

Beyond Case-Level Similarity: Multi-Defendant Legal Case Retrieval via Individual-Aware Reasoning

Yi Yang^{1*}, Ruize Ma^{1*}, Yi Feng^{1†}, Chuanyi Li¹, Vincent Ng²

¹State Key Laboratory for Novel Software Technology, Nanjing University, China

²Human Language Technology Research Institute, University of Texas at Dallas, USA

yi.yang@smail.nju.edu.cn, fy@nju.edu.cn, vince@hlt.utdallas.edu

Abstract

Legal Case Retrieval (LCR) aims to identify cases that are legally similar to a given query case. Existing research has largely recast LCR as the task of modeling the textual similarity between the query case and a historical case. While this approach can handle single-defendant cases fairly easily, such surface-level matching proves insufficient for multi-defendant cases, which typically requires modeling and aligning defendants' roles and judgment outcomes in different cases. To address the challenges associated with Multi-defendant LCR (Multi-LCR), we (1) present Multi-ELAM and Multi-LeCaRDv2, two multi-defendant datasets we create by enriching widely used benchmarks with defendant-level annotations¹, and (2) introduce a novel role- and judgment-aware framework for Multi-LCR. Experimental results show that our approach outperforms state-of-the-art baselines, highlighting its effectiveness for Multi-LCR.

1 Introduction

Given a query case, Legal Case Retrieval (LCR) seeks to identify legally similar cases from large judicial corpora (Feng et al., 2024). LCR provides a foundation for downstream tasks such as *legal judgment prediction* (Wu et al., 2023b), which involves predicting the outcomes of a legal case (e.g., the charge, the term penalty, and the law articles), and *legal question answering* (Martinez-Gil, 2023), which involves answering a legal question in the form of a short phrase/paragraph. LCR is critical for promoting judicial consistency and fairness, and increasing work efficiency for legal professionals.

Most existing studies on LCR formulate the task as a case-level similarity matching problem (Saravanan et al., 2009; Zeng et al., 2005; Sun et al.,

*Equal contribution.

†Corresponding author.

¹Our datasets and code are available on <https://github.com/yiy1x/MultiLCR>.

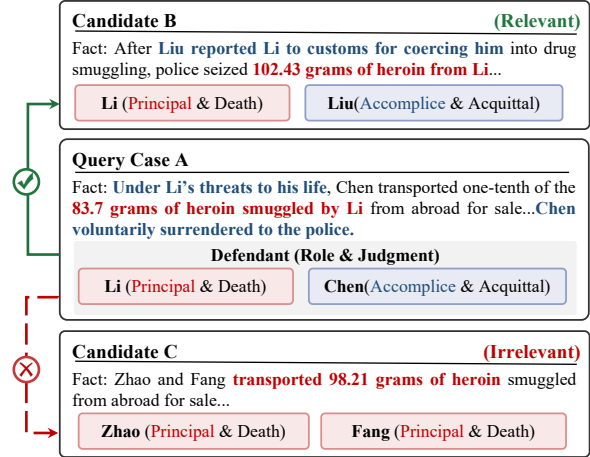


Figure 1: Example of Multi-LCR. Note that a LCR system takes as input the facts of a query as well as the facts of one or more candidate cases. In particular, the role and judgment of each defendant shown in this example are *not* part of the input to a LCR system: they are shown in this example to illustrate why cases A and C are dissimilar and why cases A and B are similar.

2023; Qin et al., 2024; Li et al., 2023, 2025), where similarity is measured based on the overall case content, such as fact descriptions (Jiang et al., 2019; Westermann et al., 2020) and key elements (Sun et al., 2023; Qin et al., 2024; Li et al., 2023, 2025). Though effective for single-defendant cases, this formulation proves insufficient for multi-defendant cases, where multiple defendants participate in the same factual context but may assume distinct legal roles, engage in different behaviors, and ultimately receive different judicial outcomes (Pan et al., 2019; Tang, 2009; Chen, 2016). As a result, two cases that appear similar at the factual level may differ significantly in their legal substance from the perspective of individual defendants.

To exemplify, consider Figure 1, which shows a query case (A) and two candidate cases (B and C), all of which are drug trafficking cases. In the query, Chen transported only one-tenth of the 83.7 grams of heroin smuggled by Li under explicit

threats to his life, and later voluntarily surrendered to the police, leading to his recognition as an accessory offender. In contrast, in candidate C, Zhao and Fang jointly transported 98.21 grams of heroin smuggled from abroad for sale, both being identified as principal offenders. Although these cases exhibit strong factual similarity in terms of crime type, conduct, and drug quantity, they differ fundamentally in the defendants’ legal roles (Fang is a Principal in C while Chen is an Accomplice in A) and corresponding judgments (Fang’s outcome is Death while Chen’s outcome is Acquittal). Hence, while cases A and C are factually similar, from the perspective of LCR they are not considered similar.

This observation reveals that case-level similarity alone is insufficient for Multi-defendant Legal Case Retrieval (Multi-LCR). In multi-defendant scenarios, effective retrieval requires additionally modeling and aligning the involved individuals across cases. Specifically, similarity must be evaluated at the *defendant* level by considering (i) each defendant’s Crime Contribution Portraits (CCPs), which captures their legal role in the case (e.g., principal, accomplice), and (ii) their Legal Judgment Outcomes (LJOs), e.g., charges, law articles, sentencing terms. Two cases should be regarded as similar only if their defendants can be aligned and exhibit consistent roles and judgment outcomes.

Addressing Multi-LCR introduces several key challenges. First, similarity modeling is complex, as Multi-LCR requires matching individual-level information, including each defendant’s CCPs and LJOs. This process demands (1) *defendant-level alignment*, which performs a structural comparison between the defendants in a query case and a candidate case; and (2) *holistic aggregation alignment*, which integrates fine-grained defendant-level alignments into a single similarity score that represents the overall similarity between the two cases. Second, there is a lack of annotations for training defendant-level alignment models. While datasets like ELAM (Yu et al., 2022) and LeCaRDv2 (Li et al., 2024) contain multi-defendant cases, these cases are only labeled with the overall case similarity, and do not provide explicit annotations for each defendant’s CCPs or LJOs.

To address Multi-LCR, we first construct two multi-defendant legal case datasets with annotations for each defendant’s CCPs and LJOs. We then design an “individual-first, case-second” retrieval approach where we (1) perform defendant-level understanding by predicting each defendant’s

roles and judgment outcomes from the case fact, and (2) incorporate role alignment and judgment consistency into the computation of overall case similarity. This integration of individual-level legal reasoning into the retrieval process enables more accurate similarity estimation for multi-defendant cases. Extensive experiments validate the importance of individual-aware similarity modeling.

2 Related Work

While early work treated LCR as a retrieval task using retrieval models such as TF-IDF and BM25 (Jones, 2004; Robertson and Zaragoza, 2009), later work introduced heuristic solutions, using legal ontologies to enable domain inferences (Saravanan et al., 2009) and developing models that expand semantic sub-elements of cases, such as issues and factors (Zeng et al., 2005), to refine the matching logic. With the advent of the deep learning era, LCR models have been developed to construct similarity from different perspectives. For instance, Law-Match (Sun et al., 2023) reconstructs case representations by incorporating law articles. GEAR (Qin et al., 2024) leverages judgment prediction as an auxiliary task to enhance retrieval. IOT-Match (Yu et al., 2022) and LCM-LAI (Xu et al., 2025) improve legal case matching via rationale extraction and law article prediction. SAILER (Li et al., 2023) exploits the logical flows across sections in legal documents. DELTA (Li et al., 2025) applies word alignment to isolate essential facts. Large Language Models (LLMs) have opened new opportunities for LCR. General LLMs, including the LLaMA family (Grattafiori et al., 2024), GPT series (OpenAI et al., 2024), and Gemini series (Comanici et al., 2025), show strong semantic understanding and transferability. Legal LLMs, such as ChatLaw (Cui et al., 2026), LawyerLLaMA (Huang et al., 2023), and DISC-Law (Yue et al., 2024), demonstrate competitive performance on the LawBench (Fei et al., 2024) benchmark.

By recasting LCR as a text similarity task, existing approaches retrieve factually similar cases, but lack the mechanisms to model the fine-grained, defendant-level alignment of each defendant’s role and LJO, a central gap our paper seeks to address.

3 Preliminaries

3.1 Problem Formulation

Following existing work (Sun et al., 2023; Li et al., 2025), we formulate Multi-LCR as Legal Case

Statistic	Multi-ELAM	Multi-LeCaRDv2
Training set pairs (LCM & LCR)	3,563	18,970
Testing set pairs (LCM)	551	4,735
Testing set queries (LCR)	150	160
Avg. candidates per query (LCR)	100	99.14
Avg. length (query)	1,147	544
Avg. length (candidate)	1,125	5,729
Avg. defendants	3.61	4.15

Table 1: Statistics on the two datasets.

Matching (LCM) and Legal Case Retrieval (LCR). LCM evaluates a model’s ability to assess similarity between an individual query–candidate case pair, and is typically formulated as a classification task. In contrast, LCR assesses a model’s ranking capability by requiring it to identify the top- k most relevant cases from a pool of candidates.

For the LCM setting, given a query case C_q and a candidate case C_c , the objective is to learn a matching function $f_{\text{match}}(C_q, C_c) \rightarrow z$, where $z \in Z = \{0, 1, 2\}$ represents human-annotated similarity labels. For example, 0 means mismatch, 1 means partially match, and 2 means match.

For the LCR setting, given a query case C_q , the task is to retrieve the top- k most legally relevant cases from a candidate ranking pool $C = \{C_1, \dots, C_N\}$. The objective is to learn a similarity function $\phi(C_q, C_c)$, where ϕ outputs a relevance score. The final retrieval result consists of the top- k candidates with the highest scores.

Note that a legal case is represented as $C = (\mathbf{F}, D)$, where $\mathbf{F}$ denotes the fact description and $D = \{d_1, \dots, d_n\}$ is a set of defendants involved. Each defendant d_i is represented by a representation $\mathbf{r}_i$ containing CCP $\mathbf{c}_i$ and LJO $\mathbf{l}_i$.

3.2 Dataset Construction

To provide annotations for CCPs and LJOs, we construct two datasets, Multi-ELAM and Multi-LeCaRDv2, by augmenting two LCR benchmarks, ELAM (Yu et al., 2022) and LeCaRDv2 (Li et al., 2024). Table 1 shows their statistics.

Annotating defendants and LJOs. We create Multi-ELAM from ELAM. Since ELAM does not provide annotations of the defendants involved and their judgment outcomes (charges, law articles, sentencing terms), we (1) obtain the full judgment texts by retrieving the original documents from *China Judgment Online*², (2) manually extract the defendants from the full texts; (3) automatically

extract the LJOs for each defendant via regular expressions, and (4) hand-verify the correctness of these LJOs.³ We create Multi-LeCaRDv2 from LeCaRDv2 in a similar fashion.⁴

Annotating CCPs. For CCPs, we first design a hierarchical scheme for CCPs, which is partially motivated by Kim et al. (2025). For a defendant d_i , its CCP $\mathbf{c}_i$ is designed as a structured representation rather than a single, flat label. This design is motivated by two considerations. First, coarse-grained tags (e.g., “Accomplice”) are insufficient to capture the fine-grained distinctions required for Multi-LCR, such as differentiating a “coerced accomplice” from a “profit-driven technical supporter”. Second, a defendant’s role may be inherently multi-faceted; for example, an individual may act as a “Principal” while simultaneously exhibiting “mitigating behaviors”. We define $\mathbf{c}_i$ as a set of role entries, $\{e_{i,1}, \dots, e_{i,K}\}$, where each entry e follows the four-level hierarchical schema summarized in Figure 2, comprising **Role**, **Tier**, **Patterns**, and **Attributes**. Specifically, the hierarchy consists of: (1) **Role Type** (R), which provides a coarse-grained categorization of defendants into functional roles, including *Principal*, *Accessory*, and *Mitigating* roles; (2) **Contribution Tier** (T) **and Weight** (w), where each role is mapped to a tier ranging from T1 (leading contribution) to T5 (mitigating contribution). To quantify the tiers, we assign each tier a numerical weight reflecting its effect on sentencing, indicating whether it increases or decreases the sentence term. For example, we set $w = 1.0$ for T1 and $w = -0.5$ for T5, where positive values increase the sentencing term and negative values decrease it; (3) **Behavioral Patterns** (P), which instantiate each tier through concrete actions (e.g., *Planner*, *Organizer*, *Confession of Key Facts*), grounding abstract roles in factual behavior; and (4) **Attributes** (Phase & Motivation), which further contextualize each pattern by specifying the temporal stage (*pre-crime*, *post-crime*) and the defendant’s subjective intent (e.g., *profit-seeking*, *remorse*).⁵ This structure encodes the defendant’s multi-faceted nature, providing a rich, qualitative semantic basis for the downstream role encoding.

Given the CCP scheme, we use DeepSeek-V3 to

³Steps 2 and 4 involve human intervention. See Appendix A for details.

⁴Before resolving disagreements, we computed the inter-annotator agreement (IAA) on our defendant and LJO annotations. Details can be found in Appendix B.

⁵See Appendix C for details of the hierarchical scheme. The IAA of CCP annotations can be found in Appendix B.

²<https://wenshu.court.gov.cn/>

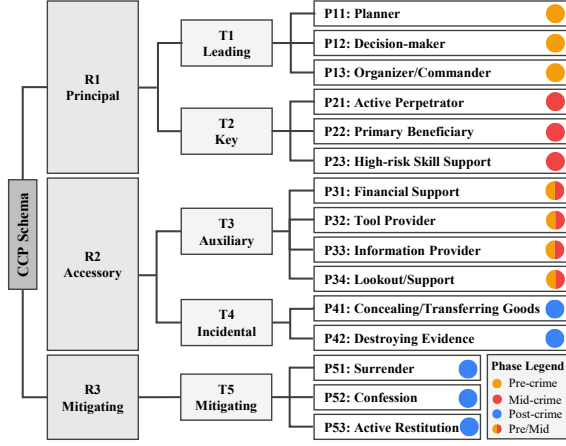


Figure 2: Hierarchical schema for CCP. While Attributes are composed of Phase and Motivation, only Phase is shown in this figure due to space limitations. The Motivation categories can be found in Appendix L.1.

generate defendant-level CCPs under several constrained prompts, ensuring consistency with the hierarchy (e.g., T5 corresponds to R3). Since manually verifying CCPs for all defendants is infeasible, we randomly sample a subset of cases for human annotation checks to ensure the quality and reliability of the automatically generated CCPs.⁶

4 Approach

Figure 3 overviews our Multi-LCR framework, DSLMLCR, with its components detailed below.

4.1 Multi-LLM Debate

To generate CCPs and LJOs for the query and candidate cases, we employ a multi-step debate trick (Tao et al., 2025) using three powerful LLMs (DeepSeek-V3, Qwen3-235B-A22B, and Kimi-K2) to mitigate single-model bias and leverage diverse reasoning capabilities.⁷

Stage 1: Independent analysis. All models are employed to independently propose initial analyses. The objective is to generate a broad pool of CCPs and LJOs by capturing diverse reasoning paths. To ensure this reasoning is structured and interpretable, each model follows a unified two-step chain-of-thought process. First, it analyzes the raw fact description $\mathbf{F}$ to generate a detailed CCPs $\mathbf{c}_i$ for each defendant. Second, it uses this generated $\mathbf{c}_i$ as an explicit reasoning scaffold, combining it with the original facts $\mathbf{F}_q$, to predict the final LJO

$\mathbf{l}_i$. This method ensures that predicting LJOs is not merely a black-box prediction but is explicitly grounded in the structured analysis of the CCP.

Stage 2: Summarization. The outputs from the three models are processed in an initial convergence step: DeepSeek-V3 serves as the summarizer whose role is to synthesize divergent analyses into a single, coherent draft by evaluating points of agreement and disagreement among the initial hypotheses. To ensure legal grounding, the summarizer is provided with the texts of the law articles referenced in the Stage 1 LJO outputs. The result of this stage is a draft of the CCPs and LJOs.

Stage 3: Peer critique. This draft is then passed to the other two models, Qwen3-235B-A22B and Kimi-K2, for a crucial adversarial check. This step uses different models than the summarizer to prevent confirmation bias. The core of this stage is a rigorous critique process: their prompts enforce that any proposed revisions, particularly those affecting sentence severity, must be rigorously justified by specific evidence from the case facts or explicit provisions within the same legal texts provided in Stage 2. Each model then outputs a structured critique, detailing its agreement or disagreement and offering these legally-grounded revisions.

Stage 4: Final decision. Finally, the consolidated draft (from Stage 2) and the two structured critiques (from Stage 3) are passed back to DeepSeek-V3 for a final resolution. This model’s task is to perform a final critical analysis, weighing the draft against the critiques. It resolves any remaining conflicts by strictly adhering to the specific evidence in the case facts and the same legal texts provided in Stage 2. The output of this stage is the representation $\mathbf{r}_i = (\mathbf{c}_i, \mathbf{l}_i)$ for each defendant d_i .

4.2 Individual-Aware Reasoning

Next, we integrate defendant-level features into case similarity modeling.

Textual Encoding for Fact Description and Defendant d_i . To embed the fact description $\mathbf{F} \in \{\mathbf{F}_q, \mathbf{F}_c\}$, we employ a shared pre-trained encoder (SAILER) to map it into dense embeddings. Specifically, for the text sequence representing the case facts, s , we derive its representation $\mathbf{e}_F \in \mathbb{R}^d$ via mean pooling over the encoder’s hidden states:

$$\mathbf{e}_F = \text{MeanPooling}(\text{Encoder}(s)) \quad (1)$$

We derive the quantitative LJO embedding $\mathbf{e}_{LJO,i}$ for each defendant d_i using the same en-

⁶Appendix D presents the dataset construction procedure.

⁷See Appendix M for detailed prompts.

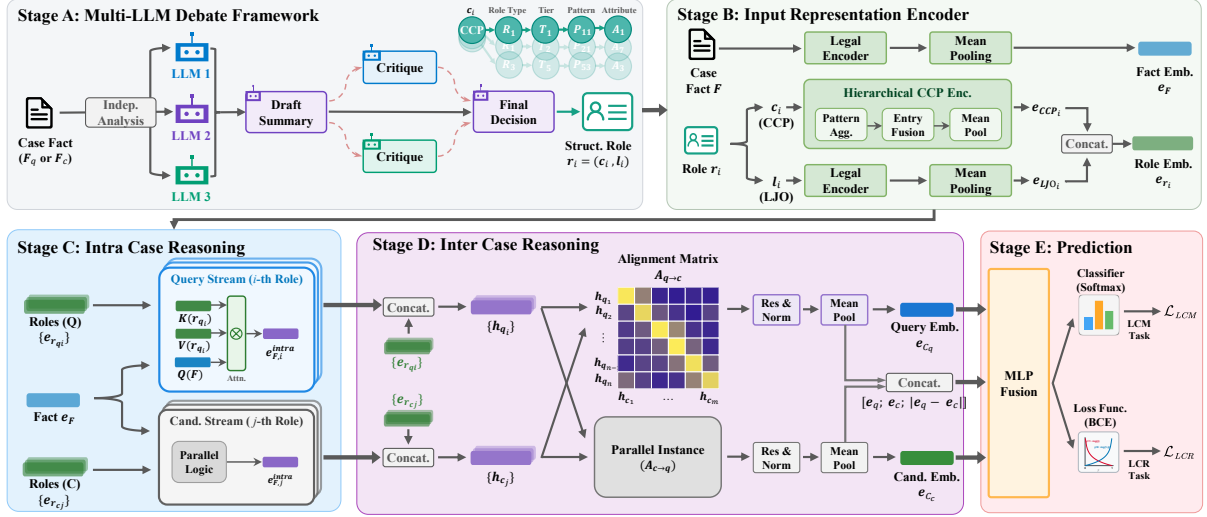


Figure 3: The overall architecture of the proposed DS-MLCR framework.

coding method, where the LJOs are initially represented as a text sequence comprising the texts of law articles, charges, and sentencing terms.

Hierarchical Role Encoding. We represent the qualitative CCP c_i for defendant d_i using a bottom-up aggregation module. A pretraining embedding matrix is first obtained to encode CCP-related information. For each role entry e_{ij} , we first synthesize its behavioral patterns and attributes (phase and motivation) into a pattern-aware vector $\mathbf{v}_{ij} \in \mathbb{R}^d$ via an attention-based pooling:

$$\mathbf{v}_{ij} = \text{Attn} \left(\left\{ \text{MLP} \left([\mathbf{e}_{ijk}^{\text{patt}}; \mathbf{e}_{ijk}^{\text{phase}}; \mathbf{e}_{ijk}^{\text{moti}}] \right) \right\}_{k=1}^P \right) \quad (2)$$

where P is the number of patterns in entry j . Then, these entry-level vectors are fused with the role type, tier embeddings, and the scalar weight w_{ij} to form the defendant-level CCP representation:

$$\mathbf{e}_{CCP,i} = \frac{1}{N_r} \sum_{j=1}^{N_r} \text{MLP} \left([\mathbf{e}_{ij}^{\text{type}}; \mathbf{e}_{ij}^{\text{tier}}; \mathbf{v}_{ij}; w_{ij}] \right) \quad (3)$$

where N_r is the total number of entries for defendant d_i . Finally, the qualitative and quantitative features are concatenated to form the comprehensive defendant embedding $\mathbf{e}_{r_i} \in \mathbb{R}^{d_r}$ (where $d_r = d_c + d$):

$$\mathbf{e}_{r_i} = [\mathbf{e}_{CCP,i}; \mathbf{e}_{LJO,i}] \quad (4)$$

Intra-case reasoning module. We utilize an intra-case reasoning module that integrates factual and defendant semantics to produce individual-aware fact representations. Specifically, we define

an augmented representation $\mathbf{x}_i = [\mathbf{e}_F; \mathbf{e}_{r_i}]$ and compute the contextualized representation $\mathbf{e}_{F,i}^{\text{intra}}$ via a cross-attention mechanism:

$$\mathbf{e}_{F,i}^{\text{intra}} = \text{softmax} \left(\frac{(\mathbf{W}_q \mathbf{e}_F)(\mathbf{W}_k \mathbf{x}_i)^\top}{\sqrt{d}} \right) (\mathbf{W}_v \mathbf{x}_i) \quad (5)$$

where $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v$ are learnable projection matrices. This module ensures the resulting embedding $\mathbf{e}_{F,i}^{\text{intra}} \in \mathbb{R}^d$ captures the specific legal involvement of defendant d_i by integrating factual semantics with individual roles.

Inter-case reasoning module. To model the cross-case defendant alignment, we first construct a unified representation $\mathbf{h}_i = [\mathbf{e}_{F,i}^{\text{intra}}; \mathbf{e}_{r_i}]$ for each defendant. Let $\mathbf{H}_q \in \mathbb{R}^{n \times d_h}$ and $\mathbf{H}_c \in \mathbb{R}^{m \times d_h}$ be the defendant matrices for the query and candidate cases. We employ a bidirectional Cross-Attention mechanism to align individuals across cases:

$$\tilde{\mathbf{H}}_q = \text{LayerNorm} \left(\mathbf{H}_q + \text{Softmax} \left(\frac{\mathbf{Q}_q \mathbf{K}_c^\top}{\sqrt{d_h}} \right) \mathbf{V}_c \right) \quad (6)$$

where $\mathbf{Q}_q, \mathbf{K}_c, \mathbf{V}_c$ are linear projections of $\mathbf{H}_q$ and $\mathbf{H}_c$. The updated representation $\tilde{\mathbf{H}}_c$ is obtained symmetrically. This process explicitly matches defendants with similar roles and LJOs (e.g., aligning principal offenders) between cases.

To synthesize a global similarity measure, we aggregate the aligned individual representations into case-level embeddings $\mathbf{e}_{Cq}$ and $\mathbf{e}_{Cc}$ via mean pooling over $\tilde{\mathbf{H}}_q$ and $\tilde{\mathbf{H}}_c$. The final prediction $\mathbf{z}_{\text{out}}$ is computed by an MLP that takes the case-level interaction features as input:

$$\mathbf{z}_{\text{out}} = \text{MLP}([\mathbf{e}_{Cq}; \mathbf{e}_{Cc}; |\mathbf{e}_{Cq} - \mathbf{e}_{Cc}|]) \quad (7)$$

where $\mathbf{z}_{\text{out}}$ yields the logits for multi-class matching (LCM) or binary relevance (LCR).

4.3 Model Training

The training procedure consists of two stages:

Stage 1: CCP Module Pretraining. To bridge the modality gap between structured CCP data and unstructured legal texts, we pretrain a structured CCP encoder using contrastive learning to obtain high-quality role representations. We construct a positive training pair for each defendant d_i , comprising their structured CCP representation and a corresponding textual context $\mathbf{e}_{\text{text}_i}$, which is formed by concatenating the fact description embedding $\mathbf{e}_{F_i}$ with the LJO embedding $\mathbf{e}_{LJO_i}$, i.e., $\mathbf{e}_{\text{text}_i} = [\mathbf{e}_{F_i}; \mathbf{e}_{LJO_i}] \in \mathbb{R}^{2d}$.

To enable contrastive pretraining, we follow the InfoNCE framework (van den Oord et al., 2019) and design a dual-tower architecture where one branch encodes the concatenated fact and LJO texts in an unstructured manner using a frozen legal language model (SAILER), while the other maps structured data into role embeddings. Both outputs are linearly projected into a shared space and normalized with L_2 to facilitate representation alignment:

$$\hat{\mathbf{z}}_{\text{text}_i} = \frac{\mathbf{W}_{\text{text}} \mathbf{e}_{\text{text}_i}}{\|\mathbf{W}_{\text{text}} \mathbf{e}_{\text{text}_i}\|_2}, \quad \hat{\mathbf{z}}_{\text{CCP}_i} = \frac{\mathbf{W}_{\text{CCP}} \mathbf{e}_{\text{CCP}_i}}{\|\mathbf{W}_{\text{CCP}} \mathbf{e}_{\text{CCP}_i}\|_2} \quad (8)$$

where $\mathbf{W}_{\text{text}} \in \mathbb{R}^{d_p \times 2d}$ and $\mathbf{W}_{\text{CCP}} \in \mathbb{R}^{d_p \times d_c}$ are linear projection matrices for the textual and structured representations respectively, and $\hat{\mathbf{z}}$ denotes the L_2 -normalized vectors. Then, we optimize the model using a symmetric InfoNCE loss defined as:

$$\mathcal{L}_{\text{pretrain}} = -\frac{1}{2N} \sum_{i=1}^N \left(\log \frac{\exp(\hat{\mathbf{z}}_{\text{text}_i} \cdot \hat{\mathbf{z}}_{\text{CCP}_i}^\top / \tau)}{\sum_{j=1}^N \exp(\hat{\mathbf{z}}_{\text{text}_i} \cdot \hat{\mathbf{z}}_{\text{CCP}_j}^\top / \tau)} + \log \frac{\exp(\hat{\mathbf{z}}_{\text{CCP}_i} \cdot \hat{\mathbf{z}}_{\text{text}_i}^\top / \tau)}{\sum_{j=1}^N \exp(\hat{\mathbf{z}}_{\text{CCP}_i} \cdot \hat{\mathbf{z}}_{\text{text}_j}^\top / \tau)} \right) \quad (9)$$

where N is the batch size and τ is a learnable temperature parameter that controls the concentration level of the distribution. This loss aligns paired text and CCP representations while separating unpaired ones, enabling the CCP module to capture role semantics consistent with legal context.

Stage 2: End-to-End Optimization. Next, we perform end-to-end optimization of the DS-MLCR framework to jointly model factual and role-based

semantics for Multi-LCR. Following pre-training, the CCP module is frozen and the DS-MLCR is fine-tuned end-to-end. All model parameters are optimized by minimizing a task-specific loss function, $\mathcal{L}_{\text{task}}$. For LCM, we employ the cross-entropy loss; For LCR, we use the binary cross-entropy loss to discriminate the relevance of case pairs.⁸

5 Evaluation

5.1 Experimental Setup

Baselines. We compare our approach with three categories of SOTA baselines: (1) **legal-oriented retrieval methods** (e.g., SAILER (Li et al., 2023) and DELTA (Li et al., 2025)), which are specifically designed for legal case retrieval; (2) **general LLMs** (e.g., Llama3-Chinese-8B-Instruct (Cui et al., 2024), Qwen3-8B/32B (Yang et al., 2025), Gemini-2.5-flash (Comanici et al., 2025)), which possess strong reasoning capabilities for general tasks; and (3) **legal LLMs** (e.g., Llama3-Chinese-8B-Law-Lora, DeepSeek-R1-Chinese-Law (Guo et al., 2025), DISC-Law-7B (Yue et al., 2024)), which are optimized for legal tasks.⁹

Evaluation Metrics. For LCM, we employ Accuracy, Macro-Precision, Macro-Recall, and Macro-F1 as evaluation metrics. For LCR, we adopt Recall@K with $k \in \{1, 5, 10, 20, 30\}$, following the setup in (Li et al., 2024; Qin et al., 2024).

Implementation Details. We adopt SAILER¹⁰ as the encoder with a max input length of 512 tokens. The CCP embedding dimension is set to 128. Adam (Kingma and Ba, 2017) is employed for parameter updates. During CCP pretraining, the text encoder is kept frozen and the module is trained for 40 epochs with a batch size of 16 and a learning rate of 5×10^{-5} . The temperature τ in the InfoNCE loss is initialized to 0.07. For the downstream fine-tuning stage, we limit the max number of defendants per case to five and perform grid search over batch sizes $\{2, 4, 8, 16\}$ and learning rates $\{1 \times 10^{-6}, 3 \times 10^{-6}, 5 \times 10^{-6}, 1 \times 10^{-5}, 3 \times 10^{-5}\}$ to determine the optimal hyperparameters configuration. All results are averaged over three runs with the best configuration. Experiments were conducted on four NVIDIA A100 GPUs, totaling ~ 800 GPU hours.¹¹

⁸See Appendix H for a discussion on these losses and how using other losses for LCR impact its performance.

⁹See Appendix E for details on the baselines.

¹⁰https://huggingface.co/CSHaitao/SAILER_zh

¹¹See Appendix F for details on the settings.

Models	Legal Case Matching								Legal Case Retrieval									
	Multi-ELAM				Multi-LeCaRDv2				Multi-ELAM					Multi-LeCaRDv2				
	Acc	P	R	F1	Acc	P	R	F1	@1	@5	@10	@20	@30	@1	@5	@10	@20	@30
SAILER	75.9	67.1	67.0	67.0	64.5	47.3	35.9	38.1	7.4	30.3	50.0	74.3	86.2	1.8	7.8	14.5	29.0	43.3
DELTA	76.6	68.8	68.0	68.2	65.2	44.6	36.4	38.4	9.1	30.0	51.3	73.0	84.1	1.5	7.9	15.7	30.0	45.5
LawMatch-SAILER	75.9	67.0	68.1	67.5	66.3	46.7	36.3	38.5	6.9	30.1	50.4	78.2	88.3	2.0	10.3	18.8	34.2	48.6
LawMatch-DELTA	77.3	67.3	67.3	66.9	65.2	46.1	32.4	33.5	9.4	33.8	51.4	74.6	85.2	1.8	8.5	17.0	32.7	47.2
Llama3-8B-Instruct	31.9	33.9	35.2	30.1	22.4	23.2	22.3	16.9	2.1	9.2	15.3	27.3	36.1	1.2	5.0	10.4	20.5	30.8
Qwen3-8B	74.4	61.5	52.1	53.8	14.3	28.1	25.3	6.7	8.4	27.5	43.5	68.1	85.4	1.7	8.8	17.1	34.2	49.1
Qwen3-32B	78.4	68.3	66.8	66.8	16.6	35.3	34.4	18.7	10.1	37.4	60.0	86.9	94.1	2.1	9.5	17.6	32.0	46.7
Gemini-2.5-flash	78.4	69.1	67.7	68.0	22.9	36.2	34.9	22.5	11.3	38.8	62.9	87.8	94.9	2.2	10.3	19.7	35.5	49.7
Llama3-8B-LawLora	62.3	27.7	32.9	26.3	15.5	27.1	25.3	10.4	2.4	26.4	51.4	57.6	57.9	1.0	5.8	11.7	23.5	35.0
DeepSeek-R1-Law	71.1	58.9	56.4	57.4	20.3	28.4	32.4	19.7	7.6	34.7	62.2	78.0	85.9	1.9	8.4	16.0	31.1	44.0
DISC-Law-7B	71.1	54.5	61.7	54.7	52.4	28.3	33.2	30.1	10.8	39.3	57.6	82.0	94.4	2.2	9.5	18.1	34.2	48.6
DS-MLCR (Oracle)	80.0	72.8	72.5	72.6	73.4	61.5	53.8	56.6	12.1	42.1	69.3	90.2	95.9	2.6	11.9	21.9	38.8	54.3
DS-MLCR	79.1	70.4	71.7	70.8	69.1	52.6	41.3	41.4	11.4	40.1	67.2	89.2	95.1	2.5	10.8	20.1	36.6	52.0
-w/o CCP	77.9	67.7	68.5	67.8	68.0	46.5	41.0	40.7	10.9	36.8	65.6	86.8	94.4	2.2	9.2	17.3	32.4	46.2
-w/o LJO	76.2	69.2	69.7	69.4	65.3	47.3	37.0	37.3	10.8	36.2	52.1	78.9	93.2	1.5	6.5	13.3	24.4	35.3
-w/o INTRA	77.7	68.4	67.8	67.9	67.8	49.0	38.8	39.6	9.4	35.1	58.7	81.0	90.3	2.1	10.3	18.7	33.7	47.8
-w/o INTER	77.3	68.4	68.6	68.4	67.6	47.7	35.7	37.0	7.8	39.4	61.7	89.8	90.4	2.2	10.5	19.0	35.6	49.3

Table 2: Main results for Legal Case Matching and Legal Case Retrieval. For LCR, "@k" refers to the R@k results.

5.2 Results and Discussion

Table 2 shows the results of LCM (left) and LCR (right) on Multi-ELAM and Multi-LeCaRDv2. For LCM, specialized retrieval models (e.g., SAILER, DELTA) outperform general LLMs. Interestingly, general-purpose LLMs outperform legal LLMs on Multi-ELAM. In contrast, legal LLMs excel on Multi-LeCaRDv2, where tasks demand deeper legal reasoning. However, all LLMs face challenges with complex legal relevance, often producing errors when factual similarity does not correspond to legal similarity. Overall, both general and legal LLMs lag behind dedicated retrieval models, highlighting the inherent difficulty of Multi-LCR. DS-MLCR, however, achieves the best performance.

For LCR, Multi-LeCaRDv2 poses greater challenges due to more complex facts, leading to performance drops across models. However, DS-MLCR consistently surpasses competing methods.¹²

Ablations. We perform ablation experiments to assess the contributions of each component in DS-MLCR. As can be seen in the last four rows of Table 2, removing CCP (-w/o CCP) or LJO modules (-w/o LJO) leads to clear performance drops, highlighting the importance of integrating both qualitative and quantitative defendant information. Similarly, removing either the intra-case or inter-case reasoning module (-w/o INTRA, -w/o INTER) reduces retrieval performance, confirming the neces-

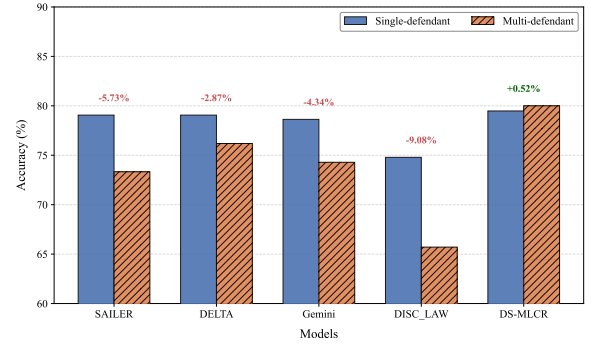


Figure 4: Performance comparison between single and multi-defendant cases on Multi-ELAM.

sity of the individual-aware reasoning.

Single vs. Multi-Defendant Comparison. We report the results of DS-MLCR separately on the single-defendant and multi-defendant subsets of the Multi-ELAM LCM test set. Compared with strong baselines (Figure 4), DS-MLCR yields much larger accuracy gains in the multi-defendant cases.

Upper Bound Analysis. We conduct oracle experiments to estimate the upper-bound performance of DS-MLCR, where ground-truth CCPs and LJOs are used. As shown in Table 2, the Oracle setting yields significant improvements, particularly for LCM on Multi-LeCaRDv2, where Macro-F1 rises from 41.44% to 56.64%, a +15.2% increase.

Generalizability We evaluate cross-jurisdiction generalizability of DS-MLCR on COLIEE

¹²See Appendix G for detailed experimental analysis.

Models	R@1	R@5	R@10	R@20	R@30
SAILER	10.3	30.0	44.9	63.6	75.7
DELTA	10.6	30.3	44.6	64.4	75.8
Qwen3-8B	7.1	24.3	35.9	49.4	56.6
Gemini-2.5-flash	10.3	32.0	46.2	62.9	74.1
DeepSeek-R1-Law	6.5	22.0	31.3	45.7	56.8
DISC-Law-7B	10.4	32.7	43.6	59.1	67.4
DS-MLCR	11.4	33.1	46.6	65.5	75.9

Table 3: Generalizability results on COLIEE 2026.

2026 (Goebel et al., 2024), which focuses on English case law retrieval in Canadian administrative law. This setting differs substantially from the Chinese criminal law context of Multi-ELAM and Multi-LeCaRDv2, posing both cross-lingual and cross-jurisdiction challenges. To adapt to COLIEE, we replace the CCP with the Judicial Review Portrait (JRP), which characterizes the conduct of government respondents in administrative decisions. Generated via the same multi-LLM debate framework, JRP captures respondent behavior through a structured hierarchy of *Role Types* (procedural, substantive, lawful) and *Error Tiers* ranging from procedural unfairness to lawful conduct, and is integrated into the input representation encoder. More generally, the architecture of our structured schema is universal across legal settings: both CCP and JRP organize individual-level legal information using the same hierarchical structure of Role Type, Tier, and Behavioral Pattern, while the definitions of these labels are jurisdiction-specific. For example, JRP contains three Role Types and four Error Tiers: R1 (Procedural Role) corresponds to T1 (Procedural Unfairness); R2 (Substantive Role) corresponds to T2 (Legal/Factual Errors) or T3 (Evidentiary Errors); and R3 (Lawful Role) corresponds to T4 (Lawful/Justified Conduct). Thus, adapting the schema to a new jurisdiction requires redefining the legal meaning of its roles, tiers, and patterns, while retaining the same hierarchical organization. As shown in Table 3, DS-MLCR achieves SOTA results across all Recall metrics.¹³ These findings demonstrate DS-MLCR’s ability to generalize to another jurisdiction.

Further Analysis. To examine the decision boundary between similar and dissimilar cases un-

¹³For the COLIEE experiments, we employ the English-pretrained versions of SAILER and DELTA to ensure linguistic consistency with the English case law dataset.

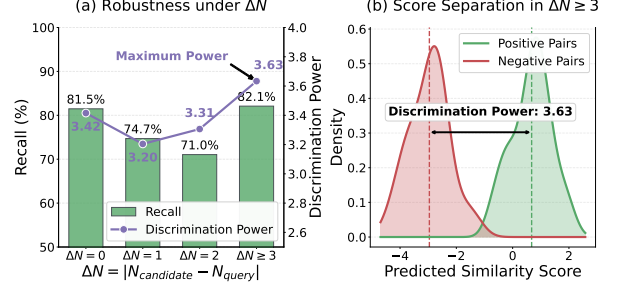


Figure 5: Performance Analysis across Defendant-Count Discrepancies (ΔN) on Multi-ELAM. (a) Robustness under ΔN . (b) Score separation in $\Delta N \geq 3$.

der varying defendant-count discrepancies ($\Delta N = |N_{\text{candidate}} - N_{\text{query}}|$), we analyze matched and mismatched pairs from the Multi-ELAM test set for LCM. These pairs are grouped according to ΔN into four categories: $\Delta N = 0$, 1, 2, and $\Delta N \geq 3$.

Large defendant discrepancies introduce a more challenging scenario for multiple-defendant case matching. As ΔN increases, the model must identify a small subset of relevant defendants from a larger pool of candidate participants. This substantially enlarges the alignment space and introduces additional distractor defendants, making it difficult for similarity models based on holistic representations to capture the correct correspondences.

Despite this increased difficulty, DS-MLCR maintains strong performance when ΔN is large. In Figure 5(a), the model achieves a recall of 82.1% for cases where $\Delta N \geq 3$, suggesting that the role- and judgment-aware mechanism can effectively identify the correct defendant correspondences while filtering out irrelevant participants.

While recall measures retrieval success, it does not reflect how clearly the model separates matched and mismatched pairs. So, we analyze the raw similarity logits using Kernel Density Estimation (KDE) (Parzen, 1962), which estimates the probability density of similarity scores and allows us to visualize their distributions under different ΔN ’s.

As shown in Figure 5(b), the score distributions of matched and mismatched pairs form distinct peaks with limited overlap, indicating that DS-MLCR effectively separates positive and negative pairs in the similarity space. We further quantify this separation using discrimination power, defined as the absolute difference between the expected similarity scores of the two distributions.

Notably, the discrimination power reaches its peak (3.63) when $\Delta N \geq 3$, exceeding the margins observed for $\Delta N = 1$ or 2. A possible ex-

Crime Type	n	Acc. (%)
Gambling	561	47.8
Embezzlement	327	48.3
Fraudulent Issuance of VAT Invoices	60	51.7
Illegal Business Operations	476	55.9
Intentional Damage to Property	359	62.1
Illegal Logging	90	65.6
Forgery of Official Documents/Seals	30	70.0
Drug Manufacturing and Trafficking	28	82.1

Table 4: Performance of DS-MLCR across different crime types on the Multi-LeCaRDv2 LCM test set.

Role Composition / Structure	n	Acc. (%)
Single-defendant	2,735	71.7
Co-principals (Multiple R1)	173	72.4
Hierarchical (R1 + R2/R3)	1,827	68.3
<i>Fine-grained subsets</i>		
- Adjacent Contribution Tiers	1,202	65.3
- Principal–Accessory Imbalance	1,016	62.7

Table 5: Performance of DS-MLCR under different CCP structural complexities on the Multi-LeCaRDv2 LCM test set.

planation is that small discrepancies often correspond to minor participant variations in multiple-defendant cases, where most defendants remain alignable across cases. In contrast, when ΔN is large, many candidate defendants cannot be aligned with the query case. The defendant-level alignment therefore imposes stronger penalties on mismatched cases, resulting in a larger separation between similar and dissimilar pairs.

Efficiency Analysis. The CCPs and LJOs of candidate cases can be pre-computed offline and cached, so the Multi-LLM Debate is only required for the query case during retrieval. We also employ three efficiency optimizations during Multi-LLM Debate inference. (1) Stage 1 (Independent Analysis) and Stage 3 (Peer Critique) execute their respective LLM calls in parallel. (2) Stage 2 (Summarization) synthesizes the outputs of the three proposer LLMs into a single coherent draft, allowing subsequent stages to operate on the draft rather than repeatedly processing all preceding outputs. (3) We use defendant-level streaming execution, passing Stage 2’s intermediate results to Stage 3 as soon as they are available.¹⁴

Crime-Type Error Analysis. As shown in Table 4, DS-MLCR performs strongly on Drug Manufacturing and Trafficking (82.1%) and Forgery of Official Documents/Seals (70.0%), but worse on

Gambling (47.8%), Embezzlement (48.3%), and Fraudulent Issuance of VAT Invoices (51.7%). Error cases reveal three recurring patterns. First, in financially complex crimes such as fraudulent VAT invoice issuance and illegal business operations, cases with similar legal conclusions may differ substantially in their fact descriptions, including transaction paths, intermediaries, and operational details. Second, different defendants may play functionally similar roles, such as facilitators or coordinators, but exhibit heterogeneous behaviors in the fact descriptions, making defendant-level alignment more difficult. Third, some defendants exhibit multifaceted roles across different stages of the crime, which remain challenging to capture even though CCPs support multiple role entries.

CCP Structural Complexity Analysis. In Table 5, we partition the 4,735 test pairs by overall role composition into single-defendant cases, co-principal cases with multiple R1 roles, and hierarchical cases containing R1 together with R2/R3 roles. DS-MLCR achieves comparable accuracy on single-defendant cases (71.7%) and co-principal cases (72.4%), suggesting that simple or symmetric role structures are relatively easier to handle. In contrast, accuracy decreases to 68.3% for hierarchical structures, where defendant-level alignment requires distinguishing different contribution levels within the same case. We additionally examine two fine-grained subsets involving adjacent Contribution Tiers and Principal–Accessory imbalance. Performance further decreases to 65.3% and 62.7%, respectively, indicating that fine-grained distinctions between closely related or asymmetric defendant roles remain challenging.

6 Conclusion

To address the challenges of Multi-LCR, we constructed two datasets, Multi-ELAM and Multi-LeCaRDv2, and presented DS-MLCR, a role- and judgment-aware framework for handling multi-defendant cases. We introduced a multi-LLM debate mechanism to generate defendant representations in terms of CCPs and LJOs and employ them for case similarity modeling through individual-aware reasoning. Extensive experiments demonstrated DS-MLCR’s effectiveness.¹⁵

¹⁵Further results that show that all stages of the multi-LLM debate mechanism contribute positively to performance are shown in Appendix I, and a case study that illustrates the importance of CCPs and LJOs is presented in Appendix K.

¹⁴More efficiency analyses are provided in Appendix J.

Limitations

DS-MLCR, similar to other LCR models, depends on a transformer-based encoder that struggles with overly long multi-defendant cases, where truncation may omit critical details. Furthermore, constructing role representations for every defendant and applying a multi-LLM debate ensures robustness but requires considerable computational resources.

Ethics Statement

Bias, Fairness, and Ethical Considerations.

The case data used in this study were sourced from publicly available judicial decision databases. To safeguard privacy, all sensitive details (e.g., defendant names, locations, etc.) were anonymized prior to being processed by externally deployed large language models. The primary objective of this research is to address the technical challenges in Multi-LCR, and our proposed model is intended as an auxiliary decision-support tool. We emphasize that any improvements in model performance are intended to assist human legal experts, not to replace them.

AI systems applied to legal data may inherit or amplify biases in historical judicial records, and their outputs may be misinterpreted. In the Multi-LCR setting, such biases may affect the inferred defendant roles and legal judgment outcomes, especially for underrepresented case types or role patterns in historical judicial records. Moreover, retrieval systems may introduce automation bias if legal professionals over-rely on highly ranked cases as precedents, even when important differences between defendants remain. Therefore, DS-MLCR should be used under human supervision rather than as an automated basis for legal judgment.

Although such risks cannot be fully mitigated, future work will focus on robustness, fairness, and transparency to better understand system behavior and limitations. We emphasize that any deployment of these systems in practice must be supported by proper professional supervision.

Future Societal Consequences. Advances in multi-defendant legal case retrieval can enhance legal research by providing more fine-grained, defendant-level comparisons in complex cases. By explicitly modeling heterogeneous roles and legal outcomes, such systems can help surface more relevant precedents while preserving distinctions that

are critical for human legal judgment. Importantly, these models are intended to function as decision-support tools within a human-in-the-loop framework, assisting legal professionals rather than replacing their expertise. By reducing the burden of large-scale case screening, they may allow practitioners to focus on higher-level legal reasoning and contextual interpretation. In this way, role-aware retrieval systems can contribute to more informed, transparent, and accountable legal analysis.

Terms of Use. This dataset is consistent with the terms of use and the privacy regulations of the original ELAM and LeCaRDv2 datasets upon which our work is based. All processed cases and annotations are intended strictly for non-commercial academic research. There is nothing about the composition of the dataset or the way it was collected, preprocessed, and labeled that might negatively impact its future research applications or infringe upon intellectual property rights.

Data Distribution. We open-sourced the data produced from this work at <https://github.com/yiy1x/MultiLCR>.

Acknowledgments

We thank the reviewers for their valuable comments on an earlier draft of this paper. This work was supported by National Natural Science Foundation of China (No. 62406139), State Key Laboratory for Novel Software Technology at Nanjing University (KFKT2026A09, ZZKT2026A43).

References

- Ping-Hsun Chen. 2016. Joinder of unrelated infringers as defendants in patent litigation under the jurisprudence of the united states district court for the eastern district of texas—a critical review. *Journal of the Patent and Trademark Office Society*, 98:155–181.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Maris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. *Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities*. *Preprint*, arXiv:2507.06261.
- Jiaxi Cui, Munan Ning, Zongjian Li, Hao Li, Yang Ya, Bohua Chen, Bin Ling, Yonghong Tian, and

- Li Yuan. 2026. [Chatlaw: A multi-agent legal assistant based on a role-aligned mixture-of-experts architecture](#). *Fundamental Research*.
- Yiming Cui, Ziqing Yang, and Xin Yao. 2024. [Efficient and effective text encoding for chinese LLaMA and Alpaca](#). *Preprint*, arXiv:2304.08177.
- Chenlong Deng, Zhicheng Dou, Yujia Zhou, Peitian Zhang, and Kelong Mao. 2024. [An element is worth a thousand words: Enhancing legal case retrieval by incorporating legal elements](#). In *Findings of the Association for Computational Linguistics*, pages 2354–2365.
- Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Alan Huang, Songyang Zhang, Kai Chen, Zhixin Yin, Zongwen Shen, Jidong Ge, and Vincent Ng. 2024. [Lawbench: Benchmarking legal knowledge of large language models](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7933–7962.
- Yi Feng, Chuanyi Li, and Vincent Ng. 2024. [Legal case retrieval: A survey of the state of the art](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 6472–6485.
- Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.
- Weifeng Ge, Weilin Huang, Dengke Dong, and Matthew R. Scott. 2018. [Deep metric learning with hierarchical triplet loss](#). In *Computer Vision - ECCV 2018 - 15th European Conference*, volume 11210 of *Lecture Notes in Computer Science*, pages 272–288.
- Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Julian Rabelo, Ken Satoh, and Masaharu Yoshioka. 2024. [Overview and discussion of the competition on legal information, extraction/entailment \(COLIEE\) 2023](#). *The Review of Socionetwork Strategies*, 18(1):27–47.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The Llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. [Deepseek-r1 incentivizes reasoning in llms through reinforcement learning](#). *Nature*, 645(8081):633–638.
- Quzhe Huang, Mingxu Tao, Chen Zhang, Zhenwei An, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. 2023. [Lawyer LLaMA technical report](#). *Preprint*, arXiv:2305.15062.
- Wanhong Huang, Yi Feng, Chuanyi Li, Honghan Wu, Jidong Ge, and Vincent Ng. 2024. [CMDL: A large-scale chinese multi-defendant legal judgment prediction dataset](#). In *Findings of the Association for Computational Linguistics*, pages 5895–5906.
- Jyun-Yu Jiang, Mingyang Zhang, Cheng Li, Michael Bendersky, Nadav Golbandi, and Marc Najork. 2019. [Semantic text matching for long-form documents](#). In *The World Wide Web Conference, WWW 2019*, pages 795–806.
- Thorsten Joachims. 2002. [Optimizing search engines using clickthrough data](#). In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142.
- Karen Spärck Jones. 2004. [A statistical interpretation of term specificity and its application in retrieval](#). *J. Documentation*, 60(5):493–502.
- Chaeun Kim, Jinu Lee, and Wonseok Hwang. 2025. [Legalsearchlm: Rethinking legal case retrieval as legal elements generation](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4521–4554.
- Diederik P. Kingma and Jimmy Ba. 2017. [Adam: A method for stochastic optimization](#). *Preprint*, arXiv:1412.6980.
- Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Yueyue Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2023. [SAILER: structure-aware pre-trained language model for legal case retrieval](#). In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1035–1044.
- Haitao Li, Qingyao Ai, Xinyan Han, Jia Chen, Qian Dong, and Yiqun Liu. 2025. [DELTA: pre-train a discriminative encoder for legal case retrieval via structural word alignment](#). In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence*, pages 27072–27080.
- Haitao Li, Yunqiu Shao, Yueyue Wu, Qingyao Ai, Yixiao Ma, and Yiqun Liu. 2024. [Lecardv2: A large-scale chinese legal case retrieval dataset](#). In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2251–2260.
- Hui Li, Jin Lu, Yuquan Le, and Jiawei He. 2022. [IACN: interactive attention capsule network for similar case matching](#). *Intelligent Data Analysis*, 26(2):525–541.
- Jorge Martinez-Gil. 2023. [A survey on legal question-answering systems](#). *Computer Science Review*, 48:100552.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin,

- Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. [GPT-4 technical report](#). *Preprint*, arXiv:2303.08774.
- Sicheng Pan, Tun Lu, Ning Gu, Huajuan Zhang, and Chunlin Xu. 2019. [Charge prediction for multi-defendant cases with multi-scale attention](#). In *Computer Supported Cooperative Work and Social Computing - 14th CCF Conference*, pages 766–777.
- Emanuel Parzen. 1962. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076.
- Weicong Qin, Zelin Cao, Weijie Yu, Zihua Si, Sirui Chen, and Jun Xu. 2024. [Explicitly integrating judgment prediction with legal document retrieval: A law-guided generative approach](#). In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2210–2220.
- Stephen E. Robertson and Hugo Zaragoza. 2009. [The probabilistic relevance framework: BM25 and beyond](#). *Found. Trends Inf. Retr.*, 3(4):333–389.
- M. Saravanan, Balaraman Ravindran, and S. Raman. 2009. [Improving legal information retrieval using an ontological framework](#). *Artificial Intelligence and Law*, 17(2):101–124.
- Zhongxiang Sun, Jun Xu, Xiao Zhang, Zhenhua Dong, and Ji-Rong Wen. 2023. [Law article-enhanced legal case matching: A causal learning approach](#). In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1549–1558.
- Zheng Sophia Tang. 2009. Multiple defendants in the european jurisdiction regulation. *European Law Review*, 34:80–102.
- Mingxu Tao, Dongyan Zhao, and Yansong Feng. 2025. [Chain-of-discussion: A multi-model framework for complex evidence-based question answering](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 11070–11085.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. [Representation learning with contrastive predictive coding](#). *Preprint*, arXiv:1807.03748.
- Hannes Westermann, Jaromír Savelka, and Karim Benyekhlef. 2020. [Paragraph similarity scoring and fine-tuned BERT for legal information retrieval and entailment](#). In *New Frontiers in Artificial Intelligence - JSAI-isAI 2020 Workshops*, pages 269–285.
- Shiguang Wu, Zhongkun Liu, Zhen Zhang, Zheng Chen, Wentao Deng, Wenhao Zhang, Jiyuan Yang, Zhitao Yao, Yougang Lyu, Xin Xin, Shen Gao, Pengjie Ren, Zhaochun Ren, and Zhumin Chen. 2023a. fuzi.mingcha. <https://github.com/irlab-sdu/fuzi.mingcha>.
- Yiquan Wu, Siying Zhou, Yifei Liu, Weiming Lu, Xiaozhong Liu, Yating Zhang, Changlong Sun, Fei Wu, and Kun Kuang. 2023b. [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.
- Nuo Xu, Pinghui Wang, Zi Liang, Junzhou Zhao, and Xiaohong Guan. 2025. [How vital is the jurisprudential relevance: Law article-intervened legal case retrieval and matching](#). *ACM Transactions on Information Systems*, 43(4):85:1–85:32.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Weijie Yu, Zhongxiang Sun, Jun Xu, Zhenhua Dong, Xu Chen, Hongteng Xu, and Ji-Rong Wen. 2022. [Explainable legal case matching via inverse optimal transport-based rationale extraction](#). In *the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 657–668.
- Shengbin Yue, Shujun Liu, Yuxuan Zhou, Chenchen Shen, Siyuan Wang, Yao Xiao, Bingxuan Li, Yun Song, Xiaoyu Shen, Wei Chen, Xuanjing Huang, and Zhongyu Wei. 2024. [Lawllm: Intelligent legal system with legal reasoning and verifiable retrieval](#). In *Proceedings of the 29th International Conference on Database Systems for Advanced Applications*, pages 304–321.
- Yiming Zeng, Ruili Wang, John Zelezniak, and Elizabeth A. Kemp. 2005. [Knowledge representation for the intelligent legal case retrieval](#). In *Proceedings of the 9th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, pages 339–345.

A Human Involvement in Annotating Defendants, LJOs, and CCPs

To ensure the authority and consistency of legal semantic annotations, we have established a rigorous human involvement mechanism.

Annotator Recruitment and Compensation. Our annotation team comprises three postgraduate law students (serving as primary annotators) and one senior lawyer (serving as the legal expert and adjudicator).

- **Recruitment Criteria:** The selection of annotators was strictly based on their proficiency in the Chinese criminal law system.

- **Compensation:** The postgraduate students received research assistantship stipends proportional to their workload, whereas the senior lawyer was compensated via professional consultation. All participants provided informed consent.

Training and Guidelines. Prior to the formal annotation phase, all team members participated in a systematic training session.

- **Training Objectives:** The session aimed to standardize objective criteria for defendant extraction and for verifying LJOs and CCPs.
- **Annotation Guidelines:** We developed a comprehensive operation manual (detailed in Appendix D), which mandates that annotators strictly verify all information against the raw text of the judgment documents.

Annotation Procedure. Steps 2 and 4 of our annotation procedure required direct human intervention. For Step 2, the students independently extracted the defendants. Disagreements were resolved via discussion facilitated by the lawyer. In Step 3, the LJOs were extracted for every defendant agreed upon in Step 2. In Step 4, the students independently verified the LJOs, and disagreements were resolved through discussion facilitated by the lawyer. Additionally, this team is also responsible for the human verification of the LLM-generated CCPs, following a similar independent-review-and-adjudication.

B Inter-Annotator Agreement for the Defendant Annotations

The inter-annotator agreement (IAA) for the defendant annotations was measured using pairwise precision and recall between two annotators, averaged over the three pairs of annotators. For Multi-ELAM, the average precision and recall were 0.92 and 0.89, respectively. For Multi-LeCaRDv2, the corresponding values were 0.91 and 0.87. For LJO annotations (i.e., law articles, charges and sentencing terms), we evaluated IAA using Fleiss’ Kappa (Fleiss, 1971), yielding high consistency across both datasets: Multi-ELAM—Law Article 0.82, Charge 0.79, Sentencing Terms 0.76; Multi-LeCaRDv2—Law Article 0.80, Charge 0.77, Sentencing Terms 0.74. For CCP verification (i.e., Role Type, Contribution Tier, and Behavioral Patterns),

we calculated the IAA on a randomly sampled subset of 1,000 cases (500 from Multi-ELAM and 500 from Multi-LeCaRDv2). For defendants involved in multiple role entries, the three human auditors performed independent judgments for each entry separately to ensure structural alignment. We evaluated agreement across our hierarchical schema using Fleiss’ Kappa, yielding high consistency: Multi-ELAM—Role Type 0.84, Contribution Tier 0.78, Behavioral Patterns 0.75; Multi-LeCaRDv2—Role Type 0.81, Contribution Tier 0.76, Behavioral Patterns 0.74.

C Details of the Hierarchical Schema

- **Level 1: Role Type (R).** This is the highest level of abstraction, categorizing the defendant into functional prototypes: $R1$ *Principal Role* (e.g., primary actor), $R2$ *Accessory Role* (e.g., supporting actor), and $R3$ *Mitigating Role* (e.g., factors reducing involvement).
- **Level 2: Contribution Tier (T) and Weight (w).** To quantify the degree of involvement, each Role Type is mapped to a specific Contribution Tier with a corresponding default numerical **weight** (w). These weights serve as the baseline for the downstream encoder:
 - $T1$: *Leading Contribution* ($w = 1.00$)
 - $T2$: *Key Contribution* ($w = 0.75$)
 - $T3$: *Auxiliary Contribution* ($w = 0.50$)
 - $T4$: *Incidental Contribution* ($w = 0.30$)
 - $T5$: *Mitigating Contribution* ($w = -0.50$)

The weight w is a variable value; while it defaults to the tier-specific baseline, it can be adjusted (e.g., within ± 0.05) based on the specific role entry to enable fine-grained differentiation of legal impact.

- **Level 3: Behavioral Patterns (P).** Under each tier, specific criminal acts are defined. For instance, a $T1$ contribution may manifest as $P11$: *Planner* or $P13$: *Organizer*, while a $T5$ contribution includes $P52$: *Confession of Key Facts*. This level grounds the abstract role in concrete factual descriptions.
- **Level 4: Attributes (Phase & Motivation).** Finally, two attributes add context to the patterns:

1. **Phase:** Specifies the temporal stage of the behavior (e.g., *Pre-crime*, *Post-crime*).
2. **Motivation:** Captures subjective intent using a controlled vocabulary (e.g., *Profit-seeking*, *Remorse*; see Appendix L.1 for the complete list).

To exemplify, consider the real-world case of defendant Xing (see Appendix L.2 for a description of this case), his CCP c_i is represented as a composite of two entries: (1) an R1-T1 entry ($w = 1.0$) characterized by *P11: Planner* (driven by *Profit-seeking*); and (2) an R3-T5 entry ($w = -0.5$) characterized by *P52: Confession* (driven by *Remorse*).

D Dataset Construction Details

Regex-driven Extraction of golden LJO. To extract the golden LJO, we developed an automated parsing pipeline built upon a sophisticated regular expression (regex) framework. The system precisely identifies and extracts critical legal information following the "Judgment Outcome" section. The core logic consists of three primary dimensions:

- **Entity Alignment:** Based on the manually extracted list of defendant entities, the pipeline performs targeted matching within the "Court's View" and "Judgment Outcome" sections. This alignment mechanism ensures that complex criminal behavior descriptions are accurately mapped to the predefined defendant entities, establishing a clear behavior-individual correlation.
- **Legal Judgment Outcome (LJO) Extraction:** With the defendants already identified, the system automatically extracts the charges, applicable law articles, and penalties for each defendant. To facilitate quantitative computation and structured storage, sentencing outcomes are first mapped into numerical fields, specifically including public surveillance, criminal detention, and fixed-term imprisonment (all normalized into total months), alongside categorical indicators for life imprisonment and the death penalty. For downstream textual encoding, these numerical values are deterministically formatted back into standardized natural language templates (e.g., "Sentenced to fixed-term imprisonment of six months")

before being concatenated with the charges and law articles.

- **Processing of Fines and Combined Punishments:** To address the diverse linguistic expressions in legal texts, the framework incorporates normalization modules to handle complex fine format conversions (e.g., converting Chinese numeric characters into standard Arabic numerals). Furthermore, for cases involving "combined punishment for multiple crimes", the pipeline recursively parses sentences for individual crimes as well as the final executed sentence, ensuring the logical integrity and accuracy of the judicial data.

While the regex-driven pipeline captures the majority of standardized judicial patterns, all results are subjected to a rigorous manual verification process to ensure the resulting golden LJO labels achieve the highest possible fidelity. During this manual audit, annotators were also instructed to identify and redact any inadvertently collected offensive language or lingering personal identifiers to ensure the dataset's safety and privacy. The verification follows a multi-stage protocol:

- **Independent Labeling:** Three postgraduate law students independently review and verify the outputs generated by the regex-driven pipeline.
- **Dispute Resolution:** In instances where discrepancies or disputes arise among the three annotators, a senior legal expert leads a collective discussion with the students to adjudicate the final outcome and reach a consensus.
- **Inter-annotator Agreement:** Before resolving disagreements, we computed the inter-annotator agreement (IAA) on our defendant and LJO annotations. The IAA for the defendant annotations was measured using pairwise precision and recall between two annotators, averaged over the three pairs of annotators. For Multi-ELAM, the average precision and recall were 0.92 and 0.89, respectively. For Multi-LeCaRDv2, the corresponding values were 0.91 and 0.87. For LJO annotations (i.e., law articles, charges and sentencing terms), we evaluated IAA using Fleiss' Kappa, yielding high consistency across both datasets: Multi-ELAM—Law Article 0.82,

Charge 0.79, Sentencing Terms 0.76; Multi-LeCaRDv2—Law Article 0.80, Charge 0.77, Sentencing Terms 0.74.

Guidelines for LJO Verification. Annotators were tasked with validating the regex-driven pipeline’s outputs against the full judgment texts through a rigorous verification process. First, a completeness check was required to verify that every defendant listed in the “Decision” section was present, necessitating the addition of missing defendants or the removal of incorrectly extracted non-defendant entities, such as witnesses. Subsequently, annotators performed attribute validation by cross-referencing extracted fields with the raw text. This involved ensuring standard nomenclature for charges (explicitly marking acquittals), validating law article citations while excluding procedural laws, and confirming the quantitative precision of sentencing terms, including probation periods and fines. Throughout this process, the raw judgment text served as the absolute ground truth; any mismatch between the regex output and the document required correction based strictly on the original text.

LLM-driven Generation and Verification of golden CCP. To construct the golden CCP, we utilize DeepSeek-V3 to generate defendant-level profiles based on our hierarchical legal schema. The generation is governed by a Logic-Constrained Generation protocol, employing specialized prompts (detailed in Table 10) to enforce structural consistency between tiers. These constraints mandate specific legal mappings. For instance, a Principal Role (*R1*) must correspond to either a Leading Contribution (*T1*) or a Key Contribution (*T2*) and their associated patterns (e.g., *P11*: Planner, *P21*: Active Perpetrator), while a Mitigating Role (*R3*) is strictly aligned with a Mitigating Contribution (*T5*). After generation, we apply an additional rule-based regularization check enforcing the same logical constraints. If violations are detected, the CCP is regenerated by the LLM under identical constraints, repeated for up to three iterations. Then, we perform a rigorous human quality audit on a randomly sampled subset of 1,000 (500 per dataset) cases to ensure the reliability of the automated generation:

- **Human Verification:** Three postgraduate law students act as auditors to independently review the LLM-generated CCPs, checking for

factual alignment with the case facts and adherence to the logical hierarchy.

- **Correction of Misalignments:** In cases where the LLM fails to capture a nuanced behavioral pattern or violates a structural constraint, the students manually rectify the CCP entry to match the judicial reality.
- **Expert-led Adjudication:** Any disputes regarding the corrections are referred to a senior legal expert, who leads a collective discussion with the students to finalize the Golden CCP.
- **Inter-annotator Agreement:** Before resolving disagreements, we computed the inter-annotator agreement on the LLM-generated CCP annotations for this 1,000-case subset (500 per dataset). To ensure structural alignment, auditors performed independent judgments for each role entry separately, particularly for defendants with multi-faceted roles involving multiple entries. We evaluated inter-annotator agreements using Fleiss’ Kappa across the three core dimensions of our hierarchical schema, yielding high consistency: Multi-ELAM—Role Type 0.84, Contribution Tier 0.78, Behavioral Patterns 0.75; Multi-LeCaRDv2—Role Type 0.81, Contribution Tier 0.76, Behavioral Patterns 0.74.
- **Accuracy Metrics:** Our audit demonstrates an initial generation accuracy of 95.4% for the CCP profiles. All identified errors in the sampled set are corrected through the protocol to ensure 100% fidelity in the final golden labels.

Task-Specific Data Partitioning We utilize these augmented benchmarks to support both the LCM and LCR subtasks. The data is partitioned as follows:

- **Training Set (for LCM and LCR):** The training sets for both tasks are identical. For the LCM task, we treat the multi-level similarity labels as distinct classification targets (a 3-class task for Multi-ELAM and a 4-class task for Multi-LeCaRDv2). For the LCR task, we binarize these labels, treating only high-similarity pairs as relevant (Label 2 for Multi-ELAM; Labels 2 and 3 for Multi-LeCaRDv2).

- Testing Set (for LCM): For the classification task, we use a held-out set of query-candidate pairs, as detailed in Table 1.
- Testing Set (for LCR): For the retrieval task, evaluation is performed on unseen query cases. We assess performance using a 100-candidate pool for each query. For Multi-LeCaRDv2, this pool is adopted directly from the original LeCaRDv2 benchmark. However, due to the filtering of certain invalid or missing documents, the actual pool size averages 99.14 candidates per query. For Multi-ELAM, the pool is constructed by merging all known relevant documents (Label 2) with the highest-ranked non-relevant documents retrieved via BM25 to form a pool of 100.

E Details of Baselines

(1) Legal-oriented retrieval methods: **SAILER** (Li et al., 2023) is the first pre-trained LCR model that leverages an asymmetric encoder-decoder architecture to capture the structural information of legal documents. **DELTA** (Li et al., 2025) improves discriminative ability for LCR by using word alignment and contrastive learning to focus its representation on key legal facts. **Law-Match** (Sun et al., 2023) is a model-agnostic causal learning framework that employs law articles as instrumental variables to reconstruct case embeddings.

(2) General large language models: Llama3-Chinese-8B-Instruct (Cui et al., 2024) is an adapted version of the LLaMA 3 (Grattafiori et al., 2024) model, optimized for Chinese language tasks via instruction fine-tuning on a large multilingual corpus. **Qwen3-8B/32B** (Yang et al., 2025) are models from the latest Qwen3 family, which integrates thinking and non-thinking modes into a unified framework. **Gemini-2.5-flash** (Comanici et al., 2025) is a capable general model designed to offer powerful reasoning performance while operating at low latency and cost.

(3) Legal large language models: Llama3-Chinese-8B-Law-Lora¹⁶ is a legal-domain LLM built by applying LoRA to the base Llama3-Chinese-8B model, fine-tuned on a specialized legal corpus of statutes, case law, and judicial opinions. **DeepSeek-R1-Chinese-Law**¹⁷ (Guo et al.,

2025) is a version of DeepSeek-R1 distilled onto Llama3-8B, and subsequently fine-tuned on a Chinese legal corpus. **DISC-Law-7B** (Yue et al., 2024) is an LLM-powered intelligent Chinese legal system that outperforms all existing legal-specific large models on the LawBench (Fei et al., 2024) benchmark. The latest version of the model used in this work is finetuned from Qwen2.5-7B-Instruct.

For dense retrieval methods, we finetune these models on our dataset with their optimal hyperparameter configurations. To ensure reproducibility and fairness, we employ a unified prompt design and the officially recommended hyperparameter settings for all LLM experiments. We access Gemini via its official API, while the other LLMs are run locally. To improve robustness, we implement a three-round retry mechanism for cases where the model fails to produce a valid output. In the LCM task, we utilized a few-shot prompt that provides label definitions and an in-context example per similarity class to elicit the classification. For the LCR task, a zero-shot prompt instructed the models to output a similarity score on a scale of 0 to 100. The complete prompt template for Multi-ELAM is shown in Table 15, while Multi-LeCaRDv2 differs only in its label definitions and case examples, with all other settings kept identical.

Due to the long-text nature of the LCR task, models such as ChatLaw (Cui et al., 2026), Lawyer-LLaMA (Huang et al., 2023), and fuzimingcha (Wu et al., 2023a) were excluded, as their limited input capacity prevents them from adequately handling such settings.

F Detailed Experimental Settings

We reserve 10% of the training set for parameter tuning. The reported ~ 800 A100 GPU hours represent the total experimental budget rather than the cost of a single training run, including CCP module pretraining, matching module training, hyperparameter search, and three-run averaging under the selected configuration. CCP module pretraining requires approximately 1.35 GPU hours on Multi-ELAM and 7.50 GPU hours on Multi-LeCaRDv2. For a single run under a fixed configuration, matching module training requires approximately 4.4 GPU hours for Multi-ELAM LCM, 16.9 GPU hours for Multi-LeCaRDv2 LCM, 7.0 GPU hours for Multi-ELAM LCR, and 29.2 GPU hours for Multi-LeCaRDv2 LCR. Thus, running all four task

¹⁶https://ollama.com/mrhua/llama3-8b-chinese-lora-law_f16_q4_0

¹⁷<https://huggingface.co/corn6/>

DeepSeek-R1-Chinese-Law

settings once requires approximately 57 GPU hours in total. Most of the total computational budget comes from hyperparameter search and repeated evaluation, particularly on Multi-LeCaRDv2. The cost of the Multi-LLM Debate data generation stage is not included in the GPU-hour estimate, as this stage is performed through API calls rather than local GPU training.

G Detailed Experimental Analysis

LCM Results Analysis LCM task is designed to assess a model’s ability to accurately classify the similarity between case pairs. We derive the following observations from the LCM experiment results:

- Fine-tuned retrieval models exhibit superior classification accuracy compared to LLMs. Traditional retrieval methods like SAILER and DELTA, when fine-tuned on task-specific training data, effectively learn the discriminative features that define each similarity category. In contrast, LLMs, operating in a few-shot setting with only label definitions and in-context examples, struggle to consistently map the complex nuances of case pairs to the correct labels.
- General-purpose LLMs can outperform specialized legal LLMs on benchmarks with semantically intuitive labels. For instance, on Multi-ELAM, Gemini achieves a 7.26% higher accuracy than the legal-specific DISC-Law-7B. This is likely because Multi-ELAM’s labels are defined as ‘dissimilar’, ‘partially similar’ and ‘fully similar’, which rely primarily on general notions of similarity. These labels align well with the broad semantic knowledge of general-purpose models, making the task less dependent on deep legal expertise.
- Legal-domain LLMs demonstrate a distinct advantage on tasks requiring specialized legal reasoning. This is evident on Multi-LeCaRDv2, where DISC-Law-7B surpasses Gemini’s accuracy by a significant margin of 29.48%. Unlike Multi-ELAM, the four similarity labels in Multi-LeCaRDv2 are defined by key legal elements and circumstances, so correctly identifying these requires specialized legal domain knowledge.
- Limitations in discerning complex legal relevance. Despite the reasoning potential of LLMs, they struggle with the intricate mapping between facts and law required for Multi-LCM. We observe that LLMs often produce false positives/negatives across four distinct dimensions of similarity. Specifically, they fail to discern cases that are factually similar but legally irrelevant (where different criminal degrees are involved) or factually distinct but legally similar. This suggests that general LLMs, and even some fine-tuned legal LLMs, remain heavily reliant on textual semantics.
- Among all baselines, DS-MLCR achieves the highest performance, outperforming both retrieval models and LLMs. The superiority of DS-MLCR lies in its novel hybrid design. It moves beyond traditional whole-text approaches by modeling defendant roles with fine granularity, which is achieved by leveraging CCP and LJO generated via multi-LLM reasoning. This synergy between structured role awareness and discriminative training empowers the model to capture subtle legal logic that other models miss, resulting in a clear advantage in assessing case similarity.

LCR Results Analysis The core objective of the LCR task is to evaluate how well a model can rank legally similar cases from a candidate ranking pool. We identify the following findings from the LCR experiment results:

- The Multi-LeCaRDv2 benchmark presents a greater challenge, leading to a significant performance drop across all model types. Compared with Multi-ELAM, the candidate pool in Multi-LeCaRDv2 features richer case backgrounds and more complex factual scenarios, along with a higher density of ground-truth relevant cases, thereby substantially increasing the difficulty of the ranking task.
- Powerful LLMs demonstrate strong performance in ranking tasks, which we attribute to their fundamental design. The LCM task imposes a classification framework, requiring a model to first assess similarity and then perform an additional interpretive step of categorizing that assessment into rigid, human-defined labels. In contrast, the LCR task is a more direct measure of semantic relevance. It

permits LLMs to operate on a seamless spectrum of similarity, translating their nuanced comprehension directly into a relevance score. This inherent suitability to their native reasoning capabilities explains their strong performance in retrieval.

- Scoring instability and discrete clustering effects. The relevance scores produced by LLMs often suffer from a form of benchmark fluidity where the internal assessment criteria shift according to the specific documents present in the candidate pool. This dynamic scale drift makes it difficult to maintain a consistent ranking standard across different queries. Moreover, although the models are instructed to provide single-digit precision on a 0-100 scale, they often gravitate toward specific numerical milestones such as multiples of five or ten. This rounding phenomenon leads to a sparse distribution of scores that fails to differentiate between cases with high factual overlap.
- Objective misalignment between generation and retrieval. The performance gap stems from a fundamental conflict between the generative training objectives of LLMs and the discriminative requirements of the retrieval task. While LLMs are optimized for the task of predicting the next token in a sequence, legal case retrieval depends on mapping case pairs into a shared representation space for precise ranking. This creates a significant train-test gap because the generative process is not explicitly designed to identify the contrastive boundaries between relevant and irrelevant precedents.
- The performance of LLMs is highly variable. For example, models such as Llama3-Chinese-8B-Instruct fall notably behind their counterparts. This variability indicates that advanced legal reasoning capabilities are not uniformly present across LLMs, highlighting that the selection of an appropriate base model is a critical factor for this task.
- The LCR results further confirm the superiority of DS-MLCR. Unlike competing methods that rely solely on surface-level factual similarity, our framework captures deeper structural correspondences between multi-defendants,

leading to consistently higher RecallK scores. This establishes DS-MLCR as the state-of-the-art system for multi-defendant retrieval tasks.

H Analysis on Loss Functions

Recall that for the LCR task, we follow prior work (Li et al., 2022; Deng et al., 2024) and adopt the Binary Cross-Entropy (BCE) loss. We further evaluate BCE against several representative ranking losses commonly used in retrieval tasks, including Margin Ranking Loss (Joachims, 2002), Adaptive Margin Ranking Loss (Ge et al., 2018), and InfoNCE Loss (van den Oord et al., 2019). We apply these loss functions to both our DS-MLCR model and its oracle variant. Specifically, the adaptive margin ranking loss scales the margin proportionally to the numerical distance between discrete relevance labels (e.g., 0-2 for Multi-ELAM and 0-3 for Multi-LeCaRDv2). However, experimental results show that these ranking losses consistently underperform the pointwise BCE loss. This observation can be attributed to several factors. First, the limited candidate pool size (4 for Multi-ELAM and 30 for Multi-LeCaRDv2) provides insufficient contrastive diversity, restricting the model’s ability to learn a well-calibrated ranking function. With such a small comparison set, the model receives limited negative variation, which weakens the learning signal for fine-grained discrimination. Second, the low training data utility further hinders optimization. Only 58.48% of queries in Multi-ELAM and 70.47% in Multi-LeCaRDv2 contain both positive and negative samples, meaning that pairwise ranking objectives effectively discard a substantial portion of available supervision. This incomplete contrast reduces convergence stability and limits the effectiveness of contrastive learning. Third, LLM-generated defendant CCPs and LJOs inevitably introduce noise into the alignment process. Although margin-based losses encourage strong separation, they are highly sensitive to annotation or reasoning inaccuracies. Even minor misalignments in the role-aware reasoning stage can violate strict margin constraints, resulting in unstable gradients. In contrast, BCE provides a smoother optimization landscape and exhibits greater tolerance to noisy supervision. Overall, the pointwise classification paradigm proves to be more robust in the complex Multi-LCR setting.

Debate Stage	Multi-ELAM							Multi-LeCaRDv2						
	Charges			Law Articles			Sent.	Charges			Law Articles			Sent.
	P	R	F1	P	R	F1	Acc	P	R	F1	P	R	F1	Acc
Proposer 1 (DeepSeek)	0.932	0.936	0.934	0.927	0.927	0.926	0.450	0.665	0.662	0.652	0.613	0.581	0.578	0.251
Proposer 2 (Qwen)	0.951	0.955	0.952	0.952	0.954	0.952	0.436	0.547	0.535	0.533	0.545	0.518	0.520	0.249
Proposer 3 (Kimi)	0.928	0.931	0.930	0.923	0.921	0.921	0.480	0.712	0.718	0.703	0.705	0.706	0.686	0.294
Summarizer Stage	0.947	0.951	0.948	0.945	0.947	0.945	0.463	0.735	0.730	0.721	0.714	0.698	0.690	0.304
Final Decision (Ours)	0.971	0.976	0.973	0.971	0.972	0.971	0.488	0.746	0.739	0.732	0.723	0.707	0.699	0.301

Table 6: Performance evaluation of different stages in the multi-LLM debate. Metrics for crime and law article prediction are weighted P/R/F1. The metric for sentence prediction is weighted Accuracy. The best results are highlighted in bold.

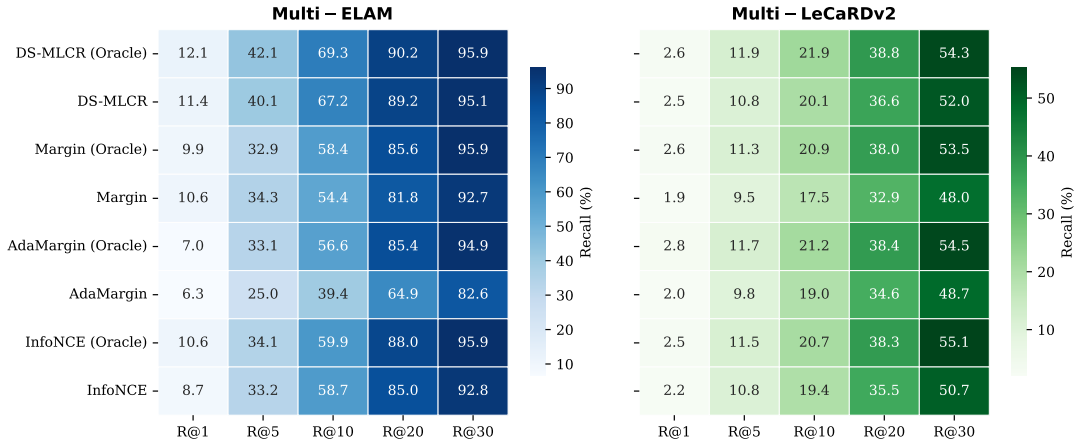


Figure 6: Performance Heatmap of Various Retrieval Losses.

I Effectiveness of the Multi-LLM Debate Mechanism

To quantitatively validate the effectiveness of our multi-LLM debate, we evaluated legal judgment prediction at each stage of the framework. We assessed three proposer LLMs (DeepSeek-V3, Qwen3, Kimi-K2), the Summarization stage output, and the final Critique and Decision output on three core LJO sub-tasks: crime prediction, law article prediction, and sentence prediction. Following prior work (Huang et al., 2024), we used weighted Precision, Recall, and F1 for crime and law article prediction, and weighted Accuracy for sentence prediction, with weights proportional to the number of defendants ($w = \log_2(N_{\text{defendants}} + 1)$) to account for case complexity. As shown in Table 6, the results reveal consistent, progressive improvements in prediction quality across the stages of the debate framework.

J Detailed Efficiency Analysis

To examine the efficiency of the Multi-LLM Debate, we compare DS-MLCR with a single-proposer setting, where DeepSeek-V3 alone generates the CCPs and LJOs while the same Individual-Aware Reasoning module is used for case similarity modeling. We report the average inference time and token usage per query under the LCM setting in Table 7. Compared with the single-proposer setting, the Multi-LLM Debate increases the inference time from 14.05s to 41.59s and the token usage from 2,756 to 14,631. While this introduces additional computational cost, the Multi-LLM Debate substantially improves the quality of CCPs and LJOs. As shown in Table 6, the Final Decision consistently improves over the individual proposer LLMs on the three LJO sub-tasks. For example, compared with DeepSeek-V3, the weighted F1 for crime prediction on Multi-ELAM increases from 0.934 to 0.973. This improvement further translates into better downstream case similarity modeling, with Macro-F1 increasing from 68.0% to 70.8% on

Models	Multi-ELAM				Multi-LeCaRDv2				Time (s)	Tokens	Cost (\$)
	Acc	P	R	F1	Acc	P	R	F1			
DeepSeek-V3	78.6	69.2	68.1	68.0	66.1	44.8	37.8	39.0	14.05	2,756	0.0028
DS-MLCR	79.1	70.4	71.7	70.8	69.1	52.6	41.3	41.4	41.59	14,631	0.0146
DS-MLCR (Oracle)	80.0	72.8	72.5	72.6	73.4	61.5	53.8	56.6	–	–	–

Table 7: Performance and inference efficiency of the Multi-LLM Debate under the LCM setting. The inference time and token usage are averaged per query. The cost is estimated at USD 1.00 per 1M tokens.

Models	Multi-ELAM	Multi-LeCaRDv2
SAILER	16.60 ms	15.57 ms
DS-MLCR	42.36 ms	39.13 ms

Table 8: Inference latency of SAILER and DS-MLCR on Multi-ELAM and Multi-LeCaRDv2.

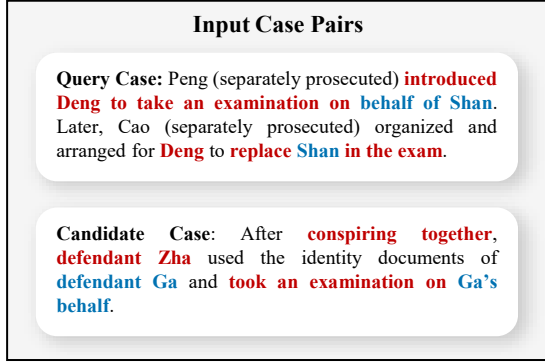


Figure 7: Case Study on Exam Impersonation with Different Defendant Roles

Multi-ELAM and from 39.0% to 41.4% on Multi-LeCaRDv2. Furthermore, the Oracle results in Table 2 show substantial remaining headroom when ground-truth CCPs and LJOs are used, particularly on Multi-LeCaRDv2, where Macro-F1 increases from 41.4% to 56.6%. These results indicate that the additional computational cost is targeted at improving the quality of CCPs and LJOs, which directly contributes to downstream matching performance, rather than being redundant overhead.

We further evaluate the inference latency of the Individual-Aware Reasoning module after CCP/LJO generation. As shown in Table 8, DS-MLCR requires 42.36 ms and 39.13 ms on Multi-ELAM and Multi-LeCaRDv2, respectively, compared with 16.60 ms and 15.57 ms for SAILER. Thus, once the defendant representations are available, the additional latency introduced by Individual-Aware Reasoning is limited.

K Case Study

The critical importance of structural role alignment in Multi-LCR is vividly illustrated by the case pair in Figure 7. Although both cases involve examination impersonation and thus appear factually similar, their defendant structures differ significantly. The query case reveals a multi-layered, hierarchical chain of command where the initiator, Shan (identified as Planner/Organizer, R1-T1-P11/P13), utilized intermediaries to arrange for the executor, Deng (Active Perpetrator, R2-T2-P21). In contrast, the candidate case features a simple pair conspiracy where Ga (Planner, R1-T1-P11) and Zha (Active Perpetrator, R1-T2-P21) acted as symmetric co-principals.

A baseline model like Gemini, blind to these structural differences in defendant roles, would erroneously classify the pair as a “full match” based on surface-level facts. However, DS-MLCR identifies the misalignment in defendant roles by leveraging qualitative CCP and quantitative LJO, thereby correctly outputs a “partial match”.

L More Details of Methodology

L.1 Controlled Vocabulary for CCP Motivation

The 10 distinct motivations used in the controlled vocabulary for the CCP are: *Profit-seeking, Revenge, Remorse, Duress, Protective, Emotion-driven, Following Orders, No Malicious Intent, Evasion of Law, and Unknown.*

L.2 Example of a Complete CCP Entry

Table 9 is a complete, structured CCP entry for defendant Xing as generated by our Multi-LLM Debate module.

M Prompt Templates for the Multi-LLM Debate Framework

This appendix provides the detailed prompt templates used in our Multi-LLM Debate framework. Each prompt is designed for a specific stage of

Case Fact Description (Input) Defendant Xing purchased specialized equipment for eavesdropping via the internet, subsequently providing it to Wang and others for use in cheating on the National Postgraduate Entrance Examination, from which he obtained illegal profits.... After being apprehended, Defendant Xing truthfully confessed his crimes....
Structured CCP Data (Output) <pre> { "Defendant": "Xing", "Entries": [{ "Role": "R1 Principal Role", "Tier": "T1: Leading Contribution", "Weight": 1.0, "Patterns": [{ "Pattern": "P11: Planner", "Phase": "Pre-crime", "Motivation": "Profit-seeking" }] }, { "Role": "R3 Mitigating Role", "Tier": "T5: Mitigating Contribution", "Weight": -0.5, "Patterns": [{ "Pattern": "P52: Confession of Key Facts", "Phase": "Post-crime", "Motivation": "Remorse" }] }] } </pre>

Table 9: Example of a Complete CCP Structure (Defendant: Xing)

the reasoning process, guiding the Large Language Models to perform structured analysis, summarization, critique, and final decision-making. Notably, during the adversarial critique (Stage 3) and final decision (Stage 4) phases, LLMs may exhibit a tendency to arbitrarily inflate sentences or hallucinate aggravating factors during free-form debate. To ensure the debate remains strictly grounded in objective evidence (as emphasized in Section 4.1), we explicitly incorporate judicial guardrails into the prompts, such as the "principle of prudence" and "strict limits on sentence adjustment." These constraints force the models to rely solely on statutory provisions and factual evidence rather than speculative penalization. Specifically, Table 10 to Table 14 show the detailed prompts for each stage of the debate mechanism.

Background Setting You are an expert legal analysis assistant responsible for analyzing the Crime Contribution Portrait of all defendants in criminal cases. Your task is to identify all defendants from the case description and evaluate the role and degree of contribution of each defendant in the crime.
Analysis Steps Identify Defendants: Carefully read the case description to identify all defendants involved and record their names. Assess Crime Contribution Portrait (CCP): Based on the case description and in conjunction with the "Contribution Tier" and "Behavioral Patterns" provided below, generate a <code>crime_contribution_portrait</code> field for each defendant to quantify their role and the severity of their involvement in the joint crime. Each defendant may have multiple role contributions (e.g., principal and mitigating roles); please model each role separately and include them in the <code>crime_contribution_portrait</code> list.
Strict Constraints Important: You must strictly select from the predefined options below. Do not create any new IDs or names! Role Type (<code>role_type</code>): Must be one of <i>R1 Principal Role</i>, <i>R2 Accessory Role</i>, <i>R3 Mitigating Role</i>. Contribution Tier (<code>tier</code>): Must be one of <i>T1: Leading Contribution</i>, <i>T2: Key Contribution</i>, <i>T3: Auxiliary Contribution</i>, <i>T4: Incidental Contribution</i>, <i>T5: Mitigating Contribution</i>. Weight (<code>weight</code>): Assign a numerical weight based on the Tier: T1 (1.00), T2 (0.75), T3 (0.50), T4 (0.30), T5 (-0.50). You may adjust this baseline value within a range of ± 0.05 to reflect the specific factual nuances of the defendant's actions. Behavioral Patterns (<code>patterns</code>): T1-level: P11: Planner, P12: Decision-maker, P13: Organizer/Commander. T2-level: P21: Active Perpetrator, P22: Primary Beneficiary, P23: High-risk Skill Support. T3-level: P31: Financial Support, P32: Tool Provider, P33: Information Provider, P34: Lookout/Support. T4-level: P41: Concealing/Transferring Stolen Goods, P42: Destroying Evidence. T5-level: P51: Voluntary Surrender, P52: Confession of Key Facts, P53: Active Restitution/Compensation. Phase (<code>phase</code>): Must be one of <i>Pre-crime</i>, <i>Mid-crime</i>, <i>Post-crime</i>. Motivation (<code>motivation</code>): Must be one of <i>Profit-seeking</i>, <i>Revenge</i>, <i>Remorse</i>, <i>Duress</i>, <i>Protective</i>, <i>Emotion-driven</i>, <i>Following Orders</i>, <i>No Malicious Intent</i>, <i>Evasion of Law</i>, <i>Unknown</i>.
Output Format ### Case Description <pre>{case_description} { "defendants_portraits": [// Strictly follow the structured JSON schema demonstrated in Table 9], "explanation": "Analysis process and rationale" }</pre>

Table 10: Prompt Template for Stage 1 (Round 1): Crime Contribution Portrait (CCP) Generation

Background Setting You are an expert legal analysis assistant. You now need to provide a detailed Legal Judgment Prediction for the defendants by integrating the case description with the previously generated Crime Contribution Portrait.
Analysis Steps Predict Judgment Based on CCP: For each defendant, predict the applicable law articles, charges, and sentence based on the case description and their CCP. Sentencing Considerations: Refer to the role type, contribution tier, and weight in the CCP. Principal roles (R1) typically bear greater criminal responsibility. Accessory roles (R2) may be eligible for sentence reduction as accomplices. Mitigating roles (R3) should be considered for lenient or reduced punishment.
Output Format ### Case Description {case_description} ### Crime Contribution Portrait Analysis {crime_contribution_portrait_result} <pre>{ "case_LJO": [{ "name": "<Defendant Name>", "law": "<Applicable Law Articles>", "crime": ["<Charge(s)>"], "sentence": ["<Specific Predicted Sentence>"], "crime_contribution_portrait": <The CCP for this defendant from the first round> }], "explanation": "Rationale for the judgment based on the CCP" }</pre>

Table 11: Prompt Template for Stage 1 (Round 2): Legal Judgment Outcome (LJO) Generation

Task You need to conduct an in-depth analysis of the judgment results and CCP analyses provided by three legal LLMs, synthesize them with the case facts and legal texts, and produce the most reasonable consolidated judgment and CCP summary.
Input Content #### Case Facts {case_description} #### Judgment from LLM-1 (e.g., DeepSeek-V3): {llm1_result} #### Judgment from LLM-2 (e.g., Qwen3-235B-A22B): {llm2_result} #### Judgment from LLM-3 (e.g., Kimi-K2): {llm3_result} #### Relevant Legal Texts {law_texts}
Requirements LJO Judgment Summary Analysis: Analyze each model's judgment on defendants, law articles, charges, and sentences. Evaluate the accuracy of each judgment against the provided legal texts. For any discrepancies, determine the most reasonable judgment through a thorough analysis of the legal texts. CCP Summary Analysis: Analyze the CCP assessment for each defendant from the three models. Compare the similarities and differences in their judgments on role types, contribution tiers, and behavioral patterns. For any discrepancies, provide the most reasonable summary based on the case facts.
Output Format Please output your analysis in the following format: <pre>{ "summary": { ... }, "explanation": "Comprehensive rationale for the judgment based on legal texts and CCP analysis" }</pre>

Table 12: Prompt Template for Stage 2: Summarization

Task As a legal expert, please evaluate the following case judgment summary and CCP summary based on the case facts. Indicate the parts you agree or disagree with and provide your rationale.
Case Facts {case_description} ### Initial Summary {summary_result} ### Relevant Legal Texts {law_texts}
Important Constraints: Strict Limits on Sentence Adjustment 1. Prohibition of Arbitrary Sentence Increases: Strictly avoid recommending a higher sentence unless there is explicit legal basis proving the initial summary's sentence is clearly too lenient and fails to meet the statutory minimum. 2. Principle of Leniency: Prioritize mitigating and extenuating circumstances in sentencing recommendations, adhering to the policy of balancing punishment with leniency. 3. Cautious Approach: Any recommendation for sentence adjustment must be supported by sufficient legal text evidence and detailed legal analysis. 4. Conclusive Evidence: Only recommend a sentence adjustment if you are 100% certain the initial sentence violates statutory provisions. 5. Preference for Reduction: If there is ambiguity regarding the sentence, lean towards recommending a reduced sentence rather than an increased one.
Output Format { "opinion": "Support/Partial Support/Object", "analysis": [{ "point": "...", "agreement": true/false, "reason": "...", "suggestion": "..." }], "explanation": "Overall evaluation of the LJO and CCP summaries, with special emphasis on the cautious approach to sentencing assessment." }

Table 13: Prompt Template for Stage 3: Peer Critique

Task Based on the case facts, the initial summary, and the critiques from two legal experts, conduct an independent critical analysis to provide the final case judgment and Crime Contribution Portrait.
Case Facts {case_description} ### Initial Summary {summary_result} ### Expert Opinion 1 {expert1_opinion} ### Expert Opinion 2 {expert2_opinion} ### Relevant Legal Texts {law_texts}
Key Decision Principles: Independent Judgment and Critical Thinking 1. Do Not Blindly Follow Expert Opinions: Expert opinions are for reference only. You must make an independent judgment based on the legal texts and case facts. 2. Principle of Prudence in Sentencing: Tend to maintain a moderate sentence. Be highly skeptical of recommendations to increase sentences unless rigorously supported by law. Prioritize leniency when there is ambiguity. 3. Critical Analysis: Analyze the reasonableness and evidentiary support of the expert opinions. Identify any potential for excessive severity. 4. Policy of Balancing Leniency and Severity: Strictly implement the criminal policy of balancing leniency and severity to avoid disproportionately heavy sentences.
Output Format Only output the result in the following JSON format: { "case_LJO": [{ ... }], "explanation": "Rationale for the independent judgment based on legal texts and CCP, explaining how expert opinions were critically handled, with special emphasis on the basis and prudence of the sentencing decision." }

Table 14: Prompt Template for Stage 4: Final Decision

Prompt Template for the LCM Task(Multi-ELAM)
#Background# You are an expert specializing in legal case retrieval for intelligent justice systems. #Requirements# You are required to determine the similarity label between the query case and the candidate case provided in the #Input# section. Both cases are enclosed in XML-style tags. Assess the similarity and relevance between the candidate case and the query case based on the fact description and key criminal elements. Provide your resulting label from the set {0, 1, 2}, where 0 means dissimilar, 1 means partially similar, and 2 means highly similar. Please note that you must only output the final numerical label: 0, 1, or 2. #Input# <Query Case>{Query}</Query Case> <Candidate Case>{Candidate}</Candidate Case> #Example# <Example Query Case> One day in 2015, the defendant, Jin Long, found Mr. Chen’s motor vehicle driver’s license at the Chengbei Gas Station in Hunchun City and replaced the photo on it with his own. On June 5, 2017, near the Heshan Forest Farm in Hunchun City, Jin Long was caught by officers of the Hunchun Traffic Police Brigade for driving under the influence of alcohol. Jin Long used the altered driver’s license to accept administrative punishment in Hunchun City, which resulted in 12 points being deducted from Mr. Chen’s driver’s license..... </Example Query Case> <Candidate Case with Label 2> On April 26, 2019, the defendant, Li Wenhua, was stopped by the police of the Holingol City Brigade near the PetroChina gas station on Heremute Street while driving a red Ouman brand eight-wheeled truck using a fake A2 driver’s license..... </Candidate Case with Label 2> <Candidate Case with Label 1> In June 2018, the defendant, Chao Meng, who had not obtained a motor vehicle driver’s license, purchased a counterfeit C1 motor vehicle driver’s license online from an unidentified person for a price of 870 RMB. At approximately 18:00 on July 10, 2019, the defendant, Chao Meng, was driving a "Hyundai" sedan and was stopped by on-duty traffic police from the Shangqiu City Traffic Police Detachment at the Henan-Shandong border toll station of the Ji-Guang Expressway in Liangyuan District, Shangqiu City. A search in the national public security traffic management information system yielded no driver’s license information for the defendant, Chao Meng..... </Candidate Case with Label 1> <Candidate Case with Label 0> One day in October 2018, the defendant, Liang, knowing that Meng (separately prosecuted) and Zhang1 (separately prosecuted) were running a casino in partnership, colluded with Han (separately prosecuted) who worked at the casino. They found Zhao (separately prosecuted) and Zhang2 (separately prosecuted), who were also staff at the casino. They demanded that these two individuals not testify that Meng had a stake in the casino, and gave each of them one thousand yuan in cash as hush money..... </Candidate Case with Label 0>

Table 15: The prompt templates for the Legal Case Matching and Legal Case Retrieval tasks

Prompt Template for the LCM Task(Multi-LeCaRDv2)
#Background# You are an expert specializing in legal case retrieval for intelligent justice systems. #Requirements# You are required to determine the similarity label between the query case and the candidate case provided in the #Input# section. Both cases are enclosed in XML-style tags. Assess the similarity and relevance between the candidate case and the query case based on the fact description and key criminal elements. Provide your resulting label from the set {0, 1, 2, 3}. Specifically, 0 indicates that neither the key criminal elements nor the fact description is relevant; 1 indicates that the fact description is relevant but the key criminal elements are irrelevant; 2 indicates that the key criminal elements are relevant but the fact description is irrelevant; and 3 indicates that both the key criminal elements and the fact description are relevant. Please note that you must only output the final numerical label: 0, 1, 2, or 3. #Input# <Query Case>{Query}</Query Case> <Candidate Case>{Candidate}</Candidate Case> #Example# <Example Query Case> At 1:00 on January 10, 2018, Zhang1 drove a car and collided with Sun Shaopeng, causing Sun's death, and then escaped. Zhang1 asked Zhang Jiandong to find someone to take the blame. Zhang Jiandong, Jia Gaofei, and Liu Bowei arranged for Shang Yanxiang to impersonate the driver for 20,000 RMB. Shang Yanxiang surrendered to the police and falsely claimed to be the perpetrator..... </Example Query Case> <Candidate Case with Label 3> On August 27, 2014, Lv drove a vehicle without a license, caused an accident injuring three people, and fled. Lv contacted Yuan, who found the defendant Yang to impersonate the driver. Yang made a false confession to the police claiming to be the driver, but video evidence later revealed Yang was not the actual driver..... </Candidate Case with Label 3> <Candidate Case with Label 2> On December 19, 2018, Huang Shanwei drove a heavy truck and caused a fatal accident. Worried that insurance would not cover the damages due to Huang's internship period, the car owner, defendant Sun Wenhua, falsely claimed to the police that he was driving, shielding Huang from prosecution. Sun later surrendered and confessed the truth..... </Candidate Case with Label 2> <Candidate Case with Label 1> On December 22, 2012, Xu, Tu, and Zhao killed a man in a fight. They contacted the defendant Yang, informing him of the crime and asking to leave the city. Knowing this, Yang drove them to Suzhou to help them escape and received 400 RMB. Yang was convicted of harboring criminals..... </Candidate Case with Label 1> <Candidate Case with Label 0> On January 31, 2015, the defendant Tao sold 0.17 grams of methamphetamine to Lin for 100 RMB in a hotel room and was caught by the police on the spot. Tao was convicted of drug trafficking..... </Candidate Case with Label 0>
Prompt Template for the LCR Task
#Background# You are an expert specializing in legal case retrieval for intelligent justice systems. #Requirements# You are required to determine the similarity between the candidate case and the query case provided in the #Input# section. Both cases are enclosed in XML-style tags. Assess the similarity and relevance between the query case and the candidate case based on the fact description and key criminal elements, providing a similarity score on a scale of 0 to 100. Please note that you must only output the final similarity score. The score must be an integer between 0 and 100. Do not use extreme values such as 0 or 100. #Input# <Query Case>{Query}</Query Case> <Candidate Case>{Candidate}</Candidate Case>

Table 15: The prompt templates for the Legal Case Matching and Legal Case Retrieval tasks (Continued)