# A Circumstance-Aware Neural Framework for Explainable Legal Judgment Prediction

Linan Yue, Qi Liu, *Member, IEEE*, Binbin Jin, Han Wu, and Yanqing An

***Abstract*—Massive legal documents have promoted the application of legal intelligence. Among them, Legal Judgment Prediction (LJP) has emerged as a critical task, garnering significant attention. LJP aims to predict judgment results for multiple subtasks, including charges, law articles, and terms of penalty. Existing studies primarily focus on utilizing the entire factual description to produce judgment results, overlooking the practical judicial scenario where judges consider various crime circumstances to decide verdicts and sentencing. To this end, in this article, we propose a circumstance-aware LJP framework (i.e., NeurJudge) by exploring the circumstances of crime. Specifically, NeurJudge first separates the factual description into different circumstances with the predicted results of intermediate subtasks and then employs them to yield results of other subtasks. Besides, as confusing verdicts may degrade the performance of LJP, we further develop a variant of NeurJudge (NeurJudge+) that incorporates the semantics of labels (charges and law articles) into facts to yield more expressive and distinguishable fact representations. Finally, to provide explanations for LJP, we extend NeurJudge to an explainable LJP framework E-NeurJudge with a cooperative teacher-student system. The teacher system is NeurJudge which exploits legal particularities well but lacks explanation capability. The student system is a rationalization method that provides explainability but fails to utilize legal particularities. To combine the advantages of the above methods, we use a transferring function to transfer legal particularities from the teacher to the student, making a trade-off between yielding LJP results and rendering them explainable. Extensive experimental results on real-world datasets validate the effectiveness of our proposed frameworks.**

***Index Terms*—Explainable legal document mining, fact separation, label embedding, rationalization, text classification.**

## I. Introduction

In recent years, with the development of data engineering, the legal field has accumulated an increasing amount of data, such as legal documents. By mining these data, legal intelligence can be effectively promoted, which can provide legal assistance to laymen and legal professionals. Among them, Legal Judgment Prediction (LJP) has received increasing attention, which is a fundamental task in legal document mining. Researchers [1], [2], [3], [4] formalize LJP as three text classification subtasks (i.e., charges, articles, and terms of penalty prediction). Although they achieve promising results in LJP, these methods still suffer from several limitations on judgment process modeling, confusing verdicts distinctions, and explainable judgment prediction.

Manuscript received 18 December 2022; revised 13 February 2024; accepted 28 March 2024. Date of publication 11 April 2024; date of current version 27 September 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFF0901003 and in part by the Fundamental Research Funds for the Central Universities. Recommended for acceptance by X. Chen. *(Corresponding author: Qi Liu.)*

Linan Yue and Yanqing An are with the State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China, Hefei, Anhui 230026, China (e-mail: lnyue@mail.ustc.edu.cn; anyq@mail.ustc.edu.cn).

Qi Liu is with the State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China, Hefei, Anhui 230026, China, and also with the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, Anhui 230026, China (e-mail: qiliuql@ustc.edu.cn).

Binbin Jin is with Huawei Cloud Computing Technologies Company Ltd., D04 E516 Dublin, Ireland (e-mail: jinbinbin1@huawei.com).

Han Wu is with Career Science Lab, Boss Zhipin, Beijing 100028, China (e-mail: ustcwuhan@gmail.com).

Digital Object Identifier 10.1109/TKDE.2024.3387580

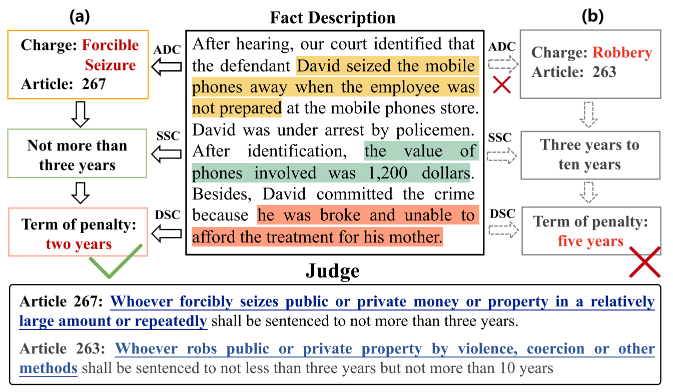
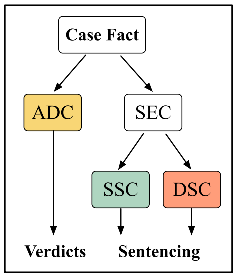

(a)
Fact Description
(b)
Charge: Forcible Seizure
Article: 267
ADC
After hearing, our court identified that the defendant David seized the mobile phones away when the employee was not prepared at the mobile phones store. David was under arrest by policemen. After identification, the value of phones involved was 1,200 dollars. Besides, David committed the crime because he was broke and unable to afford the treatment for his mother.
ADC
Charge: Robbery
Article: 263
SSC
Not more than three years
SSC
Three years to ten years
DSC
Term of penalty: two years
DSC
Term of penalty: five years
Judge
Article 267: Whoever forcibly seizes public or private money or property in a relatively large amount or repeatedly shall be sentenced to not more than three years.
Article 263: Whoever robs public or private property by violence, coercion or other methods shall be sentenced to not less than three years but not more than 10 years

Fig. 1. (a). A fine-grained judgment process, where judges decide judgment results in sequence according to ADC, SSC, and DSC, respectively. (b). A misjudgment process. Since Article 263 and Article 267 are confusable, it seems to misjudge the results of articles and lead to errors in subsequent tasks easily.

First, when modeling the judgment process, existing approaches mostly exploit the entire factual description to predict all subtasks [2], [3], [5]. However, they ignore the practical judgment process, where human judges decide the verdicts and sentencing based on the circumstances of crime [6]. Among them, different circumstances are commonly located in different parts of the factual description. Specifically, in Fig. 1(a), human judges first decide that the defendant committed *forcible seizure* and apply *law article 267* based on "*David seized the mobile phones away when the employee was not prepared*" in fact.

Such part of the fact is referred to as ***AD****judging* ***C****ircumstance* (ADC) which decides the corresponding verdicts (i.e., charges and law articles). Then, judges further determine the term of penalty according to ***SE****ntencing* ***C****ircumstance* (SEC) which consists of ***S****tatutory* ***S****entencing* ***C****ircumstance* (SSC) and ***D****iscretionary* ***S****entencing* ***C****ircumstance* (DSC).

Specifically, based on the fact "*the value of phones involved was 1,200 dollars*" which is consistent with the *law article 267* and referred to as SSC, the judges decide the defendant shall



(a)

| Circumstances of Crime | Abbreviation | Definition |
|---|---|---|
| ***AD**judging **C**ircumstance* | ADC | A part of fact description which is in line with the definition of charge and decides the corresponding verdicts. |
| ***SE**ntencing **C**ircumstance* | SEC | A part of fact description which determines the terms of penalty under the premise that a certain act has constituted a crime. |
| ***S**tatutory **S**entencing **C**ircumstance* | SSC | The relevant circumstances expressly provided for in the Criminal Law, which must be taken into account in sentencing. |
| ***D**iscretionary **S**entencing **C**ircumstance* | DSC | Circumstances that need to be considered when sentencing according to the legislative spirit and trial practical experience. |

(b)

Fig. 2. (a) Relationships between the Circumstances of Crime and judgment results. (b) Details about the Circumstances of Crime.

be sentenced to not more than three years in prison. Finally, considering "*he was broke and unable to afford the treatment for his mother*." which is referred to as DSC, the judges make the decision at their discretion (i.e., the defendant shall be sentenced to two years). Details about the circumstances of crime are shown in Fig. 2. In summary, we can infer the judgment process based on crime circumstances is complex. Therefore, it is significant to design a strategy to simulate this process.

Besides, several confusing verdicts including charges and law articles may degrade the overall performance of LJP. Specifically, high similarities of the charges or law articles descriptions make their corresponding verdicts vulnerable to misjudge and further lead to incorrect sentences. For example, in Fig. 1(b), *Articles 267* and *263* both describe offenses of violating property. The only distinguishing factor lies in *Article 263* also describes violence while *Article 267* does not. Therefore, there exists a high likelihood of misjudging the verdict as *Article 263*, thereby adversely impacting subsequent tasks. Similarly, the problem exists in confusing charges. Hence, it is challenging to distinguish the semantics of confusing charges and articles for LJP.

Last but not least, both legal practitioners and laymen usually care about not only the predicted judgment results but also the rationales for these results. However, although existing LJP methods have achieved promising performance, the judgment results are still unexplainable, rendering these methods unreliable. One feasible approach is that employing the selective rationalization methods [7], [8] to improve the explainability of LJP. Particularly, it first extracts a subsequence in fact (i.e., rationale) and encodes it as rationale representations. The results are then derived solely based on these representations, with the extracted rationale serving as an explanation for the prediction. For example, in Fig. 1, the goal of rationalization is to yield the charge as *forcible seizure* while simultaneously extracting "*David seized the mobile phones away when the employee was not prepared*" from the fact as evidence to support the prediction. Finally, the extracted sentence is employed as an explanatory rationale for the charge prediction. However, when employing the rationalization method to LJP, it may degrade the performance of predicting judgments since it fails to utilize legal particularities well [8].

To directly achieve the primary goal of yielding judgment results by addressing the first two challenges, in our preliminary work [9], we propose a circumstance-aware neural framework (i.e., NeurJudge) for LJP to simulate the practical judgment logic. Based on the judgment process, NeurJudge employs the predictions of intermediate subtasks to categorize the facts into distinct circumstances. The initial step involves separating the ADC and SEC from the fact, using the relevant charge as a basis. Then, the ADC is employed to predict the corresponding article. Building upon this, the predicted article is utilized to identify the SSC and DSC within the SEC. These identified circumstances are then utilized to yield the term of penalty.

Besides, to alleviate the confusing verdict problem, we propose a variant of NeurJudge (NeurJudge+) with an external graph-based label embedding module. Particularly, we first construct two similarity graphs of labels (i.e., charges and articles). For the label similarity graph, we calculate the weight between two label nodes by cosine similarity and keep the nodes and edges with similarity greater than 0.6 as the final label similarity graph. For example, in Fig. 1, *Forcible Seizure* and *Robbery* are two interconnected nodes in the charge similarity graph, and *Article 267* and *263* are two interconnected nodes in the article similarity graph. Then, we employ a decomposition strategy to extract special label features from graphs. These features are obtained by analyzing the interaction between the graph features and the factual description. This process enables the identification and capture of distinguishable components within the case. By incorporating these more expressive fact representations into NeurJudge, we can enhance its overall performance.

In NeurJudge, it fails to explicitly explain what circumstances in case facts drive NeurJudge to make certain predictions. Therefore, in this paper, we extend NeurJudge and propose an explainable NeurJudge framework (E-NeurJudge) to generate explanations to support the predicted judgment results. Specifically, E-NeurJudge consists of a two-pass system, including the original LJP system (teacher) and an explanatory system (student), which are trained cooperatively. Among them, the teacher model is the original NeurJudge (or NeurJudge+), which generates the crime circumstances representations well and fully exploits legal particularities but fails to explain yielded judgment results. The student model is a selective rationalization method that can provide the explainability for LJP but does not utilize the legal particularities well. E-NeurJudge combines the advantages of the two above models with a transferring function, which encourages the circumstance and rationale representations generated by the two models to encode the identical information. This alignment process enables the student model to effectively match the circumstance representations and learn the legal intricacies from the teacher model. Then, the extracted rationales can be

TABLE I
CHARACTERISTICS OF ALL MODELS

| Model | Legal Particularities | Explainability |
|---|---|---|
| NeurJudge | ✓ | × |
| Rationalization | × | ✓ |
| E-NeurJudge | ✓ | ✓ |

considered as the crime circumstances to support the judgment results. Table I lists detailed characteristics of the above models.

In summary, the major contributions of this paper are shown as follows:

- To simulate the judgment process for LJP, we propose a preliminary circumstance-aware framework NeurJudge. Besides, to address the confusing verdict problem, we develop a variant of NeurJudge (NeurJudge+) that fuses distinguishable verdict features into LJP.
- To provide explanations for LJP, we extend NeurJudge and propose an explainable NeurJudge framework (E-NeurJudge), which makes a trade-off between yielding accurate LJP results and rendering them explainable.
- Extensive experiments on real-world datasets validate the effectiveness of both NeurJudge and E-NeurJudge.

## II. Related Work

*Legal Judgment Prediction.* The success of deep neural networks has sparked interest in the field of legal document mining [1], [10], [11], [12], [13], [14]. Within this domain, Legal Judgment Prediction (LJP) has garnered considerable attention [1], [4], [10], [15], [16]. Existing approaches in LJP can be broadly categorized into two types. The first type focuses on leveraging rich legal knowledge to predict judgment results. For instance, [1], [5], [17] incorporated the semantics of law articles and charges into LJP using attention-based methods. Additionally, some researchers address the problem of confusing charges in prediction by manually defining legal attributes of charges [18]. The second type of approach seeks to model the judgment process by simulating human judges. For example, [2] explores multiple subtasks involved in legal judgment and models the judgment process based on the topological order of these subtasks. [3] extends this work by incorporating dependencies among prediction results of multiple subtasks in LJP. Despite the extensive research on LJP, there has been limited consideration for the fine-grained judgment process, where judges determine verdicts and sentencing based on different circumstances of a crime. Besides, there is a consideration about how to provide explanations for the predicted judgment results.

*Label Embedding.* There has been significant research [19], [20], [21], [22] conducted on exploring the rich information contained within class labels using label embedding methods. Among them, [23] proposed an approach where they embedded words and labels in the same latent space and designed an attention framework to measure the compatibility between text and labels. [24] introduced a framework that fed the concatenation of label descriptions and text inputs to the classifier. Although these studies recognized the importance of label semantics, they did not specifically address problems involving labels with confusing semantics.

TABLE II
MAIN MATHEMATICAL NOTATIONS

| Notation | Description |
|---|---|
| $s^d = \{w_1^d, \dots, w_{l_d}^d\}$ | a word sequence of the factual description |
| $Y_c = \{c_1, \dots, c_n\}$ | the set of charge labels |
| $s^{c_i} = \{w_1^{c_i}, \dots, w_{l_c}^{c_i}\}$ | a word sequence of the charge $c_i$ description |
| $Y_a = \{a_1, \dots, a_m\}$ | the set of article labels |
| $s^{a_j} = \{w_1^{a_j}, \dots, w_{l_a}^{a_j}\}$ | a word sequence of the article $a_j$ description |
| $Y_t = \{t_1, \dots, t_k\}$ | the set of term of penalty labels |
| $t_z$ | an arbitrary non-overlapping interval |

*Selective Rationalization.* Selective Rationalizations have received increasing attention for providing the model explainability [7], [25], [26], [27], [28], [29], [30]. Specifically, Lei et al. [7] first proposed a classical framework for rationalization consisting of a *selector* and a *predictor*. Among them, the *selector* extracts a text span (i.e., rationale) from the original text input, and the *predictor* yields the result based on the selected span. Along this line, Bastings et al. [31] studied a HardKuma distribution for reparameterized gradient estimates, ensuring the end-to-end differentiability of this framework. Although several researchers have conducted experiments on the LJP task [8], their objective was simply to validate their models on LJP datasets and failed to explore the legal particularities deeply. Besides, more related to selective rationalization is the argument extraction method [32], [33], [34], which identifies arguments, along with their components in text. Different from the rationalization, the argument extraction often requires annotation of which tokens or sentences in text belong to the argument, while the annotated rationale is unavailable during the whole training process of the selective rationalization.

## III. Problem Definition

Table II shows notations about the input. Given a training sample, it consists of four components, including: the factual description $s^d$, the set of charge labels $Y_c$ and their corresponding descriptions $\{s^{c_i}\}_i^n$, article labels $Y_a$ and the article descriptions $\{s^{a_i}\}_i^m$, and the set of term of penalty labels $Y_t$. The LJP task is formulated as a single label multi-task text classification problem based on massive legal data, and our goal is two-fold:

1) yield the judgment results based on the factual description, including charges, articles, and terms of penalty;
2) extract the corresponding circumstances of crime from the factual description to support the predicted results.

Taking the case in Fig. 1 for example, based on the factual description, our method yields the charge, article, and term of penalty as *Forcible Seizure*, *Article 267*, and an interval of *two to three years*, respectively. Meanwhile, it extracts ADC, SSC, and DSC to explain these results.

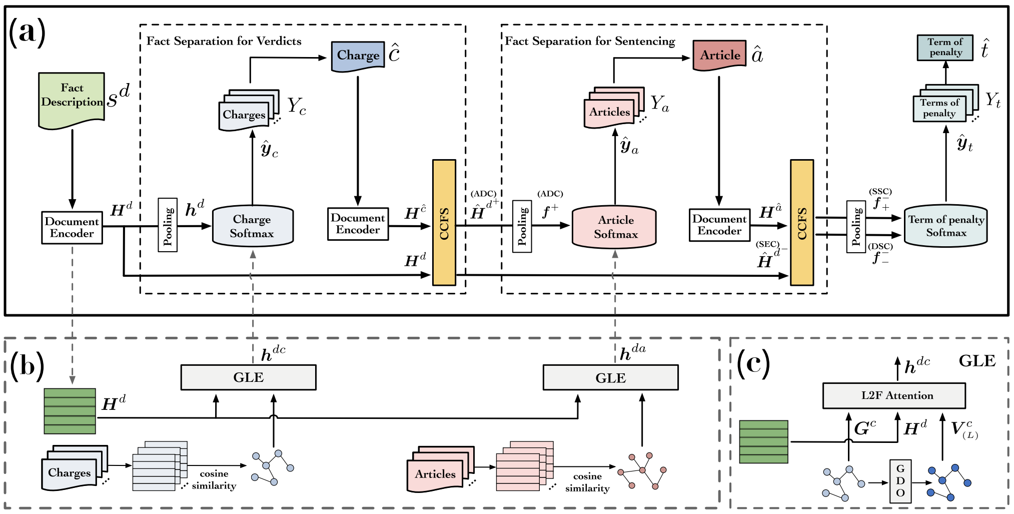


Fig. 3. (a). Overview of our NeurJudge framework. (b). More expressive fact representations in NeurJudge+. (c). The Architecture of Graph-based Label Embedding (GLE) method.

## IV. Circumstance-Aware NeurJudge

### A. NeurJudge Framework

To simulate the practical judgment process, where judges determine the verdicts and sentencing based on different crime circumstances, we propose a NeurJudge framework (Fig. 3(a)) which directly achieves the goal of yielding judgment results. NeurJudge mainly consists of two components (i.e., Document Encoder and Fact Separation). Specifically, we employ document encoders to generate the semantic vectors of the text descriptions on fact, charge, and article labels. Then, in Fact Separation, we propose a ***C***ircumstances of ***C***rime aware ***F***act ***S***eparation (**CCFS**) method to separate fact into three parts (i.e., ADC, SSC, and DSC) based on the vectors from document encoders. Finally, we adopt them to corresponding subtasks to predict the judgment results.

*1) Document Encoder:* In this section, we design the document encoder to generate vector representations for the factual description, charge labels, and article labels. We implement two types of encoders: GRU-based NeurJudge and BERT-based NeurJudge. For GRU-based NeurJudge, we use a bi-directional GRU [35] as our encoder. Given a word sequence of the factual description $s^d$, we map each word of $s^d$ to its word embedding using pre-trained word vectors, such as word2vec [36]. This results in a word embedding sequence $\boldsymbol{E}^d = \{\boldsymbol{e}_1^d, \ldots, \boldsymbol{e}_{l_d}^d\}$, where $\boldsymbol{e}_i^d \in \mathbb{R}^{d_w}$ and $d_w$ is the dimension of the word embedding. We then use a bi-directional GRU encoder to embed $\boldsymbol{E}^d$ into continuous hidden states:

$$\boldsymbol{H}^d = \text{Bi-GRU}(\boldsymbol{E}^d), \tag{1}$$

where $\boldsymbol{H}^d = \{\boldsymbol{h}_1^d, \boldsymbol{h}_2^d, \ldots, \boldsymbol{h}_{l_d}^d\} \in \mathbb{R}^{l_d \times d_s}$ and $d_s$ is twice the size of the hidden state.

For BERT-based NeurJudge, we use BERT [37] as our encoder. We input the word sequence $s^d$ into BERT. The output of BERT is denoted as $\boldsymbol{H}^d = \{\boldsymbol{h}_1^d, \boldsymbol{h}_2^d, \ldots, \boldsymbol{h}_{l_d}^d\} \in \mathbb{R}^{l_d \times d_s}$, where $d_s$ represents the dimension of the last hidden layer of BERT. Similarly, for an arbitrary charge description $s^{c_i}$ and article description $s^{a_j}$, we can obtain their hidden states as $\boldsymbol{H}^{c_i} \in \mathbb{R}^{l_c \times d_s}$ and $\boldsymbol{H}^{a_j} \in \mathbb{R}^{l_a \times d_s}$, respectively.

*2) Fact Separation:* As shown in Fig. 2(a), judges focus on different circumstances of crime. They first choose ADC from the facts to determine verdicts, and then decide the sentencing according to SEC that consists of SSC and DSC.

However, it is challenging to represent the different circumstances of the crime that are located in various parts of the fact. From the observations in Fig. 2(b), we notice that ADC is the part of the fact that is consistent with the definition of the charge. In other words, ADC serves as a similar component between the factual description and the definition of the charges. On the other hand, SEC can be considered as the dissimilar component, as it consists of SSC and DSC. Similarly, SEC is composed of SSC and DSC, where SSC is ruled by law articles and can be seen as a similar component between the articles and SEC. In contrast, DSC is the dissimilar component. Inspired by observations, we propose a ***C***ircumstances of ***C***rime aware ***F***act ***S***eparation (**CCFS**) method to separate fact for judgment prediction.

In fact separation for verdicts, we separate ADC and SEC aware fact representation with the definition of charge. Specifically, the input of CCFS is both the fact representation $\boldsymbol{H}^d$ and the charge representation $\boldsymbol{H}^{\hat{c}}$, where $\hat{c}$ is the predicted charge based the fact. To get the predicted charge $\hat{c}$, we first apply per-dimension mean-pooling over $\boldsymbol{H}^d$ to obtain the sentence-level fact representation $\boldsymbol{h}^d \in \mathbb{R}^{d_s}$:

$$\boldsymbol{h}^d = \sum_{z=1}^{l_d} \boldsymbol{h}_z^d \,/\, l_d. \tag{2}$$

Next, we predict the most related charge $\hat{c}$ in the set of charge labels $Y_c$:

$$\hat{\boldsymbol{y}}_c = \text{softmax}\left(\boldsymbol{W}_c \boldsymbol{h}^d + \boldsymbol{b}_c\right). \tag{3}$$

Then, the most relevant charge $\hat{c}$ is computed as:

$$\hat{c} = \arg\max_{i=1,\ldots,n} \hat{y}_{c_i}, \tag{4}$$

where $\hat{y}_{c_i} \in \hat{\boldsymbol{y}}_c, i = 1, \ldots, n$. After obtaining the corresponding charge's semantic vector $\boldsymbol{H}^{\hat{c}}$, we take it and the fact vectors $\boldsymbol{H}^d$ as the input of CCFS to separate case fact. Inspired by [38], [39], [40], we adopt the Vector Rejection method to separate the vectors into similar and dissimilar components. We quantify the relevance between the charge and the fact, indicating which charge words are most relevant to each fact word:

$$\boldsymbol{D} = \boldsymbol{H}^d \boldsymbol{W}_f \boldsymbol{H}^{\hat{c}\top}, \tag{5}$$

where $\boldsymbol{W}_f \in \mathbb{R}^{d_s * d_s}$. Then, we obtain $\tilde{\boldsymbol{H}}^d \in \mathbb{R}^{l_d * d_s}$ which contains the attended charges vectors for the entire fact:

$$\tilde{\boldsymbol{H}}^d = \mathrm{softmax}(\boldsymbol{D})\, \boldsymbol{H}^{\hat{c}}. \tag{6}$$

Next, we employ the vector rejection over $\boldsymbol{H}^d$ and $\tilde{\boldsymbol{H}}^d$ to yield the similar and dissimilar component between them:

$$\hat{\boldsymbol{H}}^{d^+} = \frac{\boldsymbol{H}^d \cdot \tilde{\boldsymbol{H}}^d}{\tilde{\boldsymbol{H}}^d \cdot \tilde{\boldsymbol{H}}^d} \tilde{\boldsymbol{H}}^d, \quad \hat{\boldsymbol{H}}^{d^-} = \boldsymbol{H}^d - \hat{\boldsymbol{H}}^{d^+}, \tag{7}$$

where $\boldsymbol{H}^d$ is separated into two components: the parallel vectors $\hat{\boldsymbol{H}}^{d^+}$ and the perpendicular vectors $\hat{\boldsymbol{H}}^{d^-}$. Among them, $\hat{\boldsymbol{H}}^{d^+}$ correspond to the similar component and can be considered as ADC, and $\hat{\boldsymbol{H}}^{d^-}$ could be considered as the dissimilar component (i.e., SEC).

Similarly, in fact separation for sentencing, we take the separated SEC vectors $\hat{\boldsymbol{H}}^{d^-}$ and the predicted article vectors $\hat{\boldsymbol{H}}^{d^-}$ as the input. To get the predicted article $\hat{a}$, we first obtain the sentence-level ADC vectors $\boldsymbol{f}^+$ by applying a mean-pooling over $\hat{\boldsymbol{H}}^{d^+}$, and define the following linear function to predict the most related article $\hat{a}$ in $Y_a$:

$$\hat{\boldsymbol{y}}_a = \mathrm{softmax}\left(\boldsymbol{W}_a \boldsymbol{f}^+ + \boldsymbol{b}_a\right), \tag{8}$$

where $\hat{a} = \arg\max \hat{y}_{a_j}, \hat{y}_{a_j} \in \hat{\boldsymbol{y}}_a$. Based on $\boldsymbol{H}^{\hat{a}}$, we separate SEC vectors $\hat{\boldsymbol{H}}^{d^-}$ into SSC vectors and DSC vectors, and apply mean-pooling operation over them to get sentence-level SSC vectors $\boldsymbol{f}^-_+ \in \mathbb{R}^{d_s}$ and DSC vectors $\boldsymbol{f}^-_- \in \mathbb{R}^{d_s}$.

*3) Prediction and Training:* Considering the prediction for charges and articles ((3) and (8)), we introduce the term of penalty prediction based on SSC vectors $\boldsymbol{f}^-_+$ and DSC vectors $\boldsymbol{f}^-_-$:

$$\hat{\boldsymbol{y}}_t = \mathrm{softmax}\left(\boldsymbol{W}_t[\boldsymbol{f}^-_+; \boldsymbol{f}^-_-] + \boldsymbol{b}_t\right), \tag{9}$$

where the symbol " ; " denotes the concatenation operation. Finally, we utilize the cross-entropy loss function for each subtask and compute the weighted sum as the overall loss:

$$\mathcal{L}_{ner} = -\sum_{j\in\{c,a,t\}} \lambda_j \sum_{k=1}^{|Y_j|} \boldsymbol{y}_{j,k} \log\left(\hat{\boldsymbol{y}}_{j,k}\right), \tag{10}$$

where $|Y_j|$ denotes the number of labels for subtask $j$, and $\lambda_j$ represents the adjusted hyperparameter.

## B. NeurJudge+

To address the issue of confusing verdicts, we propose a variant of NeurJudge, called NeurJudge+. This variant combines the structures of both Fig. 3(a) and (b) (i.e., NeurJudge). NeurJudge+ introduces two similarity graphs for aggregating confusing charges and articles. Then, we present the *G*raph-based *L*abel *E*mbedding (GLE) method in NeurJudge+. GLE comprises two key components: the *G*raph *D*ecomposition *O*peration (GDO) and the *L*abel-to-*F*act (L2F) attention. This method is visually depicted in Fig. 3(c). Specifically, GDO extracts label features distinguished from other similar labels on label similarity graphs. Furthermore, L2F attention facilitates the interaction between label features and fact vectors, enabling the model to focus on the distinctive components of a case and enhance the representation of facts accordingly. Incorporating this refined representation into NeurJudge, we aim to enhance the performance of charge and article prediction.

*1) Graph Construct Layer:* To address the issue of confusing verdicts, one effective approach is to improve the interaction between verdict labels and case facts using label embedding methods. Since similar labels often have indistinguishable representations, the crucial step is to extract distinctive label features by removing the shared components and preserving the dissimilarities between labels. Hence, it is essential to devise a strategy to identify which labels are similar to each other.

An intuitive way is to exploit the tree-shaped structure in Criminal Law of the civil law system [17] where we can consider the children's labels from the same parent label to be similar. However, from the perspective of legal knowledge, we argue that the tree structure fails to capture the similarity between labels. We illustrate this opinion by taking the charge labels as an example. In the tree structure, the child nodes under the same parent node may not necessarily be similar. For instance, under the parent node of *crimes endangering public security*, there are crimes such as *arson* and *traffic accidents*, which are intuitively unrelated. Furthermore, child nodes under different parent nodes may exhibit similarity. For example, the crime of *occupation* (child) under the crime of *violating property* (parent) is similar to the crime of *embezzlement* (child) under the crime of *embezzlement and bribery* (parent). These concepts are often discussed and differentiated by specialized legal literature [6]. Hence, we believe that the tree structure does not adequately reflect the similarity between charges. Therefore, based on the tree structure, we extend it into the graph structure to model the relationship between confusing charge and article labels.

Specifically, we consider the charge similarity graph as an example. Fig. 4(a) shows a hierarchical tree structure among the criminal categories (represented by **P**) and the specific charges (represented by **c**). Utilizing this tree structure, our first step involves connecting all child labels to form a fully connected graph using the charge labels, as illustrated in Fig. 4(b). Next, we calculate the weight between two nodes by employing cosine similarity, based on the word embeddings associated with each node. To be specific, we collect the embeddings of all the words and obtain the sentence-level text representations through a mean pooling method. Based on the sentence representations,

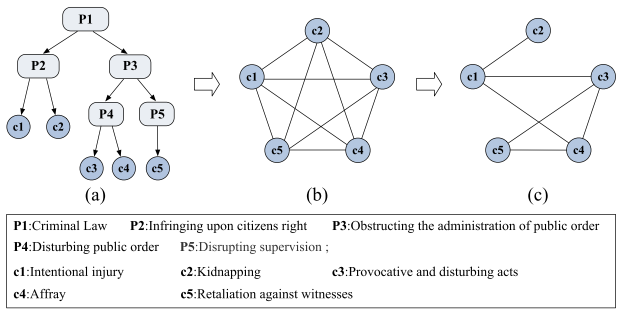

P1
P2
P3
P4
P5
c1
c2
c3
c4
c5
(a)
(b)
(c)
P1:Criminal Law P2:Infringing upon citizens right P3:Obstructing the administration of public order
P4:Disturbing public order P5:Disrupting supervision ;
c1:Intentional injury c2:Kidnapping c3:Provocative and disturbing acts
c4:Affray c5:Retaliation against witnesses

Fig. 4. Charge similarity graph construction.

we can achieve the similarity computation. Finally, to focus on the most similar charges, we eliminate edges with weights below a predefined threshold $\tau$ (as shown in Fig. 4(c)). The same approach can be applied to obtain the article similarity graph. To obtain representations for these two graphs, we employ the *D*ocument *E*ncoder, which is described in Section IV-A1. This allows us to obtain arbitrary charge or article vectors, denoted as $\boldsymbol{H}^{c_i}$ and $\boldsymbol{H}^{a_j}$, respectively. By applying the per-dimension mean-pooling operation, we obtain $\boldsymbol{h}^{c_i}$ and $\boldsymbol{h}^{a_j}$ for each charge and article, respectively. We then stack all $\boldsymbol{h}^{c_i}$ and $\boldsymbol{h}^{a_j}$ column-wise to obtain the original charge features $\boldsymbol{G}^c = \{\boldsymbol{h}^{c_1}, \ldots, \boldsymbol{h}^{c_n}\} \in \mathbb{R}^{n \times d_s}$, and the original article features $\boldsymbol{G}^a = \{\boldsymbol{h}^{a_1}, \ldots, \boldsymbol{h}^{a_m}\} \in \mathbb{R}^{m \times d_s}$, where $c_i$ represents a charge node and $a_j$ is an article node.

*2) Graph Decomposition Operation:* To extract the special features of labels from the graphs $\boldsymbol{G}^c$ and $\boldsymbol{G}^a$, we employ the graph decomposition operation (GDO). GDO focuses on learning the unique features of neighboring nodes, differentiating it from Graph Convolutional Networks (GCN) [41], which can suffer from over-smoothing issues [42] that make aggregated node representations indistinguishable. In GDO, when aggregating feature information from neighboring nodes into a central node, we aim to remove similar features and utilize the dissimilar ones to enrich the representation of the central node. Inspired by Xu et al. [5], we adopt the Vector Rejection operation to obtain similar and dissimilar features among nodes. Specifically, for an arbitrary charge node $c_i$ in $\boldsymbol{G}^c$ at the $l_{th}$ layer, the information aggregation from neighbors is as follows:

$$\boldsymbol{v}^{c_i}_{(l+1)} = \boldsymbol{v}^{c_i}_{(l)} - \sum_{c_j \in N_i} \frac{g(\boldsymbol{v}^{c_i}_{(l)}, \boldsymbol{v}^{c_j}_{(l)})}{|N_i|}, g\left(\boldsymbol{v}^{c_i}, \boldsymbol{v}^{c_j}\right) = \frac{\boldsymbol{v}^{c_i} \cdot \boldsymbol{v}^{c_j}}{\boldsymbol{v}^{c_j} \cdot \boldsymbol{v}^{c_j}} \boldsymbol{v}^{c_j}, \tag{11}$$

where $\boldsymbol{v}^{c_i}_{(l)} \in \mathbb{R}^{d_s}$ represents the vector of $c_i$ at the $l_{th}$ layer. Specifically, we set $\boldsymbol{h}^{c_i}$ as $\boldsymbol{v}^{c_i}_{(1)}$, which is the representation in the first layer. $N_i$ represents the set of neighbors of $c_i$. The function $g$ calculates the similar component between $\boldsymbol{v}^{c_i}$ and $\boldsymbol{v}^{c_j}$, and $\boldsymbol{v}^{c_i}_{(l+1)}$ represents the dissimilar component between $\boldsymbol{v}^{c_i}_{(l)}$ and its neighbors. After applying GDO for $L$ layers, we obtain the charge node representation of the last layer, denoted as $\boldsymbol{v}^{c_i}_{(L)}$, which captures the special features of $c_i$. Similarly, we obtain an article node representation $\boldsymbol{v}^{a_i}_{(L)}$.

*3) L2F Attention Layer:* To address confusing verdicts problems, we employ the L2F attention mechanism which focuses on identifying the fact words that have the closest relation to each label. For charge labels, we obtain attended fact vectors $\tilde{\boldsymbol{h}}^{dc}$ based on the original charge features $\boldsymbol{G}^c$ and $\hat{\boldsymbol{h}}^{dc}$ based on the special charge features $\boldsymbol{V}^c_{(L)}$. This helps mitigate losses in the semantics of labels. The attended fact vectors are computed as follows:

$$\tilde{\boldsymbol{h}}^{dc} = \alpha\, \boldsymbol{H}^d, \quad \hat{\boldsymbol{h}}^{dc} = \beta\, \boldsymbol{H}^d. \tag{12}$$

Among them $\alpha \in \mathbb{R}^{l_d}$ and $\beta \in \mathbb{R}^{l_d}$ are attention vectors based on the original features and the special features:

$$\alpha = \text{softmax}(max_{col}(\boldsymbol{H}^d \boldsymbol{W}_\alpha \boldsymbol{G}^{c^\mathsf{T}}), \tag{13}$$

$$\beta = \text{softmax}(max_{col}(\boldsymbol{H}^d \boldsymbol{W}_\beta \boldsymbol{V}^{c^\mathsf{T}}_{(L)})), \tag{14}$$

where $\boldsymbol{W}_\alpha \in \mathbb{R}^{d_s \times d_s}$, $\boldsymbol{W}_\beta \in \mathbb{R}^{d_s \times d_s}$, and $max_{\text{col}}$ is performed across the column. We then concatenate $\tilde{\boldsymbol{h}}^{dc}$ and $\hat{\boldsymbol{h}}^{dc}$ to obtain the charge label-aware fact representation $\boldsymbol{h}^{dc} \in \mathbb{R}^{2d_s}$. Similarly, we can get $\boldsymbol{h}^{da} \in \mathbb{R}^{2d_s}$ for the article.

*4) Prediction in NeurJudge+:* In NeurJudge+, we aim to alleviate confusing verdicts problems by enhancing the task-specific representation of facts for charge and article prediction. To achieve this, we incorporate the mixed semantic vectors obtained from the L2F attention into the prediction of the two subtasks. Specifically, we replace $\boldsymbol{h}^d$ with $[\boldsymbol{h}^{dc}\,;\boldsymbol{h}^d]$ in (3), and replace $\boldsymbol{f}^+$ with $[\boldsymbol{h}^{da}\,;\boldsymbol{f}^+]$ in (8), where " ; " represents the concatenate operation. After that, we can train NeurJudge+ by minimizing the same objective function in (10).

## V. Explainable Legal Judgment Prediction Framework: E-NeurJudge

Although NeurJudge and its variant can effectively deal with the problem of LJP, they still fail to provide the explainability for their predictions, which is the main bottleneck for applications of LJP models in actual reality. To achieve the explainability of LJP, intuitively, we can adopt the selective rationalization mechanism, consisting of a *selector* and a *predictor*. Among them, the *selector* first extracts a text span from the factual description. Then, the *predictor* yields the judgment results based only on the selected span. Finally, this span is defined as the rationale to support the predicted results. However, the rationalization method fails to consider the particularities of the LJP task, such as the judgment process modeling and confusing verdicts distinction. Inspired by the knowledge transfer [43], we propose a cooperative teacher-student framework combining the superiority of NeurJudge for LJP tasks with the explainability of rationalization methods. Particularly, we take NeurJudge which exploits the LJP particularities but lacks explainability as the teacher model, and employ the explainable rationalization method as the student model. Then, the student learns from the teacher to match the circumstance representations. Finally, the extracted rationales are considered as crime circumstances to explain yielded judgment results. Fig. 5 shows the architecture of E-NeurJudge.

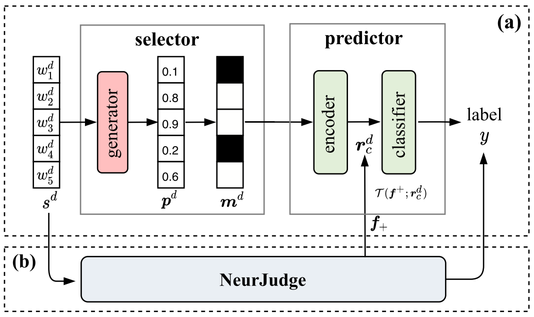


Fig. 5. E-NeurJudge framework for the charge prediction. Among them, (a) represents the framework of the rationalization. (b) is the NeurJudge. E-NeurJudge adopts NeurJudge as the teacher model and transfers the circumstance representations to the student (a rationalization method) to achieve similar results to the teacher.

### *A. Student Model: Rationalization*

Given the factual description $s^d = \{w_1^d, \dots, w_{l_d}^d\}$, where $w_i^d$ represents the $i$-th word. The goal of rationalization is applying a *selector* to extract a text span from the fact as the rationale, and then employing a *predictor* to yield the judgment results based on the rationale. It is noted that gold rationales are unavailable during training. Fig. 5(a) shows the overview of the rationalization method.

*1) Selector:* To select the rationale from the fact, the *selector* first generates a probability distribution $\boldsymbol{p}_i^d$ representing the probability of sampling $w_i^d$ as the part of the rationale. Specifically, similar to (1), we embed the fact into continuous hidden states $\boldsymbol{H}^d$ with Bi-GRU or BERT, and then the probability distribution $\boldsymbol{P}^d = \{\boldsymbol{p}_1^d, \boldsymbol{p}_2^d \dots \boldsymbol{p}_{l_d}^d\} \in \mathbb{R}^{l_d \times 2}$ can be calculated as: $\boldsymbol{P}^d = \text{softmax}(\boldsymbol{W}_s \boldsymbol{H}^d)$, where $\boldsymbol{W}_s \in \mathbb{R}^{d_s \times 2}$. Next, the *selector* samples a binary mask variable $m^d = \{m_1^d, m_2^d, \dots, m_{l_d}^d\}$ from the probability distribution $\boldsymbol{P}^d$, where $m_i^d \in \{0, 1\}$. Besides, to ensure the sampling process is differentiable, we employ the Gumbel-softmax reparameterization method [44]:

$$u_i \sim U(0,1), \quad g_i = -\log\left(-\log\left(u_i\right)\right),$$
$$m_i^d = \frac{\exp\left(\left(\log\left(\boldsymbol{p}_i^d\right) + g_j\right)/\tau\right)}{\sum_t \exp\left(\left(\log\left(\boldsymbol{p}_t^d\right) + g_t\right)/\tau\right)}, \tag{15}$$

where $U(0,1)$ represents the uniform distribution and $\tau$ is a temperature hyperparameter. Finally, the rationale can be extracted as $m^d \odot s^d = \{m_1^d \cdot s_1^d, m_2^d \cdot s_2^d, \dots, m_n^d \cdot s_n^d\}$.

*2) Predictor:* Based on the extracted rationales through the *selector*, the *predictor* yields the judgment results by re-encoding the rationales as hidden states $\boldsymbol{r}_j^d \in \mathbb{R}^{d_s}$, where $j \in \{c, a, t\}$ represents the charge, article, and term of penalty prediction task. The objective of the *predictor* is defined as:

$$\mathcal{L}_j = -\sum_{k=1}^{|Y_j|} \boldsymbol{y}_{j,k} \log\left(\hat{\boldsymbol{y}}_{j,k}\right), \hat{\boldsymbol{y}}_j = \text{softmax}\left(\boldsymbol{W}_j \boldsymbol{r}_j^d + \boldsymbol{b}_j\right). \tag{16}$$

Finally, the rationales can be considered as the explanation for judgment results.

*3) Prediction in Rationalization:* Besides the task loss $\mathcal{L}_j$, we add the sparsity and continuity constraints [7] to ensure the *selector* extracts a short and coherent text span:

$$\mathcal{L}_j^{re} = \lambda_1 \left|\gamma - \frac{1}{l_d}\sum_{i=1}^{l_d} m_i^d\right| + \lambda_2 \sum_{i=2}^{l_d}\left|m_i^{l_d} - m_{i-1}^{l_d}\right|, \tag{17}$$

where the first term makes the model extract short text span and $\gamma \in [0,1]$ represents the predefined sparsity level, and the second term ensures the coherence of selected words. The whole objective of rationalization is $\mathcal{L}_j^{rnp} = \mathcal{L}_j + \mathcal{L}_j^{re}$.

### *B. E-NeurJudge: Cooperative Teacher-Student Framework*

Combining the legal particularities of NeurJudge with the explainability of the rationalization method, we propose a cooperative teacher-student framework (denoted by E-NeurJudge). E-NeurJudge adopts NeurJudge as the teacher model and transfers the crime circumstance representations generated by NeurJudge to the rationalization model (student), ensuring the student can learn the legal particularities. Specifically, for the charge prediction, after the fact separation for verdicts, NeurJudge generates the ADC representations $\boldsymbol{f}^+$. Then, we encourage the *predictor* in the student model to match $\boldsymbol{f}^+$ for learning the legal particularities by minimizing the value of a transferring function $\mathcal{T}(\boldsymbol{f}^+; \boldsymbol{r}_c^d)$, where $\mathcal{T}(\cdot)$ ensures the two different representations encode the same information, and $\boldsymbol{r}_c^d$ is the rationale representation generated by the *predictor* (See Section V-A2 for detail). Similarly, considering the article prediction, we employ $\boldsymbol{r}_a^d$ to match $\boldsymbol{f}^+$ with $\mathcal{T}(\boldsymbol{f}^+; \boldsymbol{r}_a^d)$. For the term of penalty prediction, since NeurJudge adopts SSC and DSC to jointly predict the sentence, to enable the student model to match the two circumstances, we first let the student model extract two different rationales and represent them as $\boldsymbol{r}_{t_1}^d$ and $\boldsymbol{r}_{t_2}^d$, and adopt them to predict the term of penalty (i.e., $\hat{\boldsymbol{y}}_t = \text{softmax}(\boldsymbol{W}_t[\boldsymbol{r}_{t_1}^d; \boldsymbol{r}_{t_2}^d] + \boldsymbol{b}_t)$). Then, we employ $\mathcal{T}(\boldsymbol{f}_+^-; \boldsymbol{r}_{t_1}^d)$ and $\mathcal{T}(\boldsymbol{f}_-^-; \boldsymbol{r}_{t_2}^d)$ to achieve the transfer.

*1) Transferring Functions:* To ensure the effectiveness of transferring, in this section, we propose four different transferring functions. Besides the common MSE loss, we also employ the KL-divergence and JS-divergence. Furthermore, inspired by the information theory, we employ a mutual information (MI) maximization method as the transferring function $\mathcal{T}(\cdot)$, where MI is a measure of the mutual dependence between the two variables. Since directly computing MI value is difficult [45], we adopt the InfoNCE [46] to approximate it:

$$I_{nce} = \frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{f(x_i,y_i)}}{\frac{1}{N}\sum_{j=1}^{N} e^{f(x_i,y_j)}}, \tag{18}$$

where $\{(x_i, y_i)\}_{i=1}^N$ is a batch of sample pairs and $f(x,y)$ is a learnable score function. For example, to achieve minimizing $\mathcal{T}(\boldsymbol{f}^+; \boldsymbol{r}_c^d)$, we can maximize $I_{nce}$ (i.e, minimize $-I_{nce}$), where we set $x_i$ as $\boldsymbol{f}^+$ and $y_i$ as $\boldsymbol{r}_c^d$. Finally, we define the above transferring functions adopted in E-NeurJudge as E-NeurJudge(MSE), E-NeurJudge(KL), E-NeurJudge(JS), and E-NeurJudge(NCE).

TABLE III
THE STATISTICS OF *CAIL* DATASETS

| Dataset | CAIL-small | CAIL-big |
|---|---|---|
| #Training Set Cases | 108,619 | 1,593,982 |
| #Test Set Cases | 26,120 | 185,721 |
| #Law Articles | 99 | 118 |
| #Charges | 115 | 129 |
| #Term of Penalty | 11 | 11 |

*2) Prediction in E-NeurJudge:* Combining the loss functions of NeurJudge $\mathcal{L}_{ner}$ and rationalization $\mathcal{L}_j^{rnp}$ with the transferring function, in this section, we define the overall objective of E-NeurJudge as below, and E-NeurJudge+ is similar:

$$\mathcal{L} = \underbrace{\mathcal{L}_{ner}}_{LJP} + \underbrace{\sum_{j\in\{c,a,t\}} \mathcal{L}_j^{rnp}}_{RNP} + \underbrace{(\mathcal{T}(\boldsymbol{f}^+;\boldsymbol{r}_c^d) + \mathcal{T}(\boldsymbol{f}^+;\boldsymbol{r}_a^d) + \mathcal{T}(\boldsymbol{f}_+^-;\boldsymbol{r}_{t_1}^d) + \mathcal{T}(\boldsymbol{f}_-^-;\boldsymbol{r}_{t_2}^d))}_{Transfer}. \tag{19}$$

## VI. EXPERIMENTS

### A. Dataset

*1) Dataset Description:* In our experiments, we utilize the publicly available **C**hinese **AI** and **L**aw challenge (*CAIL*) dataset, which consists of two datasets: *CAIL-small* and *CAIL-big* [47]. The *CAIL* dataset comprises criminal cases published by the Supreme People's Court, which include factual descriptions and corresponding judgment results such as charges, law articles, and terms of penalty. To preprocess the data, we follow the approach of [2]. First, we filter out infrequent charges and law articles, retaining those with frequencies greater than 100. Additionally, we divide the terms into non-overlapping intervals. In real-world scenarios, some cases may involve multiple charges and law articles, which adds complexity to judgment prediction. To maintain consistency with state-of-the-art methods and demonstrate the effectiveness of considering the circumstances of the crime, we filter out these multi-label samples. Detailed statistics of the *CAIL* dataset are presented in Table III.

Besides, to evaluate the effectiveness of E-NeurJudge, we need to compare the extracted rationales with real rationales annotated by humans. As there exist no human annotations of rationales in *CAIL*, we first randomly sample 100 cases from *CAIL-small* and label which words refer to the ADC, SSC, and DSC, respectively. Then, in order to extend the data sources, we randomly select an additional 100 samples from the judgment website.[1] During the sample collection process, we ensure that the charges and law articles involved in these samples fell within the range covered by the *CAIL-small* and *CAIL-big* datasets. Subsequently, we annotate the corresponding circumstances for these samples. We denote this dataset as *Legal-rationale*. Table IV shows the average length of the different crime circumstances and their percentage of the case facts. Further data analysis is presented in Section VI-A2.

[1]https://wenshu.court.gov.cn/

TABLE IV
THE STATISTICS OF *LEGAL-RATIONALE*

| | fact | ADC | SSC | DSC |
|---|---|---|---|---|
| Avg.length | 236.93 | 46.08 | 39.19 | 34.05 |
| % selected | – | 19.45 | 15.54 | 14.37 |

*2) Data Analysis:* We deeply examine the *CAIL* and *Legal-rationale* datasets to uncover several supportive observations of our model. Fig. 6 provides an overview of the relative positions of various circumstances of crime in the case facts of the *Legal-rationale* dataset (Fig. 6(a)). The horizontal axis represents the progression of the cases, with 0% indicating the beginning of the cases and 100% indicating the ending. The vertical axis is the number of cases. The yellow, green, and pink areas represent the sentences in the factual descriptions related to the ADC, SSC, and DSC. The gray areas represent elements that have less direct relevance to the facts, such as "time" and "place". By analyzing this figure, we observe that the case facts predominantly consist of circumstances of the crime and the positions of these circumstances in the case facts are not fixed and can vary throughout the text. Besides, we calculate the cosine similarity between different labels based on the word embeddings of the labels. In the left of Fig. 6(b) and (c), we define that two labels are similar when their similarity is greater than 0.6. Among them, the dark squares indicate that two labels are similar and the white squares indicate that they are not similar. From the figure, we find many labels are similar. These observations prove that similar labels can not be easily distinguished, and it is necessary to obtain special label features.

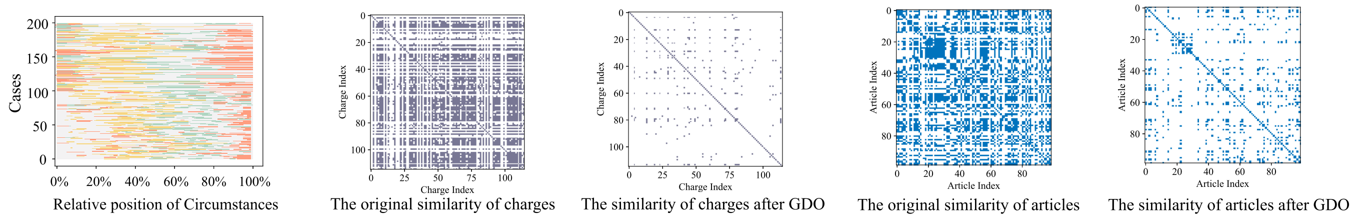
### B. Comparison Methods

We compare NeurJudge and NeurJudge+ with several representative baselines. First, we compare GRU based NeurJudge to some baselines, where GRU has gained widespread utilization in predicting LJP [1], [5]:

- **Word2Vec+SVM** uses Word2vec [36] to represent word features and employs SVM [48] for text classification.
- **FLA** [1] is an attention-based network to model the interaction between factual descriptions and articles.
- **TOPJUDGE** [2] is a multi-task learning model that captures the dependencies among subtasks in LJP.
- **Few-Shot** [18] is an attribute-attentive model that uses charge attributes to enhance the fact representation and alleviate confusing charge issues.
- **LADAN** [5] is an attention-based model that distinguishes confusing verdicts by employing a graph distillation operator.
- **CPTP** [49] is a charge-based term of penalty prediction model that utilizes deep gating networks to filter and aggregate charge-specified information gradually.

Then, considering pre-trained models' pivotal and noteworthy accomplishments within the realm of natural language processing, we choose BERT and its variants which are compared with BERT based NeurJudge as follows:

- **BERT** [50] is a language representation model based on deep bidirectional transformers. It has outperformed



**(a) Relative position of different circumstances in case facts.**

**(b) The original and distinguished similarity of charges.**

**(c) The original and distinguished similarity of articles.**

Fig. 6. (a). The relative position of different circumstances in case facts, where the vertical axis represents all cases, and the horizontal axis represents the relative position of different crime circumstances within each case's factual description. (b). The similarity between different charges. The left denotes the original charge similarity, and the right represents the similarity of charge after GDO, where the dark indicates that two labels are similar and the white indicates that they are not similar. (c). The similarity between different law articles.

state-of-the-art models on various NLP tasks. Since our experiments are conducted on Chinese datasets, we use the Chinese BERT trained by [37] as our baseline.

- **BERT-Crime** [51] is a variant of BERT that is pre-trained with crime data.

Finally, to evaluate the performance of each component in our model, we design some simplified variants:

- **NeurJudge-Mtl** is a typical multi-task learning model that simultaneously predicts all subtasks.
- **NeurJudge-ADC** replaces $[\boldsymbol{f}+^{-}; \boldsymbol{f}-^{-}]$ with $f^{+}$ in Eq.(9) to predict the term of penalty, ignoring the influence of sentencing circumstances.
- **NeurJudge-GCN** replaces GDO with GCN [41] in NeurJudge+ to demonstrate the effectiveness of GDO.
- **NeurJudge-Att** expresses NeurJudge with L2F attention, removing the GDO component of NeurJudge+.
- **NeurJudge+(Tree)** replaces the similar charge and law article graphs in NeurJudge+ as the tree structure.

For E-NeurJudge, to evaluate the selected circumstances of crime (i.e., rationales), we compare with several classical baselines in rationalization.

- **RNP** [7] first yields a Bernoulli distribution of each token and then samples the token from the distribution as rationales for predicting results.
- **HardKuma** [31] generates a HardKuma distribution based on Kumaraswamy distribution [52] for reparameterization and makes the process of selecting the rationale tokens differentiable.
- **InfoCal(HK)** [8] proposes an adversarial-based method to guide the rationalization model to extract rationales with a HardKuma distribution for reparameterization.
- **InfoCal_IB** [8] manages the trade-off between selecting short rationales and yielding accurate results by adopting an Information Bottleneck regularizer, which removes the adversarial module of InfoCal(HK).

## C. Experimental Setup

For the methodologies involving CNN or RNN and rationalization methods, we initially utilize THULAC [53] for word segmentation. Then, we employ word2vec [36] to pre-train word embeddings, utilizing an embedding size of 200. Additionally, we set the maximum document length to 350 and the hidden size to 150. Moreover, we set the value of $\gamma$ in (17) as $\{0.2, 0.2, 0.2\}$ for all three crime circumstances. Regarding the BERT-based methods, we utilize a pre-trained Chinese model provided by Chinese BERT-wwm [37]. Here, we set the maximum document length to 500 tokens. During the training process, we initialize the learning rate of the Adam optimizer [54] to $10^{-3}$. The model training is performed on a server equipped with 2 V100 GPUs, and each model is trained for 16 epochs with a batch size of 128. Finally, we evaluate the performance of predicting judgment results using accuracy (Acc), macro-precision (MP), macro-recall (MR), and macro-F1 (F1) as evaluation metrics. Furthermore, to evaluate selected rationales, we take token precision (token-P), recall (token-R), and F1 (token-F1) as the evaluation metrics, where token-P represents the number of selected tokens in the annotated rationale and token-R means how many annotated rationale tokens were selected.

## D. Experimental Results on NeurJudge

*1) Comparison With LJP Baselines on Task Prediction:* In this section, we commence by comparing our GRU encoder-based methods with several baselines, and the corresponding results are outlined in Table V. Notably, our proposed NeurJudge and NeurJudge+ consistently exhibit superior performance across both datasets. NeurJudge demonstrates a relative improvement of 4.59% and 1.90% in F1-score over the best baseline, LADAN, on average across the three subtasks in the CAIL-small dataset. Additionally, NeurJudge+ showcases an improvement of 5.69% and 4.95% in the respective metrics. Further analysis of the results reveals the following observations: (1) SVM+word2vec fails to match the performance of deep learning-based models, potentially due to its inability to capture intricate interactions between facts and labels. (2) FLA exhibits poor performance due to its two-stage nature, which can lead to error propagation. (3) TOPJUDGE, which models the judgment process using the topological order of subtasks, is outperformed by our NeurJudge, indicating that NeurJudge more effectively leverages the fine-grained judgment process by considering distinct circumstances of crime for each subtask. (4) In terms of penalty prediction, NeurJudge

TABLE V
JUDGMENT PREDICTION RESULTS ON *CAIL* (GRU BASED NEURJUDGE)

| CAIL-small | Charges | | | | Law Articles | | | | Terms of Penalty | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods | Acc | macro-P | macro-R | macro-F1 | Acc | macro-P | macro-R | macro-F1 | Acc | macro-P | macro-R | macro-F1 |
| SVM+word2vec | 83.37 | 80.78 | 77.30 | 78.25 | 84.17 | 80.74 | 75.96 | 77.09 | 33.00 | 25.56 | 25.11 | 22.50 |
| FLA | 84.72 | 83.71 | 73.75 | 75.04 | 85.63 | 83.46 | 73.83 | 74.92 | 35.04 | 33.91 | 27.14 | 24.79 |
| TOPJUDGE | 86.48 | 84.23 | 78.39 | 80.15 | 87.28 | 85.81 | 76.25 | 78.24 | 38.43 | 35.67 | 32.15 | 31.31 |
| Few-Shot | 88.15 | 87.51 | 80.57 | 81.98 | 88.44 | 86.76 | 77.93 | 79.51 | 39.62 | 37.13 | 30.93 | 31.61 |
| LADAN | 88.28 | 86.36 | 80.54 | 82.11 | 88.78 | 85.15 | 79.45 | 80.97 | 38.13 | 34.04 | 31.22 | 30.20 |
| CPTP | – | – | – | – | – | – | – | – | 39.16 | 37.06 | 33.82 | 33.79 |
| NeurJudge | 88.89 | 86.96 | 85.42 | 85.73 | 89.71 | 86.68 | 83.92 | 84.97 | 41.03 | 39.52 | 36.82 | 36.35 |
| NeurJudge+ | **89.92** | **87.76** | **86.75** | **86.96** | **90.37** | **87.22** | **85.82** | **86.13** | **41.65** | **40.44** | **37.20** | **37.27** |
| RNP | 85.76 | 82.71 | 78.04 | 79.38 | 86.45 | 81.94 | 75.92 | 77.35 | 38.19 | 33.05 | 32.10 | 30.32 |
| HardKuma | 86.26 | 86.22 | 77.83 | 80.04 | 85.76 | 80.92 | 76.85 | 77.96 | 35.94 | 35.26 | 27.52 | 26.09 |
| InfoCal_IB | 87.14 | 84.86 | 80.70 | 82.11 | 87.58 | 83.55 | 78.49 | 80.13 | 36.60 | 33.12 | 28.20 | 26.82 |
| InfoCal(HK) | 87.85 | 86.79 | 82.53 | 83.87 | 87.65 | 84.88 | 80.09 | 81.67 | 38.69 | 35.68 | 31.61 | 31.21 |
| E-TOPJUDGE | 86.91 | 85.12 | 79.31 | 80.98 | 88.01 | 86.12 | 77.12 | 79.20 | 37.99 | 35.12 | 32.01 | 30.78 |
| E-Few-Shot | 87.92 | 86.98 | 81.03 | 81.48 | 88.10 | 85.11 | 77.09 | 78.82 | 39.87 | 37.45 | 31.03 | 31.87 |
| E-LADAN | 87.87 | 85.82 | 80.01 | 81.32 | 88.21 | 84.59 | 79.08 | 79.49 | 38.88 | 34.76 | 31.98 | 30.35 |
| E-NeurJudge (MSE) | 88.13 | 85.95 | 83.79 | 84.35 | 88.59 | 85.18 | 82.37 | 83.17 | 38.52 | 36.44 | 32.27 | 31.97 |
| E-NeurJudge (KL) | 88.30 | 85.89 | 84.66 | 84.91 | 88.73 | 85.52 | 82.83 | 83.63 | 38.76 | 37.06 | 33.40 | 33.37 |
| E-NeurJudge (JS) | 88.41 | 86.64 | 84.83 | **85.43** | 88.74 | 84.77 | 82.99 | 83.59 | 38.72 | 36.87 | 33.47 | 34.22 |
| E-NeurJudge (NCE) | 88.55 | **87.05** | 84.44 | 85.17 | 88.98 | **86.33** | 82.90 | 83.93 | 39.32 | 38.06 | 34.02 | 34.26 |
| E-NeurJudge+ (MSE) | 88.28 | 85.73 | 84.52 | 84.79 | 89.31 | 86.20 | 83.63 | 84.53 | 39.02 | 37.02 | 33.27 | 33.80 |
| E-NeurJudge+ (KL) | 88.29 | 85.64 | 83.87 | 84.26 | 89.28 | 85.69 | 84.03 | 84.56 | 38.64 | 38.97 | 34.97 | 36.52 |
| E-NeurJudge+ (JS) | 88.59 | 85.75 | **85.33** | 85.26 | 89.34 | 85.44 | **84.35** | **84.66** | 38.62 | 37.87 | 35.09 | 36.07 |
| E-NeurJudge+ (NCE) | **88.80** | 86.57 | 84.57 | 85.15 | **89.65** | 85.71 | 84.14 | 84.47 | **40.84** | **39.81** | **35.69** | **36.72** |

| CAIL-big | Charges | | | | Law Articles | | | | Terms of Penalty | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods | Acc | macro-P | macro-R | macro-F1 | Acc | macro-P | macro-R | macro-F1 | Acc | macro-P | macro-R | macro-F1 |
| SVM+word2vec | 92.09 | 82.26 | 65.28 | 69.06 | 92.62 | 77.92 | 61.03 | 64.29 | 46.73 | 28.98 | 20.92 | 20.91 |
| FLA | 93.01 | 76.56 | 72.75 | 72.94 | 93.51 | 74.94 | 70.40 | 70.70 | 54.29 | 38.39 | 29.34 | 30.85 |
| TOPJUDGE | 93.19 | 79.44 | 75.52 | 75.50 | 93.24 | 74.24 | 71.19 | 70.40 | 53.52 | 44.58 | 30.41 | 30.61 |
| Few-Shot | 93.24 | 80.59 | 76.62 | 76.89 | 93.74 | 78.51 | 73.79 | 74.18 | 54.54 | 39.09 | 33.36 | 33.48 |
| LADAN | 93.26 | 81.21 | 77.65 | 77.60 | 93.27 | 75.10 | 72.04 | 71.26 | 53.62 | 41.52 | 37.53 | 36.06 |
| CPTP | – | – | – | – | – | – | – | – | 55.51 | 47.20 | 33.36 | 36.02 |
| NeurJudge | 95.33 | 84.03 | 77.54 | 78.31 | 95.46 | 81.30 | 75.37 | 76.20 | 55.29 | 44.12 | 35.30 | 36.11 |
| NeurJudge+ | **95.57** | **85.57** | **78.81** | **80.54** | **95.58** | **82.01** | **77.05** | **78.05** | **57.07** | **47.65** | **40.01** | **41.18** |
| RNP | 94.78 | 80.18 | 66.12 | 68.46 | 95.47 | 81.16 | 69.50 | 72.20 | 54.53 | 44.84 | 35.65 | 36.72 |
| HardKuma | 95.06 | 85.44 | 74.11 | 76.91 | 95.22 | 81.45 | 70.56 | 73.66 | 54.54 | 45.17 | 35.25 | 36.81 |
| InfoCal_IB | 94.99 | 85.15 | 73.96 | 76.98 | 95.18 | 81.83 | 70.71 | 73.82 | 54.64 | 45.25 | 35.34 | 36.89 |
| InfoCal(HK) | 95.55 | 88.38 | 77.99 | 81.02 | 95.80 | 84.39 | 75.85 | 78.33 | 55.56 | 46.63 | 38.02 | 39.25 |
| E-TOPJUDGE | 94.82 | 83.23 | 76.32 | 76.87 | 95.19 | 78.35 | 71.98 | 72.08 | 53.98 | 45.03 | 30.84 | 31.23 |
| E-Few-Shot | 95.48 | 84.20 | 77.13 | 77.83 | 95.51 | 81.24 | 74.87 | 75.52 | 54.27 | 40.01 | 35.90 | 35.09 |
| E-LADAN | 95.12 | 84.13 | 78.92 | 78.20 | 95.29 | 82.20 | 74.32 | 75.78 | 54.98 | 44.93 | 38.48 | 38.89 |
| E-NeurJudge (MSE) | 95.63 | 88.11 | 78.97 | 81.88 | 95.73 | 83.87 | 75.16 | 77.97 | 55.70 | 46.35 | 37.77 | 39.22 |
| E-NeurJudge (KL) | 95.57 | 87.80 | 78.44 | 81.35 | 95.63 | 84.04 | 74.40 | 77.42 | 55.71 | 46.46 | 37.92 | 39.31 |
| E-NeurJudge (JS) | 95.60 | 88.02 | 78.87 | 81.87 | 95.69 | 84.73 | 75.07 | 78.12 | 55.63 | 46.93 | 37.67 | 39.28 |
| E-NeurJudge (NCE) | 95.49 | 87.76 | 77.94 | 80.82 | 95.83 | 84.98 | 75.45 | 78.43 | 55.54 | 46.15 | 37.10 | 38.67 |
| E-NeurJudge+ (MSE) | 96.02 | 89.17 | 81.94 | 84.40 | 96.14 | 85.90 | 78.53 | **81.02** | 56.83 | 48.40 | 39.34 | 41.24 |
| E-NeurJudge+ (KL) | 96.03 | 88.82 | 81.52 | 83.92 | 96.13 | **86.49** | 78.37 | 80.82 | 56.82 | 48.16 | 40.02 | 41.41 |
| E-NeurJudge+ (JS) | 95.96 | 88.84 | 81.06 | 83.66 | 96.09 | 84.95 | 78.29 | 80.46 | 56.79 | 48.01 | 40.11 | 41.47 |
| E-NeurJudge+ (NCE) | **96.10** | **89.29** | **82.02** | **84.60** | **96.16** | 85.19 | **78.56** | 80.91 | **56.87** | **48.41** | **40.14** | **41.68** |

The bolded values in the first row indicate that NeurJudge+ performs optimally among the non-explainable legal judgment prediction methods.
The bolded values in the second row indicate values for which E-NeurJudge performs optimally among variants of E-NeurJudge.

surpasses the state-of-the-art method (CPTP) by 2.56% in F1-score, while NeurJudge+ achieves a 3.48% improvement in the CAIL-small dataset, showcasing the effectiveness of our method in further simulating human judges. (5) NeurJudge+ outperforms Few-Shot, LADAN, and NeurJudge, which we attribute to the enhanced discriminative fact feature extraction capability of our proposed extension (GLE). (6) In CAIL-big, all methods exhibit lower F1-scores compared to the CAIL-small dataset, while accuracy is higher. This discrepancy arises from the highly imbalanced training data in the CAIL-big dataset.

Next, we compare NeurJudge and NeurJudge+ based on the BERT encoder with BERT and its variants. The results are shown in Table VI. From the above observations, we find that BERT performs well on LJP tasks while behaving worse than our models, which further validates the effectiveness of our proposed models.

*2) Comparison With Variants of NeurJudge:* In this section, we conduct an analysis to evaluate the impact of the CCFS, GLE, and the graph construction operation components. We compare NeurJudge and NeurJudge+ with their respective variants on the CAIL-small dataset. To affirm the effectiveness of CCFS, we make some modifications to NeurJudge and compare it with NeurJudge-Mtl and NeurJudge-ADC. The performances of these variants are presented in Fig. 7. NeurJudge-Mtl removes the Fact Separation component, where the fact representation $\boldsymbol{h}^d$ is used to predict all three subtasks. Notably, NeurJudge-Mtl exhibits inferior performance compared to the other methods, which demonstrates the necessity of employing different circumstances for corresponding subtasks. Furthermore, we compare NeurJudge with NeurJudge-ADC, where $[\boldsymbol{f}_{-}^{+}; \boldsymbol{f}_{-}^{-}]$ in (9) is replaced with $\boldsymbol{f}^{+}$, allowing only ADC to predict the terms of penalty. It is evident that NeurJudge-ADC performs poorly, particularly in terms of penalty prediction. This further reinforces the significance of separating circumstances of crime for accurate prediction.

Besides, to validate the effectiveness of our proposed GLE, we introduce two variants, NeurJudge-Att and NeurJudge-GCN, and compare them with NeurJudge+ on the CAIL-small dataset. First, we design the NeurJudge-Att which removes the GDO and

TABLE VI
JUDGMENT PREDICTION RESULTS ON *CAIL* (BERT BASED NEURJUDGE)

| CAIL-small | Charges | | Law Articles | | Terms of Penalty | | ADC | SEC | SSC | DSC |
|---|---|---|---|---|---|---|---|---|---|---|
| Methods | Acc | macro-F1 | Acc | macro-F1 | Acc | macro-F1 | token-F1 | token-F1 | token-F1 | token-F1 |
| BERT | 90.68 | 87.69 | 90.81 | 86.06 | 40.37 | 34.09 | – | – | – | – |
| BERT-Crime | 91.26 | 87.81 | 91.30 | 85.70 | 40.90 | 34.65 | – | – | – | – |
| NeurJudge | 92.74 | 90.60 | 92.60 | 88.33 | 42.69 | 37.90 | – | – | – | – |
| NeurJudge+ | **92.91** | **90.89** | **92.64** | **88.75** | **43.81** | **39.76** | – | – | | – |
| BERT-RNP | 89.04 | 85.83 | 89.32 | 86.47 | 39.96 | 35.07 | 28.52 | 35.04 | – | – |
| E-NeurJudge | 91.80 | 90.77 | 90.64 | 88.10 | 42.38 | 38.28 | 33.18 | 44.37 | 32.35 | 11.76 |
| E-NeurJudge+ | **92.99** | **91.05** | **90.90** | **88.37** | **42.50** | **39.09** | **35.13** | **45.54** | **32.20** | **12.48** |
| **CAIL-big** | **Charges** | | **Law Articles** | | **Terms of Penalty** | | **ADC** | **SEC** | **SSC** | **DSC** |
| Methods | Acc | macro-F1 | Acc | macro-F1 | Acc | macro-F1 | token-F1 | token-F1 | token-F1 | token-F1 |
| BERT | 95.12 | 90.31 | 92.89 | 88.89 | 53.67 | 38.88 | – | – | – | – |
| BERT-Crime | 95.56 | 90.23 | 92.65 | 88.65 | 53.87 | 38.82 | – | – | – | – |
| NeurJudge | 96.02 | 91.49 | 93.34 | 89.43 | 54.89 | 41.08 | – | – | – | – |
| NeurJudge+ | 96.84 | 92.93 | 95.56 | 89.32 | 56.89 | 43.83 | – | – | – | – |
| BERT-RNP | 94.86 | 90.43 | 92.42 | 88.62 | 52.87 | 38.32 | 32.86 | 37.78 | – | – |
| E-NeurJudge | 96.39 | 91.73 | 94.84 | 88.94 | 55.68 | 42.93 | 34.52 | 45.39 | 33.68 | 12.83 |
| E-NeurJudge+ | **96.73** | **92.39** | **95.45** | **89.56** | **57.78** | **44.46** | **35.72** | **46.72** | **34.62** | **13.52** |

The bolded values in the first row indicate that NeurJudge+ performs optimally among the non-explainable legal judgment prediction methods.
The bolded values in the second row indicate values for which E-NeurJudge performs optimally among variants of E-NeurJudge.

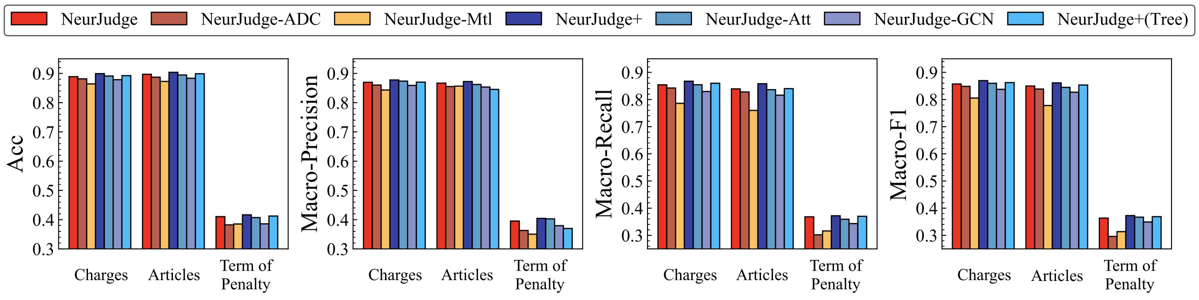


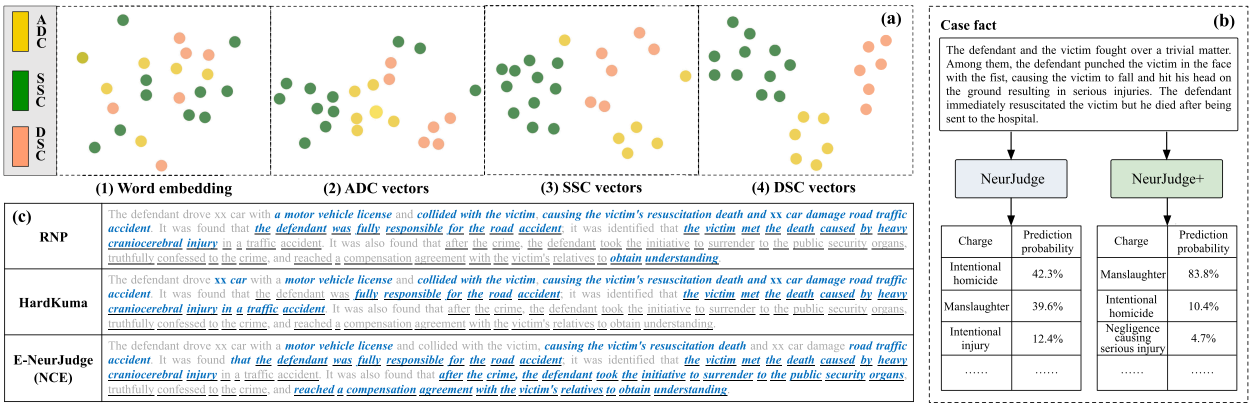
Fig. 7. Results of NeurJudge and its variants of *CAIL-small* on four metrics (GRU based NeurJudge).

instead utilizes the L2F attention operation to re-encode the fact. As shown in Fig. 7, we observe that NeurJudge-Att performs worse than NeurJudge+, which affirms the significance of the GDO component. This demonstrates that the GDO effectively captures the special label features on the label similarity graphs. Next, we introduce the NeurJudge-GCN which substitutes the GCN operation for the GDO. Notably, NeurJudge-GCN performs the worst among the three methods. This further demonstrates that the GDO is capable of effectively extracting the special label features and alleviating the over-smoothing issue, which is a common problem in GCN. These results highlight the effectiveness of our proposed GLE component in NeurJudge+. Besides, we also show the similarity among labels after GDO in the right of Fig. 6(b) and (c). From the observation, we find that similar labels are distinguished from each other after the label encoding, which also shows the effectiveness of our GDO method.

Finally, to illustrate the rationality and effectiveness of our proposed construction method (Section IV-B1) compared to the tree structure, we employ NeurJudge+(Tree) to compare with NeurJudge+ on *CAIL-small*, which modifies the Graph Construct Layer in NeurJudge+ to a tree structure. The results in Fig. 7 indicate that NeurJudge+(Tree) outperforms NeurJudge, demonstrating the necessity of modeling the relationships among label semantics. However, we find that NeurJudge+(Tree) performs worse than NeurJudge+. This observation highlights the superiority of our graph modeling approach over the tree structure.

*3) Visualizations of NeurJudge and Networks+:* In this section, we first provide a qualitative analysis of the circumstance representations generated by NeurJudge. To intuitively observe the differences among ADC, SSC and DSC vectors, we visualize these feature spaces using t-SNE [55] in Fig. 8(a). We visualize the case illustrated in Fig. 1. In the t-SNE plots, the yellow points represent the circumstance-related words that are highlighted in yellow in Fig. 1, while the green and pink points represent other words. In Fig. 8(a.1), we visualize the word embedding space of the circumstances. We can observe that the words in this space are highly fragmented. However, after the fact separation for verdicts, as shown in Fig. 8(a.2), we can see that the words related to the ADC are gathered together, indicating the effective separation of the ADC vectors. Furthermore, we separate the sentencing circumstances even further. Fig. 8(a.3) and (a.4) show the t-SNE visualizations of the SSC and DSC vectors, respectively. We can observe that these vectors are separated and gathered distinctly. The above visualizations indicate that NeurJudge can separate different circumstances of facts.

Besides, we also add a charge prediction example of NeurJudge and NeurJudge+ to investigate the effectiveness of NeurJudge+. Specifically, in Fig. 8(b), both NeurJudge and NeurJudge+ are employed to predict the charge of the given case fact. Among them, the ground truth charge is *Manslaughter*, which is similar to *Intentional homicide*. From the prediction probability, we find that NeurJudge predicts this case as *Intentional homicide* with a probability of 42.3% while predicting it as *Manslaughter*



The defendant and the victim fought over a trivial matter. Among them, the defendant punched the victim in the face with the fist, causing the victim to fall and hit his head on the ground resulting in serious injuries. The defendant immediately resuscitated the victim but he died after being sent to the hospital.

NeurJudge:

| Charge | Prediction probability |
|---|---|
| Intentional homicide | 42.3% |
| Manslaughter | 39.6% |
| Intentional injury | 12.4% |
| …… | …… |

NeurJudge+:

| Charge | Prediction probability |
|---|---|
| Manslaughter | 83.8% |
| Intentional homicide | 10.4% |
| Negligence causing serious injury | 4.7% |
| …… | …… |

| (c) | |
|---|---|
| RNP | The defendant drove xx car with ***a motor vehicle license*** and ***collided with the victim***, ***causing the victim's resuscitation death and xx car damage road traffic accident***. It was found that ***the defendant was fully responsible for the road accident***; it was identified that ***the victim met the death caused by heavy craniocerebral injury*** in a traffic accident. It was also found that after the crime, the defendant took the initiative to surrender to the public security organs, truthfully confessed to the crime, and reached a compensation agreement with the victim's relatives to ***obtain understanding***. |
| HardKuma | The defendant drove ***xx car*** with a ***motor vehicle license*** and ***collided with the victim***, ***causing the victim's resuscitation death and xx car damage road traffic accident***. It was found that the defendant was ***fully responsible for the road accident***; it was identified that ***the victim met the death caused by heavy craniocerebral injury in a traffic accident***. It was also found that after the crime, the defendant took the initiative to surrender to the public security organs, truthfully confessed to the crime, and reached a compensation agreement with the victim's relatives to obtain understanding. |
| E-NeurJudge (NCE) | The defendant drove xx car with a ***motor vehicle license*** and collided with the victim, ***causing the victim's resuscitation death*** and xx car damage ***road traffic accident***. It was found ***that the defendant was fully responsible for the road accident***; it was identified that ***the victim met the death caused by heavy craniocerebral injury*** in a traffic accident. It was also found that ***after the crime, the defendant took the initiative to surrender to the public security organs***, truthfully confessed to the crime, and ***reached a compensation agreement with the victim's relatives to obtain understanding***. |

Fig. 8. (a). T-SNE visualizations of fact vectors. (b). A charge prediction example of NeurJudge and NeurJudge+. (c). Visualized selective rationales with different methods. Among them, the underlined tokens represent the real rationales, and the blue is the predicted rationales.

with a probability of 39.6%. This indicates that NeurJudge has difficulty distinguishing between *Manslaughter* and *Intentional homicide*. In contrast, the probability that NeurJudge+ predicts this case to be *Manslaughter* is 83.8%, which is much higher than the predicted probability of *Intentional homicide* (10.4%). This observation shows that NeurJudge+ can effectively distinguish the similar charges.

## *E. Experimental Results on E-NeurJudge*

*1) Comparison With Rationalization Baselines:* To evaluate the performance of our E-NeurJudge, we show results from two aspects. First, we can clearly see that both E-NeurJudge and E-NeurJudge+ with different transferring functions outperform all rationalization methods in yielding judgment results in Table V, demonstrating that E-NeurJudge exploits legal particularities well by learning from NeurJudge. Meanwhile, compared with LJP approaches, E-NeurJudge(+) improves considerably, making a good trade-off between yielding LJP results and providing explainability. Besides, although E-NeurJudge(+) does not perform better than its teacher model NeurJudge(+), the difference is not significant. Considering that E-NeurJudge(+) provides explainability for LJP, we argue that the minor performance degradation is acceptable.

Second, to further evaluate extracted circumstances, we conduct experiments on *Legal-rationale* and show results in Tables VI and VII. Among them, in Table VI, BERT-RNP represents RNP implemented with BERT, and E-NeurJudge is achieved with InfoNCE. Besides, since rationalization methods fail to extract SSC and DSC, and conversely, they are only able to extract SEC, we combine the DSC and SSC extracted by E-NeurJudge to obtain SEC and compare it with rationalization methods. From the table, we can observe that E-NeurJudge still performs the best in extracting circumstances, illustrating the effectiveness of E-NeurJudge.

*2) Transferring Function Study:* To ensure the effectiveness of transferring the legal particularities from NeurJudge to E-NeurJudge, we employ four transferring functions. In this section, we conduct an experiment to analyze the performance on yielding judgment results (Table V) and selecting rationales (Table VII) with the four functions. From the results, we observe

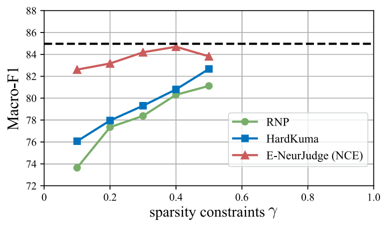


Fig. 9. The F1 score on the law article prediction with an increasing $\gamma$, where the black dashed line represents the F1 score of E-NeurJudge.

that the InfoNCE loss outperforms the other loss functions in the comparative study, especially in terms of penalty prediction and the extraction of SEC, indicating that employing the InfoNCE method is a better way to transfer crime circumstance representations to the rationalization than other methods.

*3) Model Performance With Different Sparsity Constraints:* In Section V-A3, we design a sparsity constraint $\gamma$ to ensure extracting short rationales. In this section, we further present the relation between the model performance on yielding judgment results and sparsity constraint $\gamma$. Specifically, we show the F1 score in the law article prediction (*CAIL-small*) with an increasing $\gamma$ in Fig. 9. From the observation, we can conclude that setting $\gamma = 0.2$ can achieve a balance between yield judgment results and extracting circumstances. Then, we observe that E-NeurJudge achieves a similar result to NeurJudge when $\gamma = 0.4$. Besides, with an increasing $\gamma$, E-NeurJudge consistently performs better than these previous models (RNP and HardKuma), demonstrating the effectiveness of our approach.

*4) Structural Generalizability of E-NeurJudge:* In E-NeurJudge, we utilize NeurJudge as the teacher model to transfer its acquired legal particularities to rationalization, ensuring that the student system learns these particularities while providing explainability for LJP. However, an intriguing question arises: *What would be the impact of replacing the teacher model with other LJP models?* To address this question, we perform experiments where we substitute the teacher model with alternative models such as TOPJUDGE, Few-Shot, and LADAN. These replacements are denoted as E-TOPJUDGE, E-Few-Shot, and E-LADAN, where we employ InfoNCE as the transferring

TABLE VII
TOKEN PRECISION, RECALL AND F1 OF THE SELECTED ADC, SEC, SSC AND DSC

| Training in CAIL-small | ADC | | | SEC | | | SSC | | | DSC | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods | token-P | token-R | token-F1 | token-P | token-R | token-F1 | token-P | token-R | token-F1 | token-P | token-R | token-F1 |
| RNP | **41.20** | 16.57 | 23.63 | 43.09 | 27.40 | 33.50 | – | – | – | – | – | – |
| HardKuma | 25.36 | 27.69 | 26.47 | 38.37 | 32.22 | 35.03 | – | – | – | – | – | – |
| InfoCal_IB | 37.23 | 21.42 | 27.19 | 43.54 | 30.72 | 36.02 | – | – | – | – | – | – |
| InfoCal(HK) | 34.45 | 26.46 | 29.93 | 40.33 | 33.88 | 36.82 | – | – | – | – | – | – |
| E-TOPJUDGE | 30.37 | 22.42 | 25.80 | 40.73 | 40.58 | 40.65 | – | – | – | – | – | – |
| E-Few-Shot | 31.47 | 21.94 | 25.85 | 39.96 | 41.63 | 40.78 | – | – | – | – | – | – |
| E-LADAN | 33.46 | 23.86 | 27.86 | 40.19 | 43.90 | 41.96 | – | – | – | – | – | – |
| E-NeurJudge (MSE) | 32.37 | **28.31** | 30.20 | 41.63 | 48.62 | 44.85 | 32.57 | 34.04 | 33.29 | 8.86 | 11.80 | 10.12 |
| E-NeurJudge (KL) | 29.41 | 22.59 | 25.55 | 43.34 | 46.56 | 44.89 | **35.39** | 37.02 | 36.19 | 12.27 | 15.45 | 13.68 |
| E-NeurJudge (JS) | 34.57 | 28.03 | 30.96 | 42.43 | 48.39 | 45.21 | 29.55 | 33.40 | 31.36 | 10.57 | 14.87 | 12.36 |
| E-NeurJudge (NCE) | 36.10 | 27.25 | **31.06** | 41.11 | 59.44 | 48.60 | 28.26 | 37.74 | 32.32 | 13.73 | 23.96 | 17.46 |
| E-NeurJudge+ (MSE) | 31.21 | 26.83 | 28.85 | **45.10** | 43.22 | 44.14 | 30.70 | 31.96 | 31.32 | 10.09 | 12.62 | 11.21 |
| E-NeurJudge+ (KL) | 29.60 | 22.62 | 25.64 | 41.16 | 50.99 | 45.55 | 33.39 | **40.88** | **36.76** | 10.04 | 14.92 | 12.00 |
| E-NeurJudge+ (JS) | 31.57 | 24.45 | 27.56 | 41.14 | 51.47 | 45.73 | 29.77 | 37.83 | 33.32 | 10.44 | 14.82 | 12.25 |
| E-NeurJudge+ (NCE) | 36.41 | 26.47 | 30.65 | 41.47 | **60.11** | **49.08** | 27.98 | 39.93 | 32.90 | **13.84** | **25.32** | **17.90** |
| Training in CAIL-big | ADC | | | SEC | | | SSC | | | DSC | | |
| Methods | token-P | token-R | token-F1 | token-P | token-R | token-F1 | token-P | token-R | token-F1 | token-P | token-R | token-F1 |
| RNP | 37.73 | 18.16 | 24.52 | 42.80 | 35.64 | 38.89 | – | – | – | – | – | – |
| HardKuma | 36.27 | 27.66 | 31.39 | 39.31 | 37.26 | 38.26 | – | – | – | – | – | – |
| InfoCal_IB | 36.26 | 21.41 | 26.92 | 39.17 | 39.25 | 39.21 | – | – | – | – | – | – |
| InfoCal(HK) | **41.76** | 20.71 | 27.69 | 40.30 | 41.05 | 40.67 | – | – | – | – | – | – |
| E-TOPJUDGE | 31.44 | 24.19 | 27.34 | 39.58 | 43.13 | 41.28 | – | – | – | – | – | – |
| E-Few-Shot | 32.91 | 25.39 | 28.67 | 41.22 | 44.09 | 42.61 | – | – | – | – | – | – |
| E-LADAN | 35.39 | 23.30 | 28.10 | 42.73 | 46.75 | 44.65 | – | – | – | – | – | – |
| E-NeurJudge (MSE) | 33.18 | 30.77 | 31.93 | 38.24 | 50.64 | 43.58 | 18.99 | 17.72 | 18.33 | 11.19 | 15.75 | 13.08 |
| E-NeurJudge (KL) | 32.65 | 30.45 | 31.51 | 39.25 | 53.99 | 45.45 | 17.99 | 17.76 | 17.87 | 11.82 | 15.72 | 13.49 |
| E-NeurJudge (JS) | 31.88 | 29.96 | 30.89 | 36.86 | 49.31 | 42.19 | 20.40 | 17.00 | 18.55 | 13.46 | 17.17 | 15.09 |
| E-NeurJudge (NCE) | 30.64 | 28.97 | 29.78 | **43.35** | **56.34** | **49.00** | 25.82 | 31.35 | 28.32 | 11.61 | 15.69 | 13.35 |
| E-NeurJudge+ (MSE) | 33.83 | **32.46** | **33.13** | 41.72 | 43.93 | 42.80 | 28.00 | 30.43 | 29.16 | 17.24 | 23.36 | 19.84 |
| E-NeurJudge+ (KL) | 32.61 | 30.69 | 31.62 | 41.54 | 47.57 | 44.35 | **30.54** | **34.20** | **32.27** | 20.31 | 27.36 | 23.31 |
| E-NeurJudge+ (JS) | 32.03 | 29.50 | 30.71 | 41.41 | 43.44 | 42.40 | 17.30 | 16.98 | 17.14 | 15.95 | 22.88 | 18.80 |
| E-NeurJudge+ (NCE) | 31.41 | 29.71 | 30.54 | 40.52 | 50.79 | 45.08 | 29.16 | 30.00 | 29.57 | **21.85** | **29.40** | **25.07** |

The bolded values indicate the optimal results in the CAIL-small and CAIL-big datasets, respectively.

function. We conduct experiments using these models on all datasets.

During task prediction, we observe that the prediction performance of E-TOPJUDGE, E-Few-Shot, and E-LADAN is positively correlated with the original teacher model's prediction effectiveness in Table V. This indicates that if a more powerful LJP model emerges in the future, our explainable framework can incorporate it as the teacher model. This improves the prediction performance of the model while maintaining explainability. It also demonstrates the generalizability of our framework. Furthermore, in the task of rationalization (Table VII), E-TOPJUDGE, E-Few-Shot, and E-LADAN achieve better results compared to traditional rationalization methods. These results underscore the significance of introducing legal particularities in the process. In summary, by replacing the teacher model with other LJP models, our framework showcases the potential for utilizing more powerful models while ensuring explainability.

*5) Visualizations of Rationalization:* To visually show the extracted circumstances, we provide an example of the extracted SEC with different methods in Fig. 8(c). From the observation, compared with other rationalization methods, E-NeurJudge selects key SEC tokens that are sufficient to support the predicted results, indicating E-NeurJudge exploits legal particularities well.

## VII. ETHICAL DISCUSSION

Since LegalAI is an emerging but sensitive technology, there are certain ethical concerns worth discussing. Although E-NeurJudge has achieved excellent performance on real datasets and provided corresponding explanations, it is still worth noting that the method is not intended to replace offline criminal litigation. It is assistance for human judges, which can help judges adopt the relevant law articles quickly and play a huge role in ensuring the principle of *"treating like cases alike"* [56], [57]. Besides, the goal of our algorithm is to select rationales to support judgment results, but whether the algorithm makes an appropriate analysis of cases remains doubtful. Therefore, judges need to check the judgment results from algorithms [58].

## VIII. CONCLUSION

In this article, we comprehensively studied the legal judgment prediction problem based on massive legal data. To be specific, we first proposed a circumstance-aware NeurJudge framework that simulated the judgment process. By utilizing the results of intermediate subtasks, NeurJudge employed CCFS to separate the facts into different circumstances and used them to yield corresponding judgment results. Then, to address confusing verdict problems, we designed a variant of NeurJudge (NeurJudge+). With the interaction between extracted label features from label similarity graphs and the factual description, more expressive fact representations have been incorporated into NeurJudge to make more accurate predictions.

Besides, although NeurJudge effectively predicted judgment results, it could not provide explanations. Therefore, we extended NeurJudge to an explainable LJP framework E-NeurJudge with a two-pass system, including the teacher (NeurJudge) and the student system (a rationalization method). Then, we adopted four types of transferring functions to ensure the student could learn legal particularities from the teacher while providing explainability for LJP. Extensive experiments on real-world datasets demonstrated the superiority of NeurJudge and E-NeurJudge.

## References

[1] B. Luo, Y. Feng, J. Xu, X. Zhang, and D. Zhao, "Learning to predict charges for criminal cases with legal basis," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2017, pp. 2727–2736.

[2] H. Zhong, Z. Guo, C. Tu, C. Xiao, Z. Liu, and M. Sun, "Legal judgment prediction via topological learning," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2018, pp. 3540–3549.

[3] W. Yang, W. Jia, X. Zhou, and Y. Luo, "Legal judgment prediction via multi-perspective bi-feedback network," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 4085–4091.

[4] Y. Feng, C. Li, and V. Ng, "Legal judgment prediction: A survey of the state of the art," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, 2022, pp. 5461–5469.

[5] N. Xu, P. Wang, L. Chen, L. Pan, X. Wang, and J. Zhao, "Distinguish confusing law articles for legal judgment prediction," in *Proc. 58th Assoc. Comput. Linguistics*, 2020, pp. 3086–3095.

[6] M. Zhang et al., *Criminal Law*. Beijing, China: Law Press, 2003.

[7] T. Lei, R. Barzilay, and T. Jaakkola, "Rationalizing neural predictions," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2016, pp. 107–117.

[8] L. Sha, O.-M. Camburu, and T. Lukasiewicz, "Learning from the best: Rationalizing prediction by adversarial information calibration," in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 13771–13779.

[9] L. Yue et al., "NeurJudge: A circumstance-aware neural framework for legal judgment prediction," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2021, pp. 973–982.

[10] X. Zhou, Y. Zhang, X. Liu, C. Sun, and L. Si, "Legal intelligence for e-commerce: Multi-task learning by leveraging multiview dispute representation," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 315–324.

[11] B. Liu et al., "Conversational versus traditional: Comparing search behavior and outcome in legal case retrieval," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2021, pp. 1622–1626.

[12] Y. Shao, Y. Wu, Y. Liu, J. Mao, M. Zhang, and S. Ma, "Investigating user behavior in legal case retrieval," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2021, pp. 962–972.

[13] Y. Ma et al., "LeCaRD: A legal case retrieval dataset for Chinese law system," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2021, pp. 2342–2348.

[14] L. Ma et al., "Legal judgment prediction with multi-stage case representation learning in the real court setting," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2021, pp. 993–1002.

[15] P. Wang, Z. Yang, S. Niu, Y. Zhang, L. Zhang, and S. Niu, "Modeling dynamic pairwise attention for crime classification over legal articles," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 485–494.

[16] Y. Feng, C. Li, and V. Ng, "Legal judgment prediction via event extraction with constraints," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 648–664.

[17] P. Wang, Y. Fan, S. Niu, Z. Yang, Y. Zhang, and J. Guo, "Hierarchical matching network for crime classification," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 325–334.

[18] Z. Hu, X. Li, C. Tu, Z. Liu, and M. Sun, "Few-shot charge prediction with discriminative legal attributes," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 487–498.

[19] H. Zhang, L. Xiao, W. Chen, Y. Wang, and Y. Jin, "Multi-task label embedding for text classification," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2018, pp. 4545–4553.

[20] S. Wang, G. Peng, and Z. Zheng, "Capturing joint label distribution for multi-label classification through adversarial learning," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 12, pp. 2310–2321, Dec. 2020.

[21] C. Du, Z. Chen, F. Feng, L. Zhu, T. Gan, and L. Nie, "Explicit interaction model towards text classification," in *Proc. 33rd AAAI Conf. Artif. Intell.*, vol. 33, pp. 6359–6366, 2019.

[22] X. Jia, Z. Li, X. Zheng, W. Li, and S.-J. Huang, "Label distribution learning with label correlations on local samples," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 4, pp. 1619–1631, Apr. 2021.

[23] G. Wang et al., "Joint embedding of words and labels for text classification," in *Proc. 56th Assoc. Comput. Linguistics*, 2018, pp. 2321–2331.

[24] D. Chai, W. Wu, Q. Han, F. Wu, and J. Li, "Description based text classification with reinforcement learning," in *Proc. 37th Int. Conf. Mach.Learn.*, 2020, pp. 1371–1382.

[25] M. Yu, S. Chang, Y. Zhang, and T. S. Jaakkola, "Rethinking cooperative rationalization: Introspective extraction and complement control," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2019, pp. 4094–4103.

[26] B. Paranjape, M. Joshi, J. Thickstun, H. Hajishirzi, and L. Zettlemoyer, "An information bottleneck approach for controlling conciseness in rationale extraction," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2020, pp. 1938–1952.

[27] Y. Cui, T. Liu, W. Che, Z. Chen, and S. Wang, "ExpMRC: Explainability evaluation for machine reading comprehension," *Heliyon*, vol. 8, 2022, Art. no. e09290.

[28] L. Yue, Q. Liu, Y. Du, Y. An, L. Wang, and E. Chen, "Dare: Disentanglement-augmented rationale extraction," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 26603–26617, 2022.

[29] L. Yue, Q. Liu, L. Wang, Y. An, Y. Du, and Z. Huang, "Interventional rationalization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2023, pp. 11404–11418.

[30] L. Yue, Q. Liu, Y. Du, L. Wang, W. Gao, and Y. An, "Towards faithful explanations: Boosting rationalization with shortcuts discovery," 2024, *arXiv:2403.07955*.

[31] J. Bastings, W. Aziz, and I. Titov, "Interpretable neural predictions with differentiable binary variables," in *Proc. 57th Assoc. Comput. Linguistics*, 2019, pp. 2963–2977.

[32] C. Sardianos, I. M. Katakis, G. Petasis, and V. Karkaletsis, "Argument extraction from news," in *Proc. 2nd Workshop Argumentation Mining*, 2015, pp. 56–66.

[33] X. Du and C. Cardie, "Event extraction by answering (almost) natural questions," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 671–683.

[34] S. Li, H. Ji, and J. Han, "Document-level event argument extraction by conditional generation," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics*, 2021, pp. 894–908.

[35] K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 1724–1734.

[36] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[37] Y. Cui et al., "Pre-training with whole word masking for Chinese bert," 2019, *arXiv: 1906.08101*.

[38] Z. Wang, H. Mi, and A. Ittycheriah, "Sentence similarity learning by lexical decomposition and composition," in *Proc. 26th Int. Conf. Comput. Linguistics*, 2016, pp. 1340–1349.

[39] Q. Liu et al., "Finding similar exercises in online education systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1821–1830.

[40] X. Wang et al., "Fine-grained similarity measurement between educational videos and exercises," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 331–339.

[41] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[42] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. 32st AAAI Conf. Artif. Intell.*, 2018, pp. 3538–3545.

[43] Y. Cui, T. Liu, W. Che, Z. Chen, and S. Wang, "Teaching machines to read, answer and explain," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 1483–1492, 2022.

[44] E. Jang, S. Gu, and B. Poole, "Categorical reparametrization with gumble-softmax," 2016, *arXiv:1611.01144*.

[45] M. I. Belghazi et al., "Mutual information neural estimation," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2018, pp. 531–540.

[46] A. V. D. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv: 1807.03748*.

[47] C. Xiao et al., "Cail2018: A large-scale legal dataset for judgment prediction," 2018, *arXiv: 1807.02478*.

[48] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," in *Neural Processing Letters*. Berlin, Germany: Springer, 1999.

[49] H. Chen, D. Cai, W. Dai, Z. Dai, and Y. Ding, “Charge-based prison term prediction with deep gating network,” in *Proc. Conf. Empir. Methods Natural Lang. Process. Joint Conf. Natural Lang. Process.*, 2019, pp. 6362–6367.
[50] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2019, pp. 4171–4186.
[51] H. Zhong, Z. Zhang, Z. Liu, and M. Sun, “Open chinese language pre-trained model zoo,” 2019. [Online]. Available: https://github.com/thunlp/openclap
[52] P. Kumaraswamy, “A generalized probability density function for double-bounded random processes,” *J. Hydrol.*, vol. 46, pp. 79–88, 1980.
[53] M. Sun, X. Chen, K. Zhang, Z. Guo, and Z. Liu, “THULAC: An efficient lexical analyzer for Chinese, *Retrieved Jan*, vol. 10, pp. 2022, 2016.
[54] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, *arXiv:1412.6980*.
[55] L. V. D. Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.
[56] C. Sun, Y. Zhang, Q. Zhang, and X. Liu, “Legal artificial intelligence - have you lost a piece from jigsaw puzzle?,” in *Proc. AAAI Spring Symp. Combining Mach. Learn. Knowl. Eng.*, 2020.
[57] C. Rigano, “Using artificial intelligence to address criminal justice needs,” *Nat. Inst. Justice J.*, vol. 280, pp. 1–10, 2019.
[58] Y. Wu et al., “De-biased court’s view generation with causality,” in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2020, pp. 763–780.

**Linan Yue** received the BE degree in computer science from Hehai University, Nanjing, China, in 2019, where he is currently working toward the PhD degree in data science with the University of Science and Technology of China under the advisory of Prof. Q. Liu. He has published several papers in referred conference proceedings, such as NeurIPS, ICLR, SIGIR, SIGKDD and WWW conference. His current research interests include data mining, AI for Law and trustworthy AI.

**Qi Liu** (Member, IEEE) received the PhD degree from the University of Science and Technology of China (USTC), Hefei, China, in 2013. He is currently a professor with the School of Computer Science and Technology, University of Science and Technology of China (USTC), Hefei, China. His research interests include data mining, machine learning, and recommender systems. He was the recipient of KDD’ 18 Best Student Paper Award and ICDM’ 11 Best Research Paper Award. He was also the recipient of China Outstanding Youth Science Foundation, in 2019.

**Binbin Jin** received the BE and PhD degrees in computer science from the University of Science and Technology of China (USTC), Hefei, China, in 2016 and 2021, respectively. He is currently working in Huawei Cloud Computing Technologies, Company Ltd., Hangzhou, China. He has published several papers in referred journals and conference proceedings, such as the *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *IEEE Transactions on Knowledge and Data Engineering*, *ACM Transactions on Information Systems*, NeurIPS, and AAAI. His current research interests include data mining, recommender systems and natural language processing.

**Han Wu** received the PhD degrees in computer science from the University of Science and Technology of China (USTC), Hefei, China, in 2024. She is currently working with the Career Science Lab, Boss Zhipin. She has published several papers in refereed conference proceedings, such as IJCAI, ICDM, SIGIR, and SIGKDD. Her current research interest includes data mining in business intelligence, with a focus on patent analysis such as patent litigation prediction and patent technology tracing.

**Yanqing An** received the BE degree in computer science from Taiyuan University of Technology, Taiyuan, China, in 2019, where he is currently working toward the PhD degree in data science with University of Science and Technology of China under the advisory of Prof. Q. Liu. He has published several papers in referred conference proceedings, such as SIGKDD and SIGIR conference. His current research interests include data mining, user modeling and AI for legal applications.