Navigating Legal Case Matching via Heterogeneous Graph Learning and Matching Path Decomposition

Anonymous ACL submission

Abstract

Legal case matching (LCM) endeavors to determine the relevance between query cases and target cases, which plays a pivotal role in supporting legal decisions. In legal practice, query cases typically contain only fact descriptions, while target cases, being historical cases, often include additional case analysis that provides a new perspective for the LCM task beyond semantic similarity. In statutory law systems (e.g., China), such analysis relies on law article interpretation, while in case law systems (e.g., US, UK), it relies on precedent case references. Based on these observations, we propose a relation-driven framework called RedMatch, under which target cases are intrinsically connected to one another and associated with cited laws. First, it constructs a global heterogeneous graph for all target cases to extract case-case and case-law relations. Then, a graph transformer integrates these relations in the matching prediction model to enhance the case representation. Finally, a path learning task is designed to navigate the model to decompose multiple matching paths to reach target cases by leveraging these relations. RedMatch also introduces a law article matching task via multitask learning to align LCM outcomes and enhance the method's versatility. Experiments on three publicly available datasets, including Chinese and English language, demonstrate state-of-the-art performance of RedMatch, highlighting its effectiveness and generalizability.

1 Introduction

Legal case matching (LCM) plays a vital role in legal systems, which provides guidance and justification for legal decisions. For example, when faced with query cases, if the judge needs to find historical cases for reference, LCM helps determine the relevance between them. In addressing this task, some studies formulate LCM as matching two long-form text documents based on semantic similarity. Early rule-based methods (Zeng et al., 2005; Saravanan et al., 2009) evolved into the recent learning-based methods like BERT-based methods (Chalkidis et al., 2020; Shao et al., 2020; Xiao et al., 2021a; Li et al., 2023) and graph-based methods (Bhattacharya et al., 2020; Sun et al., 2023), aiming to capture complex semantics.

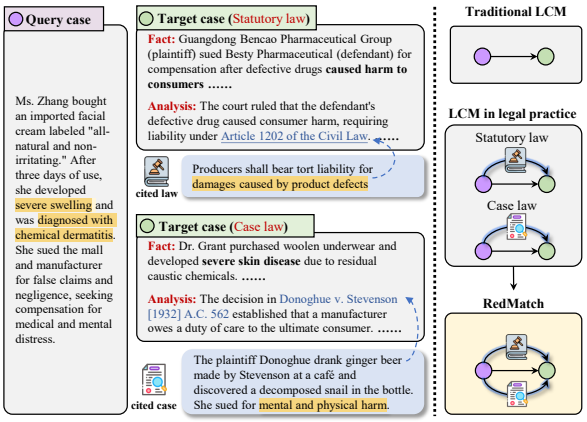

Figure 1: Illustration of using case analysis in the Legal Case Matching (LCM) task under statutory law and case law systems. RedMatch takes case analysis (e.g., cited case and cited law article) into consideration during the matching process.

While effective, treating legal cases merely as general long-form text documents leaves room for further enhancement. In legal practice, query cases typically contain only fact descriptions, while target cases, being historical cases, often include additional case analysis. This analysis offers an in-depth examination, facilitating a more accurate and comprehensive case representation. The focus in case analysis differs in different legal systems. As shown in Fig. 1, case law (Legal Information Institute, Cornell Law School, 2020; Lewis et al., 2025) (e.g., Australia, the UK, and the US) heavily relies on precedents, often including analyses of other cases within target cases. In contrast, statutory law (Coolidge, 2023) (e.g., China and Most of Europe) emphasizes interpretations of law articles. Case analysis in both legal systems reveals the intrinsi-

cal relations within target cases and their relations to law articles, which can assist the LCM task. Recently, there has been a trend towards integrating these systems to more comprehensively address legal issues (Harvard Law Review, 2016).

Based on these observations, we argue that LCM should account for these case-case and case-law relations together. For instance, if a query case cannot directly match a target case due to low semantic similarity, it may first match a cited law or case, and then reach the target case through these relations. In essence, these relations provide alternative matching paths, increasing matching flexibility. However, realizing such an idea presents two challenges: 1) How to extract effective case-case and case-law relations? 2) How to navigate the matching process with the help of these relations?

To address these challenges, we propose RedMatch, a Relation-driven framework that models case-case and case-law relations to navigate legal case matching. First, a heterogeneous graph is constructed with cases and laws as distinct nodes, connected through case references and law citations to extract relational information. This relational information is then integrated into the model via a graph transformer that aggregates information from neighboring nodes to generate graph representation of cases, which are fused with semantic representation. To leverage relations for navigating the matching process, a path learning task enumerates and decomposes multiple relation-driven matching paths into sub-paths, treating those from correct matches as positive and failed matches as negative. A margin loss is employed to encourage positive selection, guiding the model towards better matching direction. During inference, multiple matching paths are combined to compute probability distribution for final predictions. RedMatch also handles the law article matching task through multi-task learning, aligning LCM results and enhancing the method's practical versatility.

Extensive experiments on three public datasets (two Chinese datasets for criminal and civil cases, respectively, and one English dataset) show that RedMatch achieves state-of-the-art performance and significantly improves law article matching.

In summary, our contributions include:

- We investigate the legal case matching task from the perspectives of case law and statutory law, modeling the case-case and case-law relations into a heterogeneous graph.
- We propose RedMatch, a relation-driven framework that uses a graph transformer to integrate relational information and introduces a path learning task with margin loss to navigate the matching process.
- Experiments on public Chinese and English datasets show the state-of-the-art performance of RedMatch on both tasks, validating the effectiveness and generalizability of relation navigation. The code is publicly available[1].

2 Related Work

2.1 Legal Case Matching

Legal case matching (LCM), which calculates the similarity between paired cases, is at the core of legal case retrieval (LCR) to find relevant cases. Early methods relied on manual knowledge and feature engineering (Zhong et al., 2020), lacking adaptability across legal contexts. Recent works have leveraged deep learning, with BERT-PLI (Shao et al., 2020) using BERT for paragraph-level matching and Lawformer (Xiao et al., 2021a) utilizing Longformer (Beltagy et al., 2020) for long documents. Some studies have also considered structured legal knowledge. SAILER (Li et al., 2023) reformulates queries and generates case summaries for alignment. Law-Match (Sun et al., 2023) introduces causal learning, using law articles as instrumental variables to separate their mediation effect from case embeddings. Beyond text similarity, legal factor modeling has been explored. CaseGNN (Tang et al., 2023) constructs text-attributed graphs for legal cases and applies graph neural networks for retrieval. CaseLink (Tang et al., 2024; Donabauer and Kruschwitz, 2025) constructs a global homogeneous case graph to link all query and target cases and uses inductive graph learning. While prior works improve LCM through semantic learning, causal inference, they overlook the complex relational information between cases and laws. In contrast, our work models case-case and case-law relations, capturing legal dependencies to guide the matching process.

2.2 Graph Neural Networks

Graph Neural Networks (GNNs) have proven effective in modeling structured data by aggregating neighborhood information for meaningful representations (Hamilton et al., 2017). Traditional

[1] https://anonymous.4open.science/r/RedMatch-257B

models like Graph Convolutional Networks (GCN) (Kipf and Welling, 2016) use convolution in a transductive setting, while Graph Attention Networks (GAT) (Veličković et al., 2017) introduce attention mechanisms to weigh neighboring nodes' importance. Heterogeneous Graph Transformers (HGT) (Hu et al., 2020) further enhance these models by handling multi-relational graphs with adaptive attention, making them suitable for complex applications. GNNs have also been applied in text matching tasks. Liu et al. (2018) introduced a Concept Interaction Graph, applying GCN to compute matching scores, while (Pang et al., 2021) integrated PageRank into a transformer for sentence similarity in long-form text matching. When addressing the LCM task, existing GNN-based methods focus mainly on word-wise or sentence-wise relations (Tang et al., 2023; Yu et al., 2022; Bi et al., 2022; Ma et al., 2023) or using a global homogeneous graph to model the relations (Tang et al., 2024). Our work connects case and law nodes to construct a heterogeneous graph, enabling the exploration of multiple matching paths and enhancing matching flexibility.

3 Background and Preliminaries

3.1 Task Definition

We define the LCM task as follows: given a labeled tuple (c^q, c^t, y^c), where c^q is a query case, c^t is a target case from the historical cases base $C = \{c_1, c_2, \ldots, c_n\}$ and y^c is the human-annotated matching label. The label y^c can be selected from a predefined set $\mathcal{Y}^c$ (e.g. $\mathcal{Y}^c = \{0, 1, 2\}$, where 0 means mismatch, 1 means partially match, 2 means match). Both query cases and target cases are textual documents. The objective of LCM is to learn a decision function $f(c^q, c^t) \rightarrow y^c$ that assigns a matching label y^c to each paired case. In this work, we also introduce the law article matching (LAM) task. Similar to the LCM task, its objective is to learn a decision function $f(c^q, a^t) \rightarrow y^a$, where a^t is the target law article and y^a is the matching label from $\mathcal{Y}^a$. In practice, query cases are mainly fact descriptions, while target cases additionally contain case analysis.

3.2 Statutory Law and Case Law

Statutory law (Coolidge, 2023) refers to a legal system formally enacted by legislative bodies, such as Congress or state legislatures. These laws are written and codified, providing specific regulations and guidelines that govern various aspects of society. For example, traffic laws and drug possession laws are statutory laws established through legislation. Statutory law is predominantly used in China, most of Europe and South America. Case law (Legal Information Institute, Cornell Law School, 2020; Lewis et al., 2025), also known as common law, is the legal system derived from judicial decisions made in courts. It relies on precedents set by previous cases to guide future decisions, evolving over time as courts address new issues. The United Kingdom and the United States are prime examples of case law jurisdictions.

As shown in Appendix Fig. 5, the two legal systems are widely used worldwide, with varying applicability depending on the context. In recent years, there has been a growing trend toward integrating these systems to address legal issues more comprehensively (Harvard Law Review, 2016; Palermo and Kössler, 2017). In line with the trend of legal system integration, we believe that legal case matching should consider not only the content of the cases themselves but also case-case relations and case-law relations.

4 Methodology

In this section, we introduce details of RedMatch, whose illustration is shown in Fig. 2.

4.1 Heterogeneous Graph Construction

First, our goal is to extract case-case and case-law relations for all target cases, which naturally fits the representation of these relations using a global heterogeneous knowledge graph $G = (V, E)$, where V represents the node set and E denotes the edge set. We define two node types and three edge types to model relations, as outlined below.

Case Node Considering rich information in target cases, each target case c_i is converted into a node, where its feature representation $x_{c_i} \in \mathbb{R}^d$ is obtained by encoding the full case text with a legal text encoder. d is the dimensionality of the feature space. The encoder can be any pre-trained model that encodes case text into features, such as Legal-BERT (Chalkidis et al., 2020), SAILER (Li et al., 2023), or Lawformer (Xiao et al., 2021a).

Law Node Let $A = \{a_1, a_2, \ldots, a_m\}$ represent the set of law articles. The statutory texts from the legislation (e.g., Article 1202 of the Civil Law in Fig. 1) are converted into law nodes. Each

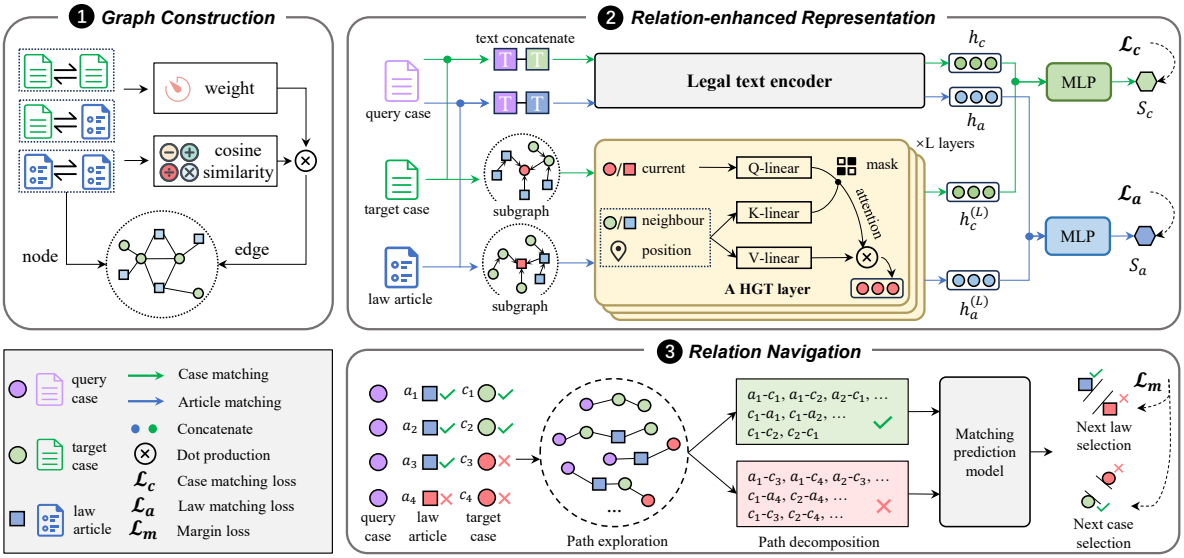

Figure 2: The training framework of RedMatch, consists of three modules: 1) A heterogeneous graph is constructed with legal cases and law articles as distinct node types, capturing relational information. 2) A heterogeneous graph transformer aggregates neighboring case and law information to integrate relational information for matching prediction. 3) A path learning task decomposes multiple matching paths, with margin loss encouraging the selection of positive paths, enhancing matching flexibility.

law node's feature representation $x_{a_i} \in \mathbb{R}^d$ is also encoded by the legal text encoder.

To enhance message passing, we not only extract relations based on cited cases or laws, but also explore between all matching nodes. Edges combine type-specific weights and semantic similarity to reduce reliance on the lexical matching method and capture complex legal relations.

Case-Case Edge Two related cases often cite the same law articles in their case analysis (MacCormick, 1994). The edge weight between two case nodes c_i and c_j is defined:

$$w_{c_i,c_j} = J\left(a_{c_i}, a_{c_j}\right) \times \mathrm{Sim}\left(x_{c_i}, x_{c_j}\right)$$
$$J(a_{c_i}, a_{c_j}) = \frac{|a_{c_i} \cap a_{c_j}|}{|a_{c_i} \cup a_{c_j}|}, \quad (1)$$

where Sim denotes cosine similarity, and J is the Jaccard similarity between two cited law article sets a_{c_i} and a_{c_j}.

Case-Law Edge The edge weight between a case node c_i and a cited law node a_j reflects the citation importance of the law in the case (Bacchus, 2002). An earlier rank of cited law yields a higher weight.

$$w_{c_i,a_j} = \frac{1}{\mathrm{Pos}_{c_i}(a_j)} \times \mathrm{Sim}\left(x_{c_i}, x_{a_j}\right), \quad (2)$$

$\mathrm{Pos}_{c_i}(a_j)$ is the rank of the cited law in the case. If a law is not cited in the case, its edge weight is 0.

Law-Law Edge The edge weight between two law nodes a_i and a_j is defined based on their co-occurrence frequency (Small, 1973) in the case set:

$$w_{a_i,a_j} = \frac{\mathrm{freq}(a_i, a_j)}{\sum_{k=1}^{m} \mathrm{freq}(a_i, a_k)} \times \mathrm{Sim}\left(x_{a_i}, x_{a_j}\right), \quad (3)$$

where $\mathrm{freq}(a_i, a_j)$ is the number of cases in which both a_i, a_j are cited together.

Relation Adjacency Matrix The adjacency matrix $M \in \mathbb{R}^{(n+m)\times(n+m)}$ integrates case-case, case-law, and law-law relationships as:

$$M = \begin{bmatrix} M_{\text{case-case}} & M_{\text{case-law}} \\ M_{\text{case-law}}^{\top} & M_{\text{law-law}} \end{bmatrix}, \quad (4)$$

where $^{\top}$ is the transpose operation. The overall adjacency matrix, M, is symmetric.

4.2 Relation-enhanced Representation via Graph Learning

Here, our objective is to integrate relational information into the matching prediction model based on the constructed heterogeneous knowledge graph, enriching the target case to enhance matching comprehensiveness. As shown in Fig. 2, with the inclusion of law articles, we can address the law article matching (LAM) task in parallel through multitask learning, which can be applied to align the prediction of the LCM task. Given a query case c^q and

a target case c^t, we introduce the model structure using the LCM task as an example.

Text Encoding The query case c^q and target case c^t are concatenated and fed into the legal text encoder (as with the same model in Sec. 4.1) to encode their joint semantic representation:

$$\mathbf{h}_c = \text{Encoder}(\text{concat}(c^q, c^t)). \quad (5)$$

Heterogeneous Graph Encoding In the LCM task, RedMatch extracts a subgraph centered around the target case c^t from the heterogeneous graph. Neighbor nodes within d hops, denoted as $\mathcal{N}^{(d)}(c^t)$, are ranked based on the edge weights of the neighbor node and the target case. The neighbor node can be a case node or law node. Then, top k neighbor nodes are aggregated using a heterogeneous graph transformer (HGT):

$$\mathbf{h}_c^{(l+1)} = \text{HGT}\left(\mathbf{h}_c^{(l)}, \text{TopK}(\mathcal{N}^{(d)}(c^t))\right), \quad (6)$$

where TopK is the rank function, $\mathbf{h}_c^{(l)}$ is the graph representation at the l-th layer, which is initialized with original node representation x_{c^t}. After propagating through L-layer HGT, we view $\mathbf{h}_c^{(L)}$ as the final graph representation.

Matching Score Calculation The semantic representation $\mathbf{h}_c$ and graph representation $\mathbf{h}_c^{(L)}$ are concatenated and fed into a task-specific MLP to predict the matching score s_c:

$$s_c = \text{MLP}\left([\mathbf{h}_c; \mathbf{h}_c^{(L)}]\right) \quad (7)$$

MLP is a three-layer perceptron with sigmoid activation functions. Then, the prediction model determines the matching prediction label $\hat{y}^c = \text{argmax}(s_c)$ and is trained by a cross-entropy loss:

$$\mathcal{L}_c = -\sum(\hat{y}^c, y^c). \quad (8)$$

For the LAM task, it shares the legal text encoder and HGT with the LCM task but predicts the law article matching score s_a using a separate MLP, trained with a cross-entropy loss $\mathcal{L}_a$.

4.3 Relation Navigation via Path Decomposition

To avoid failure from single-path retrieval, RedMatch introduces a path learning task to navigate the model in exploring multiple relation-based paths to target cases.

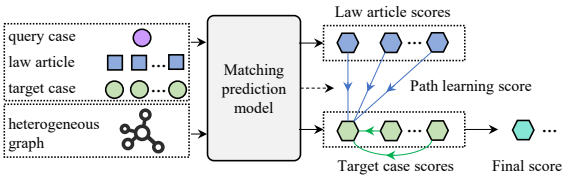

Figure 3: The inference of RedMatch.

Path Decomposition RedMatch explores potential paths connecting c^q to target case c^t through the intermediate node, which can be a case node or law node. Each path is decomposed into ordered node pairs, forming a sequence of sub-paths. For instance, a path $c^q \to a_1 \to c^t$ is decomposed into (c^q, a_1) and (a_1, c^t). To encourage the model to select nodes towards better matching direction, node pairs are categorized as positives and negatives. In positive pairs set $\mathbb{P}$, both nodes are matching. In negative pairs set $\mathbb{N}$, at least one node is not matching (details in Appendix A).

Path Learning Our goal is to ensure the model assigns higher scores to positive pairs than to negative pairs. For each node pair (u, v), we can compute the score with the matching prediction model described in Sec. 4.2. However, since the use of task-specific MLP, the score $s_{uv} \in \mathbb{R}^L$. We apply a projection function $\psi(\cdot)$ and a softmax operation to convert it into a scalar value:

$$\tilde{s}_{uv} = \text{softmax}(\psi(s_{uv})). \quad (9)$$

To adaptively navigate the model, we introduce a dynamic margin loss:

$$\mathcal{L}_m = \max(0, \delta(v, v') - \tilde{s}_{uv} + \tilde{s}_{uv'}), \quad (10)$$

where $(u, v) \in \mathbb{P}$, $(u, v') \in \mathbb{N}$, $\delta(v, v')$ is the dynamic margin, which is computed based on their disparity on the annotated matching label, $\delta(v, v') = |y_v - y_{v'}|$.

4.4 Training and Inference

In the training stage, the model jointly optimizes the LCM, LAM and path learning task by minimizing the sum of their losses $\mathcal{L} = \mathcal{L}_c + \mathcal{L}_a + \mathcal{L}_m$.

In the inference stage shown in Fig. 3, given a query case and several target cases H_c. LCM determines the relevance between each paired query-target case. RedMatch processes them together as follows: 1) calculate the matching scores between each paired query-target case. 2) calculate the path scores $\tilde{s}_{ij}$ between target cases. 3) For i-th target case, fuse the matching score s_i with multiple

paths ending at the case i with path scores as the weight to obtain the final score, which is then used to predict the matching label.

$$\hat{y}_i^c = \text{argmax}(s_i + \tau \cdot \sum_{j \in H_c, H_a} \tilde{s}_{ij} \cdot s_j), \quad (11)$$

τ is the temperature. By incorporating the LAM task, the model also computes path scores between target cases and law articles, and LAM labels are also predicted as above process. We provide an example in Fig. 6 and details in Appendix D for better illustration.

5 Experiments

5.1 Datasets and Metrics

The experiments are conducted based on three publicly available legal case matching datasets. LeCaRD (Ma et al., 2021) is a Chinese criminal case dataset with 107 query cases and 43,000 target cases, each paired with 30 manually annotated target cases assigned a 4-level relevance label (i.e., 0, 1, 2, 3). C3RD (Ye and Li, 2024) is a Chinese civil case dataset comprising 1,146 query cases, each associated with 100 candidate cases labeled with a binary relevance label (0 for mismatch, 1 for match). LeCaRD and C3RD follow the statutory system. COLIEE (Goebel et al., 2023) is an English legal benchmark following the caselaw system. It contains 959 query cases and a pool of 4,400 candidate cases. Following Yu et al. (2022), to reduce computational complexity, we include all matching cases for each query and randomly sample twice as many mismatched cases to form the candidate set. We provide data statistics in Tab. 6.

We randomly divide the dataset into training set, validation set and test set according to the ratio of 8: 1: 1. Following previous works (Yu et al., 2022; Sun et al., 2023), we employ Accuracy, Macro-Precision, Macro-Recall, and Macro-F1 as the evaluation metrics.

5.2 Baselines

We consider three types of baselines in this study.

1) Pretrained Embedding Methods. BGE is a pre-trained embedding model designed for general-purpose text retrieval tasks. It leverages supervised contrastive learning to align text representations effectively. **LegalBERT** refers to several BERT-based models pre-trained on legal texts, with specific models used in different datasets. **Lawformer** (Xiao et al., 2021b) is a Longformer-based model pre-trained on millions of Chinese legal cases, that captures representations of long legal documents. We feed both cases as input and perform mean pooling over the output to compute similarity.

2) LLM Methods. We test the performance of LLMs, including **GPT-4o mini** and **Qwen2.5-turbo**. We call the API and prompt the LLM to predict the matching labels between query cases and target cases. For example, on the Lecard dataset, the prompt is: *"You are a case similarity scoring assistant. Your task is to assess the similarity between two cases based on the following rules: 3 for highly similar, 2 for moderately similar, 1 for slightly similar, and 0 for not similar. Please output one of 0, 1, 2, or 3, and nothing else."*

3) Legal Case Matching Methods. SAILER (Li et al., 2023) adopts an asymmetric encoder-decoder to embed structural information of legal documents into dense vectors. **BERT-PLI** (Shao et al., 2020) captures paragraph-level semantics via BERT and RNN with attention, using an MLP to compute case relevance. **Law-Match** (Sun et al., 2023) introduces a causal learning framework that separates and reweights law-related and unrelated components of a case for better prediction. **CaseGNN** and **CaseLink** (Tang et al., 2023, 2024) are graph-based models that exploit legal structural and inter-case relationships via text-attributed graphs and inductive graph learning, respectively.

5.3 Implementation details

We finetune RedMatch using grid search on the test set with the Adam optimizer. The batch size is set to 4, and the learning rate is fixed at 3e-5. For law articles, we use the statutory texts from both PRC criminal and civil law for the Chinese dataset LeCaRD, C3RD, as well as the Courts Act and Rules of Canada caselaws for the English dataset COLIEE, following Tang et al. (2024). For query cases lacking law article citations, in the LeCaRD dataset, the most frequently cited articles among candidate cases are selected as the query case's articles. We evaluate the quality of the constructed graph using both human and automated evaluation in Appendix E. When aggregating neighbor information within the graph, a two-layer HGT is employed, the neighbor hop d is tuned from $\{1, 2, 3\}$ and the neighbor number k is tuned from $\{5, 10, 20, 50, 100\}$. During inference, the multi-path fusion temperature τ is tuned from $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$. The RedMatch implementation in the paper is based on Lawformer as

Models	LeCaRD				C3RD				COLIEE			
	Acc.(%)	P.(%)	R.(%)	F1(%)	Acc.(%)	P.(%)	R.(%)	F1(%)	Acc.(%)	P.(%)	R.(%)	F1(%)
Pretrained Embedding Methods												
BGE	60.68	60.67	59.67	59.86	96.47	93.05	88.66	90.69	76.32	74.62	77.24	75.01
LegalBERT	58.51	59.71	55.25	55.97	96.82	92.30	91.67	91.98	86.86	85.02	85.87	85.41
Lawformer	61.92	62.90	59.81	60.67	96.70	92.74	90.34	91.50	-	-	-	-
LLM Methods												
Qwen2.5-turbo	41.57	42.55	41.57	37.67	90.70	55.92	76.58	64.64	80.82	70.44	73.15	71.77
GPT-4o mini	34.70	38.21	34.70	28.93	92.22	61.11	45.83	52.38	80.88	79.92	56.96	66.52
Legal Case Matching Methods												
SAILER	62.54	62.88	64.05	62.93	96.65	92.40	90.48	91.41	77.58	75.23	77.13	75.82
BERT-PLI	64.09	63.63	64.58	64.03	95.43	90.33	85.76	87.86	82.48	80.24	82.03	80.92
Law-Match	61.61	63.23	59.88	60.68	96.69	91.84	91.53	91.68	-	-	-	-
CaseGNN	59.85	61.32	58.40	59.83	95.83	89.30	89.87	89.58	84.12	84.33	83.70	84.01
CaseLink	61.42	62.75	60.35	61.53	96.88	91.02	91.15	91.08	85.36	85.90	84.61	85.25
RedMatch	**67.18**†	**66.71**†	**65.21**†	**65.74**†	**97.75**†	**93.79**†	**91.87**	**92.88**†	**87.46**†	**85.73**†	**86.62**†	**86.14**†

Table 1: Performance comparisons between RedMatch and the baselines. The boldface represents the best performance. '†' indicates the model outperforms all baselines significantly in paired t-test at $p < 0.05$ level (with Bonferroni correction). '-' denotes results are not available.

Models	LeCaRD		C3RD	
	Acc.(%)	F1(%)	Acc.(%)	F1(%)
RedMatch (Full)	**67.18**	**65.74**	**97.75**	**92.88**
w/o HGT	66.26	65.04	96.13	90.08
w/o LAM	64.09	62.30	96.66	92.05
w/o margin loss	64.09	62.62	95.45	87.33
w/o multi-path	66.63	65.30	97.10	92.52

Table 2: We perform the ablation study on LeCaRD and C3RD datasets by removing: the HGT graph representation (w/o HGT), law article matching task (w/o LAM), dynamic margin loss (w/o margin loss), matching with multiple paths (w/o multi-path).

the legal text encoder. Baseline models are fine-tuned with optimal parameters reported in their papers. More implementation details are provided in Appendix C.

5.4 Main Results

Results of Legal Case Matching From Tab. 1, we have the following observations: 1) Pretrained models generally underperform compared to specialized legal case matching methods on most datasets, highlighting that legal cases should not be treated as generic long texts and require domain-specific considerations. 2) Our method outperforms baselines on three datasets (including Chinese and English language), demonstrating the effectiveness of using relations to navigate matching and the generalizability of our method. 3) Performance on C3RD exceeds that on LeCaRD and COLIEE, as C3RD contains richer and more useful case analysis. 4) LLMs perform poorly on most datasets, especially on statutory law datasets (e.g., LeCaRD, C3RD), which suggests that LLMs lack sufficient law article knowledge to analyze cases.

Results of Ablation Experiment From Tab. 2, we can conclude that: 1) Overall, the full RedMatch achieves the best performance across both datasets, with an F1 score of 65.74% on LeCaRD, and 92.88% on C3RD. 2) Removing the graph representation leads to a performance decrease, which proves relation integration is beneficial. 3) The LAM task can align the results of the LCM task through multitask learning. 4) The margin loss designed for path learning plays a crucial role in improving model performance. It prepares the model for matching score optimization in the inference stage, while strengthening its matching prediction ability. 5) Incorporating multiple matching paths to predict the final matching case can improve accuracy. We provide further ablation studies on graph design and GNN choice in Appendices F.1 and F.

5.5 Analysis Study

Results of Law Article Matching Tab. 3 illustrates the performance of RedMatch and its ablations and baselines on the LAM task. We have the following observations: 1) RedMatch achieves the best performance, with an F1 score of 96.36% on LeCaRD, and 95.14% on C3RD, highlighting its application utility. 2) Ablation results reveal

Models	LeCaRD		C3RD	
	Acc.(%)	F1(%)	Acc.(%)	F1(%)
BGE	80.00	66.94	92.36	91.59
LegalBERT	65.71	61.96	90.61	89.65
Lawformer	65.71	64.29	92.68	91.90
Qwen2.5-turbo	84.20	86.14	66.90	20.24
GPT-4o mini	78.16	82.83	63.20	32.60
RedMatch(Full)	**96.77**	**96.36**	**95.46**	**95.14**
w/o HGT	94.27	93.65	92.83	92.17
w/o margin loss	91.06	90.82	91.92	91.79
w/o multi-path	94.62	94.06	94.94	94.38

Table 3: Performance of law article matching task on LeCaRD and C3RD. Best results are marked bold.

Models	Acc.(%)	P.(%)	R.(%)	F1(%)
Lawformer	61.92	62.90	59.81	60.67
+ RedMatch	$65.63^{+3.71}$	$65.15^{+2.25}$	$64.63^{+4.82}$	$64.85^{+4.18}$
LegalBERT	58.51	59.71	55.25	55.97
+ RedMatch	$65.33^{+6.82}$	$64.94^{+5.23}$	$65.12^{+9.87}$	$64.54^{+8.57}$
SAILER	62.54	62.88	64.05	62.93
+ RedMatch	$71.52^{+8.98}$	$72.79^{+9.91}$	$69.93^{+5.88}$	$70.39^{+7.46}$

Table 4: Performance of RedMatch with different base models on LeCaRD.

the pivotal role of the margin loss. We attribute this to the big enhancement of relation navigation in handling the concise and complex semantics of law articles. In RedMatch, the matching path of "case-law" can be expanded to "case-case-law".

Base Model Replacement Tab. 4 shows the performance of RedMatch when integrated with different base models as the legal text encoder. The results indicate that RedMatch exhibits generalizable improvements across all base models. Specifically, RedMatch improves SAILER significantly, with Acc improving from 62.54% to 71.52% and F1 increasing from 62.93% to 70.39%.

Hyperparameter Exploration We explore the impact of key hyperparameters in RedMatch: neighbor number k, neighbor hop d, and temperature τ. As shown in Fig. 4, when aggregating neighbor information, the parameters k and d need to balance useful and redundant information integration. The best performance is achieved when $k = 20$ and $d = 1$. The temperature τ, which controls the strength of multi-path fusion, significantly affects performance. A moderate value of $\tau = 0.2$ yields the best results.

Consumption and Latency Analysis Tab. 5 presents the analysis of inference time and memory footprint for different models. RedMatch shows a

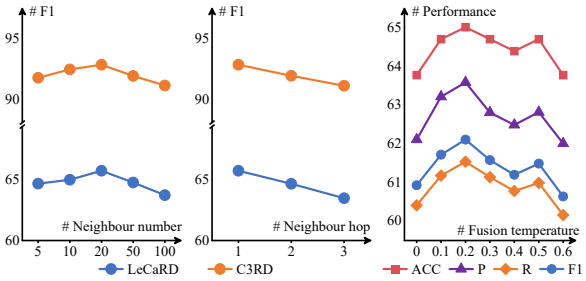

Figure 4: Performance exploration on neighbor number k, neighbor hop d, and temperature τ.

Models	Time	Space
Lawformer	0.034 s	1.35 G
GPT-4o mini	0.960 s	-
Law-Match	0.041 s	1.49 G
RedMatch	0.046 s	1.51 G

Table 5: Analysis of consumption and latency.

slightly longer inference time than Lawformer and Law-Match, while remaining competitive. Its memory consumption is marginally higher than Law-Match and Lawformer. In contrast, the LLM-based GPT-4o mini relies on API services with no local memory usage but suffers from substantially higher latency. RedMatch also exhibits favorable training efficiency: on 6 NVIDIA A100-PCIE-40GB GPUs, it requires only 1.8, 4.4, and 3.7 hours for 20 epochs on LeCaRD, C3RD, and COLIEE, respectively. These results demonstrate that RedMatch is well-suited for practical deployment.

6 Conclusion

In this paper, we propose RedMatch, a relation-driven framework for legal case matching that incorporates case-case and case-law relations. Our approach constructs a heterogeneous graph to model these relations, which is then processed using a graph transformer to integrate relational information into the matching process. Furthermore, we introduce a path learning task with margin loss to navigate the matching process effectively, enabling the model to explore multiple matching paths and enhance flexibility. By employing multitask learning, RedMatch jointly addresses legal case and law article matching, improving the overall practical applicability. Extensive experiments on three publicly available datasets in both Chinese and English show that RedMatch outperforms state-of-the-art methods in both legal case and law article matching, highlighting its effectiveness and generalizability.

7 Limitations

In this section, we discuss the limitations of our work as follows:

- We limit the proposed approach to the judicial domain, where high-quality text data (e.g., case verdicts) are available, and relevant case analysis can be extracted directly from legal documents. Application to other fields may require manual annotations and adjustments to handle domain-specific variations.
- The RedMatch model relies on a multi-stage process, where the construction of case-case and case-law relations may be affected by the quality of input data, such as incomplete or unclear legal references. This may influence the accuracy of the final matching predictions. A possible solution could be incorporating more robust preprocessing or event extraction tasks to improve the input data quality.
- New cases/articles can be incorporated by embedding them with the pretrained text encoder and connecting them via similarity-based edges, but the graph encoder (HGT) requires retraining when the data distribution changes significantly.

8 Ethics Statement

As AI becomes more integrated into the legal field, ethical concerns arise, particularly since any error could have serious consequences (Wu et al., 2020). The RedMatch framework is designed to assist legal professionals by suggesting case matches rather than making final decisions. Human judgment remains crucial, and the system allows judges and lawyers to review and adjust the suggestions, ensuring oversight and minimizing risks of bias or misinterpretation.

References

Michael Bacchus. 2002. Strung out: Legal citation, the bluebook, and the anxiety of authority. *U. Pa. L. Rev.*, 151:245.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Paheli Bhattacharya, Kripabandhu Ghosh, Arindam Pal, and Saptarshi Ghosh. 2020. Methods for computing legal document similarity: A comparative study. *Preprint*, arXiv:2004.12307.

Sheng Bi, Zafar Ali, Meng Wang, Tianxing Wu, and Guilin Qi. 2022. Learning heterogeneous graph embedding for chinese legal document similarity. *Know.-Based Syst.*, 250(C).

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*.

Xueyuan Chen, Xiao Wei, Hang Yu, and Luo Xiangfeng. 2023. Regrl: An informative graph representation via hierarchical recursive learning for legal case recommendation. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.

Todd Coolidge. 2023. Common law vs. statutory law. https://coolidgelawfirmaz.com/common-law-v-statutory-law-criminal-defense-attorney.

Gregor Donabauer and Udo Kruschwitz. 2025. A reproducibility study of graph-based legal case retrieval. *Preprint*, arXiv:2504.08400.

Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Juliano Rabelo, Ken Satoh, and Masaharu Yoshioka. 2023. Summary of the competition on legal information, extraction/entailment (coliee) 2023. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, ICAIL '23, page 472–480, New York, NY, USA. Association for Computing Machinery.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Harvard Law Review. 2016. Chinese common law? guiding cases and judicial reform. *Harvard Law Review*, 129(8).

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, pages 2704–2710.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Legal Information Institute, Cornell Law School. 2020. Common law. Last updated May 2020.

A. D. E. Lewis, Albert Roland Kiralfy, and Mary Ann Glendon. 2025. Common law. *Encyclopedia Britannica*.

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.

Bang Liu, Di Niu, Haojie Wei, Jinghong Lin, Yancheng He, Kunfeng Lai, and Yu Xu. 2018. Matching article pairs with graphical decomposition and convolutions. *arXiv preprint arXiv:1802.07459*.

Yixiao Ma, Yunqiu Shao, Yueyue Wu, Yiqun Liu, Ruizhe Zhang, Min Zhang, and Shaoping Ma. 2021. Lecard: A legal case retrieval dataset for chinese law system. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2342–2348, New York, NY, USA. Association for Computing Machinery.

Yixiao Ma, Yueyue Wu, Qingyao Ai, Yiqun Liu, Yunqiu Shao, Min Zhang, and Shaoping Ma. 2023. Incorporating structural information into legal case retrieval. *ACM Transactions on Information Systems*, 42(2):1–28.

Neil MacCormick. 1994. *Legal reasoning and legal theory*. Clarendon Press.

Francesco Palermo and Karl Kössler. 2017. *Comparative federalism: constitutional arrangements and case law*, volume 19. Bloomsbury Publishing.

Liang Pang, Yanyan Lan, and Xueqi Cheng. 2021. Match-ignition: Plugging pagerank into transformer for long-form text matching. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1396–1405.

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.

Mark Carl Rom, Masaki Hidaka, and Rachel Bzostek Walker. 2022. *Introduction to Political Science*. OpenStax, Houston, Texas.

M. Saravanan, B. Ravindran, and S. Raman. 2009. Improving legal information retrieval using an ontological framework. *Artificial Intelligence and Law*, 17(2):101–124.

Yunqiu Shao, Jiaxin Mao, Yiqun Liu, Weizhi Ma, Ken Satoh, Min Zhang, and Shaoping Ma. 2020. Bert-pli: Modeling paragraph-level interactions for legal case retrieval. In *IJCAI*, pages 3501–3507.

Henry Small. 1973. Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for information Science*, 24(4):265–269.

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.

Yanran Tang, Ruihong Qiu, Yilun Liu, Xue Li, and Zi Huang. 2023. Casegnn: Graph neural networks for legal case retrieval with text-attributed graphs. *Preprint*, arXiv:2312.11229.

Yanran Tang, Ruihong Qiu, Hongzhi Yin, Xue Li, and Zi Huang. 2024. Caselink: Inductive graph learning for legal case retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2199–2209.

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

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Jiawei Wang, Yuquan Le, Da Cao, Shaofei Lu, Zhe Quan, and Meng Wang. 2024. Graph reasoning with supervised contrastive learning for legal judgment prediction. *IEEE Transactions on Neural Networks and Learning Systems*.

Yiquan Wu, Kun Kuang, Yating Zhang, Xiaozhong Liu, Changlong Sun, Jun Xiao, Yueting Zhuang, Luo Si, and Fei Wu. 2020. De-biased court's view generation with causality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 763–780.

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

Chaojun Xiao, Xueyu Hu, Zhiyuan Liu, Cunchao Tu, and Maosong Sun. 2021b. Lawformer: A pre-trained language model for chinese legal long documents. *Preprint*, arXiv:2105.03887.

Yaming Yang, Ziyu Guan, Jianxin Li, Wei Zhao, Jiangtao Cui, and Quan Wang. 2021. Interpretable and efficient heterogeneous graph convolutional network. *IEEE Transactions on Knowledge and Data Engineering*, 35(2):1637–1650.

Fuda Ye and Shuangyin Li. 2024. Milecut: A multi-view truncation framework for legal case retrieval. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 1341–1349, New York, NY, USA. Association for Computing Machinery.

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 *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 657–668.

Yiming Zeng, Ruili Wang, John Zeleznikow, and Elizabeth Kemp. 2005. Knowledge representation for the intelligent legal case retrieval. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 339–345, Berlin, Heidelberg. Springer Berlin Heidelberg.

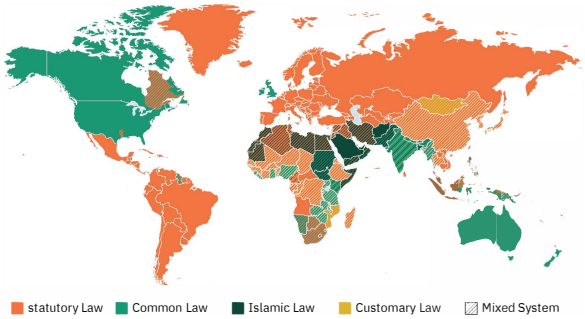

Figure 5: This map shows the different types of legal systems in place around the world (Rom et al., 2022).

Dataset	Query case	Case base	Avg. case	Avg. law article
LeCaRD	107	43,000	10.39	1.08
C3RD	1146	114,600	11.43	2.95
COLIEE	959	4,400	4.68	5.20

Table 6: Dataset details.

Yunong Zhang, Xiao Wei, and Hang Yu. 2024. Hdljp: A hierarchical dependency-based legal judgment prediction framework for multi-task learning. *Knowledge-Based Systems*, 299:112033.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. How does nlp benefit legal system: A summary of legal artificial intelligence. *arXiv preprint arXiv:2004.12158*.

A Positive-Negative Node Pair Splitting Strategy

In our heterogeneous graph, relations are symmetric. Additionally, both the LCM task and the LAM task are symmetric because they compute paired scores. For a node pair, if both nodes have matching labels, assigning a high score to this pair will correctly guide the model towards recognizing matches. Therefore, such pairs are considered positive. However, if at least one node has a mismatch label, assigning a high score to this pair would mislead the model. Consequently, these pairs are regarded as negative.

B Case Study

Furthermore, we present a case study of the legal case matching (LCM) and law article matching (LAM) tasks using RedMatch, as illustrated in Tab. 7. The first row shows a successful match, where RedMatch correctly predicted a complete match with a label of "3" between the query case involving a defendant driving under the influence of alcohol and the target case. The second row demonstrates a successful mismatch detection, where RedMatch correctly assigned a label of "0" to a completely unrelated case. Finally, the third row illustrates another correct match, where RedMatch accurately linked the query case to a relevant law article with a predicted label of "3". These case studies highlight the reliability of RedMatch in accurately assessing both case-case and case-law relevance.

C More Implementation Details

In our experiments, in baseline BGE, we use BGE-zh for the Chinese datasets LeCaRD and C3RD, and BGE-en for the English dataset COLIEE. In baseline LegalBERT, for the Chinese datasets LeCaRD and C3RD, we utilize crimeBERT and civilBERT from OpenCLaP[2], respectively. For the English dataset COLIEE, we employ the Legal-BERT[3], which is trained on diverse English legal texts, including legislation, court cases, and contracts. Lawformer cannot be applied to the COLIEE dataset due to the absence of an English pretrained embedding. Law-Match is also not applicable because the COLIEE dataset does not contain statutory law articles. We also made efforts to implement other related methods. However, some works could not be reproduced due to objective constraints. First, legal case retrieval methods (Yang et al., 2021; Bi et al., 2022; Qin et al., 2024) differ fundamentally from our legal case matching setting: the former retrieve top-k similar cases based on similarity scores, while the latter classify pairwise relevance levels (e.g., 0–3). This difference leads to incompatible model architectures and evaluation protocols. Second, several methods could not be implemented due to the lack of released code, required data files, or implementation details. Nevertheless, we still cite and discuss these works to better position our work. We conducted 3-fold cross-validation on LeCaRD and observed consistent performance across splits (F1 scores: 65.78, 65.48, 65.90).

For the ablation experiments, the three settings (w/o LAM, w/o margin loss, and w/o multi-path) reflect a **progressive degradation** of our model design. Among them, **w/o LAM** has the most substantial structural impact, rendering the multitask learning framework, margin loss, and multi-path fusion inoperative. Removing the LAM task disables the prediction of law article scores, thereby break-

[2]https://github.com/thunlp/OpenCLaP
[3]https://huggingface.co/nlpaueb/legal-bert-base-uncased

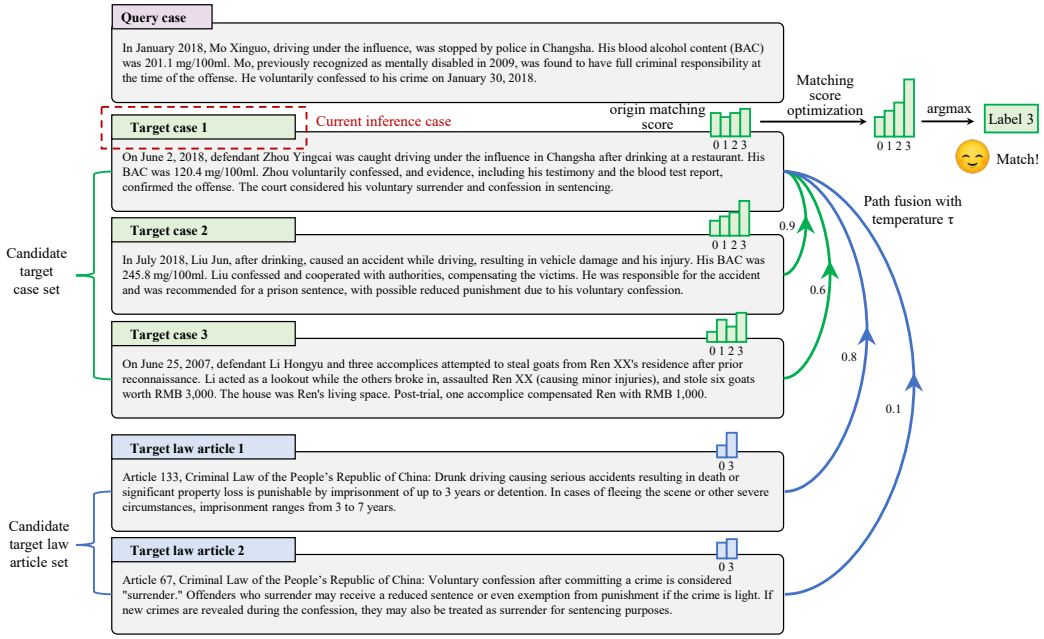

Figure 6: Detailed inference process illustrated with a real example from LeCaRD dataset.

ing transitions through law articles (i.e., case → law → case), which are essential for heterogeneous graph matching. **W/o margin loss** also disables the path learning task and multi-path fusion since margin loss is the core supervision signal for learning diverse matching paths. Without it, the model loses the ability to explore alternative paths during inference. Finally, **w/o multi-path** removes the multi-path fusion during inference but retains margin loss during training. While this setting weakens our design, the model still benefits from path-level learning signals.

D Inference Discussion

In legal practice, the LCM task typically involves a query case and multiple candidate target cases. The goal is to compute relevance scores for each paired query-target case and select the top-k highest-scoring ones. To reduce computational costs, candidate target cases are usually pre-filtered (e.g., using BM25) rather than drawn from the entire case database. Therefore, Our method RedMatch computes scores for all candidate target cases in one go while considering their mutual interactions, which does not increase computational overhead. In our implementation, the two datasets LeCaRD and C3RD provide pre-defined candidate target case sets for each query case, while for COLIEE, we use BM25 for initial filtering to obtain the candidate set.

E Evaluation of Graph Quality

To validate the quality of the heterogeneous graph constructed by RedMatch, we conduct a targeted evaluation on three edge types: (1) case–case representing similar cases, (2) case–law representing relevant law articles to cases, and (3) law-law representing law articles that are frequently cited together.

E.1 Evaluation Setup and Criteria

We randomly sample 200 case-case pairs, 200 case-law pairs, and 200 law-law pairs from the constructed graph across diverse crime types and statutes. Each sample was evaluated by both human annotators and a large language model (GPT-4o mini).

Each pair was assessed along the following dimensions: (1) Legal Relevance (LR): Whether the pair shares substantial legal basis, including applicable statutes, legal reasoning, or judgment logic. (2) Practical Usefulness (PU): Whether the link would be helpful for legal professionals in tasks such as case referencing or statutory application. (3) Final Decision: A binary judgment on whether the link is valid, i.e., whether the two cases form a similar-case pair, or whether the law article is relevant to the case. All evaluations use a 1–5 scale for the two dimensions and a Yes/No final judgment.

For human evaluation, we invite 10 Ph.D. law students to annotate the sampled links. Each pair

Task	Query case	Target case or Target law article	Predicted label
LCM	In the afternoon of August 24, 2018, the defendant Sun Xiaozhai **drunkenly drove** a black small ordinary bus with license plate No. Anhui S ××××× from Sun Wafang Village, Huagou Town, Eddy County, to Sun Heilou Natural Village.	The trial judgment found that: July 9, 2015 at 21:00, in Kenli County Shengxing Road and Jingyuan Road intersection, the defendant Ding Moumou **drunk driving** Lu EKXXX3 sedan and Chi Moumou driving Lu E3XXX7 JIANGZUO light ordinary truck collided, Ding Moumou driving to escape.	3
	In the afternoon of August 24, 2018, the defendant Sun Xiaozhai **drunkenly drove** a black small ordinary bus with license plate No. Anhui S ××××× from Sun Wafang Village, Huagou Town, Eddy County, to Sun Heilou Natural Village.	Public Prosecution alleges that at 12:00 on October 3, 2015, the defendant Zhao Yingzhi, together with Zhao Xiaokai (check no such person) rode a motorcycle to Jize County Xiaozhai Town Center Health Center, will be parked in the hospital yard of the health center of Meng Mou hall door east of the Xinfei brand of electric tricycles locks picking and then **stole**.	0
LAM	In the afternoon of August 24, 2018, the defendant Sun Xiaozhai **drunkenly drove** a black small ordinary bus with license plate No. Anhui S ××××× from Sun Wafang Village, Huagou Town, Eddy County, to Sun Heilou Natural Village.	Article 133-1 of the Criminal Law of the People's Republic of China states that anyone who drives a motor vehicle on a road under any of the following circumstances shall be sentenced to a term of imprisonment and fined: (a) chasing and racing under aggravating circumstances; **(b) driving a motor vehicle while intoxicated**.	3

Table 7: Case study of legal case matching (LCM) and law article matching (LAM). The boldface represents key information for matching.

Type	Evaluator	LR (1–5)	PU (1–5)	Valid Link Ratio (%)
case-case	Human	4.09	4.18	91.50
case-case	GPT-4o mini	4.32	4.04	95.45
case-law	Human	4.95	4.98	98.00
case-law	GPT-4o mini	4.88	4.79	96.97
law-law	Human	3.85	4.36	82.00
law-law	GPT-4o mini	4.37	4.20	85.45

Table 8: Evaluation results of graph edges by human annotators and GPT-4o mini

was independently rated by at least two annotators, and the final decision was determined by majority vote. All annotations were done blind to the source model and edge type. For LLM evaluation, we use GPT-4o mini with the following prompt: *"You are a legal expert assistant. I will give you either a pair of legal cases or a case and a law article. Please evaluate: (1) Legal Relevance (1–5) (2) Practical Usefulness (1–5) (3) Final Decision: Is this a valid link? (Yes/No)"*

E.2 Results of Graph Quality

We present the average scores and valid link ratios for both edge types evaluated by human annotators and GPT-4o mini in Tab. 8. The case-law relationships are derived directly from the actual articles cited in real cases, resulting in near-perfect evaluation scores. The case-case and law-law relationships are constructed based on our predefined thresholds (0.9 and 0.1, respectively), and the resulting edges also achieved high-quality evaluation scores. The results show high agreement between human and LLM evaluations, confirming the quality of the constructed heterogeneous graph.

F Further Study on Graph Module

We further analyze RedMatch's design choices by examining the effects of graph initialization, graph type, and graph model.

F.1 Graph Initialization

Unlike general graph initialization that relies solely on embedding-based similarity, we use a multidimensional edge weighting strategy that combines semantic and judgment similarity for case-case edges, citation ranking for case-law edges, and co-occurrence frequency for law-law edges to better capture legal relationships and reduce dependence on surface-level similarity.

We further validate the importance of this design via an ablation study on the LeCaRD dataset. Specifically, we compare RedMatch with two simplified variants: (1) w/o type-specific: Removes differentiated edge types and builds the graph purely on node similarity. (2) w/o weight: Uses a uniform graph where all edge weights are set to 1. As shown in Tab. 9, removing either type-specific design or edge weighting leads to a noticeable drop in performance, especially for w/o weight, which

Method	Acc. (%)	P. (%)	R. (%)	F1. (%)
RedMatch	67.18	66.71	65.21	65.74
w/o type-specific	65.02	64.59	62.54	63.11
w/o weight	59.99	56.73	50.26	54.97

Table 9: Ablation study on the type-specific weight design.

Method	Acc.(%)	P.(%)	R.(%)	F1.(%)
LCM (heterogeneous)	67.18	66.71	65.21	65.74
LCM (homogeneous)	57.25	60.42	57.61	52.01
LAM (heterogeneous)	96.77	96.76	96.00	96.36
LAM (homogeneous)	72.92	53.07	60.63	54.62

Table 10: Ablation study on graph types.

highlights the critical role of our edge design in modeling legal relevance.

F.2 Graph Type: Heterogeneous vs. Homogeneous

To assess the impact of using a unified heterogeneous graph, we construct separate homogeneous graphs for LCM and LAM. These graphs only retain intra-type edges and are encoded using standard GCNs. The performance comparison on the LeCaRD dataset is shown in Tab. 10. The results show that separating the tasks into individual homogeneous graphs leads to a significant drop in performance, which confirms the effectiveness of the proposed heterogeneous graph structure in modeling complex legal relationships. This highlights two key advantages of the heterogeneous graph design. (1) Nodes in the heterogeneous graph can aggregate features from both similar cases and relevant law articles, leading to richer representations. (2) A unified graph provides shared structural context, which enhances joint learning and better supports the interdependencies between LCM and LAM tasks.

F.3 GNN Model

We also explore several GNN methods from legal judgment prediction tasks for modeling the heterogeneous graph (Zhang et al., 2024; Tong et al., 2024; Chen et al., 2023; Wang et al., 2024). Our original model uses the Heterogeneous Graph Transformer (HGT) to enable type-specific, long-range message passing. As alternatives, we test (1) removing graph encoding entirely, and (2) replacing HGT with HGCN (Yang et al., 2021). In addition, we also directly apply general heterogeneous GNNs (HAN, GTN, MAGNN) to implement the LCM and LAM tasks for comparison. From Tab. 11, removing graph encoding reduces performance, showing the importance of graph structure. HGT outperforms HGCN due to its attention-based modeling of global dependencies. Directly applying general GNNs without the RedMatch framework or task-specific graph design yields much lower accuracy and F1-scores, highlighting that both the framework and domain-aware graph construction are essential for effective legal matching.

Method	Acc.(%)	P.(%)	R.(%)	F1.(%)
RedMatch (HGT)	67.18	66.71	65.21	65.74
RedMatch (HGCN)	66.47	64.37	63.69	63.48
RedMatch (w/o HGT)	66.26	64.12	62.88	65.04
HAN	47.23	55.45	64.04	42.20
GTN	47.58	55.82	64.99	42.58
MAGNN	47.56	55.80	64.94	42.56

Table 11: Ablation study on graph model.

G Distinctions from CaseLink

To better clarify the differences between our work and CaseLink (Tang et al., 2024). Specifically, we make the following distinctions:

- **Our core motivation is different from that of CaseLink.** CaseLink is designed for case law systems (e.g., UK and US) where statutory law is not explicitly involved, focusing on case-case connections for inductive learning. In contrast, our work is in line with the trend of legal system integration (i.e., case law and statutory law), aiming to leverage diverse legal relationships for graph representation and matching path decomposition. This enables our model to learn a novel transition ability between relevant elements in the matching process, such as query case $\rightarrow$ relevant law article $\rightarrow$ target case, which increases the chance of hitting a matching case.

- **Our work addresses a key limitation of CaseLink.** While CaseLink constructs a graph using case and charge information, it treats all nodes as the same type, forming a homogeneous graph. This results in overly simplistic information aggregation during the GNN process, leading to an issue where edge weights become incomparable due to significant variations. CaseLink acknowledges this drawback in its discussion section. In the legal domain, different types of legal texts follow

distinct writing conventions, making it crucial to differentiate and aggregate them appropriately. To address this, we construct a heterogeneous knowledge graph that explicitly models two types of nodes and three types of edges, enabling more fine-grained graph learning and resolving the comparability issue.

- **Our work is a comprehensive extension of CaseLink, addressing all the limitations discussed in its paper.** (1) Overcoming inductive learning constraints: Inductive learning in CaseLink introduces limitations in reference connections. We address this by constructing a global graph based on the case base while excluding the query case, thus avoiding such restrictions. (2) Improved edge weighting strategy: CaseLink determines edge weights solely based on general quantitative measurements. We extend this by introducing type-specific weighting mechanisms grounded in legal considerations. (3) Solving the edge weight imbalance issue: By leveraging a heterogeneous graph structure, we resolve CaseLink's issue of large, incomparable variations in edge weights.