	30.70	26.35	22.35	24.92
w/ Prompt$_{LC}$	53.30	97.40	89.15	38.00	39.85	32.45	27.60	28.71
w/ SFT$_{Syll}$	39.65	96.40	86.70	13.90	13.05	16.00	12.55	12.47
w/ SFT$_{LC}$	55.36	96.85	87.85	**59.00**	43.70	29.05	26.80	36.88
w/ DPO$_{LC}$	51.17	96.20	88.85	36.00	41.95	25.55	21.00	30.95
InternLM-3-8B	41.61	96.10	86.10	20.00	15.70	17.70	16.15	15.80
w/ Prompt$_{LC}$	48.58	96.85	86.20	26.40	32.95	29.65	19.55	26.10
w/ SFT$_{Syll}$	40.32	95.60	82.35	14.60	20.05	18.75	11.80	12.13
w/ SFT$_{LC}$	44.89	92.85	74.30	49.90	38.95	17.55	11.80	17.91
w/ DPO$_{LC}$	42.67	92.80	76.95	25.10	36.05	20.05	9.40	16.43
Llama-3.1-8B	37.16	97.05	85.20	2.80	6.75	13.20	13.50	11.71
w/ Prompt$_{LC}$	40.35	96.75	84.05	5.90	19.10	18.40	14.35	13.04
w/ SFT$_{Syll}$	33.61	93.70	79.75	1.40	4.70	9.65	12.45	1.48
w/ SFT$_{LC}$	51.35	97.10	86.75	48.00	39.60	24.15	23.50	25.95
w/ DPO$_{LC}$	44.57	96.55	85.20	15.70	33.30	18.85	15.75	19.84

Table 1: Evaluation results on the LexChain$_{eval}$ benchmark. Entries labeled with the model name alone (e.g., GPT-4o) represent the zero-shot inference setting. The variant "w/ Prompt$_{LC}$" denotes models prompted using LexChain-style legal reasoning prompts. Variants "w/ SFT$_{Syll}$" and "w/ SFT$_{LC}$" denote models fine-tuned on syllogism-style and LexChain-style SFT data, respectively. The variant "w/ DPO$_{LC}$" refers to models optimized with LexChain-style DPO data.

et al. (2023). In this approach, training instances are structured into the syllogistic form: a major premise representing applicable legal rules, a minor premise representing case facts, and a conclusion stating the legal outcome.

SFT$_{LC}$. Using the curated LexChain$_{train}$ data, we perform supervised fine-tuning on three open-source models, including Qwen-3-8B, InternLM-3-8B and Llama-3.1-8B.

DPO$_{LC}$. In the DPO setting, we use the LexChain$_{train}$ data as the *chosen* samples. We further generate corresponding *rejected* samples by collecting answers generated by zero-shot versions of three models (Qwen-3-8B, InternLM-3-8B and Llama-3.1-8B) on LexChain$_{train}$. The resulting (*chosen*, *rejected*) sample pairs are used for DPO training.

Experiment Setup

The experiment setup for the baseline approaches evaluated in this work is described below.

Model Inference Settings. For all inferences involving closed-source models, the temperature parameter is set to 0.7. For open-source models, inference settings are determined according to either the official suggestions from model developers or the defaults specified in each model's configuration file. For example, for the Qwen-3-8B variants, the following parameters are used: temperature is 0.6, top_k is set to 20 and top_p is set to 0.95.

For non-reasoning models, the maximum token size is set to 4096 tokens. For reasoning-enabled models (such as DeepSeek-R1 and o3-mini), the maximum token size is increased to 8000 tokens to accommodate the extended outputs required for generating reasoning traces.

SFT Experiments. Models are fine-tuned with a batch size of 16 and a learning rate of 1e-4 with AdamW optimizer (Loshchilov and Hutter 2017). Low-Rank Adaptation (Hu et al. 2022, LoRA) is used for parameter-efficient adaptation. We use 10% of the data held out for validation; the optimal checkpoint is selected based on validation set performance. The three models, Qwen-3-8B, InternLM-3-8B and Llama-3.1-8B, are trained for 120, 160 and 80 steps, respectively. All SFT experiments are carried out using LlamaFactory (Zheng et al. 2024) on 8 NVIDIA A800 GPUs.

DPO Experiments. The setting for the DPO experiments is similar to that of SFT. The effective batch size for DPO training is set to 8. For each model, the final DPO checkpoint is chosen based on the training step comparable to the optimal SFT checkpoint to enable a reasonable comparison.

Evaluation Results on LexChain$_{eval}$

The evaluation results for a broad range of state-of-the-art LLMs on LexChain$_{eval}$ are reported in Table 1. In the zero-shot setting, most models exhibit limited capability in reasoning over the multiple dimensions of tort legal analysis. The best-performing model without any task-specific

Model	Legal NER	Crim Damage
Qwen-3-8B	7.87	**91.40**
w/ SFT$_{LC}$	19.36	84.40
w/ DPO$_{LC}$	23.38	80.40
InternLM-3-8B	13.67	64.80
w/ SFT$_{LC}$	45.58	69.20
w/ DPO$_{LC}$	14.14	68.00
Llama-3.1-8B	61.88	38.80
w/ SFT$_{LC}$	**63.90**	48.60
w/ DPO$_{LC}$	60.20	48.60

Table 2: Evaluation results of baseline models on the Legal Named-Entity Recognition (Legal NER) and the Criminal Damages Calculation (Crim Damage) tasks.

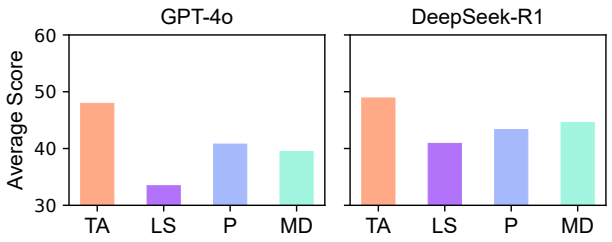

Figure 3: Evaluation results of GPT-4o and DeepSeek-R1 on the four most common types of tort liability, including liability for motor vehicle traffic accident (TA), liability for injury to providers of labor services (LS), liability for product (P) and liability for medical damage (MD).

prompt or fine-tuning is Qwen-3-8B, achieving an overall score of 48.79. Notably, as one of the most widely used LLMs, GPT-4o achieves only 41.17 overall, underscoring the challenge posed by the tort legal reasoning task.

We observe that the LexChain-style legal prompting approach (Prompt$_{LC}$) leads to consistent performance improvements across all models. The largest gain is observed for DeepSeek-R1, achieving a 34.1% increase over its zero-shot baseline and yielding the highest overall performance among all tested approaches. These results highlight the effectiveness of explicit, structured legal prompting in eliciting enhanced legal reasoning abilities from current models.

We further examine the impact of fine-tuning with reasoning-enhanced data, comparing models trained on syllogistic reasoning data (SFT$_{Syll}$), LexChain-style reasoning data (SFT$_{LC}$) and LexChain-style preference data (DPO$_{LC}$). We find that SFT$_{Syll}$ yields reduced performance compared to zero-shot baselines for all models. This suggests that generic syllogistic reasoning is insufficient to capture the domain-specific structure and nuances in tort reasoning. In contrast, both SFT$_{LC}$ and DPO$_{LC}$ exhibit consistent performance improvements across models, with significant gains[3] observed for Qwen-3-8B and Llama-3.1-8B.

The consistent performance gains achieved by Prompt$_{LC}$, SFT$_{LC}$ and DPO$_{LC}$ collectively demonstrate the effectiveness of incorporating structured legal reasoning signals, whether through legal prompting or model post-training, to improve model abilities on complex legal reasoning tasks.

Generalizability to Other Legal AI Tasks

To evaluate the generalizability of our proposed reasoning-incorporated approaches to other tasks, we conduct experiments on two related legal AI tasks: Legal Named-Entity Recognition and Criminal Damages Calculation (Fei et al. 2024). Although both tasks are situated within the context of criminal cases, they share certain similarities with the *legal element identification* and the *compensation calculation* components of our LexChain framework. We compare the performance of reasoning-enhanced models trained using SFT and DPO to their zero-shot counterparts. As shown in Table 2, both the SFT-based and DPO-based approaches consistently achieve comparable or better performance on these tasks, suggesting that explicitly integrating legal reasoning chains into model training can enhance model performance across a broader spectrum of legal tasks.

[3]Significance is tested through paired t-test (α=0.01).

Analysis of Tort Liability Types

To analyze model performance across different tort categories, we classify test instances in LexChain$_{eval}$ according to their underlying dispute types. We focus on the four most prevalent types of tort liability in real-world legal practice (China Judgments Online 2013): liability for motor vehicle traffic accident, liability for injury to providers of labor services, liability for product and liability for medical damage, and examine model performance within each category.

Figure 3 presents the average overall scores of two representative models, GPT-4o and DeepSeek-R1, on these four categories. As shown, both models achieve their highest scores on traffic accident cases, while their performance drops notably in labor services disputes, the second most common category in real-world tort litigation. In particular, GPT-4o scores below 35 in this category, indicating a significant weakness. This analysis reveals important weaknesses in existing LLMs' legal reasoning capabilities, suggesting a promising direction for future model improvements.

Conclusion

In this work, we propose LexChain, a structured reasoning framework that explicitly models legal reasoning for Chinese tort-related cases. We introduce a benchmark to evaluate LLMs' ability to identify key reasoning steps in tort case analysis, and find that existing models exhibit notable limitations on this task. By incorporating LexChain-style reasoning through legal prompting, supervised fine-tuning and direct preference optimization, we observe consistent performance improvements on tort reasoning. These reasoning-enhanced models also generalize well to other legal AI tasks. These results highlight the importance of explicitly modeling legal reasoning structures to enhance LLMs' ability for authentic, domain-sensitive legal analysis.

Acknowledgments

We thank the anonymous reviewers for their insightful feedback and suggestions. We are deeply grateful to the legal practitioners we consulted in the early stage of this research. Their generous insights and practical perspectives helped shape the legal reasoning framework proposed in this paper. This work is also supported by the AI9Stars community.

References

Anthropic. 2025. Introducing Claude 4. https://www.anthropic.com/news/claude-4. (Accessed on July 29, 2025).

Catala, P.; and Weir, T. 1965. *Delict and Torts: A Study in Parallel*, volume 2. Institute of Comparative Law of Tulane University.

China Judgments Online. 2013. China Judgments Online. https://wenshu.court.gov.cn. (Accessed on July 29, 2025).

Cohen, J. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1): 37–46.

Comanici, G.; Bieber, E.; Schaekermann, M.; Pasupat, I.; Sachdeva, N.; Dhillon, I.; Blistein, M.; Ram, O.; Zhang, D.; Rosen, E.; et al. 2025. Gemini 2.5: Pushing the Frontier With Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *arXiv preprint arXiv:2507.06261*.

Deng, W.; Pei, J.; Kong, K.; Chen, Z.; Wei, F.; Li, Y.; Ren, Z.; Chen, Z.; and Ren, P. 2023. Syllogistic Reasoning for Legal Judgment Analysis. In *EMNLP*, 13997–14009.

El-Kishky, A.; Wei, A.; Saraiva, A.; Minaiev, B.; Selsam, D.; Dohan, D.; Song, F.; Lightman, H.; Clavera, I.; Pachocki, J.; et al. 2025. Competitive Programming With Large Reasoning Models. *arXiv preprint arXiv:2502.06807*.

Fei, Z.; Shen, X.; Zhu, D.; Zhou, F.; Han, Z.; Huang, A.; Zhang, S.; Chen, K.; Yin, Z.; Shen, Z.; et al. 2024. LawBench: Benchmarking Legal Knowledge of Large Language Models. In *EMNLP*.

Fernandes, R.; Biedenkapp, A.; Hutter, F.; and Awad, N. 2025. A Llama Walks Into the 'Bar': Efficient Supervised Fine-Tuning for Legal Reasoning in the Multi-State Bar Exam. *arXiv preprint arXiv:2504.04945*.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*.

Gu, J.; Jiang, X.; Shi, Z.; Tan, H.; Zhai, X.; Xu, C.; Li, W.; Shen, Y.; Ma, S.; Liu, H.; et al. 2024. A Survey on LLM-as-a-Judge. *arXiv preprint arXiv:2411.15594*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. LoRA: Low-Rank Adaptation of Large Language Models. *ICLR*, 1(2): 3.

Huang, D.; Bu, Q.; Qing, Y.; and Cui, H. 2023. CodeCoT: Tackling Code Syntax Errors in CoT Reasoning for Code Generation. *arXiv preprint arXiv:2308.08784*.

Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; Ramesh, A.; Clark, A.; Ostrow, A.; Welihinda, A.; Hayes, A.; Radford, A.; et al. 2024. GPT-4o System Card. *arXiv preprint arXiv:2410.21276*.

InternLM. 2025. InternLM3 Model Card. https://huggingface.co/internlm/internlm3-8b-instruct. (Accessed on July 29, 2025).

Jiang, C.; and Yang, X. 2023. Legal Syllogism Prompting: Teaching Large Language Models for Legal Judgment Prediction. In *International Conference on Artificial Intelligence and Law*, 417–421.

Kontorovich, E. 2001. The Mitigation of Emotional Distress Damages. *The University of Chicago Law Review*, 68(2): 491–520.

Kuppa, A.; Rasumov-Rahe, N.; and Voses, M. 2023. Chain of Reference Prompting Helps LLM to Think Like A Lawyer. In *Generative AI Law Workshop*.

Landis, J. R.; and Koch, G. G. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 159–174.

Latin, H. A. 1985. Problem-Solving Behavior and Theories of Tort Liability. *Calif. L. Rev.*, 73: 677.

Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. 2024. Deepseek-V3 Technical Report. *arXiv preprint arXiv:2412.19437*.

Liu, Y.; Iter, D.; Xu, Y.; Wang, S.; Xu, R.; and Zhu, C. 2023. G-eval: NLG Evaluation Using GPT-4 With Better Human Alignment. *arXiv preprint arXiv:2303.16634*.

Loshchilov, I.; and Hutter, F. 2017. Decoupled Weight Decay Regularization. *arXiv preprint arXiv:1711.05101*.

Moore, M. S. 1999. Causation and Responsibility. *Social Philosophy and Policy*, 16(2): 1–51.

OpenAI. 2025. OpenAI o3-mini. https://openai.com/index/openai-o3-mini/. (Accessed on July 29, 2025).

Pearson, R. N. 1979. Apportionment of Losses Under Comparative Fault Laws - An Analysis of the Alternatives. *La. L. Rev.*, 40: 343.

Samuelson, D. R. 1997. Introducing Legal Reasoning. *J. Legal Educ.*, 47: 571.

Scharffs, B. G. 2004. The Character of Legal Reasoning. *Wash. & Lee L. Rev.*, 61: 733.

Schauer, F. F. 2009. *Thinking Like a Lawyer: A New Introduction to Legal Reasoning*. Harvard University Press.

Servantez, S.; Barrow, J.; Hammond, K.; and Jain, R. 2024. Chain of Logic: Rule-Based Reasoning With Large Language Models. *arXiv preprint arXiv:2402.10400*.

Smith, J. 1911. Legal Cause in Actions of Tort. *Harv. L. Rev.*, 25: 303.

Van Wijck, P.; and Winters, J. K. 2001. The Principle of Full Compensation in Tort Law. *European Journal of Law and Economics*, 11(3): 319–332.

Wagner, G. 2019. Comparative Tort Law. *The Oxford Handbook of Comparative Law*, 1003–1042.

Wan, M.; Zhu, F.; Amour, B.; Tang, H.; et al. 2022. *The Civil Code of the People's Republic of China*. Springer.

Wang, Y.; Yuan, J.; Chuang, Y.-N.; Wang, Z.; Liu, Y.; Cusick, M.; Kulkarni, P.; Ji, Z.; Ibrahim, Y.; and Hu, X. 2024. DHP Benchmark: Are LLMs Good NLG Evaluators? *arXiv preprint arXiv:2408.13704*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *NeurIPS*, 35: 24824–24837.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*.

Yu, F.; Quartey, L.; and Schilder, F. 2022. Legal Prompting: Teaching A Language Model to Think Like A Lawyer. *arXiv preprint arXiv:2212.01326*.

Yu, F.; Quartey, L.; and Schilder, F. 2023. Exploring the Effectiveness of Prompt Engineering for Legal Reasoning Tasks. In *Findings of ACL*, 13582–13596.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *NeurIPS*, 36: 46595–46623.

Zheng, Y.; Zhang, R.; Zhang, J.; Ye, Y.; Luo, Z.; Feng, Z.; and Ma, Y. 2024. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models. In *ACL*.