Information Processing and Management 61 (2024) 103663

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Legal Judgment Prediction via graph boosting with constraints

Suxin Tong, Jingling Yuan *, Peiliang Zhang, Lin Li

School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan, 430070, Hubei, China

ARTICLE INFO

Keywords:
Legal judgment prediction
Multi-label learning
Consistency constrained learning
Graph neural networks
Multi-task learning

ABSTRACT

Legal Judgment Prediction (LJP) is a multi-task multi-label problem in the civil law system, involving the prediction of law articles, charges, and terms of penalty based on fact descriptions. However, most existing research approaches LJP as a single-label scenario, neglecting the correlations between multiple labels and failing to consider cross-task consistency constraints in a multi-label scenario. Moreover, although previous multi-task studies have proposed expert models and coarse-grained topology construction for inter-task relationships, the former neglects rich information exchange among different tasks, and the latter, if one task's prediction is inaccurate, will affect subsequent tasks. This paper has designed legal label graphs and proposed a novel graph boosting with constraints framework, GJudge, for legal judgment prediction to address these limitations. The framework comprises a multi-perspective interactive encoder and a multi-graph attention consistency expert module. The encoder utilizes bidirectional LSTM, gated attention units, cross attention, and graph attention networks to integrate fact descriptions and label similarity relationships information from legal label graphs for multi-perspective interactive encoding. The expert module utilizes the multiple expert networks and the multi-graph attention network to differentiate between confusing labels and ensure consistent constraints across tasks, this is achieved through the fusion of label consistency constraints and confusion relationships information in the legal label graphs. Experimental results on two real-world datasets across different tasks show an improvement in F1 scores ranging from at least 0.93% to a maximum of 2.97%, illustrating the effectiveness of GJudge compared to the state-of-the-art model.

1. Introduction

In recent years, the legal domain has experienced a profound transformation due to the emergence of extensive repositories containing high-quality legal documents. This abundance of legal data has created numerous opportunities for the implementation of a diverse set of innovative technologies across various aspects of legal text processing, including Legal Judgment Prediction (LJP), evidence extraction, similar case matching, etc. The application of these technologies not only streamlines repetitive tasks but also significantly improves the efficiency of judicial processes.

One of the most prominent and crucial tasks within this domain is LJP. For jurisdictions following a civil law system, this task entails automatically predicting the legal case outcome, including law articles, charges, and terms of penalty, based on the fact descriptions and legal knowledge. The application of LJP contributes to enhancing fairness and transparency while reducing litigation costs, increasing success rates, and preventing unnecessary disputes.

With recent advancements in artificial intelligence technology, significant research efforts in LJP have focused on constructing effective neural network models. These models incorporate legal attributes and data-driven approaches (Hu et al., 2018), consider

* Corresponding author.
E-mail addresses: suxin_tong@whut.edu.cn (S. Tong), yjl@whut.edu.cn (J. Yuan), zhangpl109@whut.edu.cn (P. Zhang), cathylilin@whut.edu.cn (L. Li).

https://doi.org/10.1016/j.ipm.2024.103663
Received 15 October 2023; Received in revised form 18 December 2023; Accepted 16 January 2024
Available online 25 January 2024

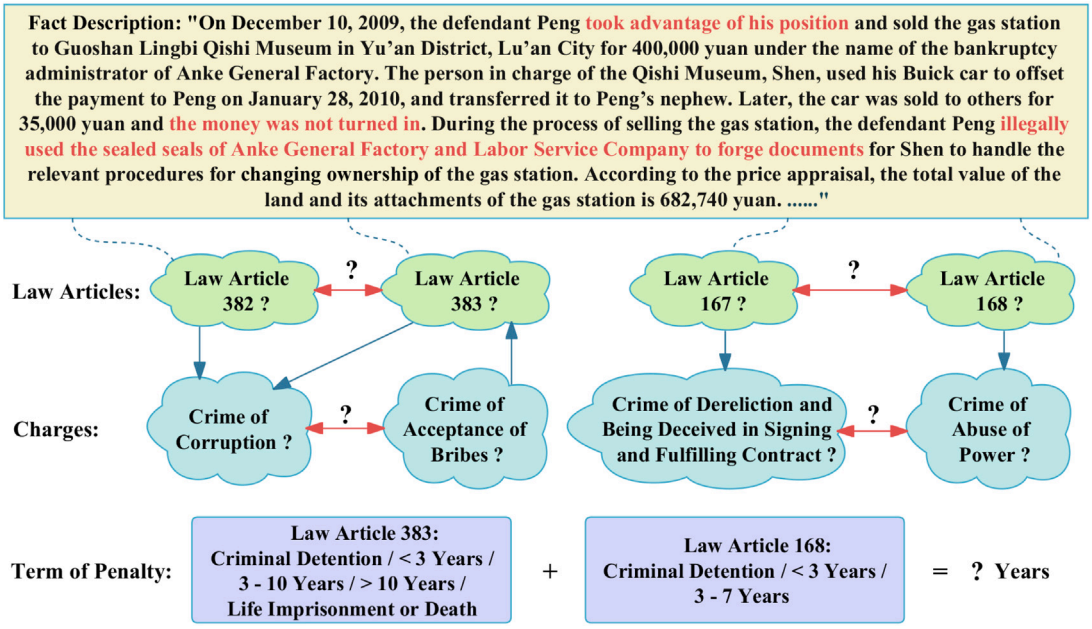

Fig. 1. A multi-label LJP relationship graph. (The Ground-truth: Law Articles: [383, 382, 168], Charges: [Crime of Corruption, Crime of Abuse of Power], Term of Penalty: [3 Years]. Article 382 provides an elucidation of the crime of corruption, while Article 383 establishes the penalties for such offenses. The crimes of corruption and acceptance of bribes are mutually confusable charges, and the punishment for acceptance of bribes is referenced from Article 383.)

dependencies among different subtasks (Zhong et al., 2018), and employ constitutive elements techniques (Zhao et al., 2022), among others. While these research endeavors have made notable progress, most studies primarily focus on single-label modeling and neglect cross-task consistency constraints. Consequently, current research efforts in LJP face two primary challenges:

Multi-label Relationship Learning: In comparison to single-label models, which only consider the one-way relationship between each label and the sample, assigning a single label to each sample, multi-label models need to consider the mutual relationships between multiple labels and predict multiple labels for each sample. As depicted in Fig. 1, in the context of multi-label legal judgment prediction, each sub-task of a case can be linked to multiple labels, and there might be interconnections and interdependencies between different task labels, thereby elevating the intricacy of model encoding. This complexity necessitates stronger expressive and reasoning abilities in the model.

Cross-Task Consistency Constrained Learning: In the context of multi-task learning, each data sample is linked to a collection of label information, which may have imbalanced distributions across different tasks. This imbalance challenges the model's ability to learn constraints associated with rare labels. Previous multi-task studies proposed expert models (Ma et al., 2018; Tang et al., 2020) and coarse-grained topology construction (Wu et al., 2022; Zhong et al., 2018) for inter-task relationships, the former neglects rich information exchange among different tasks, and the latter, if one task's prediction is inaccurate, will affect subsequent tasks.

To tackle the challenges mentioned above, we introduce a novel graph boosting with constraints framework for legal judgment prediction, GJudge, which consists of a multi-perspective interactive encoder and a multi-graph attention consistency expert module. The multi-perspective interactive encoder fuses fact descriptions and label relations by utilizing multi-perspective interactive encoding. The multi-perspective interactive encoding synthesizes information from multiple perspectives to provide a more comprehensive input representation, thereby enhancing the robustness and generalization ability of the framework. The expert module consists of multiple expert networks and a multi-graph attention network, which differentiates between confusion labels and ensures consistency constraints in cross-tasks by exploiting the correlation information among multiple labels in different tasks. By explicitly modeling the consistency constraints between labels, treating labels as nodes, and capturing label correlations through edges, the framework achieves improved performance. The paper makes the following noteworthy contributions:

- Our work is a pioneering attempt to address the problem of multi-label and cross-task relationships in the domain of LJP. While previous research has focused on either multi-label or cross-task relationships, our framework bridges this gap by providing a comprehensive approach that encompasses both aspects.
- We propose an innovative multi-perspective interactive encoder that effectively learns and integrates the relationships between multiple labels by amalgamating fact descriptions and label relevancy information.
- We develop a novel multi-graph attention consistency expert module, leveraging graph neural networks to differentiate between confusion labels and ensuring consistency constraints in cross-tasks, thereby further augmenting the efficacy of multi-task learning.
- Comprehensive experiments conducted on two real-world datasets affirm the effectiveness and efficiency of our legal judgment prediction framework, encompassing both single-label and multi-label scenarios.

2. Related work

In this section, we conduct a review of the pertinent literature within three key areas: legal judgment prediction, multi-task multi-label learning, and graph neural networks.

2.1. Legal judgment prediction

Legal Judgment Prediction (LJP) represents a vital research endeavor in the realm of intelligent judicial services. In the early stages, research endeavors predominantly centered on the examination of legal cases in specific contexts through the application of mathematical and statistical algorithms, as exemplified by previous studies (Kort, 1957; Nagel, 1963; Segal, 1984). However, with recent advancements in deep learning and neural networks, an increasing number of studies have utilized deep learning techniques for LJP (Lyu et al., 2022; Wang, Fan et al., 2019; Yang et al., 2022). Numerous investigations have delved into various facets of LJP. For example, Zhong et al. (2018) incorporated multiple subtasks and their directed acyclic graph dependencies into judgment prediction. Yang et al. (2019) introduced a multi-perspective bi-feedback network based on a sub-task topology structure and introduced a collocation attention mechanism. Xu et al. (2020) presented a graph distillation operator to autonomously discern subtle disparities among perplexing law articles, and employed attention mechanisms to extract distinguishing features from factual descriptions. Yue et al. (2021) partitioned factual descriptions into different episodes based on intermediate sub-task results and utilized them for predictions in other sub-tasks. Wu et al. (2022) designed a LJP framework that is rationale-based, where the framework generates reasoning derived from factual descriptions and employs this reasoning to predict judgments based on both the factual descriptions and the generated rationales. Feng et al. (2022) developed an event-based prediction model that incorporates constraints, used event extraction and hand-crafted constraints to improve LJP.

It can be concluded that the aforementioned studies have not taken into account the multi-label scenario, and there is a lack of research addressing classification decisions and relationships among multi-label. For example, the approach presented by Yue et al. (2021) is unable to handle conflicts and ambiguities that may arise in multi-label situations, and there are also questions about how to integrate multi-label results and select effective information for use in other subtasks. The model developed by Feng et al. (2022) is limited to the prediction based on a single event, and does not give how to fuse and extract the event information based on multiple events, and how to constrain the tasks between each other when multiple events occur. Our work does not merely represent an incremental step in the realm of single-label cross-task methodologies. Instead, we focus our efforts on a relatively uncharted territory within the domain of LJP, specifically addressing the issue of multi-label prediction while considering cross-task relationships. This decision is grounded in the practical reality of LJP scenarios, which often involve multiple labels, rendering single-label models insufficient for real-world applications.

2.2. Multi-task multi-label learning

Multi-task multi-label learning is a challenging problem that involves handling multiple tasks and multiple labels simultaneously. It typically requires effective strategies for feature sharing (Liang et al., 2023; Sun et al., 2020; Zhu et al., 2023) and learning label dependencies (Hang et al., 2022; Wang, Liu et al., 2019; Wang et al., 2021), which enhance the model's generalization and robustness. Feature sharing refers to the process of exploiting the commonalities and differences among tasks, while label dependencies refer to the relationships among labels within or across tasks. In recent years, several methods have emerged to tackle these challenges. For instance, Ma et al. (2018) designed a Multi-gate Mixture-of-Experts (MMoE) structure, which explicitly discerns task relationships from the available data. Ma et al. (2019) proposed the concept of SubNetwork Routing, enabling adaptable parameter sharing while preserving the computational efficiency of conventional multi-task neural network models. Tang et al. (2020) developed the Customized Gate Control (CGC) and progressive layered extraction models, which distinctly segregate the shared and distinct parameters within tasks to prevent conflicts arising from intricate task correlations. Hsieh and Tseng (2021) enhanced the effectiveness of multi-task learning by creating additional tasks through the amalgamation of task labels. Sadat and Caragea (2022) applied a multi-task learning strategy to enhance topic classification by integrating keyword labeling as an auxiliary task. However, to the best of our knowledge, there is still a dearth of research in the domain of multi-task multi-label learning for LJP. In this paper, we extend the prior research as presented in Tang et al. (2020), by introducing a novel framework that integrates cross-task consistency constraints and confusion label learning to address the challenges associated with multi-task multi-label learning.

2.3. Graph neural networks

Graph Neural Networks (GNNs) constitute a category of artificial neural networks that can handle data in the form of graphs. By learning node embeddings that incorporate both the node attributes and the graph configuration, GNNs can perform various tasks. Based on the diverse types of GNNs, there are five classes of existing methods: graph recurrent neural networks (Dai et al., 2017; Liu et al., 2022; Ruiz et al., 2020), graph convolutional networks (Velickovic et al., 2017; Wan et al., 2022; Wu et al., 2019), graph autoencoders (Cao et al., 2016; Tu et al., 2018), graph reinforcement learning (Do et al., 2019; You et al., 2018), and graph adversarial methods (Wang et al., 2018; Yang et al., 2023). Recently, GNNs have also been explored for LJP, which can leverage the rich information contained in legal graphs, such as law articles, charges, and penalties, and learn the relationships among them. For example, Xu et al. (2020) introduced a graph distillation operator to automatically learn fine-grained differences among confusing

Table 1
Descriptions of main notations.

Notation	Description
$\boldsymbol{X}$	A word embedding sequence of the fact description
$\boldsymbol{Y}_c$	The charge prediction result
$\boldsymbol{Y}_l$	The law article prediction result
$\boldsymbol{Y}_t$	The term of penalty prediction result
$\boldsymbol{P}$	The nodes feature matrix for all labels
$\boldsymbol{B}_{sr}$	The adjacency matrix of similarity relationship between label
$\boldsymbol{B}_{cc}$	The adjacency matrix of consistency constraints between label
$\boldsymbol{B}_{cr}$	The adjacency matrix of confusion relationships between label
$\boldsymbol{G}_{sr}$	The label similarity relationships graph
$\boldsymbol{G}_{cc}$	The label consistency constraints graph
$\boldsymbol{G}_{cr}$	The label confusion relationships graph

law articles. Yue et al. (2021) employed label descriptions to create two similarity graphs of labels and introduced a label embedding technique based on graphs. Zhao et al. (2023) utilized GNNs to extract information regarding law articles and sememes. Additionally, they employed a multi-graph fusion mechanism to combine information from diverse types of graphs. Feng et al. (2023) presented a graph structure to capture the criminal actions and their temporal relations within each case. Compared to the aforementioned works, which are only applicable to single-label tasks, we propose a novel module that combines the effectiveness of expert modules with an innovative multi-graph attention network, enabling it to simultaneously process and integrate information from multiple graphs and discern confusing labels in multi-label scenarios while ensuring consistency constraints among tasks.

3. Problem formulation

Within this section, we present various notations and terminologies, followed by the formulation of the LJP assignment. The descriptions of the main notations are as shown in Table 1.

Definition 1 (*Law Cases*)**.** Law cases are legal documents that record cases in the courts of law. Law cases typically include fact descriptions and multiple judgment factors, such as charges, law articles and terms of penalty.

Definition 2 (*Legal Label Graphs*)**.** Legal label graphs are a novel representation of the complex and diverse relationships among legal labels, such as charges, articles, and penalties. Legal label graphs can capture the similarity, consistency, and confusion among different labels, which can help to improve the precision and interpretability of LJP. Legal label graphs comprise three graphs: the label similarity relationships graph $\boldsymbol{G}_{sr} = \{\boldsymbol{P}, \boldsymbol{B}_{sr}\}$, the label consistency constraints graph $\boldsymbol{G}_{cc} = \{\boldsymbol{P}, \boldsymbol{B}_{cc}\}$, and the label confusion relationships graph $\boldsymbol{G}_{cr} = \{\boldsymbol{P}, \boldsymbol{B}_{cr}\}$.

The nodes feature matrix $\boldsymbol{P} \in \mathbb{R}^{n_l \times d}$ represents the nodes feature matrix for all labels, where n_l represents the number of graph nodes, and d indicates the vector dimensionality. $\boldsymbol{B}_{sr}^{ij} \in \mathbb{R}^{n_l \times n_l}$ denotes the adjacency matrix of similarity relationship between label i and j, $i, j \in [1, n_l]$. The similarity between labels is calculated by the union of the results of cosine similarity and Word Rotator's Distance (WRD) (Yokoi et al., 2020), and the optimal similarity threshold is determined through expert evaluation conducted by legal professionals. If the similarity score exceeds the specified threshold, $\boldsymbol{B}_{sr}^{ij} = 1$; otherwise, $\boldsymbol{B}_{sr}^{ij} = 0$. In the training label set, if label i and j coexist in the outcome of the same sample, $\boldsymbol{B}_{cc}^{ij} = 1$; otherwise, $\boldsymbol{B}_{cc}^{ij} = 0$. As for $\boldsymbol{B}_{cr} \in \mathbb{R}^{n_l \times n_l}$, since relationships of confusion often fall within the subset of similarity relationships, if $\boldsymbol{B}_{sr}^{ij} = 1$ and $\boldsymbol{B}_{cc}^{ij} = 0$, $\boldsymbol{B}_{cr}^{ij} = 1$; otherwise, $\boldsymbol{B}_{cr}^{ij} = 0$. The algorithm for constructing the legal label graphs is described in Algorithm 1.

Definition 3 (*Legal Judgment Prediction*)**.** Legal judgment prediction in the civil law system involves the task of forecasting and assessing the outcome of legal proceedings based on the descriptions of existing cases.

Our objective is to acquire knowledge of an LJP model, denoted as δ, which, given a word embedding sequence $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\} \in \mathbb{R}^{n \times d}$ of the fact description, where n represents the length of the padded fact description, as well as additional legal label graphs $\boldsymbol{G}_{sr}$, $\boldsymbol{G}_{cc}$, $\boldsymbol{G}_{cr}$, then generate multiple judgments $\boldsymbol{Y}_c$, $\boldsymbol{Y}_l$, $\boldsymbol{Y}_t$. This can be formally expressed as $\delta(\boldsymbol{X}, \boldsymbol{G}_{sr}, \boldsymbol{G}_{cc}, \boldsymbol{G}_{cr}) \Rightarrow (\boldsymbol{Y}_c, \boldsymbol{Y}_l, \boldsymbol{Y}_t)$.

4. Method

In this section, we provide a comprehensive explanation of our proposed methodology for LJP. We commence with a model overview (Section 4.1) to delineate the framework's structure and objectives. Subsequently, we delve into the details of the multi-perspective interactive encoder (Section 4.2), which encodes the fact description and the legal labels and computes their interaction features from multiple perspectives. Then we elucidate the multi-graph attention consistency expert module (Section 4.3), which leverages the legal label graphs to distinguish between confusion labels and ensure consistency constraints in cross-tasks. Finally, we describe the training (Section 4.4) procedure used to optimize our model's performance.

Algorithm 1: Construction of Legal Label Graphs

Input: The nodes feature matrix for all labels $\boldsymbol{P}$, training label set Γ, similarity threshold γ
Output: Legal label graphs $\boldsymbol{G} = \{\boldsymbol{G}_{sr}, \boldsymbol{G}_{cc}, \boldsymbol{G}_{cr}\}$

Initialize $\boldsymbol{B}_{sr} \in \mathbb{R}^{n_l \times n_l}$ and all elements are set to zero;
for $i \in [1, n_l]$ **do**
  for $j \in [1, n_l]$ **do**
    Calculate the similarity score between label i and j by the union of cosine similarity and WRD results;
    if *similarity score* $\geq \gamma$ **then**
      Set $\boldsymbol{B}_{sr}[i, j] = \boldsymbol{B}_{sr}[j, i] = 1$;
    end
  end
end
Initialize $\boldsymbol{B}_{cc} \in \mathbb{R}^{n_l \times n_l}$ and all elements are set to zero;
for *each sample in* Γ **do**
  for *label* i *and* j *coexist in the outcome of the same sample* **do**
    Set $\boldsymbol{B}_{cc}[i, j] = \boldsymbol{B}_{cc}[j, i] = 1$;
  end
end
Initialize $\boldsymbol{B}_{cr} \in \mathbb{R}^{n_l \times n_l}$ and all elements are set to zero;
for $i \in [1, n_l]$ **do**
  for $j \in [1, n_l]$ **do**
    if $\boldsymbol{B}_{sr}[i, j] = 1$ *and* $\boldsymbol{B}_{cc}[i, j] = 0$ **then**
      Set $\boldsymbol{B}_{cr}[i, j] = \boldsymbol{B}_{cr}[j, i] = 1$;
    end
  end
end
return $\boldsymbol{G} = \{\boldsymbol{G}_{sr}, \boldsymbol{G}_{cc}, \boldsymbol{G}_{cr}\}$, *where* $\boldsymbol{G}_k = \{\boldsymbol{P}, \boldsymbol{B}_k\}$ *for* $k = sr, cc, cr$;

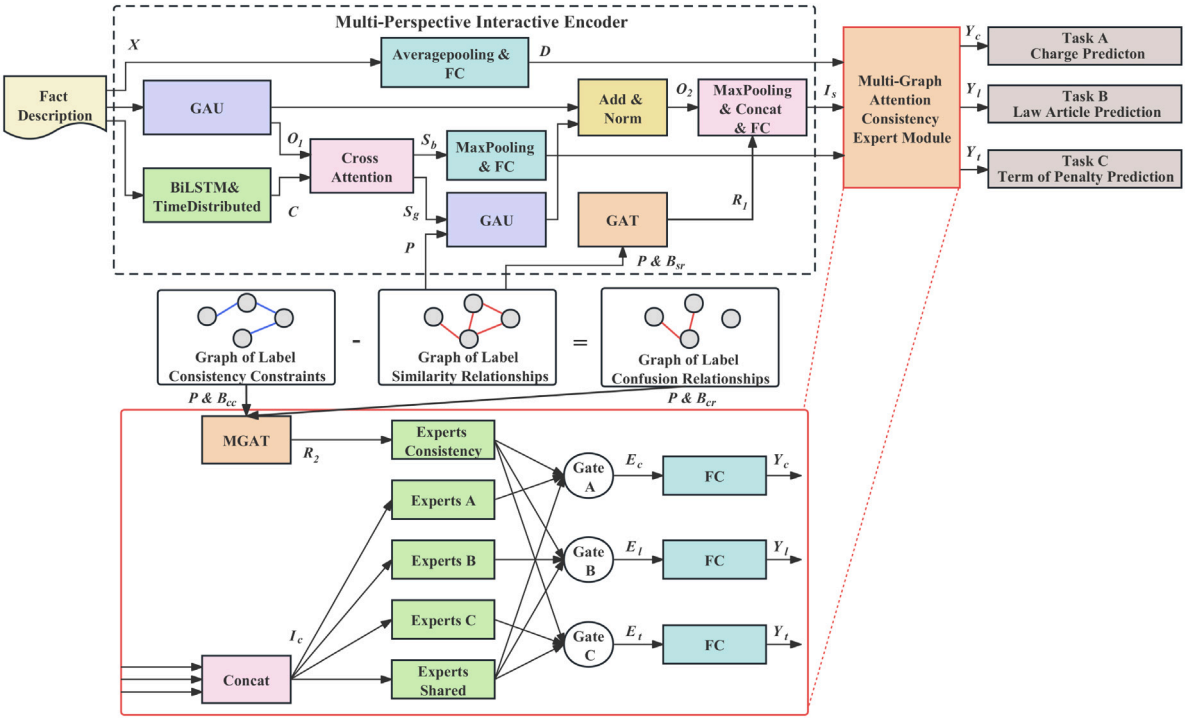

Fig. 2. The overall framework of GJudge.

4.1. Model overview

To address the task of LJP in a multi-label scenario, we introduce a novel framework called GJudge, as illustrated in Fig. 2, illustrating how the multi-perspective interactive encoder and the multi-graph attention consistency expert module work together. The multi-perspective interactive encoder comprises four main components: BiLSTM (Schuster & Paliwal, 1997), Gated Attention Units (GAU) (Hua et al., 2022), Cross Attention, and Graph Attention Networks (GAT) (Velickovic et al., 2017). These components work together to extract relevant information from the fact descriptions and incorporate label relationships. The BiLSTM and the first layer of GAU are utilized to extract information from the fact descriptions. The Cross Attention network is utilized to merge information from the BiLSTM and the first layer of GAU, it learns attention weights for each word in the fact descriptions by leveraging information from both encoders, enabling effective fusion of different perspectives. The second layer of GAU takes the sentence vectors of each label concept as queries, the output of the first GAU as keys and values, and then fuses the output of the second GAU with the GAT constructed using the graph of label similarity relationships. This step captures the label-specific information and incorporates the label relationships encoded in the graph. Finally, the outputs from different perspectives are concatenated to obtain a comprehensive representation of the fact description, which enhances the robustness and generalization ability of the framework. The multi-graph attention consistency expert module mainly consists of multiple expert modules and a Multi-Graph Attention Network (MGAT). Each task combines the information from its corresponding experts, the shared experts, and the consistency experts of the MGAT. This module leverages graph neural networks to extract meaningful correlation information among multiple labels in different tasks and explicitly models the consistency constraints between labels.

4.2. Multi-perspective interactive encoder

As shown in Fig. 2, The fact descriptions $\boldsymbol{X}$ are fed into BiLSTM and output through the TimeDistributed layer, which performs fully connected operations at each time step, thereby enhancing the ability to extract information from fact descriptions. The resulting hidden layer output for a fact description is denoted as:

$$\boldsymbol{C} = TimeDistributed\big(BiLSTM(\boldsymbol{X})\big), \tag{1}$$

where $\boldsymbol{C} \in \mathbb{R}^{n\times d_u}$, d_u represents the number of units in the hidden layer. Subsequently, the fact descriptions are encoded from different perspectives using the first layer of GAU, it involves computing matrices $\boldsymbol{Q}$ (Query), $\boldsymbol{K}$ (Key), and $\boldsymbol{V}_1$ (Value) can be calculated:

$$\boldsymbol{Z} = \phi(\boldsymbol{X}\boldsymbol{W}_z), V_1 = \phi(\boldsymbol{X}\boldsymbol{W}_v), \tag{2}$$

$$\boldsymbol{Q}, \boldsymbol{K} = RoPE(\boldsymbol{X}, \mathcal{Q}(\boldsymbol{Z}), \mathcal{K}(\boldsymbol{Z})), \tag{3}$$

$$\boldsymbol{A}_1 = \frac{1}{\sqrt{d}} relu^2(\boldsymbol{Q}\boldsymbol{K}^\top + \boldsymbol{b}_f), \tag{4}$$

where $\boldsymbol{A}_1 \in \mathbb{R}^{n\times n}$ represents the attention matrix that integrates information among different words. $\boldsymbol{W}_z \in \mathbb{R}^{d\times d}$, $\boldsymbol{W}_v \in \mathbb{R}^{d\times n}$ is the weight matrix, $\boldsymbol{b}_f$ represents the relative positional bias, $\mathcal{Q}$ and $\mathcal{K}$ are affine transformations, $RoPE$ is the Rotary Position Embedding (Su et al., 2021), $relu^2$ denotes the squared value after applying the ReLU activation function, while ϕ refers to the swish activation function. The output $\boldsymbol{O}_1$ is calculated as:

$$\boldsymbol{O}_1 = (\boldsymbol{U}_1 \odot \boldsymbol{A}_1\boldsymbol{V}_1)W_o, \boldsymbol{U}_1 = \phi(\boldsymbol{X}\boldsymbol{W}_u), \tag{5}$$

where $\boldsymbol{W}_o \in \mathbb{R}^{n\times d_u}$, $\boldsymbol{W}_u \in \mathbb{R}^{d\times n}$ represent the corresponding weight matrices. $\odot$ denotes element-wise multiplication, and $\boldsymbol{O}_1 \in \mathbb{R}^{n\times d_u}$ encapsulates the interaction information among the words.

Subsequently, the results $\boldsymbol{C}$ originating from the TimeDistributed layer, along with the output $\boldsymbol{O}_1$ produced by the first layer of GAU, are combined and employed as the input for the Cross Attention layer. This further enhances the encoding effectiveness by facilitating the interaction of information among different perspectives. The computation equations for Cross Attention are presented as follows:

$$\boldsymbol{J} = tanh\big(([\boldsymbol{C}; \boldsymbol{O}_1]\boldsymbol{W}_c + \boldsymbol{b}_c)\boldsymbol{W}_q + \boldsymbol{b}_t\big), \tag{6}$$

$$\boldsymbol{M} = softmax\big(Flatten(\boldsymbol{J})\big), \tag{7}$$

$$[\boldsymbol{S}_1; \boldsymbol{S}_2] = \boldsymbol{S} = [\boldsymbol{C}; \boldsymbol{O}_1] \odot [\boldsymbol{M}^1; \ldots; \boldsymbol{M}^{d_u}], \tag{8}$$

$$\boldsymbol{S}_b = LayerNorm(\boldsymbol{C} + \boldsymbol{S}_1), \tag{9}$$

$$\boldsymbol{S}_g = LayerNorm(\boldsymbol{O}_1 + \boldsymbol{S}_2), \tag{10}$$

where $\boldsymbol{W}_c \in \mathbb{R}^{d_u\times d_u}$ and $\boldsymbol{W}_q \in \mathbb{R}^{d_u\times 1}$ represent the corresponding weight matrices, $\boldsymbol{b}_c \in \mathbb{R}^{d_u}$ and $\boldsymbol{b}_t \in \mathbb{R}^{d_u}$ denote the relative positional biases, $\boldsymbol{J} \in \mathbb{R}^{2n\times 1}$, and $\boldsymbol{M} \in \mathbb{R}^{2n}$ are the attention matrices for multiple perspectives. $[\cdot]$ denotes the concatenation

operation, $[\boldsymbol{C};\boldsymbol{O}_1] \in \mathbb{R}^{2n\times d_u}$ and $[\boldsymbol{M}^1;\ldots;\boldsymbol{M}^{d_u}] \in \mathbb{R}^{2n\times d_u}$ represent the accumulation and concatenation of $\boldsymbol{M}$ for d_u times, $\boldsymbol{S}_1$, $\boldsymbol{S}_2$ correspond to the partitioning of $\boldsymbol{S} \in \mathbb{R}^{2n\times d_u}$, where $\boldsymbol{S}_b$ and $\boldsymbol{S}_g$ represent the interaction encoding of BiLSTM and GAU perspectives, respectively. Then, the formulas for the GAT are shown as:

$$\boldsymbol{T}_1 = LeakyReLU\left(\boldsymbol{P}\boldsymbol{W}_t\boldsymbol{H}_s + (\boldsymbol{P}\boldsymbol{W}_t\boldsymbol{H}_n)^\top\right), \tag{11}$$

$$\boldsymbol{F}_1 = softmax\left(\boldsymbol{T}_1 - 10^9(1-\boldsymbol{B}_{sr})\right), \tag{12}$$

$$\boldsymbol{R}_1 = \phi\left(\frac{1}{K}\sum_{k=1}^{N}\boldsymbol{F}_1^k\boldsymbol{P}\boldsymbol{W}_t^k\right), \tag{13}$$

where $\boldsymbol{W}_t \in \mathbb{R}^{d\times d_u}$ is the weight matrix, $\boldsymbol{H}_s$, $\boldsymbol{H}_n$ denote the weight matrices for the current node and neighboring nodes, respectively. $\boldsymbol{T}_1 \in \mathbb{R}^{n_l\times n_l}$ represents the pre-output feature regarding the nodes, N represents the number of graph attention heads, while $\boldsymbol{R}_1 \in \mathbb{R}^{n_l\times d_u}$ signifies the aggregation and averaging of all graph attention heads. Then, the second layer of GAU involves querying the fact description information encoded by the first layer of GAU with the relevant label information which can be calculated:

$$\boldsymbol{L}_t = \phi(\boldsymbol{L}\boldsymbol{W}_l), \boldsymbol{Z}_s = \phi(\boldsymbol{S}_g\boldsymbol{W}_s), \tag{14}$$

$$\boldsymbol{A}_2 = \frac{1}{\sqrt{d}}relu^2\left(\mathcal{Q}(\boldsymbol{L}_t)\mathcal{K}(\boldsymbol{Z}_s)^\top + \boldsymbol{b}_s\right), \tag{15}$$

$$\boldsymbol{U}_2 = \phi(\boldsymbol{S}_g\boldsymbol{W}_{u_2}), \boldsymbol{V}_2 = \phi(\boldsymbol{S}_g\boldsymbol{W}_{v_2}), \tag{16}$$

$$\boldsymbol{O}_2 = LayerNorm\left(\boldsymbol{O}_1 + (\boldsymbol{U}_2 \odot \boldsymbol{A}_2\boldsymbol{V}_2)\boldsymbol{W}_{o_2}\right), \tag{17}$$

where $\boldsymbol{W}_l$, $\boldsymbol{W}_s \in \mathbb{R}^{d_u\times d_u}$, $\boldsymbol{W}_{o_2} \in \mathbb{R}^{n\times d_u}$, $\boldsymbol{W}_{u_2}$, $\boldsymbol{W}_{v_2} \in \mathbb{R}^{d_u\times n}$ represent the corresponding weight matrices, and $\boldsymbol{b}_s$ represents the bias vector, $\boldsymbol{L} \in \mathbb{R}^{n\times d}$ is obtained by padding $\boldsymbol{P}$ with zeros. We then calculate the mean of all fact description word vectors and feed it to a fully connected layer to obtain deeper information. The specific formula is computed as follows:

$$\boldsymbol{D} = \frac{1}{n}\left(\sum_{i=1}^{n}\boldsymbol{x}_i\right)\boldsymbol{W}_d + \boldsymbol{b}_d, \tag{18}$$

where $\boldsymbol{W}_d \in \mathbb{R}^{d_u\times d_u}$ denotes the weight matrix, $\boldsymbol{b}_d$ denotes the relative positional bias. Then, $\boldsymbol{R}_1$ and $\boldsymbol{O}_2$ are concatenated together, as shown in the detailed formulas:

$$\boldsymbol{I}_s = \phi\left([max(\boldsymbol{R}_1); max(\boldsymbol{O}_2)]\boldsymbol{W}_g + \boldsymbol{b}_g\right), \tag{19}$$

where $\boldsymbol{W}_g \in \mathbb{R}^{(n+n_l)\times d_u}$ represent the corresponding weight matrices, and $\boldsymbol{b}_g$ represent the bias vectors. The $max(\cdot)$ operation performs max pooling along the penultimate dimension.

4.3. Multi-graph attention consistency expert module

The multi-graph attention consistency expert module is depicted in Fig. 2. This module comprises a multi-graph attention network and multiple expert networks. Similarly to Eqs. (11)–(13), the pre-output features, denoted as $\boldsymbol{T}_a$ for the label consistency constraints graph and $\boldsymbol{T}_b$ for the label confusion relationships graph, can be calculated based on the feature matrices of the labels. The specific formulas are calculated as follows:

$$\boldsymbol{B}_{cr} = -min(0, \boldsymbol{B}_{cc} - \boldsymbol{B}_{sr}), \tag{20}$$

$$\boldsymbol{F}_2 = softmax[\left(\boldsymbol{T}_a - 10^9(1-\boldsymbol{B}_{cc})\right); \left(\boldsymbol{T}_b - 10^9(1-\boldsymbol{B}_{cr})\right)], \tag{21}$$

$$\boldsymbol{R}_2 = max\left(\phi\left(\frac{1}{K}\sum_{k=1}^{N}\boldsymbol{F}_2^k\boldsymbol{P}\boldsymbol{W}_m^k\right)\right), \tag{22}$$

where $\boldsymbol{B}_{cc}$, $\boldsymbol{B}_{cr} \in \mathbb{R}^{n_l\times n_l}$ represents the adjacency matrix for the label consistency constraints graph and label confusion relationships graph, $\boldsymbol{R}_2 \in \mathbb{R}^{n_l\times d_u}$ represents the average of attention heads for all multi-graphs. The formulas for the expert module in the charge prediction task are as follows:

$$\boldsymbol{I}_c = [\phi\left(max(\boldsymbol{S}_b)\boldsymbol{W}_i + \boldsymbol{b}_i\right), \boldsymbol{I}_s, \boldsymbol{D}], \tag{23}$$

$$\boldsymbol{Y}_k = \boldsymbol{I}_c\boldsymbol{W}_k + \boldsymbol{b}_k, \boldsymbol{Y}_f = \boldsymbol{I}_c\boldsymbol{W}_f + \boldsymbol{b}_f, \boldsymbol{Y}_p = \boldsymbol{R}_2\boldsymbol{W}_p + \boldsymbol{b}_p, \tag{24}$$

$$\boldsymbol{E}_c = softmax(\boldsymbol{I}_c\boldsymbol{W}_e)[\boldsymbol{Y}_k^1;\ \ldots;\boldsymbol{Y}_k^N;\boldsymbol{Y}_f^1;\ \ldots;\boldsymbol{Y}_f^N;\boldsymbol{Y}_p^1;\ \ldots;\boldsymbol{Y}_p^N], \tag{25}$$

$$\boldsymbol{Y}_c = \boldsymbol{E}_c\boldsymbol{W}_{cy} + \boldsymbol{b}_{cy}, \tag{26}$$

Table 2
Statistics of the experimental datasets.

DataSet	CAIL-E	CAIL-M
Training set cases	154 592	142 635
Test set cases	32 508	17 630
Validation set cases	17 131	17 808
Charges	202	100
Law articles	183	105
Term of penalty	11	11

$$\boldsymbol{Y}_l = \boldsymbol{E}_l \boldsymbol{W}_{ly} + \boldsymbol{b}_{ly}, \tag{27}$$

$$\boldsymbol{Y}_t = sigmoid(\boldsymbol{E}_t \boldsymbol{W}_{ty} + \boldsymbol{b}_{ty}), \tag{28}$$

where $\boldsymbol{W}_k$, $\boldsymbol{W}_f \in \mathbb{R}^{3d_u \times d_u}$, $\boldsymbol{W}_i$, $\boldsymbol{W}_p \in \mathbb{R}^{d_u \times d_u}$, $\boldsymbol{W}_e \in \mathbb{R}^{3d_u \times 3N}$, $\boldsymbol{W}_{cy} \in \mathbb{R}^{d_u \times n_c}$, $\boldsymbol{W}_{ly} \in \mathbb{R}^{d_u \times n_l}$, and $\boldsymbol{W}_{ty} \in \mathbb{R}^{d_u \times n_t}$ represent the corresponding weight matrices. $\boldsymbol{b}_k$, $\boldsymbol{b}_f$, $\boldsymbol{b}_i$, $\boldsymbol{b}_p$, $\boldsymbol{b}_{cy}$, $\boldsymbol{b}_{ly}$, $\boldsymbol{b}_{ty}$ represent the bias vectors. n_c, n_l, and n_t indicate the count of labels for charge, law article, and term of penalty, respectively. $\boldsymbol{I}_c \in \mathbb{R}^{1 \times 3d_u}$ represents the concatenated output of the multi-perspective interactive encoder. The final expert output for the charge prediction task is $\boldsymbol{E}_c \in \mathbb{R}^{1 \times d_u}$, similarly, $\boldsymbol{E}_l$, $\boldsymbol{E}_t \in \mathbb{R}^{1 \times d_u}$ can be calculated for the law article and term of penalty prediction tasks, respectively. Finally, the outputs $\boldsymbol{Y}_c \in \mathbb{R}^{1 \times n_c}$, $\boldsymbol{Y}_l \in \mathbb{R}^{1 \times n_l}$, and $\boldsymbol{Y}_t \in \mathbb{R}^{1 \times n_t}$ for the charge, law article, and term of penalty prediction tasks, respectively, are obtained through fully connected layers.

4.4. Training

Given the need to handle both multi-label classification and single-label classification within our task, we employ ZLPR (Su et al., 2022) to compute the loss for law article and charge prediction tasks, whereas the binary cross-entropy loss is employed to compute the term of penalty prediction task. ZLPR compares the target class score with the non-target class scores pairwise, and automatically balances the weights of each item. Only classes with scores greater than zero are outputted in prediction. The ZLPR loss can be calculated by:

$$\ell_z = log\big((1 + \sum_{i \epsilon \Omega_n} e^{s_i})(1 + \sum_{j \epsilon \Omega_p} e^{-s_j})\big), \tag{29}$$

where the collections of class labels for the samples are denoted by Ω_p for positive and Ω_n for negative, where the negative class set contains the score s_i for each class i, and the positive class set contains the score s_j for each class j. The loss function for the term of penalty prediction task is calculated as follows:

$$\ell_t = -\sum_{n_t} \big(y_{n_t} log(\boldsymbol{Y}_t) + (1 - y_{n_t}) log(1 - \boldsymbol{Y}_t)\big), \tag{30}$$

where n_t represents the n_tth classification for the output neuron, y_{n_t} is the corresponding one-hot encoded label. The aggregate loss of the model, ℓ_{total}, is the combination of the losses from each sub-task with different weights, where ℓ_l and ℓ_c are the law article prediction and charge prediction losses calculated using Eq. (29), and ℓ_t represents the term of penalty prediction loss. The loss weights λ_l, λ_c, and λ_t for different tasks were learned using the method proposed by (Chen et al., 2018).

$$\ell_{total} = \lambda_l \ell_l + \lambda_c \ell_c + \lambda_t \ell_t. \tag{31}$$

5. Experiments

In this section, we perform experiments to assess the performance of our proposed GJudge for LJP. We commence by introducing the dataset in Section 5.1 that is utilized in our experiments, we then explore the baselines (Section 5.2) employed for comparison. After that, we provide an overview of the experimental settings (Section 5.3). Finally, we proceed to present the experimental results and discussion (Section 5.4) to analyze and interpret the outcomes of our evaluations.

5.1. Dataset

We utilized the publicly available CAIL2018 dataset (Xiao et al., 2018) from the 2018 China AI and Law Challenge.[1] The dataset consists of "exercise_contest", "first_stage" and "restData". Due to the predominance of single-label samples in "exercise_contest", which lacked multi-label samples, we extracted multi-charge or multi-article labeled samples from both "first_stage" and "restData" to construct the CAIL-M. However, the scarcity of data made it difficult to train the model to estimate a few cases precisely. Therefore, during the data processing stage, we removed charge and law article labels with fewer than 100 samples and shuffled the data.

[1] http://cail.cipsc.org.cn/index.html.

We renamed the "exercise_contest" as "CAIL-E" to differentiate it from the CAIL-M. The specifics regarding the two datasets are presented in Table 2.

As depicted in Table 3, we conducted a statistical analysis of the training datasets for two distinct datasets. From Table 3, it can be observed that in the charge and law article prediction task on CAIL-E, single-label instances constitute the majority. Besides, due to the extraction of multiple charge or multiple law article labels on CAIL-M, there are also samples where one task is multi-label while the other task is single-label. One noteworthy observation is that the law articles tend to have more multi-label cases than the charges on CAIL-M. This indicates that some samples have multi-label law articles but their corresponding charges are not necessarily multi-label. Furthermore, we split the penalties into distinct ranges to aid the examination of predictions related to penalties by citing the prior work (Zhong et al., 2018).

5.2. Baselines

We evaluated the performance of these models in our experiments:

- **TopJudge** (Zhong et al., 2018): a framework based on topology for multi-task learning, which carries out multiple tasks simultaneously by representing the relationship among subtasks as a directed acyclic graph.
- **MPBFN** (Yang et al., 2019): a network that uses feedback from both directions and multi-perspective to exploit the relationship between different subtasks.
- **LADAN** (Xu et al., 2020): a method for LJP that extracts features by using the variations among comparable law articles to obtain features from the facts of legal cases.
- **NeurJudge** (Yue et al., 2021): a neural framework for LJP that is sensitive to different circumstances, which divides fact descriptions into different scenarios using the results of intermediate sub-tasks, facilitating predictions for other sub-tasks.
- **RLJP** (Wu et al., 2022): a two-stage method that divides the LJP procedure into creating rationale and predicting judgment.

Furthermore, we replace the multi-graph attention consistency expert module with **MLP**, **MMoE** (Ma et al., 2018), and **CGC** (Tang et al., 2020), respectively, and employ **Transformer** (Vaswani et al., 2017) to replace the GAU in the multi-perspective interactive encoding module to implement the GJudge framework.

5.3. Experimental settings

In this paper, we employ Pyltp (Che et al., 2021) for Chinese word segmentation, with a maximum sentence length set to 600. We use the Skip-Gram model (Mikolov et al., 2013) to create word embeddings from the data we train on. The number of words in the vocabulary is 150,000 and the dimension of the word embeddings is 300. The multi-perspective interactive encoder uses a BiLSTM with one layer and 256 hidden units. The graph attention employs 8 heads for attention. We set the value of dropout to 0.2, and train each model for up to 20 iterations with a batch size of 128. The loss weights λ_l, λ_c, and λ_t are set to 1.0, 0.7, and 0.1, respectively. We apply the TensorFlow framework and choose the Adam optimizer. Our experiments were conducted on two Tesla T4 GPUs, and the implementation code is publicly available.[2] Additionally, we use accuracy (Acc.), macro-precision (MP), macro-recall (MR), and macro-F1 (F1) as the metrics for evaluation.

5.4. Experiment results and discussion

In this subsection, we assess the effectiveness of GJudge on the CAIL-E and CAIL-M and highlight the advantages of our framework from these perspectives: (1) the comparison against baselines on various evaluation metrics; (2) the removal of each part of our framework to measure its effect; (3) the examination of the influence of different methods to build the graph of label similarity; (4) the comparison of the convergence between our framework and baselines; (5) the case study provides two real-world cases as illustrative examples; (6) the error analysis of the sources of errors and limitations of our model.

5.4.1. Comparison against baselines

We conducted experiments on CAIL-E and CAIL-M, evaluating the performance of three LJP subtasks: charge prediction, law article prediction, and term of penalty prediction. The experimental results are presented in Tables 4 and 5.

Based on the results, GJudge outperforms other frameworks in terms of F1 score for three tasks on two datasets. On CAIL-E, GJudge achieves F1 scores that surpass the state-of-the-art RLJP, by 0.95%, 1.73%, and 2.97% for law article, charge, and term of penalty prediction, respectively. Similarly, on CAIL-M, GJudge improves the F1 scores by 0.93%, 1.82%, and 2.23% for the three tasks compared to RLJP.

On the CAIL-M, MPBFN achieves lower F1 scores than other methods. This can be attributed to MPBFN's bi-feedback network with both forward prediction and backward verification. Specifically, the prediction results of each subtask not only serve as input features for other subtasks but also act as input features for the respective subtask itself to enhance its predictive performance. However, this bi-feedback mechanism can result in inconsistencies among results from different subtasks. Particularly in the context of handling multi-label tasks, where relationships between subtasks are more complex and crucial, the MPBFN may exhibit a bias

[2] https://anonymous.4open.science/r/GJudge-D6FA/.

Table 3
The number of labels for different datasets in multi-label tasks.

Tasks	Law articles		Charges	
Number of labels	CAIL-E	CAIL-M	CAIL-E	CAIL-M
1	75.803%	17.315%	77.931%	53.781%
2	18.747%	66.292%	19.943%	43.355%
3	3.846%	13.807%	1.885%	2.594%
4	1.279%	2.175%	0.186%	0.223%
>4	0.325%	0.411%	0.055%	0.047%

Table 4
Judgment prediction results on CAIL-E.

Tasks	Law articles				Charges				Term of penalty			
Metrics	Acc.	MP	MR	F1	Acc.	MP	MR	F1	Acc.	MP	MR	F1
TopJudge	80.85	75.53	71.61	72.41	83.05	75.66	74.70	74.09	38.57	38.36	36.56	37.05
MPBFN	79.71	**81.60**	67.38	72.01	82.68	84.25	71.17	75.42	39.94	40.21	36.68	36.70
LADAN	80.05	75.77	72.01	72.60	83.31	79.92	75.58	76.32	38.00	37.94	37.12	36.64
NeurJudge	82.44	79.04	72.88	74.42	82.29	79.74	78.91	77.94	39.24	40.11	37.55	37.75
RLJP(Transformer)	83.74	79.39	76.69	76.73	85.49	83.05	79.89	80.28	40.66	40.23	38.40	37.42
GJudge-MLP	82.65	78.80	74.91	75.25	85.65	83.12	80.57	80.61	38.40	39.83	38.57	38.83
GJudge-MMoE	81.36	79.24	74.84	75.26	84.76	84.09	79.01	80.16	39.07	39.65	37.30	38.32
GJudge-CGC	81.76	79.51	76.67	76.72	84.08	83.25	79.78	80.06	38.48	39.61	37.84	38.47
GJudge-Transformer	82.30	78.52	**77.22**	76.94	85.03	82.47	81.62	80.78	39.98	42.26	38.05	39.24
GJudge	**83.77**	81.48	76.84	**77.68**	**85.87**	**84.44**	**81.68**	**82.01**	**40.88**	**43.44**	**39.94**	**40.39**

Table 5
Judgment prediction results on CAIL-M.

Tasks	Law articles				Charges				Term of penalty			
Metrics	Acc.	MP	MR	F1	Acc.	MP	MR	F1	Acc.	MP	MR	F1
TopJudge	81.54	82.04	73.42	76.40	83.47	83.04	73.62	76.83	42.34	46.02	42.68	43.89
MPBFN	25.71	85.94	48.50	59.76	55.16	89.56	46.28	57.98	39.61	42.24	40.25	40.29
LADAN	81.40	82.84	73.57	76.70	83.45	86.89	73.37	78.46	40.94	42.16	43.19	42.12
NeurJudge	82.78	85.02	74.71	78.65	82.73	85.75	73.27	78.21	42.04	45.13	44.02	44.26
RLJP(Transformer)	82.87	85.41	76.77	79.92	85.28	88.53	76.59	81.32	43.42	46.33	45.07	45.05
GJudge-MLP	82.68	86.01	76.52	79.74	85.01	**90.06**	76.38	81.54	42.99	46.26	43.27	44.02
GJudge-MMoE	82.66	85.87	76.97	79.90	84.97	89.07	77.75	82.01	43.15	46.85	42.97	44.11
GJudge-CGC	83.31	86.21	76.88	80.04	85.33	88.49	78.10	82.14	43.14	46.94	43.41	44.75
GJudge-Transformer	**83.54**	83.84	77.01	79.36	85.59	85.90	78.18	81.42	44.26	**48.98**	43.14	45.24
GJudge	83.45	**87.30**	**77.61**	**80.85**	**85.80**	88.65	**79.40**	**83.14**	**45.19**	48.09	**46.72**	**47.28**

towards predicting only those simple samples that do not present conflicts between the results of different subtasks, leading the model to adopt a conservative approach. This ultimately leads to a high Macro-Precision, while other metrics such as (such as Acc., MR and F1) are comparatively lower. However, on CAIL-E, where single-label instances predominate, MPBFN's metrics do not show a significant decrease compared to other frameworks, indicating the difference between multi-label tasks and single-label tasks.

Furthermore, the experimental results demonstrate that GJudge outperforms GJudge-CGC, GJudge-CGC outperforms GJudge-MMoE, and GJudge-MMoE outperforms GJudge-MLP. This suggests that when dealing with complex relationships between labels, transforming a shared parameter matrix into multiple combination-gating shared experts, adding individual experts for each task, or incorporating label consistency constraints graph and label confusion relationships graph could prove effective in enhancing the performance of each task. Additionally, the results of GJudge-Transformer show that GAU, by integrating Attention and FFN structures, achieves greater improvements in the F1 scores of different tasks in LJP. Compared to the Transformer, GAU features a more lightweight structure with only one attention head, making it a feasible alternative to the Transformer.

5.4.2. Ablation analysis

To assess the efficacy of the GJudge framework, we performed ablation experiments on CAIL-E and CAIL-M, as shown in Tables 6 (The ablation experiments and analysis for the multi-graph attention consistency expert module follow the same procedure as the comparative experiments of GJudge-MLP, GJudge-MMoE, and GJudge-CGC in Section 5.4.1, so we omit them here). In these tables, "w/o Cross Attention" indicates the removal of the Cross Attention module from the framework, "w/o GAU" represents the removal of all GAU modules and the input GAU modules (including Cross Attention and GAT), and "w/o GAU* & GAT" denotes the removal of the GAU and GAT structure after Cross Attention. After removing these components, GJudge's performance shows varying degrees of decline, which indicates that increasing the information interaction between encoding modules, expanding the encoding perspective, and integrating legal label graphs effectively enhance LJP's predictive performance. Furthermore, "w/o BiLSTM" represents the removal of the BiLSTM & TimeDistributed structure from the framework (including Cross Attention), resulting in the most significant

Table 6
Ablation experiment on CAIL-E and CAIL-M.

Dataset	Tasks	Law articles		Charges		Term of penalty	
	Metrics	Acc.	F1	Acc.	F1	Acc.	F1
CAIL-E	w/o Cross Attention	83.29	75.30	86.77	80.37	39.34	36.38
	w/o GAU	83.04	75.49	**86.92**	80.55	39.89	37.47
	w/o BiLSTM	73.98	64.13	75.13	68.03	34.58	29.31
	w/o GAU* & GAT	82.60	75.94	85.03	80.78	**40.90**	38.94
	GJudge	**83.77**	**77.68**	85.87	**82.01**	40.88	**40.39**
CAIL-M	w/o Cross Attention	82.54	79.06	85.21	81.62	42.71	44.30
	w/o GAU	83.06	80.36	85.46	81.75	42.69	44.56
	w/o BiLSTM	68.18	62.51	72.85	61.29	32.27	30.35
	w/o GAU* & GAT	83.07	79.68	85.50	81.75	43.19	45.20
	GJudge	**83.45**	**80.85**	**85.80**	**83.14**	**45.19**	**47.28**

Table 7
Performance comparison of different label similarity relationship construction methods on CAIL-E and CAIL-M.

Dataset	Tasks	Law articles		Charges		Term of penalty	
	Metrics	Acc.	F1	Acc.	F1	Acc.	F1
CAIL-E	Cosine Similarity	81.76	77.00	84.71	81.43	39.64	38.42
	WRD	81.50	77.09	84.44	81.29	39.11	38.66
	Cosine Similarity ∩ WRD	81.08	76.29	84.83	80.61	39.26	38.41
	Cosine Similarity ∪ WRD	**83.77**	**77.68**	**85.87**	**82.01**	**40.88**	**40.39**
CAIL-M	Cosine Similarity	83.33	80.34	85.63	82.59	44.90	46.59
	WRD	83.29	80.30	85.65	82.61	44.94	46.86
	Cosine Similarity ∩ WRD	83.16	80.27	85.27	82.02	45.01	46.84
	Cosine Similarity ∪ WRD	**83.45**	**80.85**	**85.80**	**83.14**	**45.19**	**47.28**

decrease in all metrics, indicating that the BiLSTM & TimeDistributed structure is crucial for the multi-perspective interactive encoder. This structure plays a core role in capturing fine-grained information and integrating it into the encoding process.

5.4.3. Analysis of label similarity relationship construction

With the aim of more effectively calculating the similarity between labels, we combined cosine similarity with WRD. This combination allows us to consider both semantic and word-order similarities between labels, thereby enhancing the accuracy and robustness of the calculated results. Instead of utilizing a weighted sum, we employ a combined approach by integrating the results of cosine similarity and WRD. This decision is driven by the potential limitation of a weighted sum approach, which may fail to capture labels with high similarity scores in one method but low scores in the other. The effectiveness of the combined approach lies in its inclusion of any label pair exhibiting a high similarity score in either method, ensuring that relevant labels with distinct semantic or word-order similarities are not overlooked. However, it is crucial to select appropriate thresholds for each method in order to prevent the inclusion of excessive irrelevant labels, which could introduce noise or ambiguity into the graph. Consequently, we established the optimal threshold for each method based on expert evaluations conducted by legal professionals. To validate the impact of different label similarity relationship construction methods on the performance of the model, we performed additional experiments, as shown in Table 7.

From the experimental results, it is evident that the method of using the union of cosine similarity and WRD (Cosine Similarity ∪ WRD) as the basis for constructing multi-label similarity relationships achieved the best performance across all three tasks. This indicates that this approach can effectively leverage the advantages of both similarity calculation methods, overcoming the limitations and biases inherent in a single method. For example, cosine similarity may overlook word order information, while WRD may neglect semantic information.

Furthermore, we observed that the method of using the intersection of cosine similarity and WRD (Cosine Similarity ∩ WRD) results for multi-label relationship construction yielded the poorest performance across all three tasks. This suggests that this method is overly strict, filtering out some meaningful label relationships. Therefore, we consider using the union of cosine similarity and WRD results as a reasonable and effective choice for multi-label relationship construction.

5.4.4. Convergence comparison

To evaluate the performance of various methods during the training process at various epochs, we have collected and presented the evolution of F1 scores for the different prediction tasks. As depicted in Fig. 3, Figs. 4 and 5, the GJudge exhibits a significant advantage by achieving a high F1 score in the very first training epoch and steadily reaching optimal results with a limited number of epochs. This advantage underscores the efficiency and effectiveness of the GJudge, which is capable of learning crucial features in legal texts in a shorter training time frame. Leveraging the advantages of graph neural networks and cross-task consistency learning, it shows great effectiveness on the LJP task.

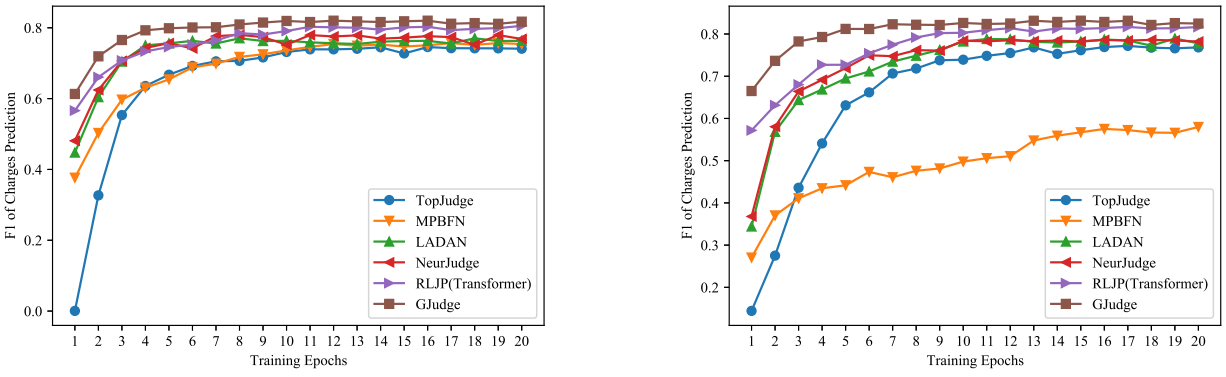

Fig. 3. F1 score evolution of different methods for charges prediction task.

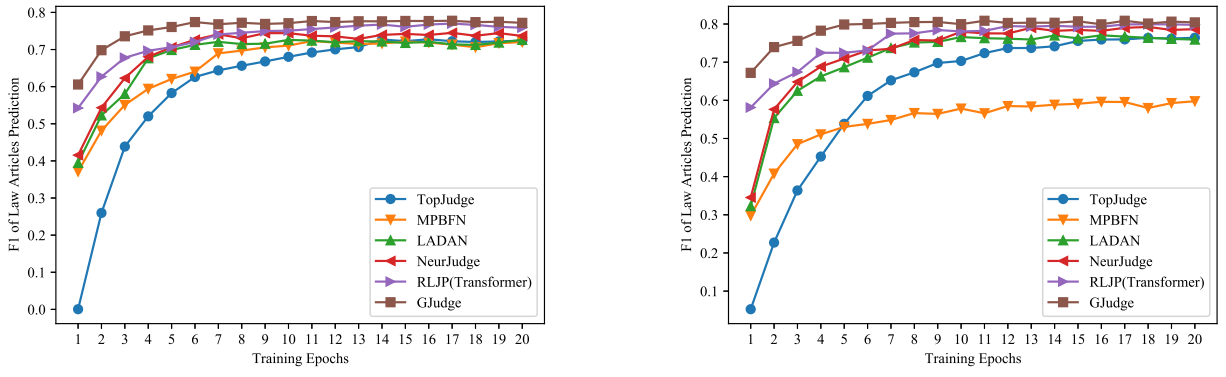

Fig. 4. F1 score evolution of different methods for law articles prediction task.

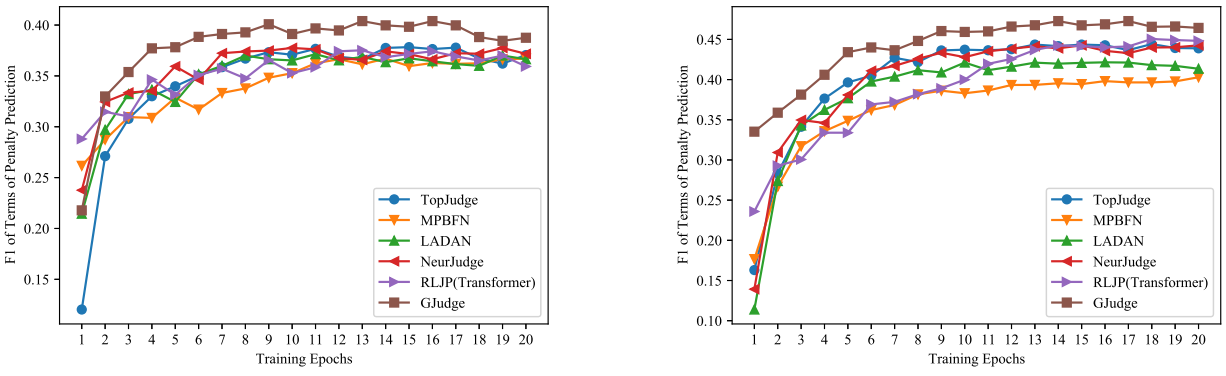

Fig. 5. F1 score evolution of different methods for terms of penalty prediction task.

Regarding the TopJudge, it generally demonstrates inferior performance in the initial epochs on both datasets. This could be attributed to the design of the TopJudge, which incorporates a directed acyclic graph to represent dependencies between sub-tasks. Consequently, when the performance of the preceding tasks is subpar during the early stages of training, it adversely affects subsequent tasks, ultimately leading to a slower convergence rate.

Furthermore, MPBFN exhibits relatively poor convergence when applied to CAIL-M. This difference can be attributed to MPBFN's additional backward verification process, in contrast to TopJudge, which solely relies on forward prediction. In the backward

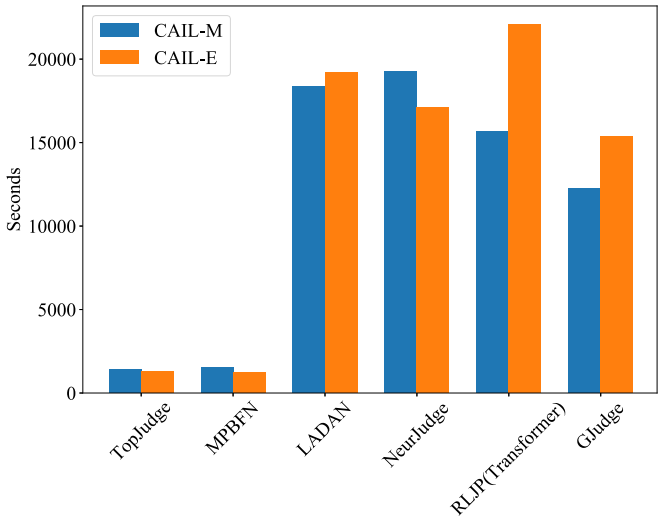

Fig. 6. Comparison of the training time of different models on two datasets.

Table 8
Analysis of confusing single-label cases.

Fact Description: The defendant, Liu, is accused by the Botou City People's Procuratorate of **utilizing their position** at Anji Hongye Limited Company in Botou City, starting in early 2012, to **sell** 20 car heating controllers manufactured by the company to Zhao (who has already been sentenced), with the assessed value of the controllers being **3520 yuan**. The prosecution contends that Liu has breached the stipulations of the Criminal Law of the People's Republic of China, which amounts to the crime of XXX.
GJudge: Crime of Theft, Article 264, less than 6 months **RLJP(Transformer)**: Crime of Duty Encroachment, Article 271, 2–3 years **ChatGPT 3.5**: Crime of Corruption and Crime of Encroachment, Article 260 and Article 261, 3 years **Ground-truth**: Crime of Theft, Article 264, 4 months

verification process, the prediction results of each subtask are also input into the respective subtask itself. Although this yields some initial effectiveness during the early stages of training, it may lead to inconsistencies or conflicts between the results of different subtasks, ultimately making it challenging for the training effectiveness to improve.

Additionally, we present a comparative chart of the training time required for different models to reach their optimal performance in Fig. 6. It can be observed that our model exhibits a notable advantage in terms of training time compared to most models developed in recent years. The shorter training times for TopJudge and MPBFN are attributed to their relatively simpler model structures. Finally, we present the time complexity of our GJudge. GJudge primarily consists of a multi-perspective interactive encoder and a multi-graph attention consistency expert module, the time complexity of the multi-perspective interactive encoder is $O(\max(|n|, |n^2d + d^2n|) + |n^2| + |n^2d + d^2n|) = O(|n^2|)$, where $O(|n|)$, $O(|n^2d + d^2n|)$ and $O(|n^2|)$ represent the complexity of the BiLSTM and TimeDistributed, GAU, and Cross Attention, respectively. Since the time complexity of GAT and MGAT is related to the nodes and edges of the graph but independent of n, this part of the time complexity is considered constant and thus not taken into account. Consequently, we derive the time complexity of the expert module as $O(|n|)$. Therefore, the overall time complexity of GJudge is $O(|n^2|)$.

5.4.5. Case study

In this subsection, we demonstrate some real examples of how our approach deals with confusing cases, both single-label and multi-label. These cases illustrate the performance and advantages of our GJudge, along with a comparison to other models, including RLJP (Transformer) and the large-scale model ChatGPT 3.5.

We present the comparative results of GJudge, RLJP (Transformer), and ChatGPT 3.5 on a confusing single-label case in Table 8. Regarding the confusing single-label case, RLJP (Transformer) incorrectly classified it as embezzlement due to the multiple occurrences of the term "company" in the case description. It failed to understand that embezzlement must involve the exploitation of one's official position in handling or managing funds, rather than just job convenience. ChatGPT, on the other hand, incorrectly classified it as a multi-label case, also failing to understand that corruption requires the use of convenience on duty rather than at work, and failing to understand that encroachment is to transfer the property of others that one possesses to oneself. Additionally, ChatGPT inaccurately associated Articles 260 and 261 with charges of abuse and abandonment, respectively, which do not correspond to the predicted charges. In contrast, our model correctly predicted all tasks in this case without explicit cues such as the "stolen", and there were no mismatches between the predicted charges and corresponding law articles.

Table 9
Analysis of confusing multi-label cases.

Fact Description: (1)...The victim, Jiang XX, was approached from behind by the defendant, Tan XX, who **forcefully grabbed** the victim's wallet (**containing 1100 yuan**) from their hand. As the wallet fell to the ground, Jiang XX attempted to pick it up, but Tan XX stepped on Jiang XX's hand, **pushed** him/her away, and seized the wallet. Tan XX managed to escape from the scene... (2)...The defendant, Tan XX, **snatched** a small bag containing an iPhone 4 from the victim's hand. Tan XX successfully fled the scene with the stolen iPhone 4, which was valued at **2025 yuan**...
GJudge: Crime of Robbery and Crime of Snatching; Article 263 and Article 267; 3–5 years **RLJP(Transformer)**: Crime of Robbery; Article 263, Article 267 and Article 269; 3–5 years **ChatGPT 3.5**: Crime of Robbery and Crime of Intentional Injury; Article 234 and Article 251; 12 years **Ground-truth**: Crime of Robbery and Crime of Snatching; Article 263 and Article 267; 3 years and 10 months

Table 10
Analysis of different error types.

Error type	CAIL-E	CAIL-M
Over-predicting the count of charge labels	6.37%	3.49%
Under-predicting the count of charge labels	1.61%	7.11%
Over-predicting the count of law article labels	6.96%	5.30%
Under-predicting the count of law article labels	2.50%	7.81%
Consistent charge label count but incorrect results	6.15%	3.60%
Consistent law article label count but incorrect results	6.77%	3.44%
Correct law articles with incorrect charges	5.35%	5.78%
Correct charges with incorrect law articles	7.37%	8.10%
Correct law articles and charges with incorrect terms of penalty	44.17%	39.59%

The results of GJudge, RLJP (Transformer), and ChatGPT 3.5 on a confusing multi-label case are shown in Table 9. In this case, GJudge correctly identifies the offenses of robbery and snatching, associating them with Article 263 and Article 267, respectively. The predicted sentence is 3–5 years, which is close to the ground-truth sentence of 3 years and 10 months. However, RLJP (Transformer) only identifies the crime as robbery and includes additional irrelevant articles (Article 269) in its prediction. ChatGPT 3.5 incorrectly predicts the crime of intentional injury, along with associated articles (Article 251) that do not match the facts. It also assigns an excessively long sentence of 12 years. These results highlight the importance of label consistency constraints in multi-label tasks, as demonstrated by the incorrect associations made by RLJP (Transformer) and ChatGPT 3.5. Furthermore, it underscores the necessity of designing domain-specific frameworks, such as GJudge, tailored to the legal field to ensure accurate and reliable predictions.

5.4.6. Error analysis

We conducted an in-depth analysis of errors in GJudge using the validation dataset, with a specific focus on two critical aspects: multi-label prediction and cross-task consistency.

Multi-label category errors: There are primarily three types of errors in multi-label prediction: over-predicting the count of labels, under-predicting the count of labels, and having consistent label count but incorrect results. As evident from the results in Table 10, it can be observed that on CAIL-E, there is a higher prevalence of errors associated with over-predicting the count of labels, whereas on CAIL-M, errors are more pronounced in under-predicting the count of labels. This phenomenon can be attributed to several factors. In the case of CAIL-E, the dataset primarily comprises single-label cases, which may lead the model to overly focus on the features of single-label cases. Consequently, when handling multi-label cases, the model might mistakenly apply the same logic or features without considering label relationships, resulting in an excess of predicted labels. On CAIL-M, as shown in the results from Table 3, the number of labels primarily falls into single-label, two-label, and two-or-more-label, with the two-or-more-label category being the least prevalent. Therefore, the model tends to lean towards under-predicting the count of labels. Furthermore, a portion of errors falls into the category of consistent label count but incorrect results, which is more pronounced on CAIL-E. This could be explained by the scarcity of training data with multi-label samples, leading to model confusion when predicting multi-label outcomes on a primarily single-label-focused CAIL-E.

Cross-task consistency errors: Our analysis also extended to cross-task consistency errors. As shown in Table 10, it is noticeable that the proportion of correct law articles with incorrect charges is lower than the proportion of correct charges with incorrect law articles. This suggests that in the context of LJP, the task of predicting law articles holds relatively more significance. This observation aligns with the real-world judicial process, where law articles are typically established first and serve as the basis for determining charges. Furthermore, it is noteworthy that even in cases where both law articles and charges are predicted correctly, there remains a high proportion of errors in predicting terms of penalty. This phenomenon could be attributed to the multitude of factors influencing terms of penalty judgments, such as the offender's identity, attitude, societal impact, etc., coupled with inherent subjectivity and flexibility in terms of penalty. Consequently, models struggle to precisely capture the patterns and characteristics of terms of penalty judgments and are susceptible to noise and biases.

To address these issues and improve model performance, strategies such as augmenting training data with more multi-label samples, enhancing consideration of label relationships, and addressing the specific challenges posed by single-label-dominant datasets like CAIL-E should be considered. Additionally, to reduce cross-task consistency errors, we recommend incorporating more information from the factual descriptions and reasoning processes of the cases and applying more rigorous criteria for evaluating the coherence and validity of predicted outcomes across different tasks.

6. Discussion and limitations

In this section, we discuss the innovative aspects and research significance of this paper, as well as the limitations of our study.

6.1. Discussion

In this subsection, the discussion on innovations includes the research significance of multi-task and multi-label aspects, the importance of legal labels, the model's generality, recommendations for related work on LJP and future research directions.

The significance of multi-task multi-label research: This study emphasizes the importance of LJP as a multi-task multi-label problem. Compared to modeling LJP as a single-label scenario, multi-task multi-label methods can more accurately predict charges, applicable legal articles, and penalty terms, thereby providing a comprehensive legal judgment outcome. This approach aids legal professionals and decision-makers in better understanding cases and making more accurate judgments.

The importance of legal labels research: This study designs the legal label graphs to capture correlation relationships between different labels. It offers a new perspective and method for LJP, fully considering the complex relationships between different tasks and labels. It better captures the complexity and diversity of multi-task multi-label scenarios and the professionalism and standardization of the legal domain. By introducing a multi-graph attention consistency expert module, the graph enhancement framework facilitates information exchange and sharing between multiple tasks, enhancing prediction accuracy and maintaining consistency constraints.

Model generality: Our model can adjust the legal label graphs and expert module based on different output objectives. Specifically, the model can handle different legal systems, such as civil law, criminal law, administrative law, etc., by constructing corresponding legal label graphs and datasets. The model can also adapt to varying numbers of tasks and labels by adjusting corresponding model parameters and constructing label information. Thus, our model exhibits strong flexibility and scalability, accommodating diverse LJP scenarios and requirements.

Recommendations for LJP and future directions: We suggest focusing on several aspects for further exploration and attention. Firstly, how to leverage additional external knowledge and data to enrich and supplement inputs and outputs of LJP, such as legal documents, regulations, case law, legal experts, legal commentaries, etc., to enhance the coverage and applicability of LJP. Secondly, designing more reasonable and effective evaluation metrics and methods to measure the performance of LJP. This could involve considering weights for different tasks and labels, accounting for case difficulty, and factoring in variations in judicial preferences to enhance fairness and comparability. Lastly, ensuring the security and reliability of LJP, such as preventing model leakage, tampering, attacks, misinformation, etc., to enhance the reliability and controllability of LJP.

6.2. Limitations

In this subsection, concerning the limitations of the paper, we explore issues such as the subjectivity of the constructed legal label graphs, limited consideration of legal elements, lack of model interpretability, and suboptimal performance in predicting penalties.

Subjectivity in constructing the legal label graphs: Our model depends on manually designed legal label graphs, which may introduce subjective and incomplete aspects. Enhancing the comprehensiveness and objectivity of the legal label graphs can be achieved by incorporating a wider range of data sources, such as legal literature, case databases, legal Q&A, etc. This approach could potentially improve the coverage and quality of the legal label graphs, enhancing the generalization ability and robustness of our model.

Limited consideration of legal elements: Our model only considers three tasks: charges, law articles, and terms of penalty, without addressing other legal elements such as fact determination, evidence evaluation, and principles of legal application. Including these elements within the model's predictive scope could potentially enhance its completeness and practicality.

Lack of model interpretability: Although this study introduces a novel graph boosting with constraints LJP framework, there is room for improvement in the model's interpretability. Further research could explore methods to explain the model's predictive results and the decision-making process of the expert module. Introducing more prior knowledge and constraint conditions could enhance the model's interpretability and rationality. For instance, incorporating legal principles, logical reasoning, causality, etc., to guide and validate the model's predictions.

Suboptimal performance in term of penalty prediction: Our model exhibits suboptimal performance in predicting penalties, possibly due to various factors influencing penalty determination, such as the personal preferences of judges, case complexity, defendant's motives, attitudes, and societal harm. Extracting and quantifying these factors from factual descriptions can be challenging. Improving term of penalty prediction may involve leveraging additional features or data, potentially improving the accuracy and rationality of the model.

7. Conclusion

In this paper, we introduce a novel graph boosting approach with constraints framework for LJP, GJudge, which incorporates fact descriptions and label similarity relationships graphs to enhance robustness and improve performance. It also utilizes a multi-graph attention consistency expert module to handle confusion labels and maintain consistency constraints in cross-tasks. The experimental results show the GJudge's effectiveness in comparison with several competitive baselines, offering an innovative solution for LJP with enhanced robustness and performance.

Building upon the GJudge framework, our future research will concentrate on the following key aspects. Firstly, we will explore techniques to further enhance the term of penalty prediction. This may involve refining the modeling approach, incorporating additional contextual information, or leveraging external resources to improve the predictions. Secondly, we will investigate methods to interpret and explain these relationships, aiming to provide insights into the prediction process and enhance the interpretability of the GJudge framework.

CRediT authorship contribution statement

Suxin Tong: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing review & editing. **Jingling Yuan:** Conceptualization, Supervision, Writing – review & editing, Funding acquisition. **Peiliang Zhang:** Conceptualization, Writing – review & editing, Formal analysis. **Lin Li:** Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by the National Natural Science Foundation of China No. 62276196 and the Hubei Key Laboratory of Big Data in Science and Technology (Wuhan Library of Chinese Academy of Science) No. 20211h0437.

References

Cao, S., Lu, W., & Xu, Q. (2016). Deep neural networks for learning graph representations. In *Proceedings of the 30th AAAI conference on artificial intelligence*. URL: https://doi.org/10.1609/aaai.v30i1.10179.

Che, W., Feng, Y., Qin, L., & Liu, T. (2021). N-LTP: An open-source neural language technology platform for Chinese. In *Proceedings of the 26th conference on empirical methods in natural language processing* (pp. 42–49). Association for Computational Linguistics, URL: https://doi.org/10.18653/v1/2021.emnlp-demo.6.

Chen, Z., Badrinarayanan, V., Lee, C., & Rabinovich, A. (2018). GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th international conference on machine learning* (pp. 793–802). URL: http://proceedings.mlr.press/v80/chen18a.html.

Dai, H., Khalil, E. B., Zhang, Y., Dilkina, B., & Song, L. (2017). Learning combinatorial optimization algorithms over graphs. In *Proceedings of the 31st international conference on neural information processing systems* (pp. 6351–6361). URL: https://proceedings.neurips.cc/paper/2017/hash/d9896106ca98d3d05b8cbdf4fd8b13a1-Abstract.html.

Do, K., Tran, T., & Venkatesh, S. (2019). Graph transformation policy network for chemical reaction prediction. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 750–760). URL: https://doi.org/10.1145/3292500.3330958.

Feng, Y., Li, C., & Ng, V. (2022). Legal judgment prediction via event extraction with constraints. In *Proceedings of the 60th annual meeting of the association for computational linguistics (Volume 1: Long papers)* (pp. 648–664). Association for Computational Linguistics, URL: https://doi.org/10.18653/v1/2022.acl-long.48.

Feng, G., Qin, Y., Huang, R., & Chen, Y. (2023). Criminal Action Graph: A semantic representation model of judgement documents for legal charge prediction. *Information Processing & Management*, *60*(5), Article 103421, URL: https://doi.org/10.1016/j.ipm.2023.103421.

Hang, J., Zhang, M., Feng, Y., & Song, X. (2022). End-to-end probabilistic label-specific feature learning for multi-label classification. In *Proceedings of the 36th AAAI conference on artificial intelligence* (pp. 6847–6855). AAAI Press, URL: https://ojs.aaai.org/index.php/AAAI/article/view/20641.

Hsieh, M., & Tseng, V. (2021). Boosting multi-task learning through combination of task labels - with applications in ECG phenotyping. In *Proceedings of the 35th AAAI conference on artificial intelligence* (pp. 7771–7779). AAAI Press, URL: https://ojs.aaai.org/index.php/AAAI/article/view/16949.

Hu, Z., Li, X., Tu, C., Liu, Z., & Sun, M. (2018). Few-shot charge prediction with discriminative legal attributes. In *Proceedings of the 27th international conference on computational linguistics* (pp. 487–498). Association for Computational Linguistics, URL: https://aclanthology.org/C18-1041/.

Hua, W., Dai, Z., Liu, H., & Le, Q. V. (2022). Transformer quality in linear time. In *Proceedings of the 39th international conference on machine learning* (pp. 9099–9117). URL: https://proceedings.mlr.press/v162/hua22a.html.

Kort, F. (1957). Predicting supreme court decisions mathematically: A quantitative analysis of the "Right to Counsel" cases. *American Political Science Review*, *51*(1), 1–12, URL: https://doi.org/10.2307/1951767.

Liang, J., Tang, J., Gao, F., Wang, Z., & Huang, H. (2023). On region-level travel demand forecasting using multi-task adaptive graph attention network. *Information Sciences*, *622*, 161–177, URL: https://doi.org/10.1016/j.ins.2022.11.138.

Liu, X., Huang, H., & Zhang, Y. (2022). End-to-end event factuality prediction using directional labeled graph recurrent network. *Information Processing & Management*, *59*(2), Article 102836, URL: https://doi.org/10.1016/j.ipm.2021.102836.

Lyu, Y., Wang, Z., Ren, Z., Ren, P., Chen, Z., Liu, X., Li, Y., Li, H., & Song, H. (2022). Improving legal judgment prediction through reinforced criminal element extraction. *Information Processing & Management*, *59*(1), Article 102780, URL: https://doi.org/10.1016/j.ipm.2021.102780.

Ma, J., Zhao, Z., Chen, J., Li, A., Hong, L., & Chi, E. H. (2019). SNR: Sub-network routing for flexible parameter sharing in multi-task learning. In *Proceedings of the 33rd AAAI conference on artificial intelligence* (pp. 216–223). AAAI Press, URL: https://doi.org/10.1609/aaai.v33i01.3301216.

Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., & Chi, E. H. (2018). Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th international conference on ACM knowledge discovery and data mining* (pp. 1930–1939). ACM, URL: https://doi.org/10.1145/3219819.3220007.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 27th annual conference on neural information processing systems* (pp. 3111–3119). URL: https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html.

Nagel, S. S. (1963). Applying correlation analysis to case prediction. *Texas Law Review*, *42*, 1006.

Ruiz, L., Gama, F., & Ribeiro, A. (2020). Gated graph recurrent neural networks. *IEEE Transactions on Signal Processing*, *68*, 6303–6318, URL: https://doi.org/10.1109/TSP.2020.3033962.

Sadat, M., & Caragea, C. (2022). Hierarchical multi-label classification of scientific documents. In *Proceedings of the 27th conference on empirical methods in natural language processing* (pp. 8923–8937). Association for Computational Linguistics, URL: https://ojs.aaai.org/index.php/AAAI/article/view/16949.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681, URL: https://doi.org/10.1109/78.650093.

Segal, J. A. (1984). Predicting supreme court cases probabilistically: The search and seizure cases, 1962–1981. *American Political Science Review*, *78*(4), 891–900, URL: https://doi.org/10.2307/1955796.

Su, J., Lu, Y., Pan, S., Wen, B., & Liu, Y. (2021). RoFormer: Enhanced transformer with rotary position embedding. CoRR abs/2104.09864 URL: https://arxiv.org/abs/2104.09864.

Su, J., Zhu, M., Murtadha, A., Pan, S., Wen, B., & Liu, Y. (2022). ZLPR: A novel loss for multi-label classification. CoRR abs/2208.02955 URL: https://doi.org/10.48550/arXiv.2208.02955.

Sun, T., Shao, Y., Li, X., Liu, P., Yan, H., Qiu, X., & Huang, X. (2020). Learning sparse sharing architectures for multiple tasks. In *Proceedings of the 34th AAAI conference on artificial intelligence* (pp. 8936–8943). AAAI Press, URL: https://ojs.aaai.org/index.php/AAAI/article/view/6424.

Tang, H., Liu, J., Zhao, M., & Gong, X. (2020). Progressive layered extraction (PLE): A novel multi-task learning (MTL) model for personalized recommendations. In *Proceedings of the 14th ACM conference on recommender systems* (pp. 269–278). ACM, URL: https://doi.org/10.1145/3383313.3412236.

Tu, K., Cui, P., Wang, X., Yu, P. S., & Zhu, W. (2018). Deep recursive network embedding with regular equivalence. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2357–2366). URL: https://doi.org/10.1145/3219819.3220068.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st international conference on neural information processing systems* (pp. 6000–6010). URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2017). Graph attention networks. CoRR abs/1710.10903 URL: http://arxiv.org/abs/1710.10903.

Wan, Y., Yuan, C., Zhan, M., & Chen, L. (2022). Robust graph learning with graph convolutional network. *Information Processing & Management*, *59*(3), Article 102916, URL: https://doi.org/10.1016/j.ipm.2022.102916.

Wang, P., Fan, Y., Niu, S., Yang, Z., Zhang, Y., & Guo, J. (2019). Hierarchical matching network for crime classification. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval* (pp. 325–334). ACM, URL: https://doi.org/10.1145/3331184.3331223.

Wang, H., Liu, W., Zhao, Y., Zhang, C., Hu, T., & Chen, G. (2019). Discriminative and correlative partial multi-label learning. In *Proceedings of the 28th international joint conference on artificial intelligence* (pp. 3691–3697). ijcai.org, URL: https://doi.org/10.24963/ijcai.2019/512.

Wang, R., Su, X., Long, S., Dai, X., Huang, S., & Chen, J. (2021). Meta-LMTC: Meta-learning for large-scale multi-label text classification. In *Proceedings of the 26th conference on empirical methods in natural language processing* (pp. 8633–8646). Association for Computational Linguistics, URL: https://doi.org/10.18653/v1/2021.emnlp-main.679.

Wang, H., Wang, J., Wang, J., Zhao, M., Zhang, W., Zhang, F., Xie, X., & Guo, M. (2018). Graphgan: Graph representation learning with generative adversarial nets. In *Proceedings of the 32nd AAAI conference on artificial intelligence*. URL: https://doi.org/10.1609/aaai.v32i1.11872.

Wu, Y., Liu, Y., Lu, W., Zhang, Y., Feng, J., Sun, C., Wu, F., & Kuang, K. (2022). Towards interactivity and interpretability: A rationale-based legal judgment prediction framework. In *Proceedings of the 27th conference on empirical methods in natural language processing* (pp. 4787–4799). Association for Computational Linguistics, URL: https://aclanthology.org/2022.emnlp-main.316.

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., & Weinberger, K. (2019). Simplifying graph convolutional networks. In *Proceedings of the 36th international conference on machine learning* (pp. 6861–6871). URL: http://proceedings.mlr.press/v97/wu19e.html.

Xiao, C., Zhong, H., Guo, Z., Tu, C., Liu, Z., Sun, M., Feng, Y., Han, X., Hu, Z., Wang, H., & Xu, J. (2018). CAIL2018: A large-scale legal dataset for judgment prediction. CoRR abs/1807.02478 URL: http://arxiv.org/abs/1807.02478.

Xu, N., Wang, P., Chen, L., Pan, L., Wang, X., & Zhao, J. (2020). Distinguish confusing law articles for legal judgment prediction. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 3086–3095). Association for Computational Linguistics, URL: https://doi.org/10.18653/v1/2020.acl-main.280.

Yang, S., Chang, X., Zhu, G., Cao, J., Qin, W., Wang, Y., & Wang, Z. (2023). GAA-PPO: A novel graph adversarial attack method by incorporating proximal policy optimization. *Neurocomputing*, *557*, Article 126707, URL: https://doi.org/10.1016/j.neucom.2023.126707.

Yang, W., Jia, W., Zhou, X., & Luo, Y. (2019). Legal judgment prediction via multi-perspective bi-feedback network. In *Proceedings of the 28th international joint conference on artificial intelligence* (pp. 4085–4091). ijcai.org, URL: https://doi.org/10.24963/ijcai.2019/567.

Yang, S., Tong, S., Zhu, G., Cao, J., Wang, Y., Xue, Z., Sun, H., & Wen, Y. (2022). MVE-FLK: A multi-task legal judgment prediction via multi-view encoder fusing legal keywords. *Knowledge-Based Systems*, *239*, Article 107960, URL: https://doi.org/10.1016/j.knosys.2021.107960.

Yokoi, S., Takahashi, R., Akama, R., Suzuki, J., & Inui, K. (2020). Word rotator's distance: Decomposing vectors gives better representations. CoRR abs/2004.15003 URL: https://arxiv.org/abs/2004.15003.

You, J., Liu, B., Ying, R., Pande, V., & Leskovec, J. (2018). Graph convolutional policy network for goal-directed molecular graph generation. In *Proceedings of the 32nd international conference on neural information processing systems* (pp. 6412–6422). URL: https://proceedings.neurips.cc/paper/2018/hash/d60678e8f2ba9c540798ebbde31177e8-Abstract.html.

Yue, L., Liu, Q., Jin, B., Wu, H., Zhang, K., An, Y., Cheng, M., Yin, B., & Wu, D. (2021). NeurJudge: A circumstance-aware neural framework for legal judgment prediction. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval* (pp. 973–982). ACM, URL: https://doi.org/10.1145/3404835.3462826.

Zhao, Q., Gao, T., & Guo, N. (2023). LA-MGFM: A legal judgment prediction method via sememe-enhanced graph neural networks and multi-graph fusion mechanism. *Information Processing & Management*, *60*(5), Article 103455, URL: https://doi.org/10.1016/j.ipm.2023.103455.

Zhao, J., Guan, Z., Xu, C., Zhao, W., & Chen, E. (2022). Charge prediction by constitutive elements matching of crimes. In *Proceedings of the 31st international joint conference on artificial intelligence* (pp. 4517–4523). ijcai.org, URL: https://doi.org/10.24963/ijcai.2022/627.

Zhong, H., Guo, Z., Tu, C., Xiao, C., Liu, Z., & Sun, M. (2018). Legal judgment prediction via topological learning. In *Proceedings of the 23rd conference on empirical methods in natural language processing* (pp. 3540–3549). Association for Computational Linguistics, URL: https://doi.org/10.18653/v1/d18-1390.

Zhu, G., Cao, J., Chen, L., Wang, Y., Bu, Z., Yang, S., Wu, J., & Wang, Z. (2023). A multi-task graph neural network with variational graph auto-encoders for session-based travel packages recommendation. *ACM Transactions on the Web*, *17*(3), 1–30, URL: https://doi.org/10.1145/3577032.