Collaborative Reasoning with Reranked Multi-facet Retrieval for Legal Judgment Prediction

Wei Cai[1] * **Zihan Mei**[1] * **Haojun Huang**[1] **Bang Wang**[1] †
[1]School of Electronic Information and Communications,
Huazhong University of Science and Technology, Wuhan, China
{caiw, meizh, hjhuang, wangbang}@hust.edu.cn

Abstract

Legal Judgment Prediction (LJP) aims to predict the charge, relevant law article, and term of penalty from case descriptions. Existing LLM-as-Judge solutions apply retrieved historical cases to enhance prediction, but the retrieval might contain irrelevant and redundant information, thereby limiting their accuracy. Moreover, these solutions rely on a single LLM for prediction and lack mechanisms to address potential bias and hallucination in its inference. To address these limitations, we introduce the *Collaborative Reasoning with Reranked Multi-facet Retrieval* (CoRe) framework, which retrieves historical cases from multiple facets, reranks them as candidate precedents, and employs two collaborative LLM judges to make the final prediction. The two LLM judges are inspired by the collegiate bench mechanism in real-world judicial practices, where Judge A makes a preliminary prediction that guides Judge B in re-selecting precedents and producing the final prediction. Experiments on public datasets validate the effectiveness of our CoRe framework in terms of better prediction performance over the competitors.

1 Introduction

Legal Judgment Prediction (LJP) aims to automatically make the judgment for a legal case based on its description, thereby helping legal practitioners to improve the efficiency of legal decision making (Luo et al., 2017; Chalkidis et al., 2020; Niklaus et al., 2021; Benedetto et al., 2023). The legal judgment Y consists of the *charge* y_c, law article y_l, and the term of penalty y_t. The LJP task is often modeled as a multi-task prediction, i.e., output the $Y = \{y_c, y_l, y_t\}$ based on the case description.

For the LJP task, many neural network-based approaches have been proposed in a multi-task setting (Zhong et al., 2018; Yue et al., 2021; Dong and Niu, 2021; Xu et al., 2020; Tong et al., 2024; Yue et al., 2024), as illustrated by Figure 1 (a). For example, (Zhong et al., 2018) integrated multiple subtasks and their directed acyclic graph dependencies into judgment prediction. (Yue et al., 2021) designed the NeurJudge+ to exploit the intermediate subtask results for the predictions of other subtasks. Besides the large training overhead, the trained networks may still be insufficient for understanding case facts and reasoning about their relationships to the judgment. Furthermore, their predictions only relied on individual cases, ignoring similarities among cases.

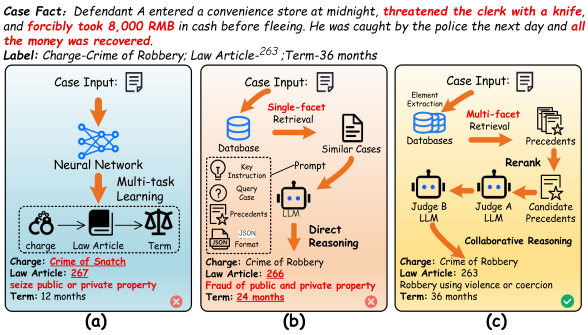


Figure 1: Examples of deficiencies in the LJP task.

Recently, large language models (LLMs) have shown remarkable proficiency in understanding and reasoning, allowing them to make legal judgments without undergoing additional training processes (Chalkidis et al., 2020; Ma et al., 2021; Zhao et al., 2026; Wu et al., 2023; Dong and Fan, 2025; Tang et al., 2025). As illustrated in Figure 1 (b), when using an LLM as a judge, the common approach is to feed it with a structured task prompt with some retrieved historical cases as demonstrations. Specifically, the prompt generally consists of the task instruction, a list of precedents, the query case, and the output format.

Although the LLM-as-Judge seems promising for the LJP task, we note that two limitations exist in the current approaches. The first is that the precedent retrieval is merely based on the semantic sim-

*Equal contribution.
†Corresponding author.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 57666–57683
October 24-29, 2026 ©2026 Association for Computational Linguistics

ilarities between the query description and those of historical cases. Notice that current retrieval is often implemented with vector search. That is, the full description of a case is encoded as a vector and precedents are retrieved based on vector similarities. However, encoding the full description into a single vector may introduce irrelevant information and unnecessary noise, impacting the quality of demonstrations for LLM judges. Indeed, although some elements in a case description, such as behaviors and consequences, constitute only a small part of a case, they are key to judgment prediction.

The second limitation is that current LLM-based approaches rely on a single LLM for directly reasoning the judgment, which ignores a common yet important human judicial procedure in practice: that is, either the prosecutor or the defendant or both can appeal to the next higher-level court if they are not satisfied with the first decision. This appeal process could be more important in the LLM-as-Judge scenario, as LLMs are prone to hallucinations and may consequently produce inaccurate predictions. While some emerging LLMs-as-Multi-Agents systems introduce multiple LLMs, they typically focus on role simulation (e.g., simulating different courtroom personas or debating perspectives), rather than employing specialized tools tailored for LJP. However, such horizontal role-playing fails to capture the vertical oversight essential to LJP, leaving a gap in systematic error correction. Therefore, we employ a second LLM as a complementary judge to verify, refine, and, when necessary, correct the preliminary judgment produced by the first LLM.

To address the aforementioned limitations, we propose a novel LLM-based framework for the LJP task, called **Co**llaborative Reasoning with **Re**ranked Multi-facet Retrieval (CoRe), which retrieves historical cases from multiple facets, reranks them as candidate precedents, and applies two collaborative LLM judges for reaching final prediction. In particular, our CoRe consists of four modules: *case element extraction* (CEE), *multi-facet precedent retrieval* (MPR), *reciprocal precedent reranking* (RPR), and *collaborative reasoning via LLM judges* (CRJ).

The CEE module extracts essential legal elements, including behavior, consequence, and fact, from a case description. The MPR module retrieves three lists of historical cases from each element facet. The RPR module computes reciprocal scores for retrieved cases and reranks them as candidate precedents. Finally, the CRJ module employs two LLMs as judges for prediction in a collaborative way: Judge A generates a preliminary prediction, based on which Judge B re-selects precedents for the final judgment prediction. Experiments on two public datasets validate the effectiveness of our CoRe framework in terms of new state-of-the-art results. Ablation studies also demonstrate the effectiveness of each module in our framework. A comprehensive discussion of related work is deferred to Appendix A.

2 Methodology

2.1 CoRe Overview

Fig. 2 illustrates the proposed CoRe framework, which consists of four key modules: (1) The CEE module leverages an LLM to extract three essential elements from a case description; (2) The MPR module retrieves semantically similar precedents from different facets; (3) The RPR module reranks precedents retrieved from different facets into a single queue; and (4) The CRJ module leverages two LLMs as judges with selected precedents to collaboratively make the final decisions.

2.2 Case Element Extraction

Precedents play an important role in real-world legal judgment. For the LJP task, a widely used approach is to encode the full case description for retrieving semantically similar precedents, which, however, often suffers from redundant narratives and human judgment bias. We argue that for legal judgment, similar criminal behavior and consequence are more important than the identities of the prosecutor and the defendant, which are often different in different cases. Furthermore, a case description often contains the case-specific entities, such as name of places, which, if encoded, could compromise the quality of retrieved precedents for judgment.

We propose to extract three essential elements from a case description, namely, *behavior*, *consequence*, and *fact*, each of which will be used for independent retrieval. Behavior describes the defendant's actions, characterized by means, process, frequency, etc. The consequence captures the severity of the harm caused, including material losses, personal injury, social impact, etc. The fact element summarizes the case description after removing irrelevant details and retaining key information relevant to conviction and sentencing,

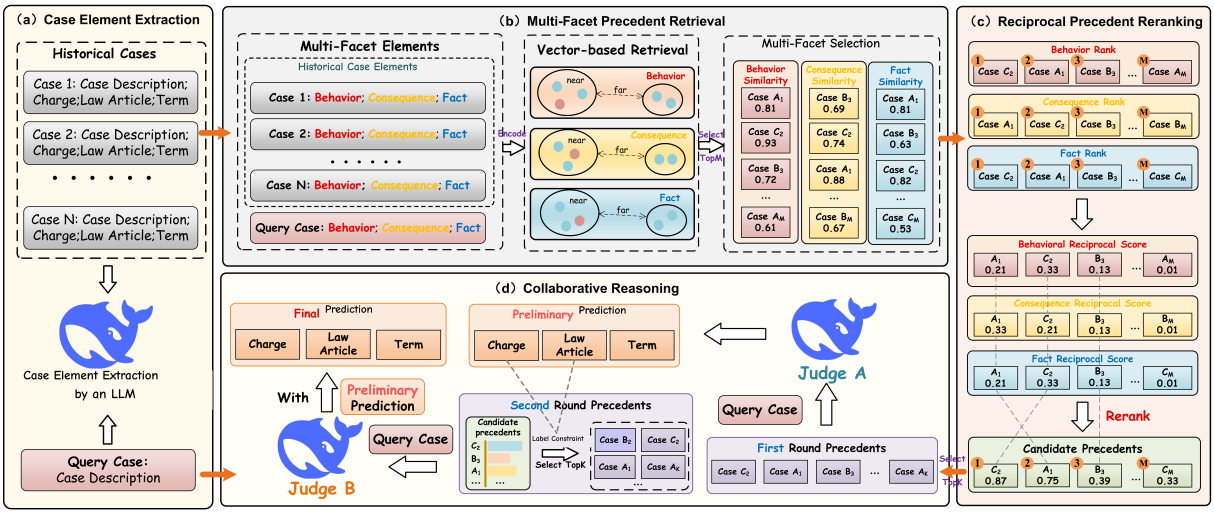


Figure 2: Overview of our proposed framework CoRe.

focusing on key information about the conviction and sentencing in the case description.

We leverage an LLM for case element extraction. Let D denote a case description. We prompt the LLM to extract its behaviors E_{b}, consequences E_{c} and facts E_{f}, all of which are composed of a few words and/or sentences. Fig. 2 (a) illustrates a few extraction examples. The evaluation of the extracted elements and the extraction prompts are detailed in Appendices F and H, respectively.

2.3 Multi-facet Precedent Retrieval

In real-world judicial practices, human judges often consult similar precedents when issuing a ruling. We argue that their practices would focus more on different aspects of the essentials in historical cases. To this end, we propose retrieving precedents from multiple facets in an independent way. That is, we retrieve three lists of precedents, focusing on similar behaviors, similar consequences, and similar facts, respectively.

We adopt the *vector search* technique for the retrieval of precedents and use a *text-embedding-ada-002 model* (OpenAI, 2022) in our experiments as the encoder to obtain the dense vector representations of case elements. Specifically, let $\mathbf{e}_{\mathrm{b}}$, $\mathbf{e}_{\mathrm{c}}$ and $\mathbf{e}_{\mathrm{f}}$ denote the encoded representation of the behaviors E_{b}, consequences E_{c} and facts E_{f}, respectively. For the query case Q for its judgment prediction, we let $\mathbf{q}_{\mathrm{b}}$, $\mathbf{q}_{\mathrm{c}}$ and $\mathbf{q}_{\mathrm{f}}$ denote its encoded representations of its behaviors Q_{b}, consequences Q_{c} and facts Q_{f}, respectively.

We compute the squared L2 distance between each query element and the corresponding element of each historical case as the retrieval score for constructing three precedent lists. For all historical cases, we compute $d_{\mathrm{b}} = \|\mathbf{e}_{\mathrm{b}} - \mathbf{q}_{\mathrm{b}}\|_2^2$, sort the cases in ascending order, and select the top-M cases to construct a *behavior precedent list* L_{b}. Similarly, we construct the *consequence precedent list* L_{c} and *fact precedent list* L_{f} for the query case.

We note that the three precedent retrievals are conducted independently from the same set of historical cases. It is possible that a historical case can appear simultaneously in one or more precedent lists. Notice that M is a hyperparameter. We set it to a large value of $M = 5000$ in our experiments. On the one hand, this allows enough precedents for references, and on the other hand filters out historical cases whose behaviors and consequences are unlikely to appear in the query. Because cases involving infrequent charges may have limited comparable precedents, the three facet-specific retrieval lists can differ in length. For each facet, we retain all available relevant precedents up to M and do not pad a shorter list with precedents retrieved from other facets.

2.4 Reciprocal Precedent Reranking

To reconcile the divergent rankings generated from the three-facet precedent lists, we propose the *reciprocal precedent reranking* (Cormack et al., 2009) to fuse the three lists, i.e., $\mathsf{L}_{\mathrm{b}}, \mathsf{L}_{\mathrm{c}}, \mathsf{L}_{\mathrm{f}}$, into a single *candidate precedent list* L. It is notable that the reciprocal reranking can leverage mutual relevance to address one-facet ranking limitations, boosting relevance and accuracy for the top candidate precedents.

Let $\mathcal{L}$ denote the set of historical cases appearing in all three precedent lists. For each case $C \in \mathcal{L}$,

we let r_{b} denote its ranking position in the list L_{b}. For example, if the case is in the 5th position of L_{b}, then $r_{\mathrm{b}} = 5$. Similarly, we obtain r_{c} and r_{f} for the case C from the list L_{c} and L_{f}, respectively. We compute its *reciprocal score* by

$$R = \frac{1}{r_{\mathrm{b}} + \alpha} + \frac{1}{r_{\mathrm{c}} + \alpha} + \frac{1}{r_{\mathrm{f}} + \alpha},$$

where α denotes the *smoothing factor* to prevent overly dominant influence from a single high ranking position. For example, the reciprocal score of a precedent ranked first in one list but poorly in other lists will not be excessively high, as the smoothing factor ensures that its final score also depends on contributions from all facets rather than being dominated by a single high rank.

We rerank all cases in $\mathcal{L}$ according to their reciprocal scores in descending order to construct the candidate precedent list L. This list integrates information from multiple facets of historical cases to provide a ranked list of balanced and representative candidate precedents to assist the next step LLM reasoning.

2.5 Collaborative Reasoning via LLM Judges

In real-world judicial practices, the *collegiate bench mechanism* assigns more than one human judge for collaboratively hearing and deciding a case to ensure the ruling fairness and accuracy. To emulate this mechanism for the LJP task, we propose to use two LLMs as two judges with a collaborative reasoning process to achieve the final judgment prediction.

Two LLM judges use structured prompts with query and selected precedents for prediction, and they work in a sequential way: **Judge A** generates preliminary predictions to guide more targeted precedent selection; **Judge B** performs secondary reasoning to correct potential biases in the preliminary results and outputs the final prediction.

Judge A (Preliminary Prediction): We construct a structured prompt for Judge A, which is designed as a sequence of components: it begins with base instructions that define the task, followed by a list of charges. Next, the facts of precedents $\mathsf{L}_{\mathrm{1st}}$ are presented in descending order of their reciprocal scores. The facts of the query case are then introduced, followed by output requirements and key instructions, which guide the LLM on how to reason. The list $\mathsf{L}_{\mathrm{1st}}$ is composed of *first round precedents* selected as the top-K precedents from the candidate list L. Appendix H provides the details of the Judge A prompt.

Judge A outputs the preliminary prediction $\hat{Y}^A = \{\hat{y}_c^A, \hat{y}_l^A, \hat{y}_t^A\}$, where $\hat{y}_c^A$, $\hat{y}_l^A$ and $\hat{y}_t^A$ respectively represent the predicted charge, relevant law article, and term of penalty.

Judge B (Final Prediction): To enhance the relevance of the precedents in its prompt, Judge B first performs a label-constrained selection of *second round precedents* based on the preliminary prediction $\hat{Y}^A$ as follows. Recall that all precedents in L are those historical cases with ground-truth labels. We select the top-K precedents in L with the same labels as $\hat{y}_c^A$ and $\hat{y}_l^A$ to construct the second round list $\mathsf{L}_{\mathrm{2nd}}$. If fewer than K training precedents match the preliminary charge and law-article predictions, we retain all available matching precedents and backfill the remaining positions with the highest-ranked precedents from $\mathsf{L}_{\mathrm{1st}}$ that have not already been selected.

The Judge B prompt is constructed similarly to Judge A, but with two key differences: the facts of precedents come from the second round precedents $\mathsf{L}_{\mathrm{2nd}}$, which are label-constrained based on $\hat{Y}^A$; the key instructions are modified to guide the LLM in revising and refining the preliminary prediction.

Specifically, for charge and law article prediction, Judge B re-examines the preliminary prediction using $\mathsf{L}_{\mathrm{2nd}}$. Unlike $\mathsf{L}_{\mathrm{2nd}}$, the first-round list $\mathsf{L}_{\mathrm{1st}}$ is constructed solely by semantic retrieval and may include precedents with different legal objectives or commission methods (e.g., contract fraud vs. fraud, or affray vs. intentional injury). By retaining only label-consistent precedents in $\mathsf{L}_{\mathrm{2nd}}$, Judge B receives more focused evidence to verify and refine the preliminary prediction. For the penalty prediction, the LLM compares the conviction and sentencing factors in the query case with those in $\mathsf{L}_{\mathrm{2nd}}$ and then re-predicts the term of the penalty. Appendix H provides the details of the Judge B prompt.

Judge B outputs the final prediction $\hat{Y}^B = \{\hat{y}_c^B, \hat{y}_l^B, \hat{y}_t^B\}$, where $\hat{y}_c^B$, $\hat{y}_l^B$ and $\hat{y}_t^B$ respectively represent the predicted charge, relevant law article, and term of penalty.

3 Experiments

3.1 Experiment Setup

Datasets: We conduct experiments on two public datasets: CAIL2018 (Xiao et al., 2018) and MultiLJP (Lyu et al., 2023). For MultiLJP, we fol-

Tasks	Charges				Law Article				Term of Penalty			
Metrics	**Acc.**	**MP**	**MR**	**F1**	**Acc.**	**MP**	**MR**	**F1**	**Acc.**	**MP**	**MR**	**F1**
TOPJUDGE (Zhong et al., 2018)	82.10	83.60	78.42	79.05	79.88	77.77	73.67	73.60	36.29	34.73	32.73	29.43
MPBFN (Yang et al., 2019)	82.14	82.28	80.72	80.72	79.12	76.30	76.02	74.78	36.02	31.94	28.60	29.85
LADAN (Xu et al., 2020)	85.07	83.42	82.52	82.74	81.20	78.24	77.38	76.47	38.29	36.16	32.49	32.65
NeurJudge+ (Yue et al., 2021)	83.69	82.65	83.59	82.55	80.88	76.95	78.29	76.37	36.78	35.04	31.79	33.25
CL4LJP (Zhang et al., 2023)	84.48	83.38	84.06	83.19	81.63	77.21	77.97	76.48	37.77	36.23	31.76	31.81
GraSCL (Wang et al., 2025)	85.55	84.10	82.83	83.06	80.81	78.85	78.06	77.34	36.82	36.08	32.37	33.12
MoCG (Mei et al., 2026)	86.29	84.67	84.74	84.33	83.29	79.07	80.12	78.68	39.36	37.79	34.68	33.82
DeepSeek-V3 (Liu et al., 2024)	62.22	27.05	20.49	21.55	77.30	61.88	49.89	51.28	20.12	32.25	26.15	17.69
CoRe	**88.51**	**89.41**	**86.47**	**87.29**	**88.22**	**85.43**	**80.73**	**82.37**	**43.20**	**43.31**	**36.80**	**38.20**

Table 1: Main results on CAIL-Small dataset. The best results are in bold, and the sub-optimal results are underlined. DeepSeek-V3 is evaluated in a zero-shot setting to validate the effectiveness of our proposed framework.

low the evaluation setting of ADAPT (Deng et al., 2024). Each multi-defendant case is organized into defendant-specific instances for inference, while the complete sets of annotated charges and applicable law articles are preserved for each defendant. Charge and law-article prediction are therefore evaluated as multi-label classification tasks. Detailed preprocessing procedures are provided in Appendix B.

During the text vectorization stage, we employ the text-embedding-ada-002 model released by OpenAI. For case element extraction and query judgment prediction, we use the DeepSeek-V3 model released on December 26, 2024. To further improve the execution efficiency of the experiments, all code implementations incorporated asynchronous parallel acceleration. We employ Accuracy (Acc.), Macro Precision (MP), Macro Recall (MR), and Macro F1 (F1) as evaluation metrics to measure the performance of all models. More details about dataset, experiment setting and the cost statistics can be found in Appendix B.

We compare our framework CoRe with the following competitors: (1) TOPJUDGE (Zhong et al., 2018), (2) MPBFN (Yang et al., 2019), (3) LADAN (Xu et al., 2020), (4) NeurJudge+ (Yue et al., 2021), (5) CL4LJP (Zhang et al., 2023), (6) GraSCL (Wang et al., 2025), (7) Lawformer (Xiao et al., 2021), (8) DeepSeek-V3 (Liu et al., 2024), (9) mT5 (Xue et al., 2021), (10) HRN (Lyu et al., 2023), (11) ADAPT (Deng et al., 2024), and (12) MoCG (Mei et al., 2026). Details of the baselines are provided in Appendix C. Not all baselines are applicable to both experimental settings. We report a baseline on a dataset only when its original formulation and available implementation support the corresponding task setting.

3.2 Main Results

Table 1 and Table 2 compare the judgment prediction performance between our CoRe framework and the competitors on the CAIL-Small and MultiLJP dataset, respectively. Underlining indicates the second-best results.

From the tables, it is observed that our CoRe achieves the best results in all performance metrics over the baselines on both datasets. In particular, CoRe improves the F1-scores of charge, law articles, and term of penalty predictions by 2.96, 3.69, and 4.38 percentage points over the second-best results, respectively, on the CAIL-small dataset. For the MultiLJP dataset, the gains reach 12.32, 2.43, and 1.09 percentage points in F1, respectively. The results validate our design motivations and objectives. The extraction of the behavior, consequence, and fact from a case can remove irrelevant or even noisy narratives but focus on key information about conviction and sentencing; the multi-facet retrieval provides a more fine-grained and systematic search on similar precedents for a query from three different perspectives; while the synthesized consideration by the reciprocal reranking boosts the selection of the more relevant and accurate candidate precedents. Finally, collaborative reasoning by two complementary LLM judges helps to reduce ruling biases and improve sentencing accuracy. A frequency-stratified analysis is provided in Appendix E.

We note that our CoRe does not require additional training overhead compared with those neural network-based approaches. Besides, we observe that using the LLM only in a zero-shot

Tasks	Charges				Law Article				Term of Penalty			
Metrics	**Acc.**	**MP**	**MR**	**F1**	**Acc.**	**MP**	**MR**	**F1**	**Acc.**	**MP**	**MR**	**F1**
TOPJUDGE (Zhong et al., 2018)	64.42	24.96	35.28	28.34	69.32	35.60	39.13	36.93	28.36	23.16	22.25	22.00
MPBFN (Yang et al., 2019)	65.59	32.79	33.20	31.59	72.47	34.73	34.22	34.35	28.32	21.59	20.91	20.70
LADAN (Xu et al., 2020)	46.62	20.68	32.42	24.74	54.57	38.09	22.40	26.64	27.05	24.05	23.43	23.16
NeurJudge+ (Yue et al., 2021)	59.51	34.19	25.36	27.55	65.21	41.72	36.96	38.15	30.06	27.56	25.63	25.95
Lawformer (Xiao et al., 2021)	65.94	38.97	29.12	32.76	75.50	36.18	35.33	34.00	32.37	22.66	20.68	18.30
mT5 (Xue et al., 2021)	81.52	63.33	49.94	52.86	87.49	74.28	53.65	58.84	33.66	39.13	24.23	23.04
HRN (Lyu et al., 2023)	83.47	63.52	58.74	60.90	84.28	64.56	59.78	62.09	34.33	37.06	30.38	33.39
ADAPT (Deng et al., 2024)	90.31	74.28	71.98	73.13	91.14	76.15	74.72	75.43	37.32	38.47	32.63	35.21
DeepSeek-V3 (Liu et al., 2024)	81.27	39.01	40.49	38.14	85.09	40.56	39.07	37.83	25.78	20.78	18.43	15.80
CoRe	**90.70**	**85.45**	**86.14**	**85.45**	**94.09**	**77.69**	**80.02**	**77.86**	**38.23**	**40.25**	**33.05**	**36.30**

Table 2: Main results on MultiLJP dataset. The best results are in bold, and the sub-optimal results are underlined. DeepSeek-V3 is evaluated in a zero-shot setting to validate the effectiveness of our proposed framework.

Tasks	Charges		Law Article		Term of Penalty	
Metrics	**Acc.**	**F1**	**Acc.**	**F1**	**Acc.**	**F1**
Sentencing CoT	85.55	83.19	85.63	77.34	36.30	31.96
Naive RAG	72.35	40.12	83.20	62.87	38.55	34.25
PAG	86.35	84.71	86.86	80.06	40.58	**38.57**
Parallel-Agents	84.83	82.74	84.13	76.26	35.70	28.43
Collaborative-Agents	86.02	84.45	85.70	79.02	41.01	36.54
CoRe	**88.51**	**87.29**	**88.22**	**82.37**	**43.20**	38.20

Table 3: Further baseline analysis on CAIL-Small.

setting yields substantially lower performance in judgment prediction: The DeepSeek-V3 in the two tables are prompted to make predictions but without any precedent demonstration (Appendix H provides its prompt details). This indicates that relying solely on an LLM for the LJP task remains highly challenging, as they struggle to capture the domain-specific knowledge and intricate reasoning required in legal contexts. Consequently, the superiority of our CoRe is not simply from the LLM reasoning capability, but rather from its task-oriented design, which better aligns with the legal domain.

3.3 Further Baseline Comparisons

To further evaluate the effectiveness of the CoRe framework, we conduct comparative experiments against several baselines: (1) **Naive RAG and PAG:** Naive RAG retrieves precedents based on the full case description, while PAG summarizes the retrieved precedents and uses the resulting summaries as contextual input; (2) **Sentencing CoT:** a single LLM performs step-by-step sentencing reasoning by first estimating a baseline penalty for the predicted crime and then adjusting it according to case-specific circumstances to obtain the final penalty; (3) **Parallel-Agents:** Judge A performs precedent-based reasoning using similar cases, Judge B independently applies the Sentencing CoT strategy, and Judge C selects the final judgment from the predictions produced by Judges A and B; and (4) **Collaborative-Agents:** this framework extends CoRe by introducing an additional Judge C to select the final judgment from the outputs of Judges A and B.

As illustrated in Table 3, we can observe that both Sentencing CoT and Naive RAG exhibit limited performance. For Sentencing CoT, it suffers from cascading errors, where inaccuracies in intermediate reasoning steps accumulate and ultimately degrade the quality of final prediction. Similarly, Naive RAG relies solely on full-text retrieval of precedents, failing to integrate multi-perspective information, which leads to lower judicial accuracy. Consequently, both methods are limited by their single reasoning paths and inadequate information sources.

Besides, neither multi-agent baseline improves the performance; instead, both show slight performance degradation. We attribute this result to two main reasons. First, all judges are instantiated from the same underlying LLM, so their predictions may lack sufficient diversity and complementarity. Second, in the three-judge architecture, Judge C makes the final decision without receiving substantially more informative or discriminative evidence than Judges A and B. As a result, Judge C is unlikely to produce more reliable judgments and may instead introduce additional noise. These results suggest that simply increasing the number of agents does

Tasks	Charges				Law Article				Term of Penalty			
Metrics	**Acc.**	**MP**	**MR**	**F1**	**Acc.**	**MP**	**MR**	**F1**	**Acc.**	**MP**	**MR**	**F1**
Full framework	**88.51**	**89.41**	**86.47**	**87.29**	**88.22**	**85.43**	**80.73**	**82.37**	**43.20**	**43.31**	**36.80**	**38.20**
w/o CEE	72.35	85.45	68.22	71.40	83.20	82.04	71.03	74.04	38.55	36.80	32.16	34.25
w/o retrieval	68.07	84.40	63.39	67.19	82.69	81.70	70.05	73.19	33.14	29.86	30.82	28.26
w/o RPR	87.01	88.23	84.06	85.34	87.14	83.87	77.21	79.38	38.60	39.91	34.25	34.87
w/o CRJ	87.33	87.98	84.95	85.74	87.00	83.37	78.61	80.22	41.63	41.74	35.61	37.10

Table 4: Ablation analysis on CAIL-Small.

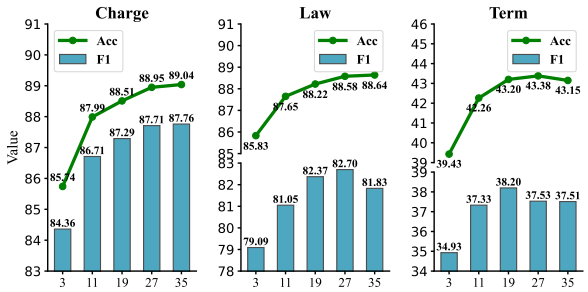


Figure 3: Impact of the number of precedents (K) in CRJ.

Retrieval setting	Charge Acc./F1	Law Acc./F1	Term Acc./F1
Behavior only	86.80/84.95	86.42/78.96	39.84/35.22
Consequence only	84.73/81.96	84.95/75.05	37.05/35.05
Fact only	86.12/83.88	86.03/78.41	40.71/36.18
w/o Behavior	86.94/85.21	86.81/79.71	42.58/37.64
w/o Consequence	87.98/86.61	87.74/81.53	41.67/36.69
w/o Fact	87.51/**87.31**	86.03/80.33	37.36/37.31
Full CoRe	**88.51**/87.29	**88.22/82.37**	**43.20/38.20**

Table 5: Effects of individual retrieval facets on CAIL-Small. Results are reported as Accuracy/Macro-F1 (%).

not necessarily improve LJP performance. The effectiveness of collaborative reasoning depends more on whether the collaboration mechanism can provide useful additional information and support reliable error correction.

3.4 Ablation and Parameter Studies

Module Ablation We perform experiments with the following ablation strategies: (1) w/o CEE: Only the CEE module is removed, and the framework retrieves precedents solely based on the full case description. (2) w/o retrieval: Historical case retrieval is not performed, and so the two LLM judges are without any precedent demonstration. (3) w/o RPR: Candidate precedents are obtained by reranking the multi-facet historical cases using the average of the three squared L2 distance. (4) w/o CRJ: Only the first LLM judge is used but with its selected first round precedents as demonstrations.

Table 4 provides the ablation study results, where each module is removed from the full CoRe framework. It can be observed that the full framework consistently achieves the best performance across all tasks and metrics, demonstrating that each module contributes to the overall effectiveness of CoRe. In particular, removing the CEE module leads to clear performance drops, especially for charge prediction and law article prediction, which indicates that the extracted elements provide useful structured information for legal reasoning.

The largest performance drops happen when removing the multi-facet precedent retrieval, indicating that precedents as demonstrations are crucial for LLM judges. It is interesting to note that the results of w/o retrieval are still much better than that of DeepSeek-V3 in Table 1. Notice that the former applies two LLM judges for collaborative reasoning, even though neither judge is provided with precedent demonstrations. This comparison again validates the effectiveness of our collaborative reasoning design.

Analysis of Facet Contributions. Table 5 evaluates the contributions of behavior, consequence, and fact in multi-facet retrieval. Single-facet retrieval consistently underperforms the full model, indicating that no individual facet provides sufficient evidence for all LJP subtasks.

The leave-one-out results show task-specific effects. Removing behavior causes the largest Macro-F1 drops for charge and law-article prediction (2.08 and 2.66 points), whereas removing consequence produces the largest decrease in term Macro-F1 (1.51 points). Removing fact notably reduces term accuracy by 5.84 points and law-article Macro-

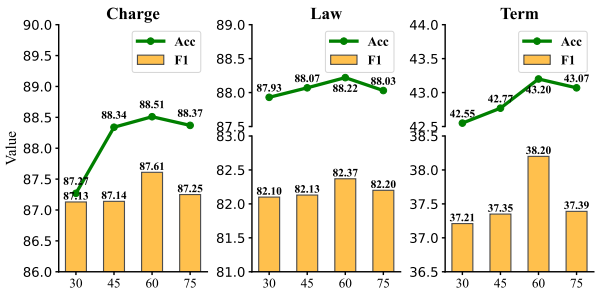


Figure 4: Impact of smoothing factor in RPR.

F1 by 2.04 points. Although w/o Fact obtains a marginally higher charge Macro-F1 (87.31 versus 87.29), its charge accuracy is lower by 1.00 point. Overall, the full model achieves the best accuracy across all three subtasks, supporting the complementary roles of the three retrieval facets.

Study of Top-K in CRJ The aforementioned results and analysis have indicated the importance of precedent as demonstrations for LLM judges. Fig. 3 plots the prediction performance against the number of precedents used in prompts. We observe that the influence of K varies across different prediction metrics. While several metrics continue to improve when $K > 19$, $K = 19$ achieves the best Term F1 and maintains competitive performance on the remaining metrics. Considering both predictive performance and retrieval efficiency, we set $K = 19$ in subsequent experiments.

Study of the Smoothing factor in RPR The smoothing factor α helps to balance the respective positional importance in the three retrieved lists. A case that ranks high in one list but low in others will not dominate the reciprocal score due to the smoothing factor. Fig. 4 presents the prediction performance against the value of smoothing factor. Results indicate that $\alpha = 60$ is the best choice for our CoRe. When $\alpha < 60$, the reciprocal score tends to be disproportionately influenced by top-ranked cases in a single facet, limiting the utilization of moderately ranked cases in other facets. Conversely, when $\alpha > 60$, the reciprocal scores become less distinguishable, diminishing the comparative importance of top-ranked cases in different facets. More parameter experiments can be found in Appendix D.

4 Case Studies

Figure 5 presents a case study to illustrate the advantages of our CoRe framework. When reasoning directly, LLMs are prone to generating incorrect

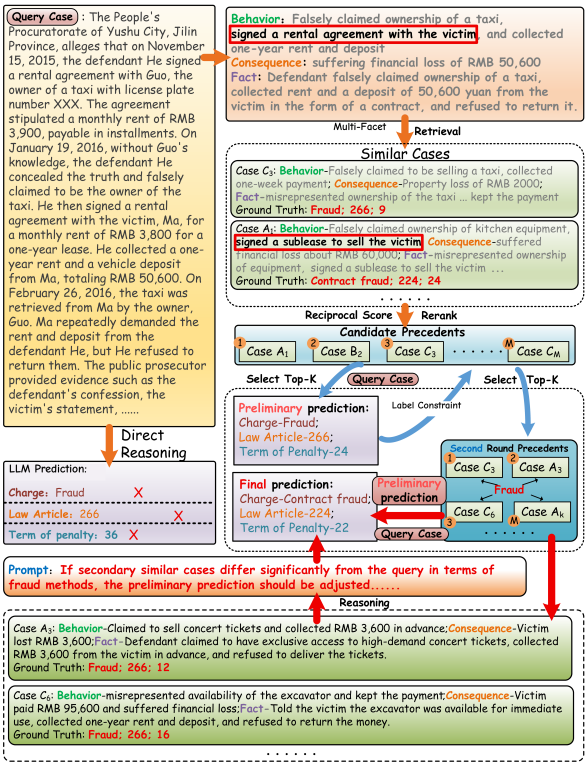


Figure 5: A case sample to demonstrate the effectiveness of our CoRe framework. Percentages in the agreement and disagreement groups are computed with respect to their corresponding group totals.

predictions, if not well prompted with appropriate precedents, such as mistakenly predicting the charge as *Fraud*, the law article as *266: Fraud of public or private property* and the term of penalty as *36*. In contrast, the element extraction, multi-facet retrieval and precedent reranking in our CoRe can help to ensure that the demonstrations in the prompt are from those multifaceted relevant precedents, which can assist the LLM-as-Judge to take more comprehensive consideration for prediction.

Judge A first makes a preliminary prediction using the selected precedents as in-context demonstrations. Note that the ground-truth labels of these precedents may differ. To provide a complementary assessment, Judge B cross-checks the preliminary prediction by examining whether the query case is consistent with the second-round precedents that share the charge and law article predicted by Judge A. For the case in Figure 5, Judge B is further guided by the observation that *"the second-round precedents differ significantly from the query case in terms of fraud methods ..."*. Based on this complementary evidence, Judge B refines the preliminary prediction and predicts the charge as *Contract fraud*, the law article as *224: Using contracts to*

Category	Charge		Law Article		Term of Penalty	
	Count	pct. (%)	Count	pct. (%)	Count	pct. (%)
Predictions Agree	25,584	95.64	25,580	95.63	19,811	74.06
Both Correct	23,021	89.98	22,930	89.64	9,353	47.21
Both Incorrect	2,563	10.02	2,650	10.36	10,458	52.79
Predictions Disagree	1,165	4.36	1,169	4.37	6,938	25.94
Only A Correct	339	29.10	340	29.08	1,783	25.70
Only B Correct	655	56.22	669	57.23	2,203	31.75
Both Incorrect	171	14.68	160	13.69	2,952	42.55
Judge A Correct	23,360	87.33	23,270	86.99	11,136	41.63
Judge B Correct	23,676	88.51	23,599	88.22	11,556	43.20
Total	26,749	100.00	26,749	100.00	26,749	100.00

Table 6: Experiment statistics of two LLM Judges.

commit fraud, and the penalty term as *22*.

Table 6 presents the experiment statistics of using only Judge A and using Judge B after Judge A on the CAIL-small dataset. It is first observed that Judge B agrees with Judge A in most cases (i.e., 95.64% agreement in charge and law article). Notice that for term of penalty, the two judges do not share a high level of agreement. On the one hand, predicting exact term of penalty is a much more complicated subtask (c.f., results in Table 1 and Table 2), as it is also dependent on the particular case-by-case facts to aggravate or mitigate the sentence. On the other hand, another design objective of Judge B is to re-examine the penalty term by referring to precedents with the same predicted charge and law article.

In our design, Judge B is allowed to override the prediction made by Judge A. Table 6 shows that, when the two judges disagree, Judge B achieves higher accuracy than Judge A across all three subtasks. Taking charge prediction as an example, among 1,165 disagreement cases, Judge A correctly predicts 339 cases (29.10%), whereas Judge B correctly predicts 655 cases (56.22%). Without Judge B, Judge A correctly predicts 23,360 cases in total. After the cross-checking process by Judge B, this number increases to 23,676. On the one hand, this suggests that even though Judge B is prompted with the preliminary prediction and those precedents of the same charge and law article, it can still make independent yet correct decisions. On the other hand, this indicates the necessity of using a second judge for dealing with potential biases of a single LLM, thereby validating our design objective.

4.1 Robustness across LLM Backbones

We evaluate CoRe with five LLM backbones: DeepSeek-V3, DeepSeek-V4-Flash, Gemini-2.5, GPT-4o-mini, and Qwen3-8B. For each backbone, we compare the *Direct* setting, where the LLM directly predicts the judgment, with the complete CoRe framework while keeping the remaining experimental settings unchanged.

Backbone	Setting	Charge Acc./MF	Law Acc./MF	Term Acc./MF
DS-V3	Direct	62.22/21.55	77.30/51.28	20.12/17.69
	CoRe	**88.51/87.29**	**88.22/82.37**	**43.20/38.20**
DS-V4-Flash	Direct	64.86/25.73	77.64/50.92	20.84/17.96
	CoRe	**86.92/84.81**	**86.75/79.46**	**40.82/35.71**
Gemini-2.5	Direct	67.41/30.82	79.05/55.12	22.36/19.27
	CoRe	**87.63/85.92**	**87.51/80.88**	**42.01/36.96**
GPT-4o-mini	Direct	63.57/26.18	76.93/49.85	20.31/17.24
	CoRe	**85.76/83.54**	**85.88/78.31**	**39.92/34.76**
Qwen3-8B	Direct	57.92/19.84	72.61/42.73	18.54/15.06
	CoRe	**82.63/79.52**	**82.97/74.68**	**36.74/31.45**

Table 7: Robustness of CoRe across LLM backbones. Results are reported as Accuracy/Macro-F1 (%). DS denotes DeepSeek.

As shown in Table 7, CoRe consistently improves Accuracy and Macro-F1 across all three LJP subtasks for every backbone. For Qwen3-8B, Macro-F1 increases by 59.68, 31.95, and 16.39 points for charge, law-article, and term prediction, respectively. Similar improvements are observed for the other backbones, especially for charge prediction and the more challenging term-of-penalty prediction. Although the absolute results vary across backbones, each CoRe variant substantially outperforms its corresponding direct-prediction setting. These results indicate that the effectiveness of CoRe is not tied to a specific LLM backbone.

5 Conclusion

In this paper, we have proposed the CoRe framework to address the limitations of existing approaches for the LJP task. The CoRe applies case element extraction and multi-facet precedent retrieval for searching task-relevant historical cases from different facets. The retrieved cases are reranked according to their reciprocal scores to be selected as demonstrations for LLM judges. The CoRe designs a collaborative reasoning mechanism by employing two LLM judges to complement each other for the final prediction, by which some potential prediction biases and hallucinations of a single LLM judge could be alleviated. Experiments on public datasets demonstrate that CoRe outperforms existing methods, with significant improvements in all three LJP subtasks. Future work will focus on the generalization of the CoRe framework to diverse languages and international legal systems, as detailed in Appendix I.

Limitations

Insufficient Multi-facet Retrieval Our framework employs the MPR module to retrieve precedents without adequately integrating the logical and temporal relationships between behaviors and consequences. This may result in the underutilization of cases that are causally or temporally relevant, thereby affecting the completeness of reasoning and the accuracy of predictions.

Limited Interpretability While the MPR and CRJ modules offer limited insights into the reasoning process, the underlying LLM fundamentally remains a black-box system. For LJP, further enhancing the transparency and interpretability of LLM-based decision-making is essential to satisfy the requirements for traceability and verifiability in AI-assisted judicial frameworks.

References

Irene Benedetto, Alkis Koudounas, Lorenzo Vaiani, Eliana Pastor, Elena Baralis, Luca Cagliero, and Francesco Tarasconi. 2023. Politohfi at semeval-2023 task 6: leveraging entity-aware and hierarchical transformers for legal entity recognition and court judgment prediction. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1401–1411.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.

Chenlong Deng, Kelong Mao, Yuyao Zhang, and Zhicheng Dou. 2024. Enabling discriminative reasoning in LLMs for legal judgment prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 784–796, Miami, Florida, USA. Association for Computational Linguistics.

Qian Dong and Shuzi Niu. 2021. Legal judgment prediction via relational learning. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 983–992.

Yubo Dong and Hehe Fan. 2025. Enhancing large language models through structured reasoning. *arXiv preprint arXiv:2506.20241*.

Zhitao He, Pengfei Cao, Chenhao Wang, Zhuoran Jin, Yubo Chen, Jiexin Xu, Huaijun Li, Kang Liu, and Jun Zhao. 2024. AgentsCourt: Building judicial decision-making agents with court debate simulation and legal knowledge augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9399–9416, Miami, Florida, USA. Association for Computational Linguistics.

Cong Jiang and Xiaolei Yang. 2023. Legal syllogism prompting: Teaching large language models for legal judgment prediction. In *Proceedings of the nineteenth international conference on artificial intelligence and law*, pages 417–421.

Hui Liao, Chuan Qin, Yongwen Ren, Hao Li, Zhenya Huang, Yanyong Zhang, and Chao Wang. 2026. Verdict: Verifiable evolving reasoning with directive-informed collegial teams for legal judgment prediction. *arXiv preprint arXiv:2603.19306*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Dugang Liu, Weihao Du, Lei Li, Weike Pan, and Zhong Ming. 2022. Augmenting legal judgment prediction with contrastive case relations. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2658–2667, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. 2017. Learning to predict charges for criminal cases with legal basis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2727–2736, Copenhagen, Denmark. Association for Computational Linguistics.

Yougang Lyu, Jitai Hao, Zihan Wang, Kai Zhao, Shen Gao, Pengjie Ren, Zhumin Chen, Fang Wang, and Zhaochun Ren. 2023. Multi-defendant legal judgment prediction via hierarchical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2198–2209, Singapore. Association for Computational Linguistics.

Yixiao Ma, Yunqiu Shao, Bulou Liu, Yiqun Liu, Min Zhang, and Shaoping Ma. 2021. Retrieving legal cases from a large-scale candidate corpus. *Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment, COLIEE2021*.

Zihan Mei, Chuanhong Zhan, Wenpeng Ye, and Bang Wang. 2026. Legal judgment prediction based on charge-anchored graph constraints. *Expert Systems with Applications*, 302:130480.

Joel Niklaus, Ilias Chalkidis, and Matthias Stürmer. 2021. Swiss-judgment-prediction: A multilingual legal judgment prediction benchmark. In *Proceedings of the Natural Legal Language Processing Workshop*

2021, pages 19–35, Punta Cana, Dominican Republic. Association for Computational Linguistics.

OpenAI. 2022. New and improved embedding model. OpenAI.

Jingyun Sun, Chengxiao Dai, Zhongze Luo, Yangbo Chang, and Yang Li. 2024. Lawluo: A multi-agent collaborative framework for multi-round chinese legal consultation. *Preprint*, arXiv:2407.16252.

Minghua Tang, Chen Bian, Liming Yang, and Xueling Zhong. 2025. Key-concept thinking prompting for improved reasoning in large language models. *Neurocomputing*, page 130986.

Suxin Tong, Jingling Yuan, Peiliang Zhang, and Lin Li. 2024. Legal judgment prediction via graph boosting with constraints. *Information Processing & Management*, 61(3):103663.

Jiawei Wang, Yuquan Le, Da Cao, Shaofei Lu, Zhe Quan, and Meng Wang. 2025. Graph reasoning with supervised contrastive learning for legal judgment prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2):2801–2815.

Yiquan Wu, Siying Zhou, Yifei Liu, Weiming Lu, Xiaozhong Liu, Yating Zhang, Changlong Sun, Fei Wu, and Kun Kuang. 2023. Precedent-enhanced legal judgment prediction with LLM and domain-model collaboration. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12060–12075, Singapore. Association for Computational Linguistics.

Chaojun Xiao, Xueyu Hu, Zhiyuan Liu, Cunchao Tu, and Maosong Sun. 2021. Lawformer: A pre-trained language model for chinese legal long documents. *AI Open*, 2:79–84.

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*.

Nuo Xu, Pinghui Wang, Long Chen, Li Pan, Xiaoyan Wang, and Junzhou Zhao. 2020. Distinguish confusing law articles for legal judgment prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3086–3095, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Wenmian Yang, Weijia Jia, Xiaojie Zhou, and Yutao Luo. 2019. Legal judgment prediction via multi-perspective bi-feedback network. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 4085–4091.

Weikang Yuan, Junjie Cao, Zhuoren Jiang, Yangyang Kang, Jun Lin, Kaisong Song, Tianqianjin Lin, Pengwei Yan, Changlong Sun, and Xiaozhong Liu. 2024. Can large language models grasp legal theories? enhance legal reasoning with insights from multi-agent collaboration. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7577–7597, Miami, Florida, USA. Association for Computational Linguistics.

Linan Yue, Qi Liu, Binbin Jin, Han Wu, and Yanqing An. 2024. A circumstance-aware neural framework for explainable legal judgment prediction. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):5453–5467.

Linan Yue, Qi Liu, Binbin Jin, Han Wu, Kai Zhang, Yanqing An, Mingyue Cheng, Biao Yin, and Dayong Wu. 2021. Neurjudge: A circumstance-aware neural framework for legal judgment prediction. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 973–982.

Han Zhang, Zhicheng Dou, Yutao Zhu, and Ji-Rong Wen. 2023. Contrastive learning for legal judgment prediction. *ACM Transactions on Information Systems*, 41(4):1–25.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Zican Dong, Yupeng Hou, Beichen Zhang, Yingqian Min, Junjie Zhang, Peiyu Liu, Xiaolei Wang, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Yiwen Hu, Jian-Yun Nie, and Ji-Rong Wen. 2026. A survey of large language models. *Frontiers of Computer Science*, 20(12).

Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal judgment prediction via topological learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3540–3549, Brussels, Belgium. Association for Computational Linguistics.

A Related Work

Neural Network-based LJP approaches Fueled by the progress of deep learning, many neural network-based approaches have been proposed for the LJP task (Zhong et al., 2018; Yue et al., 2021; Dong and Niu, 2021; Yue et al., 2024; Xu et al., 2020; Tong et al., 2024). Zhong et al. (2018) proposed a topological multi-task learning framework that incorporates dependencies among subtasks as a directed acyclic graph. Liu et al. (2022) introduced contrastive case relations to construct case triples, while refining encoding and decoding through case triples modeling. Wang et al. (2025) modeled label dependencies using graph reasoning and supervised contrastive learning with online hard negative mining. Tong et al. (2024) used legal label graphs and graph boosting with constraints to capture label correlations and ensure task consistency. However, these task-specific neural methods generally do not use retrieved precedents as in-context evidence for LLM-based legal reasoning.

LLM-based LJP approaches Recent studies have explored LLM-based legal judgment prediction through prompting, structured legal reasoning, and precedent utilization. Wu et al. (2023) proposed PLJP, which combines domain-specific models for candidate-label prediction and precedent retrieval with an LLM for final judgment reasoning. Jiang and Yang (2023) developed legal syllogism prompting, which guides LLMs to reason through a major premise, a minor premise, and a conclusion. Deng et al. (2024) proposed ADAPT, which decomposes case facts, discriminates among potential charges, and further improves LJP through fine-tuning with multi-task synthetic trajectories. These approaches demonstrate the potential of LLMs for legal judgment prediction, but they generally rely on a single LLM for final judgment reasoning. In contrast, CoRe employs two sequential LLM judges: Judge A produces a preliminary prediction based on retrieved precedents, while Judge B re-examines this prediction using label-constrained precedents to verify and refine the final judgment.

Multi-Agent Interaction in Legal Tasks Beyond single-LLM reasoning, recent studies have explored multi-agent collaboration for legal tasks by assigning LLMs complementary roles and coordinating their interactions (He et al., 2024; Sun et al., 2024; Yuan et al., 2024; Liao et al., 2026). He et al. (2024) introduced AgentsCourt, which simulates courtroom debate among multiple legal roles and incorporates legal knowledge retrieval. Sun et al. (2024) proposed LawLuo, in which specialized agents conduct multi-round Chinese legal consultation by mimicking the division of labor in law firms. Yuan et al. (2024) developed MALR to improve legal-theory understanding and reasoning through multi-agent information exchange and task decomposition. Liao et al. (2026) proposed VERDICT, which organizes specialized agents into a draft–verify–revise workflow for verifiable legal judgment prediction. Different from role-oriented collaboration, CoRe focuses on precedent-guided sequential judgment: Judge A produces a preliminary prediction based on multi-facet retrieved precedents, and Judge B re-selects label-consistent precedents to verify and refine the prediction.

Our approach explores how to use multiple LLM judges for collaborative reasoning with multi-facet retrieved precedents.

Dataset	CAIL-Small	MultiLJP
# Training Set Cases	101619	18,968
# Test Set Cases	26749	2370
# Charges	119	23
# Law Articles	103	22
# Intervals of prison term	11	11
# Average defendant per case	1	3.39
# Average length per case	440.9	3040.8

Table 8: Basic statistics of the datasets

Modules	Parameter	Value
CEE	LLMs	DeepSeek-V3
	Temperature t	0.1
MPR	Embedding Model	Text-embedding-ada-002
	Index Type	IndexFlatL2
	Dimension	1536
RPR	TopM	5000
	TopK	19
	Smoothing Factor α	60
CRJ	LLMs	DeepSeek-V3
	Judge A Temperature	0.1
	Judge B Temperature	0.1

Table 9: Detailed experimental setup values

B Experiment Settings

Details of Dataset Processing For CAIL2018, we apply the CEE pipeline to all cases in the training, validation, and test splits. The pipeline automatically extracts case elements and performs evaluation, repair, and schema validation before

Stage	Model	In-Tokens	Out-Tokens	API Requests	Cost (USD)	Time (hours)
Element Extraction (CEE)	DeepSeek-V3	60M	15M	130,000	$80	10
Vectorization (MPR)	Ada-002	20M	-	130,000	$2	4
Judge A (Preliminary)	DeepSeek-V3	48M	10M	30,000	$50	6
Judge B (Final)	DeepSeek-V3	48M	10M	30,000	$50	6
Total	**–**	**176M**	**35M**	**320,000**	**$182**	**26**

Table 10: Computational and cost analysis of CoRe on CAIL-Small over all processed training, validation, and test instances.

Prompt of Zero-Shot DeepSeek-V3

Description: This prompt is designed to guide DeepSeek in legal reasoning. The goal is to predict the Charge, relevant Law Article, and Term of penalty based on the case description.

Input: At about 20:00 on March 28, 2016, the defendant XXX found a VIVOX5 mobile phone of the victim XXX, beside the road of the football field in XXX Village, XXX District of the city. From 21:00 on March 28, 2016, the defendant secretly stole 3723 yuan from the victim XXX through the small amount secret free payment function of Alipay. After the incident, the family of the defendant XXX has compensated the victim for all losses and obtained forgiveness.

Prompt Details: "You are a legal assistant. Given a case description, your task is to perform Legal Judgment Prediction (LJP). Focus only on the case relevant to judicial reasoning and ignore rhetorical language or irrelevant details. Based on the case description, predict strictly according to the following schema:
"Charge" : the legally defined crime committed by the defendant
"Law Article" : the specific articles of law that govern the offense
"Term" : the appropriate term of penalty determined under the law
Note:
-Predict strictly using legal terminology: Charge, Relevant Law Article, and appropriate Term of Penalty.
-Do not add commentary or explanations."

Output: "Charge": Theft ✓ "Law Article": 265 ✗ "Term": 12 ✗

Figure 6: The prompt of DeepSeek-V3 in zero-shot learning

downstream retrieval. We subsequently perform task-specific preprocessing by filtering out cases containing fewer than 10 meaningful words and retaining only law articles and charges with sufficient training instances, following previous studies (Zhong et al., 2018; Xu et al., 2020; Yue et al., 2021). This procedure yields the CAIL-Small dataset used in our experiments. Detailed evaluation and repair statistics are reported only for the original CAIL test split in Appendix F . MultiLJP contains multi-defendant cases, where each defendant may be associated with multiple charges and applicable law articles. For defendant-specific inference, we reorganize each case into defendant-level instances by pairing the original case description with the target defendant. This preprocessing only changes the input organization and does not alter the original case content or judgment annotations. The complete sets of charges and law articles are retained as the ground-truth targets, and the original prison-term label is preserved. The parameter settings of all modules are shown in Table 9.

Computational and Cost Analysis As illustrated in Table 10, the reported computational cost covers the full CEE, MPR, and CRJ processing pipeline over all processed training, validation, and test instances of the CAIL-Small datasets. The total financial expenditure for the CoRe framework amounts to approximately $182, covering 320,000 API requests and the processing of 176M input tokens and 35M output tokens. The preliminary stages, including Case Element Extraction (CEE) and Multi-facet Precedent Retrieval (MPR), require 130,000 requests each. Among them, CEE consumes 60M input tokens and 15M output tokens, while MPR only involves input tokens for vectorization and therefore does not produce output tokens. The Collaborative Reasoning via LLM Judges (CRJ) module constitutes the main generative reasoning cost, where Judge A and Judge B each consume 48M input tokens and 10M output tokens across 30,000 requests. Although the two-judge structure introduces additional computa-



Figure 7: Impact of selecting LLM temperature.

tional overhead compared with single-LLM baselines, the framework leverages asynchronous parallel acceleration to maintain practical execution efficiency. Overall, the improved judgment prediction performance justifies the additional cost of this task-oriented design.

C Baselines

For each dataset, we include only baselines whose task formulation and implementation are compatible with the corresponding evaluation setting. We compare our framework CoRe with the following baselines:

(1) **TOPJUDGE** (Zhong et al., 2018): TOPJUDGE formalizes the dependencies among subtasks and proposes a topological multi-task learning framework to incorporate multiple subtasks and DAG dependencies into judgment prediction.

(2) **MPBFN** (Yang et al., 2019): MPBFN utilizes result dependencies among multiple subtasks and proposes a WCA mechanism to distinguish cases with similar descriptions but different penalties.

(3) **LADAN** (Xu et al., 2020): LADAN extracts discriminative features to effectively distinguish confusing law articles and designs a novel attention mechanism to fully exploit the discriminative features.

(4) **NeurJudge+** (Yue et al., 2021): NeurJudge+ utilizes the results of intermediate subtasks to separate the fact description into different circumstances and exploits them to make the predictions of other subtasks.

(5) **CL4LJP** (Zhang et al., 2023): CL4LJP proposes a supervised contrastive learning framework to capture the fine-grained differences between similar articles/charges for the LJP tasks

(6) **GraSCL** (Wang et al., 2025): GraSCL combines graph reasoning with supervised contrastive learning to model task and label dependencies in LJP. It applies contrastive learning at the node level to improve the discrimination of legal judgment labels.

(7) **Lawformer** (Xiao et al., 2021): Lawformer introduces a Longformer-based pre-trained language model tailored for legal texts, enabling efficient encoding of long case documents and improving downstream LJP tasks.

(8) **DeepSeek-V3** (Liu et al., 2024): DeepSeek-V3 employs a Mixture-of-Experts (MoE) architecture to dynamically select specialized experts for each input.

(9) **mT5** (Xue et al., 2021) is a multilingual encoder–decoder pre-trained language model that has been adopted as a sequence-to-sequence baseline for defendant-specific legal judgment prediction on MultiLJP.

(10) **HRN** (Lyu et al., 2023): HRN is a hierarchical reasoning framework for multi-defendant LJP. It models criminal relationships and sentencing circumstances, and predicts each defendant's law articles, charges, and term of penalty along hierarchical reasoning chains.

(11) **ADAPT** (Deng et al., 2024): ADAPT enhances LLMs via fine-tuning on multi-task synthetic trajectories, by decomposing case facts, discriminating among potential charges, and predicting final judgments.

(12) **MoCG** (Mei et al., 2026): MoCG combines a Charge Anchor Point (CAP) strategy, momentum contrastive learning, case augmentation with similar historical cases, and charge-anchored graph constraints to improve label discrimination and prediction consistency in LJP.

Stage / action	Count	Rate
Initial extraction succeeded	32,508	100.00%
Directly retained after evaluation	19,794	60.89%
Flagged for repair	12,714	39.11%
Re-extraction with feedback	9,771	30.06%
Strict-evidence re-extraction	2,445	7.52%
Consequence-only repair	344	1.06%
Automated fallback repair	154	0.47%
Successful repair calls	12,714	–
Failed repair calls	0	–
Records retained in the final file	32,508	100.00%

Table 11: Statistics of the automated CEE evaluation and repair pipeline on the original, unfiltered CAIL test split.

D Parameter Experiments

The LLM temperature governs the degree of randomness in its output. A higher temperature

Demonstration of evaluation and repair

Description: This prompt is designed to evaluate and repair extracted legal elements from case descriptions. The goal is to ensure that the extracted elements are valid, complete, and consistent with the original case description.

Input:
Case Description: At about 19:30 on May 12, 2017, the defendant XXX had a dispute with the victim XXX over a neighborhood conflict. During the dispute, the defendant struck the victim several times with a wooden stick, causing injuries to the victim's head and arm. According to the forensic examination, the victim suffered a minor injury. After the incident, the defendant compensated the victim for medical expenses and obtained forgiveness.
Extracted Elements:
Defendant's Behavior : The defendant struck the victim several times with a wooden stick during a neighborhood dispute.
Consequences : The victim suffered a minor injury to the head.
Fact : The defendant intentionally injured the victim and later voluntarily surrendered to the police.
Prompt Details: "You are a legal assistant. Given a case description and the extracted legal elements, evaluate whether the extracted elements accurately reflect the original case description and repair them if necessary. Focus on the following schema:
Defendant's Behavior : the concrete actions or conduct performed by the defendant
Consequences : the direct outcomes, damages, or losses caused by the defendant's behavior
Fact : the comprehensive description of the criminal case, including the basis for conviction and sentencing.
Note: Keep correct elements unchanged. If an element is incomplete, inconsistent, duplicated, unsupported, or incorrectly formatted, repair it according to the original case description. Do not include personal names or detailed explanations in parentheses.""

Output:
Defendant's Behavior : The defendant struck the victim several times with a wooden stick during a neighborhood dispute.
Consequences : The victim suffered injuries to the head and arm, which were identified as a minor injury.
Fact : The defendant intentionally injured the victim by striking the victim with a wooden stick, causing a minor injury. After the incident, the defendant compensated the victim for medical expenses and obtained forgiveness, which may be considered as a mitigating circumstance for sentencing.

Figure 8: The prompt of evaluation and repair.

leads to more diverse and creative text generation, whereas a value closer to zero promotes determinism and consistency. For the LJP task, rigor and accuracy are more important than creativity. Fig. 7 presents the predictive performance against the LLM temperature, where the results confirm the analysis and a good choice is setting temperature as 0.1.

E Label-Frequency Analysis

To examine performance variation across labels with different levels of test-set support, we partition charge and law-article labels into rare (1–10), few-shot (11–50), medium-frequency (51–200), and frequent ($>$ 200) groups according to their numbers of test instances. Table 13 reports the performance of CoRe for each group.

As shown in Table 13, CoRe maintains relatively stable performance for charge prediction across the few-shot, medium-frequency, and frequent groups. In contrast, law-article prediction is more sensitive to limited label support: Macro-F1 decreases from 89.07 for frequent labels to 75.75 for few-shot labels. Performance is zero for the rare groups, which contain only one charge class and three law-article classes. These results indicate that extremely low-support legal labels remain challenging.

F Evaluation of Extracted Contents

The CEE extraction, evaluation, repair, and schema-validation pipeline is applied to all processed training, validation, and test instances. To provide a consistent and held-out quality audit, Table 11 reports detailed evaluation and repair statistics only for the 32,508 cases in the original CAIL test split, before task-specific preprocessing. The evaluator assessed faithfulness to the source facts, completeness, consequence validity, summary quality, and non-hallucination on a 0–5 scale. We used DeepSeek-V4-Pro with a temperature of 0 and a maximum output length of 4,096 tokens. An extraction was directly retained if its overall score was at least 4 and no severe hallucination was detected; otherwise, it was routed to an automated targeted repair action. All evaluation, repair, and schema-validation procedures were performed automatically without human intervention.

Among the 32,508 evaluated cases in the original CAIL test split, 19,794 (60.89%) were directly

Type	Case Facts	Judge A	Judge B	Analysis
Positive correction	The defendant signed a sales contract with the victim, received the payment, and then failed to deliver the promised goods. During the transaction, the defendant used false identity information and had no actual ability to perform the contract.	**Charge:** Fraud **Law Article:** Article 266 **Term:** 36 months	**Charge:** Contract Fraud **Law Article:** Article 224 **Term:** 24 months	Judge A focuses on the illegal possession of property and predicts ordinary fraud. After second-round precedent selection, Judge B observes that the offense was committed through a contract and revises the prediction to contract fraud, which is consistent with the ground truth.
Harmful revision	The defendant secretly took the victim's mobile phone from the victim's coat pocket while the victim was resting in an Internet cafe. The victim did not notice the act at the time. The defendant later sold the phone and obtained illegal gains.	**Charge:** Theft **Law Article:** Article 264 **Term:** 10 months	**Charge:** Snatching **Law Article:** Article 267 **Term:** 12 months	Judge A correctly identifies the case as theft, since the defendant secretly took the property without the victim's awareness. However, Judge B overemphasizes that the property was taken from the victim's body and incorrectly revises the prediction to snatching. This harmful revision shows that Judge B may still override a correct preliminary prediction when it misinterprets fine-grained behavioral details.

Table 12: Representative cases of collaborative correction in CRJ.

Task	Label support	#Cls.	Acc.	Ma-F1
Charge	Rare (1–10)	1	0.00	0.00
	Few-shot (11–50)	30	82.75	86.09
	Medium (51–200)	32	87.07	88.05
	Frequent (> 200)	56	88.96	89.29
Law	Rare (1–10)	3	0.00	0.00
	Few-shot (11–50)	23	73.57	75.75
	Medium (51–200)	24	86.36	87.95
	Frequent (> 200)	52	88.92	89.07

Table 13: Performance across test-set label-support groups. Results are reported in percentage (%).

retained, while 12,714 (39.11%) were flagged for repair. The automated repair procedure successfully produced schema-valid outputs for all 12,714 flagged cases, with no repair failures. Consequently, all 32,508 audited test cases were retained after the CEE evaluation and repair pipeline. Task-specific preprocessing was subsequently applied to construct CAIL-Small; after filtering, the final test split contains 26,749 cases used for model evaluation.

G Analysis of Collaborative Corrections

To further analyze the effect of collaborative reasoning, we present two representative cases where Judge B either corrects or incorrectly revises the preliminary prediction of Judge A. In the positive correction case, Judge A makes an incorrect preliminary prediction due to misleading similarities in the first-round precedents. After label-constrained precedent selection, Judge B observes that the second-round precedents better match the query case in terms of criminal behavior and legal consequence, and therefore revises the prediction to the correct charge and law article. This example shows that the second judge can mitigate potential errors introduced by the first-round reasoning.

In contrast, the harmful revision case shows that Judge B may incorrectly override a correct preliminary prediction made by Judge A. Although the second-round precedents are selected according to the preliminary charge and law article, Judge B may still misinterpret fine-grained behavioral details during re-examination, such as confusing secret taking with public snatching. This suggests that collaborative reasoning is not always beneficial when the distinction between similar charges depends on subtle factual differences. Representative cases are presented in Table 12.

H LLM Prompt Details

In our framework, LLMs are employed in the CEE module and CRJ module, with a Chain-of-Thought (CoT) reasoning strategy incorporated to enhance decision accuracy. The process proceeds as follows: First, as illustrated in Figure 9, the LLM performs a step-by-step analysis of cases to extract key legal elements relevant to the legal judgment. Second, as illustrated in Figure 10 (a), based on

Prompt of CEE

Description: This prompt is designed to extract critical legal elements from case descriptions. The goal is to capture essential information for judicial reasoning—such as the defendant's behavior, consequences, and contextual factors—while filtering out redundant or irrelevant details.

Input: At about 20:00 on March 28, 2016, the defendant XXX found a VIVOX5 mobile phone of the victim XXX, beside the road of the football field in XXX Village, XXX District of the city. From 21:00 on March 28, 2016, the defendant secretly stole 3723 yuan from the victim XXX through the small-amount password-free payment function of Alipay. After the incident, the family of the defendant XXX has compensated the victim for all losses and obtained forgiveness.

Prompt Details: "You are a legal assistant. Given a case description, **extract the essential legal elements necessary for LJP**. Ignore rhetorical expressions, redundant details, or irrelevant information. Organize the output strictly according to the following schema:
Defendant's Behavior : the concrete actions or conduct performed by the defendant
Consequences : the direct outcomes, damages, or losses caused by the defendant's behavior
Fact : the comprehensive description of the criminal case, including the basis for conviction and sentencing.
Note: Perform a simple extraction; do not include personal names or detailed explanations in parentheses in the elements."

Output: **Defendant's Behavior** : Found the victim's mobile phone and used the Alipay small-amount password-free payment function to steal 3,723 yuan.
Consequences : The victim suffered an economic loss of 3,723 yuan.
Fact : The defendant committed theft by taking the victim's mobile phone and illegally withdrawing 3,723 yuan through Alipay. The family of the defendant compensated the victim's losses in full and obtained forgiveness, which can be considered as a mitigating circumstance for sentencing.

Figure 9: The prompt of CEE.

the first round precedents retrieved from these elements, the LLM performs analogical reasoning to generate preliminary predictions for the charge, the relevant law article, and the term of penalty. Finally, as illustrated in Figure 10 (b), the preliminary predictions are refined using second round precedents, allowing the LLM to revise and produce the final judgment. For comparison, Figure 6 shows the prompt of DeepSeek-V3 in zero-shot learning.

Notably, although we explicitly instruct the LLM to output predictions in a standardized JSON format, the generated responses may still contain redundant explanations, unstructured text, or formatting errors. To address this issue, we incorporate a format repair and validity checking mechanism in post-processing: recoverable outputs are structurally reconstructed, while irreparable invalid responses are automatically discarded, ensuring that only results conforming to the predefined JSON schema are retained.

I Future Work

Although our experiments demonstrate the effectiveness of CoRe on Chinese legal datasets, its generalizability to other languages and legal systems remains to be validated. The framework is designed to be adaptable because its core components (i.e., case element extraction, multi-facet retrieval, reranking, and collaborative reasoning) do not rely on a specific language. However, their effectiveness may vary across jurisdictions with different legal concepts, document styles, and judicial procedures.

Several components require jurisdiction-specific adaptation. In particular, the prompting templates and legal knowledge resources, such as law articles and charge definitions, must be localized to the statutes and terminology of the target jurisdiction. In addition, although `text-embedding-ada-002` supports multilingual retrieval, a domain-specific encoder for the target language, such as Legal-BERT for English legal texts, may better capture jurisdiction-specific legal semantics.

In future work, we will evaluate CoRe on legal judgment prediction benchmarks from other jurisdictions, including the European Court of Human Rights (ECtHR) dataset and the Swiss Judgment Prediction task. We will further investigate how retrieval quality, label constraints, and collaborative reasoning transfer across legal systems.

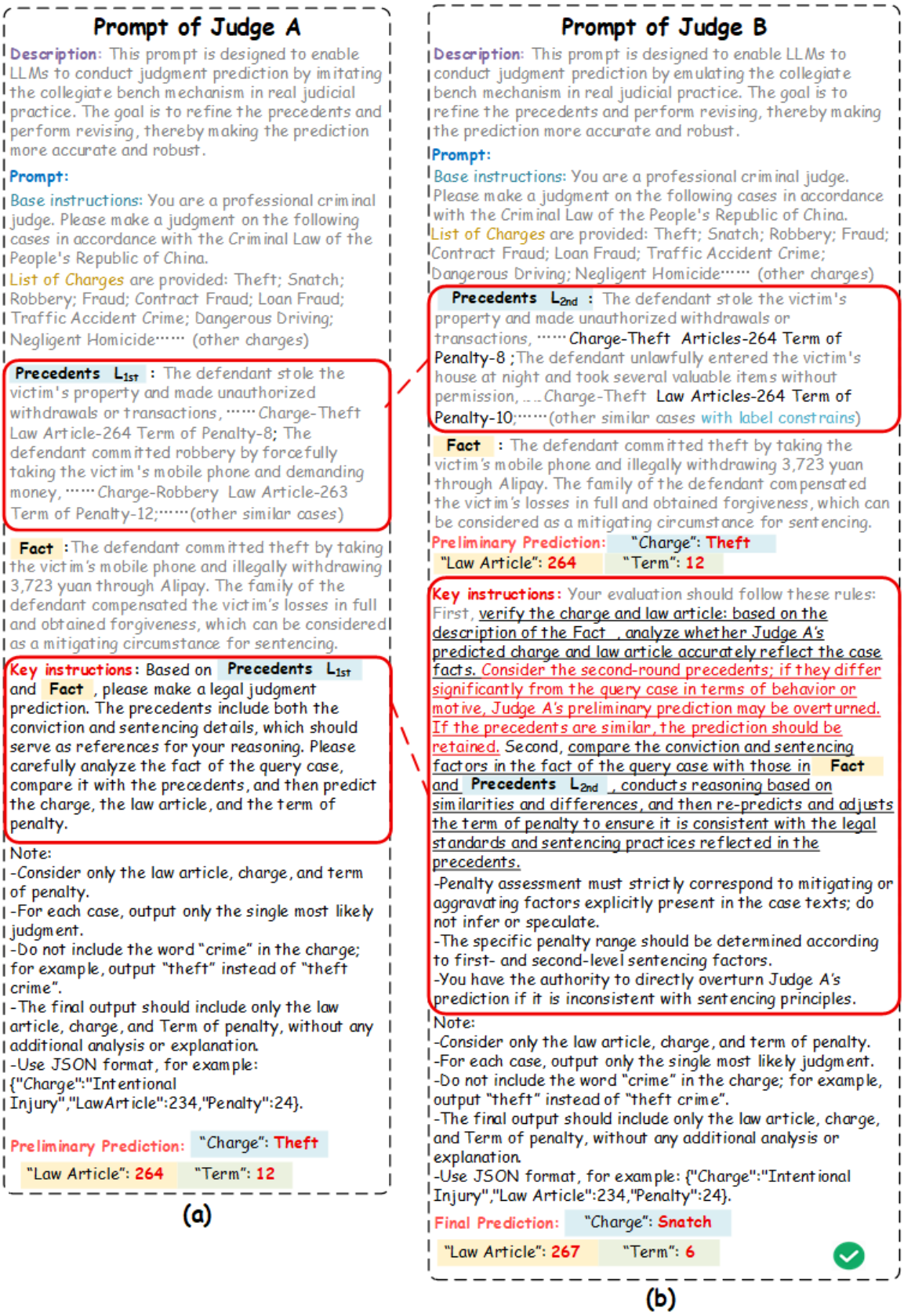


Figure 10: The prompt of collaborative reasoning