




Judicial knowledge-enhanced magnitude-aware reasoning for numerical legal judgment prediction

Sheng Bi^{1,2} · Zhiyao Zhou¹ · Lu Pan³ · Guilin Qi¹ 

Accepted: 5 October 2022 / Published online: 2 November 2022
© The Author(s), under exclusive licence to Springer Nature B.V. 2022

Abstract

Legal Judgment Prediction (LJP) is an essential component of legal assistant systems, which aims to automatically predict judgment results from a given criminal fact description. As a vital subtask of LJP, researchers have paid little attention to the numerical LJP, i.e., the prediction of imprisonment and penalty. Existing methods ignore numerical information in the criminal facts, making their performances far from satisfactory. For instance, the amount of theft varies, as do the prison terms and penalties. The major challenge is how the model can obtain the ability of numerical comparison and magnitude perception, e.g., $400 < 500 < 800$, 500 is closer to 400 than to 800. To this end, we propose a judicial knowledge-enhanced magnitude-aware reasoning architecture, called NumLJP, for the numerical LJP task. Specifically, we first implement a contrastive learning-based judicial knowledge selector to distinguish confusing criminal cases efficiently. Unlike previous approaches that employ the law article as external knowledge, judicial knowledge is a quantitative guideline in real scenarios. It contains many numerals (called anchors) that can construct a reference frame. Then we design a masked numeral prediction task to help the model remember these anchors to acquire legal numerical commonsense from the selected judicial knowledge. We construct a scale-based numerical graph using the anchors and numerals in facts to perform magnitude-aware numerical reasoning. Finally, the representations of fact description, judicial knowledge, and numerals are fused to make decisions. We conduct extensive experiments on three real-world datasets and select several competitive baselines. The results demonstrate that the macro-F1 of NumLJP improves by at least 9.53% and 11.57% on the prediction of penalty and imprisonment, respectively.

Keywords Numerical legal judgment prediction · Masked numeral prediction · Judicial knowledge · Magnitude-aware · Numerical reasoning

✉ Guilin Qi
gqi@seu.edu.cn

Extended author information available on the last page of the article

Introduction

Legal Judgment Prediction (LJP) aims to determine the judicial results based on given criminal fact description Cheng et al. (2020), Chalkidis et al. (2019), which is an elementary part of the legal assistant systems. On the one hand, people without legal background knowledge can input the fact description of their own words and find the associated processing result of cases. On the other hand, the efficiency of lawyers or legal experts has been improved since the automatic prediction of the legal assistant system can be used as a reference. Consequently, LJP has aroused extensive attention to industry and academia in recent years.

LJP contains four subtasks, i.e., the predictions of charges Luo et al. (2017), Hu et al. (2018), law articles Xu et al. (2020), prison terms Li et al. (2020), Yue et al. (2021), and penalty terms Dong and Niu (2021). The last two subtasks, denoted as numerical LJP, are the most significant and challenging in LJP and are the focus of this research. LJP is a long-standing research problem and has been undertaken for decades Kort (1957), George and Epstein (1992), Liu et al. (2015). Early studies usually propose to extract shallow textual features, such as characters, words, and phrases for LJP Liu et al. (2004), Liu and Liao (2005).

Recently, deep learning has been applied to many Natural Language Processing (NLP) tasks due to its excellent performance in automatic feature extraction. Luo et al. Luo et al. (2017) proposed a joint model with soft attention for charge prediction and relevant article extraction. Some researchers believe that there are interdependencies between multiple subtasks in LJP. Therefore, they proposed the multi-task learning frameworks to utilize judicial result dependencies for LJP Zhong et al. (2018), Yang et al. (2019), Yue et al. (2021), Dong and Niu (2021). They have achieved outstanding performances on two subtasks of LJP, i.e., the prediction of charges and law articles.

However, the numerical LJP have received little attention from researchers. As shown in Fig. 1, the numerical LJP aims to input a **Criminal Fact** description, then predict the imprisonment and penalty in the **Court Decision** (blue font). In practice, imprisonment and penalty are divided into several intervals, see Sect. 5.1.1 for more details. Although the existing approaches to solve the numerical LJP are somewhat different, their core innovations focus on learning more efficient text representations. For example, Chen et al. Chen et al. (2018) proposed a deep gating network to filter and aggregate information related to the charges from fact description to obtain charge-based representations.

To the best of our knowledge, the best evaluation scores for numerical LJP are 41.18 and 42.61 respectively Yue et al. (2021), Chen et al. (2018), which are much lower than the performance for charges and law articles (91.94 and 88.93) Dong and Niu (2021). We found empirically in these models' reproduction that the common cause of the above phenomena is that they ignore the numerals in the factual description of the crime. Specifically, they take numerals directly as plain words or replace them with a common token, such as [UNK].

For case A and case B in Fig. 1, different defendants stole property worth RMB 1,939 and 7,300, respectively. These two cases correspond to the same charge and law article, but the results of numerical LJP are quite different. Obviously,

A. **犯罪事实**: 2015年10月21日晚, 胡某驾驶一辆面包车至南田村进入到曾某家中, 胡某盗走曾某一台黑色长虹电视机。经认证中心认证, 该电视机价值人民币**1939元**。**法院判决**: 根据中华人民共和国刑法第**264**条的规定, 判决如下: 被告人胡某犯**盗窃罪**, 判处有期徒刑**7个月**, 并处罚金人民币**2000元**。
#Criminal Fact: On the evening of October 21, 2015, Hu drove a van to Nantian Village and entered Zeng's home, where Hu stole a black Changhong TV from Zeng. The TV was certified by the Value Certification Center to be worth **RMB 1,939**. **Court Decision**: According to Law Article **264** of the Criminal Law of the PRC, the judgment is as follows: the defendant Hu was sentenced to **7 months'** imprisonment and a penalty of **RMB 2,000** for the crime of **Theft**.

B. **犯罪事实**: 2016年08月13日上午, 牛某来到北戴河医院家属院, 进入到102栋3单元楼道内, 用事先准备好的开锁工具打开501室的防盗门进入室内, 将被害人王某的**7300元**现金盗走。**法院判决**: 根据中华人民共和国刑法第**264**条的规定, 判决如下: 被告人牛某犯**盗窃罪**, 判处有期徒刑**1年**, 并处罚金人民币**5000元**。
#Criminal Fact: On the morning of August 13, 2016, Niu came to the family courtyard of Beidaihe Hospital, entered the stair of Unit #3, Building #102, opened the security door of Room #501 with a pre-prepared lock picking tools to enter the interior, and stole **RMB 7,300** in cash from Wang. **Court Decision**: According to Law Article **264** of the Criminal Law of the PRC, the judgment is as follows: the defendant Niu was sentenced to **one year'** imprisonment and a penalty of **RMB 5,000** for the crime of **Theft**.

C. **犯罪事实**: 2003年7月5日, 孙某在北京市大兴区龙河路, 将被害人程某行带至京开公路东侧树林内, 对程某进行殴打后, 预谋实施抢劫, 并抢走程某现金**3890元**。**法院判决**: 根据中华人民共和国刑法第**263**条的规定, 判决如下: 被告人孙某犯**抢劫罪**, 判处有期徒刑**4年**, 并处罚金人民币**8,000元**。
#Criminal Fact: On July 5, 2003, Sun took the victim Cheng line to the woods on the east side of Jingkai Highway in Longhe Road, Daxing District, Beijing and beat Cheng. Then Sun premeditatedly committed robbery and robbed Cheng of **RMB 3,890** in cash. **Court Decision**: According to Law Article **263** of the Criminal Law of the PRC, the judgment is as follows: the defendant Sun was sentenced to **four years'** imprisonment and a penalty of **RMB 8,000** for the crime of **Robbery**.

Fig. 1 Three simplified real-world examples. Each example contains a criminal fact description and the court's decision. The red font is the numerals in the fact, and they indicate a quantitative description of the damage caused by the criminal behavior. The green font indicates the applied law article and charge, respectively. The blue font is the target of the numerical LJP, i.e., the imprisonment and the penalty, and they are divided into intervals. In the pre-processing stage, all numbers are processed into Arabic and years are converted into months

ignoring the numerals leads to the loss of the ability and justification for numerical judgments.

The numerical LJP faces two challenges, i.e., performing the numerical comparison and magnitude awareness. We use a simple and motivating hypothetical example to illustrate these two challenges intuitively.

Given the three criminal cases shown in Fig. 1, we perform a numerical judgment prediction for a simple test case *Someone stole RMB 7,000 in cash*. A straightforward idea is to compare the numerals involved in the hypothetical case with the numerals in similar criminal cases, i.e., cases A and B: $1,939 < 7,000 < 7,300$. The imprisonment should be between 7 and 12 months and the penalty should be between RMB 5,000 and 8,000 (**numerical comparison**). Contemporary numerical reasoning methods such as NumNet Ran et al. (2019) model comparison relationships between numerals with remarkable success. However, these methods have two limitations:

1. They ignore the criminal type corresponding to the numerals. In the above example, the numeral types in cases A and B are the property obtained from stealing, while the numeral type in case C is the property obtained from the robbery. If we compare 7,000 in the test case with 3,890 in case C, we will make a wrong

判决知识: 盗窃公私财物数额较大, 以 **一千元至三千元** 为起点。处三年以下有期徒刑、拘役或者管制, 并处或者单处罚金。盗窃公私财物数额巨大, 以 **三万元至十万元** 为起点。处三年以上十年以下有期徒刑, 并处罚金。盗窃公私财物数额特别巨大, 以 **三十万元至五十万元** 为起点。处十年以上有期徒刑或者无期徒刑, 并处罚金或者没收财产。# **Judicial Knowledge:** The stealing of public and private property is larger, starting with **RMB 1,000 to 3,000**. Sentenced to less than three years of imprisonment and a penalty. The stealing of public and private property in a huge amount, from **RMB 30,000 to 100,000** as a starting point. Sentenced to more than three years to ten years of fixed-term imprisonment and a fine. The stealing of public and private property is an extra huge amount, from **RMB 300,000 to 500,000** as a starting point. Sentenced to more than ten years of imprisonment or life imprisonment and a penalty or confiscation of property.

法条: 盗窃公私财物, 数额较大或者多次盗窃的, 处三年以下有期徒刑、拘役或者管制, 并处或者单处罚金; 数额巨大或者有其他严重情节的, 处三年以上十年以下有期徒刑, 并处罚金; 数额特别巨大或者有其他特别严重情节的, 处十年以上有期徒刑或者无期徒刑, 并处罚金或者没收财产。# **Law Article:** Whoever steals a relatively large amount of public or private property or commits theft repeatedly shall be sentenced to fixed-term imprisonment of not more than three years, criminal detention or public surveillance and shall also, or shall only, be fined; if the amount is huge, or if there are other serious circumstances, he shall be sentenced to fixed-term imprisonment of not less than three years but not more than 10 years and shall also be fined; if the amount is especially huge, or if there are other especially serious circumstances, he shall be sentenced to fixed-term imprisonment of not less than 10 years or life imprisonment and shall also be fined or be sentenced to confiscation of property.

Fig. 2 Judicial knowledge and the law article on Theft. The judicial knowledge is more specific than the law article and clearly states the quantitative justification for decision-making. The **numerical anchors** in this paper refer to the numerals in green font, and they compose the reference frame for the comparison

judgment. In other words, we need to determine the types of numerals involved in the criminal facts for numerical LJP.

2. They only model comparison relationships between numerals while ignoring their magnitudes, which is essential to numerical LJP. In this example, 7,000 is closer to 7,300 than 1,939, which suggests that the imprisonment of the test case should be closer to 12 months than seven months (**magnitude awareness**).

Therefore, we propose a judicial knowledge-enhanced, magnitude-aware reasoning network called **NumLJP** for numerical LJP. NumLJP consists of four main modules, i.e., a judicial knowledge selection module, a legal numerical commonsense acquisition based on a masked numeral prediction (MNP) task, a reasoning module, and a judgment prediction module. Specifically, we first select the judicial knowledge corresponding to the given criminal fact by a contrastive learning-based classifier. The identical judicial knowledge implies the same type of criminal behavior. The numerals to be compared should involve the same type of criminal facts. This module is the cornerstone of the whole model, which mimics the judge's sentencing practice. A model yields an accurate judgment only when the correct judicial knowledge is applied. Unlike the previous works of using the law article as external knowledge, which consists of a general abstract description of the criminal activity. Judicial knowledge is the practical implementation of rules in a real scenario, including quantitative criteria for the crime. We utilize an example to illustrate the differences between the law article and the judicial knowledge, as shown in Fig. 2. Then the model acquires legal numerical commonsense from the judicial knowledge selected in the previous step via a Masked Numeral Prediction (MNP) task. As we described

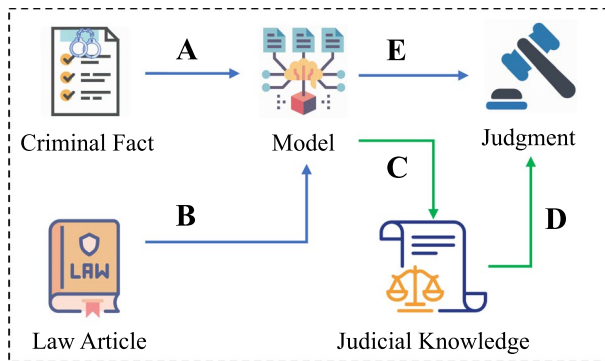


Fig. 3 The general trial process in a real scenario. **A** indicates understanding the criminal facts, **B** refers to finding the relevant law articles, and **C** means retrieving the corresponding judicial knowledge. The final decision is made in step **E** based on the appropriate sentencing criteria matched in step **D**

before, the judicial knowledge contains quantitative criteria for criminal behavior, which are called **numerical anchors** in this paper. These anchors can be used as reference points for the model to perform numerical reasoning. Inspired by Ran et al. (2019), we design a novel numerical graph. It preserves numerical comparing information and perceives the magnitude between different numerals. Based on that, we create a magnitude-aware numerical reasoning network MagNet, which is our major contribution. In the judgment prediction module, we fuse representations of numerals, fact descriptions, and judicial knowledge to predict the judgment outcome jointly. We evaluate the NumLJP on three real-world datasets and select several competitive baselines. The results illustrate that NumLJP obtains significant and consistent improvement compared to all baselines, which demonstrates the effectiveness of our method. The macro-F1 score of NumLJP improves by at least 9.53 and 11.57% on both tasks, respectively, and achieves state-of-the-art performance.

1 Motivation

When a judge faces a criminal fact, they first need to understand the description of the crime and make a qualitative analysis. Then the judge matches the defendant's illegal behavior with the corresponding law articles. Finally, the judge derives a quantitative judgment of due from the damages (or illegal gains) caused by the defendant based on judicial knowledge. The general trial process Ge et al. (2021) is shown in Fig. 3. From this, it is clear that the numerical LJP requires understanding the defendant's criminal behavior from two perspectives. i.e., qualitative and quantitative analysis. Quantitative analysis indicates quantifying the damages or illegal gains caused by the defendant. Pre-trained language models (PLMs) have been demonstrated to achieve impressive performance in context understanding. Therefore, we choose a typical PLM as a backbone for encoding the fact description in this paper. We devote more attention to the quantitative analysis part.

The numerical LJP is not simply a straightforward mathematical mapping from input to output. We assume that there is a **soft** rule $\xi(\cdot) \Rightarrow Y$ between the numeral n in fact description and final judgment Y : $\xi(n \mid n \in [a, b]) \Rightarrow Y$, where $[a, b]$ and Y refer to the range of numerals appearing in the text and the numerical judgment results, respectively. It is crucial to acquire $[a, b]$ for ξ .

Benefiting from the massive legal judgment with data-driven paradigm, deep models can derive a rough interval $[a', b']$ for numeral representation learning Lin et al. (2020). On this basis, some approaches combine the representations of comparative relations into numerical reasoning, such as NumNet Ran et al. (2019), significantly outperforming baselines that ignore numerals.

However, these methods face two insurmountable obstacles, limited training data and few numbers in the crime facts. They cannot learn an adequate representation of massive unseen numerals with limited training data. Besides, if there is only one numeral in the fact description, existing methods cannot model the relation between this numeral and the numerals in other fact descriptions. To this end, it becomes difficult to derive the proper intervals.

A clever solution is to introduce numerical anchors n^* directly from the official judicial knowledge. This method avoids directly learning infinite unseen numerals in the fact description and transfers the focus of model training to the representation of numerals in judicial knowledge. There is always corresponding judicial knowledge and fixed legal sentencing criteria for a given crime type. We cannot guarantee that the numerals in a new case are the ones the model has seen. In contrast, the numerals in judicial knowledge are limited and fixed.¹ We can learn legal numerical commonsense through judicial knowledge.²

We summarize two advantages of this method, (1) The anchors n^* provide a bridge for the numerals in fact descriptions, i.e., $n_1 < n^*, n^* < n_2 \Rightarrow n_1 < n_2$. Even if n_1 and n_2 do not appear in the same context. (2) The model can avoid comparing different types of numerals. Suppose the model has learned two samples, {theft, 5,000} and {theft, 4,000}. There is a new criminal case with {theft, 6,000}, the model has difficulty predicting the approximate outcome. Using numerical anchors to construct a reference system with the same type can solve this problem well.

We obtain a better numerical representation by introducing numerals from judicial knowledge. The previous methods can only model the comparison relationship between numerals Ran et al. (2019). The follow-up challenge is measuring the magnitude between two numerals. For example, $400 < 500 < 800$, 500 is closer to 400 than 800. It is an intuitive idea to fine-grained the intervals according to the criminal type. We define a scale for numerals in the same judicial knowledge and set a shared multiplier list for all crimes. We divide the interval with a specified scale and then combine the multiplier and the scale to measure the magnitude between two numerals.

¹ These numerals may change as the legal system is reformed, but they are fixed over a considerable period. Therefore, we assume that these numerals are fixed.

² Legal numerical commonsense indicates the judge's knowledge of the numerical features in the fact description, such as the amount of property stolen, the number of drugs sold, etc. Each of these numerals has its own range and probability distribution.

For instance, “50” is the specified scale for [400, 500, 800], the shared multiplier list is [1, ..., 8]. Then the difference between 400 and 500 is two times the scale, and the difference between 500 and 800 is six times the scale. In this way, we only need to encode the multiplier and the specified scale. The detailed calculation is defined in Sect. 4.3.

2 Related work

2.1 Legal judgment prediction

Legal artificial intelligence have attracted a lot of attention from academia and industry in recent years Zhong et al. (2020), and many legal assistant systems (LAS) have been developed Bi et al. (2019, 2019), Xu et al. (2020), Cheng et al. (2020). LJP plays an important role in the LAS and has been investigated for decades George and Epstein (1992), Liu et al. (2015). Early works usually leverage mathematical and statistical algorithms to analyze existing legal cases Segal (1984), Liu et al. (2004). Inspired by the immense success of deep learning, Luo et al. Luo et al. (2017) designed a hierarchical attention-based neural network framework to extract the most relevant law articles for charge prediction. Cheng and Xu et al. Cheng et al. (2020), Xu et al. (2020) proposed to use external law articles or legal schematic knowledge as the discriminative features to distinguish confusing cases. Some researchers have paid attention to the dependencies between the subtasks of LJP and proposed the multitask learning-based frameworks Zhong et al. (2018), Yang et al. (2019), Yue et al. (2021), Dong and Niu (2021). They have achieved outstanding performances on two subtasks of LJP, i.e., the prediction of charges and law articles.

However, few methods focus on the numerical LJP, i.e., the prediction of penalty and imprisonment. Chen et al. Chen et al. (2018) proposed a charge-based prison term prediction model that employs a deep gating network to select fine-grained features for a specific charge.

To the best of our knowledge, the best evaluation scores for numerical LJP are 41.18 and 42.61 respectively Yue et al. (2021), Chen et al. (2018). Despite these efforts put into the numerical LJP, their optimal performances are still far below the predictions of charges and law articles (91.94 and 88.93) Dong and Niu (2021).

We empirically found that the main reason for this phenomenon is that these methods ignore the numerals in the fact description. Specifically, these methods directly treat numbers as ordinary words or even simply replace them with a generic token [UNK]. The numerals in a criminal fact description are usually a quantitative description of the damage caused by the defendant’s criminal behavior, such as the value of the stolen property. Obviously, it is sufficient to rely on the text when we perform a qualitative analysis of the criminal facts, i.e., which law articles and charges should be applied. However, when performing quantitative analysis, it will be challenging to accurately predict the outcome of numerical LJP if the numerals’ semantics cannot be learned.

2.2 Contrastive learning

In recent years, contrastive learning has brought breakthroughs in the Computer Vision (CV) domain, a trend that has also attracted the attention of Natural Language Processing (NLP) community Jaiswal et al. (2021), Hénaff (2020), van den Oord et al. (2018). Wu et al. (2018) proposed to use a noise-contrast estimation Gutmann and Hyvärinen (2010) to approximate the softmax distribution and a proximal regularization methods Parikh and Boyd (2014) to resort the training process. They advocated a non-parametric approach to take instance-level discrimination as a metric learning problem, where the distance between instances is computed directly from the features. Consequently, better representations are learned for individual instances to capture their similarity. Sermanet et al. (2018) proposed to learn the representations using metric learning loss, and they observed that the similar representations attract each other in the embedding space while repelling visually identical but functionally different samples. Recently, some researchers have applied contrastive learning to fully supervised scenarios to make more effective use of label information. Khosla et al. (2020) proposed a loss function for supervised learning that considers multiple positive and negative samples on each anchor by using labeling information. Unlike self-supervised contrastive learning that employs single positive ones. These positives come from samples with the same class as the anchor, rather than data augmentation of the anchor as in self-supervised learning Tian et al. (2020). To address the challenges of sub-optimal generalization and instability that exist with cross-entropy loss, Gunel et al. (2021) proposed a supervised contrastive training objective combined with cross-entropy. The loss function is robust to fine-tuning the noise in the training data and also exhibits better generalization to relevant tasks with limited labeled task data.

Contrastive learning has powerful representation capabilities, which can effectively distinguish between different labels of samples in supervised learning scenarios. The evidence-based selection of proper judicial knowledge is crucial in the LJP. In this process, confusing cases are the main obstacle to judicial knowledge selection Cheng et al. (2020). With the help of contrastive learning, the model is able to learn the slight differences between the confusing samples Robinson et al. (2021).

2.3 Numerical reasoning

Numerical reasoning is involved in a wide range of tasks, such as information extraction Bakalov et al. (2011), retrieval Banerjee et al. (2009), math word problem Huang et al. (2016), Amini et al. (2019), Qin et al. (2020), Patel et al. (2021) and representation learning Jiang et al. (2019). In recent years, pre-trained language models have shown great promise for several tasks in natural language processing with the training of large-scale data Devlin et al. (2019), Brown et al. (2020), Lewis et al. (2020), Sanh et al. (2019). However, numerical reasoning still face significant

challenges. Researchers have tried to propose some end-to-end frameworks to inject numerical reasoning capabilities into pre-trained language models Thawani et al. (2021), Yoran et al. (2022).

More similar to our work is numerical machine reading comprehension. It requires both natural language understanding and numerical reasoning. Amrita et al. Saha et al. (2021) presented a weakly supervised neural symbolic module network using answers as learning signals. A distinctive characteristic is that discrete reasoning is avoided by employing only learnable modules with an exhaustive precomputed output space. Geva et al. Geva et al. (2020) suggested that due to the expressions evaluated as correct answers grows exponentially, the extension of existing methods to arbitrary computations requires the use of non-differentiable operations, which may lead to training difficulties. Therefore, they proposed to add two pre-training steps to the language model for automatic synthetic expressions generation, such as $3 + 4 + 11 = 18$. This gives the model the ability to understand computations in pseudo-natural language.

From the perspective of task goals, another study relevant to numerical LJP is ordinal classification or regression (Niu et al. 2016, Baly et al. 2019). These models require a combination of variability and continuity of labels. Raul et al. proposed an ordinal regression approach using multiple binary classifiers, which considers the orderliness between categories (Diaz and Marathe, 2019). To address the inconsistency among multiple binary classifiers, Cao et al. proposed a consistent rank logits framework (CORAL), which guarantees rank-conotonicity (Cao and Mirjalili, 2020). CORAL can be conveniently extended to other neural network models. Shi et al. suggested a rank consistent ordinal regression model CORN that utilizes the chain rule of conditional probability distribution to obtain unconditional rank probabilities (Shi et al. 2021). CORN relieves the constraints of weight sharing in fully connected layers in neural networks.

In legal scenarios, there is no need to perform exact calculations on the numerals in the text, and we are more concerned with the numerical comparative reasoning ability of the model. Dua et al. Dua et al. (2019) proposed a numerically-aware model to perform multiple operations on numerals, such as count, addition, and subtraction. Ran et al. Ran et al. (2019) proposed a numerical reasoning network, NumNet, which constructs a comparison-aware GNN to reason the relative information among numerals. Since the NumNet cannot identify different number types, Chen et al. Chen et al. (2020) introduced a heterogeneous directed graph to integrate the type and entity information for numerical reasoning.

Inspired by those studies, we propose a judicial knowledge-enhanced magnitude-aware reasoning architecture for the numerical LJP task. Note that the main distinction between ours and these works is that our approach does not directly perform representation or reasoning about numerals but instead shifts the main learning objective to those numerals fixed in judicial knowledge.

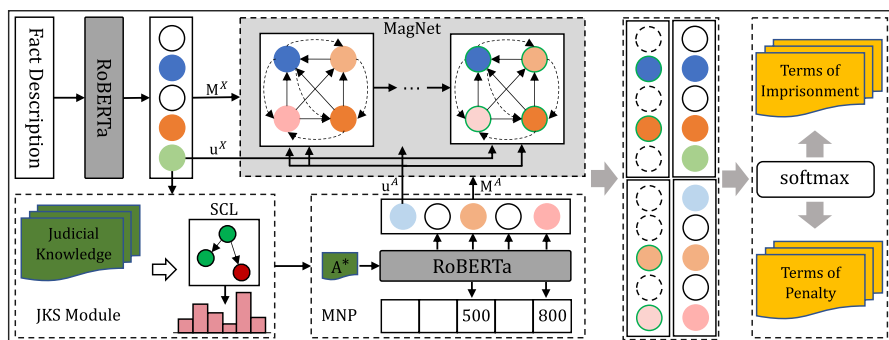


Fig. 4 The overall framework of the proposed NumLJP. The NumLJP contains four main modules, a contrastive learning-based judicial knowledge selection module (JKS), a legal numerical sense acquisition module (MNP), a numerical reasoning module incorporating judicial knowledge (MagNet), which is our main contribution, and a judgment prediction module, detailed in Sect. 4.2

3 Methodology

We first formally formulate the task of numerical Legal Judgment Prediction. Then we introduce the proposed framework **NumLJP**, followed by providing details of its modules.

3.1 Task formulation

Similar to the previous work setting Luo et al. (2017), Yang et al. (2019), we treat numerical LJP as a text classification task. As illustrated in Fig. 1, given a criminal fact description, the model automatically predicts the imprisonment term and the penalty term. Note that the imprisonment terms are divided by months. The penalty terms are divided according to the practice of Chinese criminal law, as detailed in Sect. 5.1.1. Formally, let X denote the criminal fact description. The $\mathcal{A} = \{A_1, \dots, A_{n_A}\}$ denote the judicial knowledge. A *theft*-related judicial knowledge is shown in Fig. 2. $Y_{\mathcal{A}} = \{y_i^{\mathcal{A}}\}_{i=1}^{n_A}$ denotes the set of judicial knowledge labels, where $y_i^{\mathcal{A}} \in \{0, 1\}^{n_A}$. $Y^I = \{y_i^I\}_{i=1}^{n^I}$ and $Y^P = \{y_i^P\}_{i=1}^{n^P}$ denote the terms of imprisonment and the terms of penalty, respectively, where $y_i^I \in \{0, 1\}^{n^I}$ and $y_i^P \in \{0, 1\}^{n^P}$. NumLJP aims to select the corresponding judicial knowledge \mathcal{A} based on the given fact description X and, on that basis, to predict the most probable terms of imprisonment Y^I and penalty Y^P .

3.2 Framework

The overall framework of our model NumLJP is illustrated in Fig. 4. This section will describe in detail the modules in the NumLJP and the relationships among them.

3.2.1 Backbone encoder

We utilize a pre-trained language model (PLM), RoBERTa Liu et al. (2019) as the backbone encoder, which takes the fact description as input and outputs a dense representation for each token. Here we use the vector of specific token [CLS] as the full semantic representation of the criminal fact:

$$\mathbf{u}^X, \bar{\mathbf{X}} = \text{RoBERTa}([\text{CLS}]; X), \quad (1)$$

where $([\text{CLS}]; X)$ denotes the concatenation of [CLS] and X as input, \mathbf{u}^X denotes the embedding of [CLS], and $\bar{\mathbf{X}}$ refers to the embeddings of all tokens in the fact description X .

3.2.2 Judicial knowledge selection

This module is about selecting accurate judicial knowledge based on the fact description. The major challenge lies in confusing criminal cases Cheng et al. (2020), which have minor differences in their fact descriptions. Assuming that “30,000” is mentioned in the fact description, the selection of judicial knowledge, such as “theft” or “fraud”, can significantly impact the outcome of the judgment. Therefore, it is crucial to effectively discriminate the confusing cases to apply the correct judicial knowledge.

Besides, most previous studies of numerical reasoning have ignored the numeral types, Chen et al. (2020) demonstrated that the numeral types can significantly impact the numerical comprehension and reasoning. However, they determine the types of several specific numerals only using units. In the legal domain, relying on numeral units alone cannot effectively distinguish between different types of the numeral. For example, in Fig. 1, the numerals in Cases B and C both indicate the amount of money but yield vastly different outcomes. Therefore, the judicial knowledge selection in advance is a fine-grained identification of the numeral types.

Contrastive learning has demonstrated excellent performance in several natural language processing tasks, leading to state-of-the-art results Jaiswal et al. (2021), Shorten and Khoshgoftaar (2021). Taking the idea from Gunel et al. (2021), we implement a contrastive learning-based judicial knowledge selection module (JKS), which is a classification model. In addition to the cross-entropy loss \mathcal{L}_1 , a novel supervised contrastive learning objective \mathcal{L}_2 is proposed for fine-tuning the PLM. The overall loss \mathcal{L}_{JKS} is defined as follows:

$$\begin{aligned}
\mathcal{L}_{\text{JKS}} &= (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_2, \\
\mathcal{L}_1 &= -\frac{1}{N} \sum_{i=1}^N \sum_{m=1}^{n^A} y_{i,m}^A \cdot \log \hat{y}_{i,m}^A, \\
\mathcal{L}_2 &= \sum_{i=1}^N -\frac{1}{N_{y_i^A} - 1} \sum_{j=1}^N \mathbf{1}_{i \neq j} \mathbf{1}_{y_i^A = y_j^A} \\
&\quad \log \frac{\exp(\mathbf{u}_i^X \cdot \mathbf{u}_j^X / \tau)}{\sum_{k=1}^N \mathbf{1}_{i \neq k} \exp(\mathbf{u}_i^X \cdot \mathbf{u}_k^X / \tau)},
\end{aligned} \tag{2}$$

where λ is weight hyperparameter, N denotes the batch size, τ is an adjustable scalar temperature parameter, $y_{i,m}^A$ and $\hat{y}_{i,m}^A$ denote the golden label and the prediction probability of i^{th} sample with class m , respectively. $\mathbf{1}$ denotes an indicator function. \mathbf{u}_i^X denotes the semantic representations of the i^{th} fact description in a batch, similarly with \mathbf{u}_j^X and \mathbf{u}_k^X , refer to Eq. 1. The loss of Supervised Contrastive Learning (SCL) captures the similarity between samples, which makes them with the same class are closer together than ones with different classes. This enables the model to distinguish between confusing crimes and apply proper judicial knowledge effectively.

3.2.3 Legal numerical commonsense acquisition

In this module, we design a PLM-based Masked Numeral Prediction (MNP) to help the model acquire legal numerical sense. Specifically, MNP first replaces the numerical anchors in the selected judicial knowledge A^* by JKS module with [MASK] token as input. Then it predicts the masked numerical anchors with the other non-masked tokens around them. For example,

Input: [CLS] Fraudulent public and private property amounted to RMB [MASK] will result in a six-month prison,

Output: 10,000.

Different from the existing numerical prediction, Spithourakis et al. Spithourakis and Riedel (2018) used median absolute percentage error as loss function, which is commonly applied to evaluate regression models. Since the numerals are strictly required to be accurate in judicial knowledge, we treat MNP as a classification task. We use a *softmax* function to generate a probability distribution over the vocabulary of numeral tokens.³ Similar to Eq. 1, another **RoBERTa** is used to learn the embedding of [CLS] \mathbf{u}^A and all tokens' representations $\bar{\mathbf{A}}$ in A^* :

$$\mathbf{u}^A, \bar{\mathbf{A}} = \text{RoBERTa}([\text{CLS}]; A^*). \tag{3}$$

The training objective of MNP is denoted as:

³ Here the numeral vocabulary refers to all numerical anchors that appear in a same judicial knowledge.

$$\mathcal{L}_{\text{MNP}} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^{n^A} \sum_{k=1}^{n^V} y_{i,j}^k \cdot \log \hat{y}_{i,j}^k, \quad (4)$$

where n^A denotes the number of numerals in the i^{th} judicial knowledge, and n^V denotes the size of the numeral vocabulary. $y_{i,j}^k$ and $\hat{y}_{i,j}^k$ denote the groundtruth and predicted probability of numerical anchor with k , respectively, corresponding to the j^{th} [MASK] token in the i^{th} judicial knowledge.

3.2.4 Reasoning module

We first construct a heterogeneous directed numerical graph $\mathcal{G} = \{\mathbf{V}, \mathbf{E}\}$, whose nodes in \mathbf{V} contain numerals from the fact description X and selected judicial knowledge A^* , and the edges in \mathbf{E} are used to represent the comparison and magnitude relationships between the nodes. The detailed construction is described in Sect. 4.3.

Based on the given fact description X , selected judicial knowledge A^* and numerical graph \mathcal{G} , we design a **M**agnitude-aware numerical reasoning **N**etwork, called **MagNet**. MagNet can be formally defined as follows:

$$\begin{aligned} \mathbf{M}^X &= \mathbf{W}^M \bar{\mathbf{X}}, \\ \mathbf{M}^A &= \mathbf{W}^M \bar{\mathbf{A}}, \\ \mathbf{U} &= \text{MagNet}(\mathcal{G}; \mathbf{M}^X, \mathbf{M}^A, \mathbf{u}^X, \mathbf{u}^A), \end{aligned} \quad (5)$$

where \mathbf{W}^M denotes the shared parameter matrix, \mathbf{U} is the representation of all nodes in \mathcal{G} , and the detailed definition of MagNet is clarified in Sect. 4.3.

As mentioned above, \mathbf{U} consists of the embeddings of numerals only. Since numerical LJP relies on both numerals and non-numerical tokens, we concatenate \mathbf{U} with \mathbf{M}^X and \mathbf{M}^A to obtain the final output of magnitude-aware semantic representation \mathbf{M}^O :

$$\begin{aligned} \mathbf{M}^{\text{num}} &= \mathbf{U}[\mathbf{I}^X, \mathbf{I}^A], \\ \mathbf{M}^O &= \mathbf{W}^O[\mathbf{M}^{\text{num}}; [\mathbf{M}^X; \mathbf{M}^A]], \end{aligned} \quad (6)$$

where $[\cdot; \cdot]$ denotes matrix concatenation, \mathbf{I}^X and \mathbf{I}^A denote the indexes corresponding to the numerals in X and the numerical anchors in A^* , respectively, and \mathbf{W}^O indicates the learnable parameter matrix.

3.2.5 Prediction module

Finally, we transform \mathbf{M}^O using a full-connected layer, and then use a *softmax* function to calculate the predicted probability distribution over the Y^I or Y^P . For the penalty prediction task, due to the large variation across intervals (refer to Sect. 5.1.1 for detailed settings), we follow the previous study setting and use cross-entropy loss:

$$\mathcal{L}^P = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^{n^P} y_{ij}^P \cdot \log \hat{y}_{ij}^P, \quad (7)$$

where y_{ij}^P and \hat{y}_{ij}^P denote the groundtruth and the prediction probability of i^{th} sample with the term of penalty j , respectively. According to the criteria given by legal experts, the terms of imprisonment can be divided into three main categories, i.e., life imprisonment, death, and less than 25 years (300 months). We further subdivide the “less than 25 years” into 301 categories, i.e., $\{0, \dots, 300\}$. The cross-entropy loss treats all categories equally except for the golden label. However, intuitively, the closer the prediction is to the groundtruth, the smaller its loss should be, i.e., continuity of imprisonment Niu et al. (2016). For instance, if the groundtruth of imprisonment is 23 months, a prediction of 22 months is clearly better than 8 months. Therefore, we propose a novel loss function:

$$\begin{aligned} \mathcal{L}^I = & -\frac{1}{N} \sum_{i=1}^N \{ \overbrace{y_i^\ell \cdot \log \hat{y}_i^\ell}^{\text{life imprisonment}} + \overbrace{y_i^d \cdot \log \hat{y}_i^d}^{\text{death}} \\ & + \underbrace{\sum_{k=0}^{300} y_{i,k}^l \cdot \log \hat{y}_{i,k}^l [\log(v_{i,k}^l) - \log(\hat{v}_{i,k}^l)]^2}_{\text{less than 25 years (300 months)}} \}, \end{aligned} \quad (8)$$

where y_i^ℓ and y_i^d denote the labels corresponding to life imprisonment and death penalty, respectively. $y_{i,k}^l$ and $\hat{y}_{i,k}^l$ denote the groundtruth and the prediction probability with the term of imprisonment j , and $v_{i,j}^l$ and $\hat{v}_{i,j}^l$ denote the magnitude of the groundtruth and the predicted numeral, respectively. Geva et al. (2020) suggested that the logarithmic difference gives more weight to the errors of smaller numerals, which are considered to be more severe than larger ones. The total training objective \mathcal{L}_{total} is as follows:

$$\mathcal{L}_{total} = \gamma \mathcal{L}_{JKS} + (1 - \gamma) \mathcal{L}_{MNP} + \mathcal{L}^I + \mathcal{L}^P. \quad (9)$$

where γ is hyperparameters.

3.3 Magnitude-aware numerical reasoning

3.3.1 Graph construction

As described in Sect. 1, numerical anchors in selected judicial knowledge A^* are crucial references for the numerical LJP task. Therefore, we connect numerical anchors with the numerals in the fact description X for reasoning. Moreover, the graph used in this paper contains the relationships of comparison and magnitude between these numerals.

Specifically, let $\mathcal{G} = \{\mathbf{V}, \mathbf{E}\}$ denote the numerical graph, where \mathbf{V} contains numerals from X and A^* , denoted by \mathbf{V}^X and \mathbf{V}^A , respectively, and the value corresponding

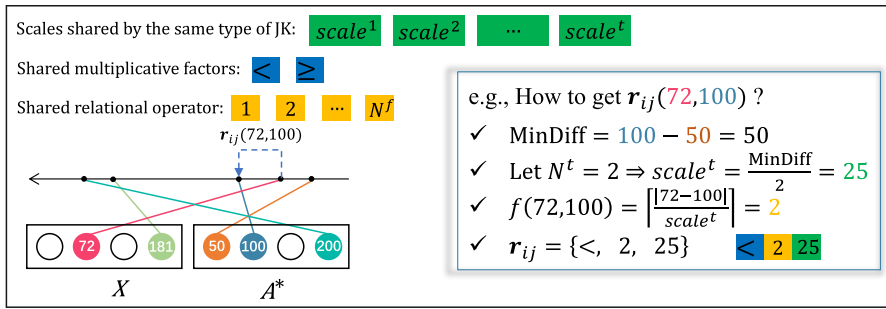


Fig. 5 A toy example illustrating the construction of the relation between two numerals (72, 100). In the data preprocessing stage, for judicial knowledge A^t , its $scale^t$ is calculated in advance

to a node $v \in \mathbf{V}$ is denoted as $n(v)$. \mathbf{E} denotes the relation r between any two numerals. Next, we will demonstrate the construction procedure of r in detail.

For two nodes $v_i, v_j \in \mathbf{V}$, if $n(v_i) < n(v_j)$ then there is a directed edge $\vec{e}_{ij} = (v_i, v_j)$ with the relation r_{ij} . r_{ij} is composed of two parts, relational operator (REL) “ $<$ ” and the magnitude relation MAG, i.e., $r_{ij} = “\leq” + “MAG”$. Conversely, there is a directed edge $\vec{e}_{ji} = (v_j, v_i)$ with the relation $r_{ji} = “\geq” + “MAG”$. That is, the relation between two nodes can be defined as REL + MAG, and $MAG(\vec{e}_{ij}) = MAG(\vec{e}_{ji})$. Next, we will elaborate on the calculation of MAG, refer to the example in Fig. 5.

For a judicial knowledge $A^t \in \mathcal{A}$, the calculation of MAG^t is divided into four steps:

1. For the numerical anchor $v_i^A, v_j^A \in \mathbf{V}^A$, we calculate the absolute minimum difference between the anchors, denoted as $MinDiff(v_i^A, v_j^A)$;
2. Calculating the scale of A^t as $scale^t = \frac{MinDiff(v_i^A, v_j^A)}{N^t}$, where N^t is a hyperparameter for interval division. We will explain the principle of setting N^t as follows. For a judicial knowledge A^t , we compute the $scale^t$ by

$$scale^t = \frac{MinDiff(v_i^A, v_j^A)}{N^t}.$$

We sort the numerical anchors in a judicial knowledge and obtain their median m^t . N^t meets the following requirements,

$$[*] \frac{m^t}{scale^t} \leq f_{max}, \quad (10)$$

where f_{max} is the maximum of multiplicative factor f . In practice we found that the larger N^t , the smaller $scale^t$, leading to higher accuracy and lower recall of the model, and vice versa.

3. Calculating the multiplicative factor for node pair (v_i, v_j) by the ceiling function $f = [*] \frac{|n(v_i) - n(v_j)|}{scale^t}$. All judicial knowledge share the same list of multiplicative factors, $f \in \{1, \dots, N^f\}$, where N^f is empirically set to 100;

4. The MAG^t of (v_i, v_j) is obtained by concatenating the multiplicative factors f and scale^t , i.e., $\text{MAG}^t = "f" + "scale"$.

There are several significant advantages of this graph construction. First, we can discriminate the difference in the interval division of diverse judicial knowledge with scale^t . Second, the shared multiplicative factor can detect the magnitude relationship between numerals, which facilitates the model to capture the similarity of numerals within the same interval. More crucially, we transform learning an exhaustive numeral representations in criminal facts into learning a fixed number of symbols and numeral representations in judicial knowledge. Thus, the proposed model can tackle the challenges posed by unseen numerals and few-shot learning.

3.3.2 MagNet

Based on a self-attention layer Vaswani et al. (2017), we perform numerical reasoning over $\mathcal{G} = \{\mathbf{V}, \mathbf{E}\}$, corresponding to the magnitude-aware numerical reasoning network MagNet in Eq. 5. And the semantic representations of fact description X and selected judicial knowledge A^* , i.e., \mathbf{u}^X and \mathbf{u}^A , as conditions for reasoning. The detailed numerical reasoning is as follows:

Initialization For each numeral node $v_i^X \in \mathbf{V}^X$, its initial embedding $\mathbf{v}_i^X = \mathbf{M}^X[\mathbf{I}^X(v_i^X)]$, where $\mathbf{I}^X(v_i^X)$ denotes the token index of v_i^X in fact description X . The initialization of each numerical anchor $v_i^A \in \mathbf{V}^A$, \mathbf{v}_i^A can be calculated in an analogous manner. As a result, the initialization of all nodes can be denoted as $\mathbf{v}^0 = \{\mathbf{v}_i^X\} \cup \{\mathbf{v}_i^A\}$

Reasoning Conditioned on \mathbf{u}^X and \mathbf{u}^A , the embedding of each node is transformed for indicating the query, key and value at each reasoning step $t \in \{1, \dots, T\}$:

$$\begin{aligned} \mathbf{u}^C &= \mathbf{W}^C[\mathbf{u}^X; \mathbf{u}^A], \\ \mathcal{Q}_i^t &= \mathbf{W}^Q[\mathbf{v}_i^t : \mathbf{v}_i^0] \odot \mathbf{W}_C^Q \mathbf{u}^C, \\ \mathcal{K}_i^t &= \mathbf{W}^K[\mathbf{v}_i^t : \mathbf{v}_i^0] \odot \mathbf{W}_C^K \mathbf{u}^C, \\ \mathcal{V}_i^t &= \mathbf{W}^V[\mathbf{v}_i^t : \mathbf{v}_i^0] \odot \mathbf{W}_C^V \mathbf{u}^C, \end{aligned} \quad (11)$$

where \odot denotes the element-wise multiplication, all of \mathbf{W} denote learnable parameter matrices, same as below.

Then we compute the attention for the edge (v_i, v_j) as $\alpha_{i,j}^t = \sigma(\mathbf{W}^E[\mathcal{Q}_i^t; \mathcal{K}_j^t])$, where σ is the *sigmoid* function. At the t step, each node aggregates message from its neighbors. To better learn the comparison relationship between numerals, each node v_i aggregates only the edges pointing to itself, i.e., (v_j, v_i) . Finally, the information about v_i and its neighbors is integrated for updating representation \mathbf{v}_i^{t+1} :

$$\mathbf{v}_i^{t+1} = \frac{1}{|\mathcal{N}_i|} \left(\sum_{j \in \mathcal{N}_i} \alpha_{i,j}^t \mathbf{M}^R \mathcal{V}_j^t \right) + \mathbf{W}^V \mathbf{v}_i^t, \quad (12)$$

where $\mathcal{N}_i = \{j \mid (v_j, v_i) \in \mathbf{E}\}$ denotes the neighbors of node v_i , \mathbf{M}^R indicates embedding matrix including the relation vector \mathbf{r}_{ij} corresponding to edge (v_i, v_j) . Each \mathbf{r}_{ij} is the concatenation of three parts, i.e., relational operator vector \mathbf{e}^{REL} , multiplicative factor vector \mathbf{e}^f and scale vector \mathbf{e}^{scale} .

4 Experiments

In this section, we conduct comprehensive experiments on three publicly available benchmark datasets to investigate the effectiveness of our proposed NumLJP.

4.1 Experimental setting

4.1.1 Datasets and preprocessing

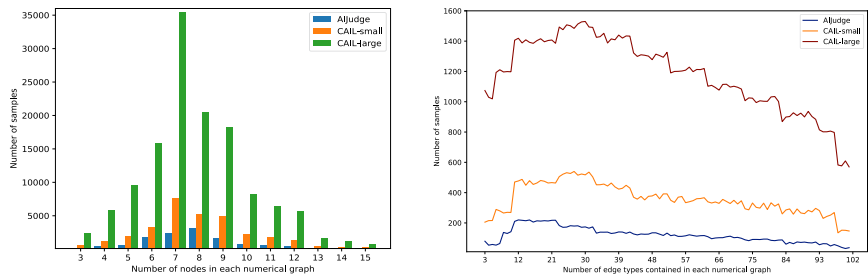
The datasets we use to conduct our experiments are released in two Chinese legal AI challenges, **CAIL2018**⁴ Xiao et al. (2018) and **AIJudge**.⁵ CAIL2018 contains two sub-datasets, CAIL-small and CAIL-large. We can obtain a four-tuple from each sample in CAIL2018, i.e., {fact description, law articles, *term of imprisonment*, *term of penalty*}. The difference is that the samples in AIJudge do not contain the term of imprisonment. Note that, the cases involving one criminal type are maintained. The cases with multiple criminal types will be reserved for future work. The data preprocessing pipeline as follows:

1. filtering out invalid samples, such as missing labels, or less than 20 words in the fact description;
2. obtaining the judicial knowledge **JK** label according to the law article;
3. converting the numerals in the fact description and judicial knowledge to Arabic numerals.

Furthermore, to avoid the negative influence of numerals irrelevant to the judgment, such as date, and time, we filter them in the preprocessing stage by the units of numerals. Specifically, we selectively keep the numerals in the criminal fact description by the units of the numerical anchors in the judicial knowledge. The frequent numerical units can be referred to Table 1. Table 1 illustrates examples of numerical anchors in judicial knowledge. Each example contains criminal charges and their corresponding numerical anchors. Each group of numerical anchors has its units and descriptions. We construct numerical graphs from each processed case and judicial knowledge according to the procedure in Sect. 4.3.1. We make statistics on the nodes and edges' types in each numerical graph, as shown in Fig. 6. Figure 6a illustrates the distribution of the nodes in each numerical graph. We can observe that

⁴ <https://github.com/thunlp/CAIL>.

⁵ <https://www.datafountain.cn/competitions/277>.



(a) Distribution of nodes in numerical graphs constructed from different datasets. **(b)** Distribution of edges' types in numerical graphs constructed from different datasets.

Fig. 6 Statistics on the nodes and edges' types in each numerical graph

most of the numerical graphs contain between 6 and 9 nodes in all three datasets. Figure 6b depicts the distribution of edges' types. The distributions of edges' types in CAIL-small and CAIL-large are similar, with most numerical graphs containing between 10 and 40 edge types. The majority of numerical graphs in AIJudge contain between 10 and 32 different edge types. Note that the edge types here denote the combination of *multiplication factors* and *scale* (refer to Sect. 4.3.1).

According to the suggestions of legal experts,⁶ the terms of penalty can be divided into 11 intervals, i.e., $[0, 1,000)$, $[1,000, 2,000)$, $[2,000, 3,000)$, $[3,000, 4,000)$, $[4,000, 5,000)$, $[5,000, 10,000)$, $[10,000, 50,000)$, $[50,000, 200,000)$, $[200,000, 500,000)$, $[500,000, 1,000,000)$, $[1,000,000, +\infty)$. We can see that there is a huge gap between the numbers in the different intervals.

4.1.2 Baselines

We choose publicly available and competitive methods as our baselines:

- **TOPJUDGE** Zhong et al. (2018) is a topological logic multi-task learning model that formalizes the dependencies between subtasks of LJP as a directed acyclic graph for prediction.
- **MPBFN** is a multi-perspective bi-feedback network based on the topology structure among subtasks, which uses attention to integrate word features from fact descriptions for distinguishing confusing cases Yang et al. (2019).
- **CPTP** is a charge-based prison term prediction model that employs a deep gating network to select fine-grained features for a specific charge Chen et al. (2018).
- **LADAN** is a graph-based representational learning framework that mines similarities between fact description and statutes as well as distinctions between confusing statutes Xu et al. (2020).
- **NeurJudge** is a circumstance-aware LJP framework, which combines the intermediate outputs of subtasks and incorporates the semantics of labels into fact embedding to learn informative representations of confusing cases Yue et al. (2021).

⁶ Existing Chinese LJP datasets are usually divided in this manner.

Table 1 Examples of criminal charges and their corresponding numerical anchors

Criminal charge	Numerical anchor	Unit	Description
Embezzlement	3, 20, 150, 300	×10,000 RMB	Amount of embezzlement
Misappropriation of public funds	3, 5, 50, 100, 200, 300, 500	×10,000 RMB	Amount of misappropriation
Active bribery	3, 20, 50, 100, 250, 500	×10,000 RMB	Amount of bribes
Smuggling, trafficking, transportation, manufacturing of drugs	10, 50	Gram	Drug quantity
Illegal possession of drugs	10, 50, 200	Gram	Drug quantity
Illegal cultivation of drug plants	500, 3,000	Plant	Amount of drug plants
Production and sale of counterfeit and shoddy products	5, 20, 50, 200	×10,000 RMB	Amount of sales
Infringement of trade secrets	30, 250	×10,000 RMB	Amount of loss or gain of the right holder
Illegal business	5, 10, 25, 50, 100, 250, 500	×10,000 RMB	Amount of sales
Credit card fraud	5, 50, 500	×10,000 RMB	Amount of fraud
Obstruction of testimony	10, 100	×10,000 RMB	Illegal possession of property
Dangerous driving	80, 200	mg/100 ml	Alcohol concentration in blood
Robbery	3, 8, 20, 40	×10,000 RMB	Illegal gains of robbery
Theft	3, 10, 50	×10,000 RMB	Illegal gains of theft
Rape	2, 3	People	Number of rapes
Organizing prostitution	3, 5, 10	People	Number of prostitutes
Soliciting, tolerating and introducing prostitution	2, 3, 5, 10	People	Number of prostitutes

- **NumNet** utilizes a numerically-aware graph neural network (GNN) to consider the comparing relationships and performs numerical reasoning over numerals in the context Ran et al. (2019). For a fair comparison, we replace the NumNet's encoder with **RoBERTa** Liu et al. (2019),⁷ and further pre-trained it on legal texts.

For all baselines, we employ their released models and train them using the parameters provided in the original paper.

4.1.3 Evaluation metrics

We employ accuracy (**Acc.**), macro-precision (**MP**), macro-recall (**MR**) and macro-F1 (**F1**) as metrics, which are widely used to evaluate text classification tasks. In addition, following literature Xiao et al. (2018), the metric **ImpScore** is used to measure the difference between the predicted imprisonment I_p and groundtruth I_g , as follows,

$$h = |\log(I_p + 1) - \log(I_g) + 1|,$$

$$\text{ImpScore} = \begin{cases} 1, & h \leq 0.2, \\ 0.8, & 0.2 < h \leq 0.4, \\ 0.6, & 0.4 < h \leq 0.6, \\ 0.4, & 0.6 < h \leq 0.8, \\ 0.2, & 0.8 < h \leq 1, \\ 0, & \text{other.} \end{cases} \quad (13)$$

4.1.4 Implementation details

We use RoBERTa-base as the backbone encoder with the 768 embedding dimensions and an Adam Kingma and Ba (2015) optimizer with a learning rate $lr = 10^{-5}$. The batch size N is 32. As the learning rate increases, graph-based models suffer from inconsistent training due to the gradient explosion of the text encoder. We set the gradient clipping to a very small maximum gradient norm ($clipping = 0.3$) to solve this issue. We observe that model achieves the best performance with the hyperparameter combination $\tau = 0.3$ and $\lambda = 0.9$. The embedding sizes of relational operator, multiplicative factor and scale are set to 256. We set the hyperparameter $N^t \in \{1, 2, 3, 4\}$, according to the type of judicial knowledge. In general, the larger the value of the numerical anchor, the smaller the N^t . The reasoning step T is set to 4. Hyperparameter γ is set to 0.45, respectively. Other standard parameters follow the default settings of the Pytorch⁸ framework.

⁷ <https://github.com/j30206868/numnet-chinese>.

⁸ <https://pytorch.org>

Table 2 Predicting the terms of penalty

Datasets	CAIL-small				CAIL-large				AIJudge			
	Acc	MP	MR	F1	Acc	MP	MR	F1	Acc	MP	MR	F1
TOPJUDGE	36.85	35.51	32.05	31.78	52.58	44.16	34.41	33.75	29.42	26.12	24.82	24.75
MPBFN	36.08	31.19	29.62	29.85	54.16	41.83	36.01	35.49	29.61	25.43	23.25	23.30
CPTP	39.06	37.80	33.59	32.91	55.50	47.13	35.59	36.89	30.55	26.27	25.04	24.56
LADAN	37.23	34.15	31.28	30.53	54.98	40.56	36.13	36.25	29.94	27.11	25.91	25.08
NeurJudge	41.62	40.41	36.20	36.17	57.01	47.62	39.12	41.05	32.08	30.39	26.83	27.79
NumNet	41.83	39.92	37.67	36.75	58.11	47.58	41.56	42.12	32.67	30.72	28.75	27.91
NumLJP	48.95	45.68	45.59	46.28	68.25	59.13	51.37	54.81	43.57	41.88	38.26	37.35

Table 3 Predicting the terms of imprisonment

Datasets	CAIL-small					CAIL-large				
	Acc	MP	MR	F1	ImpScore	Acc	MP	MR	F1	ImpScore
TOPJUDGE	38.86	38.51	36.78	35.32	58.31	52.58	45.16	41.86	43.36	64.59
MPBFN	39.12	38.18	37.67	36.77	58.26	54.16	44.87	42.32	42.45	63.20
CPTP	45.14	43.29	42.59	41.52	64.55	55.50	54.39	51.78	50.84	71.01
LADAN	42.30	41.16	39.08	38.55	62.19	54.98	51.55	48.23	47.75	68.73
NeurJudge	43.25	43.60	41.25	40.37	63.03	57.01	54.92	51.10	50.35	70.26
NumNet	44.79	42.82	42.92	41.89	63.87	58.11	56.58	53.16	52.17	72.83
NumLJP	58.15	56.30	55.12	54.36	68.05	69.38	66.06	63.86	62.41	82.37

4.2 Results and discussion

4.2.1 Main results

Tables 2 and 3 report the results of predicting the penalty terms and predicting the imprisonment terms, respectively. We can observe that our NumLJP is significantly outperforms all baselines, achieving state-of-the-art performance, as can be seen by the bolding of the optimal model performance in the tables. Specifically,

1. The performances of the multi-task learning-based models, such as TOPJUDGE, MPBFN and NeurJudge, on the numerical LJP are unsatisfactory. NumNet's MP is not always higher than other methods, but MR is consistently ahead over the small-scale datasets AIJudge and CAIL-small, which makes it outperform other

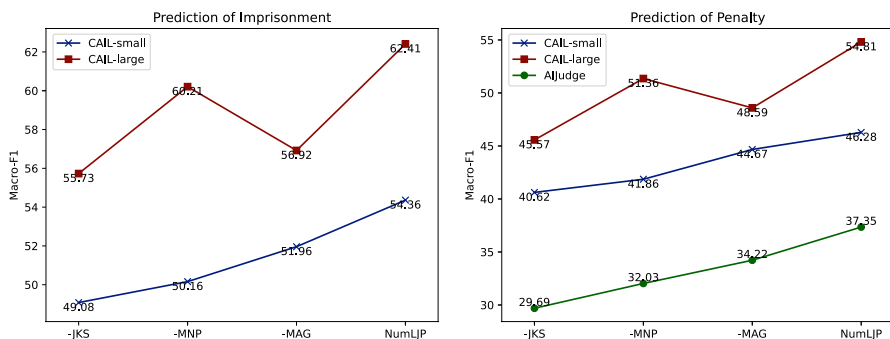


Fig. 7 Ablation analysis

baselines overall. Empirically, the PLM has a unique encoding manner⁹ for the unseen numerals instead of directly replacing them with [UNK]. This suggests that numerals are not negligible in numerical LJP. NumLJP improves macro-F1 by at least 12.84% and 9.53% on both tasks, respectively, compared to all baselines. This fully illustrates the significance of numerical reasoning on the numerical LJP tasks.

2. The advantages of the PLM-based approaches, i.e., NumLJP and NumNet, are more evident on large-scale datasets CAIL-large. NumLJP outperforms NumNet on both two tasks, which demonstrates the effectiveness of our proposed multiple modules for understanding and reasoning numerals on the numerical LJP tasks, such as JKS and MagNet, etc.
3. For predicting the penalty terms, the overall performances of the baselines are poor due to the wide gap in interval delineation of penalty terms. Since NumLJP introduces numerical anchors, which are equivalent to interpolation in these intervals. Rather than limiting to learning numerical representations in the criminal fact, we focus on numerical anchors in judicial knowledge. These numerical anchors are fixedly present in the judgment prediction for different cases, effectively alleviating the struggles posed by unseen numerals. The results illustrate that the performances of NumLJP are much higher than the baselines.
4. The exact matching results on predicting the imprisonment terms are similar to that of task penalty term prediction. Since ImpScore does not measure the exact matching error, benefiting from Huber Loss Huber (1992), the CPTP slightly outperforms NumNet on small-scale datasets. The ImpScore of NumLJP reaches 82.37, nearly 10 points higher than the sub-optimal method. Furthermore, we observed the examples that fail to match exactly by the NumLJP and found that predicted results are closer to the groundtruth than baselines. It indicates that the designed MAG learns the magnitude between the numerals.

⁹ PLMs utilize called **WordPiece** tokenizer to split words either into the full forms or into word pieces Devlin et al. (2019).

Case 1: In May 2008, Sun repeatedly violated traffic regulations without obtaining a driver's license. At noon, he drank heavily and drove his car to the intersection of Chenlong Road, then he crashed into the rear of another car. After the accident, Sun continued to drive over the speed limit, causing the death of 1 people. It was identified that the driving speed of the car was 138 km/h at the moment before the collision; the ethanol content in the blood at the time of the crime was 135mg/100 ml. GT : Penalty: RMB 120,000 (level: 8) Imprisonment: 296months NumLJP: level: 6 Imprisonment: 266months
Case 2: In July 2010, defendant Zhang sold a total of 30g of methamphetamine (one of narcotic drugs) to Gu. In the early morning, Zhang was caught red-handed with 73.9g of methamphetamine when he tried to sell it to Gu. Subsequently, police seized 93.3g of methamphetamine and 8.8g of caffeine at Zhang's residence. GT : Penalty: RMB 3,000 (level: 4) Imprisonment: 108 months NumLJP: level: 4 Imprisonment: 109months
Case 3: On January 24, Li came to the parking lot located in Hongqiao District to take Wang's bus away. At about 20:00, Li drove the car collided with a roadside utility pole, resulting in damage to the vehicle. Li abandoned the bus, and then fled. By the price certification center assessment, the vehicle damage value of RMB 12,433. On February 15, Li was arrested. GT : Penalty: RMB 15,000 (level: 6) Imprisonment: 24 months NumLJP: level: 6 Imprisonment: 23months

Fig. 8 Case Study. Those results labeled in green are the groundtruth, and those marked in blue are the predicted results of our model. The gray token indicates that the model pays more attention to it

4.2.2 Ablation analysis

To further investigate the effects of the different modules, we perform ablation studies on the three datasets, and the results (Macro-F1) are shown in Fig. 7. Specifically, **-JKS**, **-MNP**, and **-MAG** denote replacing the JKS module with direct fine-tuning via the [CLS] embedding, removing the MNP module, and removing the MAG from the graph, respectively. We can observe that:

1. **Impact of JKS** There is a significant decline in performance after replacing JKS, thus illustrating the superiority of contrastive learning. And judicial knowledge is crucial to the final judgment. We have also suggested in the Error Analysis section 5.2.4 that choosing the wrong judicial knowledge leads to incorrect predictions. Intuitively, if judge make an incorrect discrimination of a defendant's behavior, their decision is usually wrong as well.
2. **Impact of MNP and MAG** We analyze the results in combination with -MNP and -MAG. When the data volume is small, there is a dramatic decrease in the performance of -MNP. As the data volume increases, the impact of -MNP becomes decreasing. While -MAG performs the opposite. There are fewer numerals available for numerical reasoning when the data size is small, and numerical anchor becomes a shared reference frame that provides a bridge for comparing different numerals. Since the numerical graph contains many relations with MAG, a limited number of numerals cannot sufficiently learn the information embedded by the different relations. Increasing numerals are used for reasoning with which gradually weakens the effect of the numerical anchors. To this end, the large number of numerals allows MagNet to learn the comparative relationships and perceive the magnitude between them through a fine-grained scale. Intuitively, the more numerals on the number axis, the more explicit the comparative relationships and distances between them are.

4.2.3 Case study

In Fig. 8, we selected three typical cases for further analysis. Except for NumNet and NumLJP, other baselines replace the numerals with [UNK], causing them to get incorrect results.

In case 1, the penalty results predicted by NumLJP differ significantly from the gold results. The main reason is that NumLJP does not identify the numeral of deaths, i.e., the “4” marked in red. Nevertheless, the result of the imprisonment prediction is in an acceptable range ($\pm 15\%$ of the golden consequence). We revisited the model’s attention to the different tokens and found that the model paid much more attention to the “138”, “135”, “death”, “speed limit,” and “ethanol” than to the other words. This suggests that the lack of numerical understanding can lead to erroneous results, and that numerical reasoning requires a combination of text and numerals. In other words, selecting the proper judicial knowledge is indispensable for performing numerical reasoning effectively. Furthermore, it illustrates the significance of introducing types in numerical reasoning. Comparing 138 (car speed) with 135 (blood alcohol concentration) can make the reference system collapse.

The NumLJP almost exactly predicted the outcome of cases 2 and 3. Typically, the longer the imprisonment period, the more difficult it is to predict accurately. The multiple numerals in case 2 significantly improve the model’s ability to perform numerical reasoning. However, many numerals can also have a negative impact on the model, as we detailed in the error analysis 5.2.4.

Although only one numeral in the fact description of case 3, NumLJP still performs well. We believe that this is since NumLJP uses numerals from judicial knowledge as references. And even though there are fewer numerals in the criminal facts, there are still enough reference points for comparison.

To summarize, the NumLJP focuses its attention on the key tokens and numerals. The above observations illustrate that: (1) Numerals are important in numerical LJP. (2) Magnitude relationships between numerals can perceive differences between them. (3) Numerals have different meanings in judicial knowledge, further demonstrating the significance of constructing different reference frames. (4) And the MNP module helps the model learn the numerical commonsense in judicial knowledge.

4.2.4 Error analysis

Error analysis is the process of isolating, observing, and diagnosing erroneous predictions, thereby supporting understanding of the strengths and weaknesses of the model. Therefore, we revisited the samples of prediction errors in the test set (imprisonment errors of more than 15% and no exact matches for penalty) and performed an error analysis. We derived six main types of errors, and their distributions and examples are shown in Table 4. The detailed analysis is as follows.

1. Among the error cases, the most frequent cause is **selecting the wrong judicial knowledge**. Although pre-trained language models achieve impressive performance on classification tasks, they cannot entirely avoid wrong predictions. In the example, our model selects the judicial knowledge corresponding to the rob-

Table 4 Examples in Error Analysis

Error type	Ratio%	Examples
Wrong judicial knowledge	26.3	On January 8, 2009, defendant Yu met and beat the victim Xu in Anlian Town Walking East Street. Then, Yu forcibly took Xu to his home and did not allow Xu to leave while seizing Xu for RMB 20,000
Multi-type numerals	14.5	On October 21, 2012, the defendant Deng drove a motorcycle from north to south from Jiaokeng Village after drinking. When he drove to the left turn section, the motorcycle drove into the right side of the road outside, causing Wang to be injured. Deng was fully responsible for the accident. After identification, Deng's blood alcohol content was 9.27mg/ml, belonging to the state of intoxication. The motorcycle was traveling at a speed of 142km/h
Duplicate numerals	21.2	On May 18, 2011, Chen lied that he could apply for public housing and cheated Zhang and Tang of RMB 18,000 for each, a total of RMB 36,000. on February 12, 2013, Chen lied that he could apply for social security and cheated Xu of RMB 13,000 in cash
Excessive numerals	13.8	On June 3, 2013, defendant Li drove to Bazhong City and unlocked the door of the victim Chen. Li stole one iPhone, three cameras, seven lenses, and one MacBook from Chen. Identification by the certification center: the value of the iPhone was RMB 6,858; the value of the MacBook was RMB 9,378, and the values of the three cameras were RMB 5,632, 8,812, and 21,023, respectively. The value of the seven lenses were RMB 1,300, 6,800, 1,800, 3,600, 8,800, 12,100, and 15,200
Oversized numerals	6.6	During the period from October 2015 to May 2017, defendant Yang misappropriated contracted construction funds several times with the help of his role as the company's accountant. The total amount of Yang's misappropriation reached RMB 395,000.000
Implicit numerals	17.6	On June 7, 2011, defendant Huang argued with his father and mother at his home in Nangang Township because he suspected that his parents had given money and belongings to his sister. Huang hit his father on the head with an iron hoe during the argument and then hit his mother. Huang burned his parents' bodies with diesel and straw and then fled to Shuyang

bery with the information in red. However, the groundtruth focuses more on the information corresponding to kidnapping in blue. These confusing crimes are not easy to discriminate against, even for human judges. We further statistically concluded that the final judgment error rate is 98.35% if the judicial knowledge is selected incorrectly; it is evident that judicial knowledge is the cornerstone of all judgment predictions.

2. Although each criminal charge has a unique corresponding judicial knowledge, **some judicial knowledge may cover many different types of numerals**, as shown by the numerals in red in the second example. This is because judicial knowledge covers multiple types of behaviors. For instance, dangerous driving includes over-speeding, drunken driving, overloading, etc. Identifying these behaviors requires considering different types of numerals, such as speed, blood alcohol concentration, and the number of certified passengers. The model in this paper only designed for the numerical LJP in simple cases. We believe that to essentially perform numerical reasoning over different types of numerals requires learning the relationships between them with the help of multi-graph interactions. And we will carry out this research in future work.
3. Duplicate numerals refer to the repeated descriptions of numerals in the criminal fact description. As shown in the third example, 18,000 in red is already included in 36,000 in blue. We found through many cases that the model seems to have acquired the addition operation on a small number of numerals. Duplicate numerals can cause the model to yield a greater result than the real one. We are planning to filter out these numerals in the pre-processing data stage.
4. Excessive numerals indicate that some cases involve a lot of numerals, such as the numerals in the fourth example. Although we found that the model learned the additive ability, the results of dealing with a large number of numerals are often hard to satisfy. We believe that when there are many numerals in the fact description, the sum of these numerals is usually significant. The model focuses on learning the numerals in the text and ignores their totality. Excessive numerals lead to predicted judgments that are often lower than the real ones.
5. Oversized numerals are the huge numerals in the fact description, such as the 395,000,000 in the fifth example. Using the misappropriation of public funds as an example, we found by comparing all the numerals involved in this type of case that 395,000,000 is approximately 600 times larger than the second-largest one. In addition, 395,000,000 is over 1,970 times the largest numeral in the judicial knowledge of misappropriation. Although our method can measure the magnitude of different numerals by setting multipliers and typed scales, some numerals are far beyond the range.
6. Implicit numerals are not explicitly described, as shown in the sixth example. **“His parents’ bodies”** implies that the defendant killed two people. There is no well-defined scheme to detect and resolve this type of error.

Table 5 Types and examples of variant data. Where **Original text** denotes the original criminal fact description and the blue font is the modified content of the manually created variation. The types of variations include **Change values**, **Change objects**, and **Change objects and values**

Original text	On May 21, 2013, defendant Guo went to Duan's home in Anli Village. Guo stole RMB 13,700 of money from the pocket of Duan's cotton sweater, a smartphone, and a gold necklace in the bedroom. After identification, the smartphone was worth RMB 3,426, and the gold necklace was worth RMB 12,000
Change values	On May 21, 2013, defendant Guo went to Duan's home in Anli Village. Guo stole RMB 20,550 of money from the pocket of Duan's cotton sweater, a smartphone, and a gold necklace in the bedroom. After identification, the smartphone was worth RMB 5,139, and the gold necklace was worth RMB 18,000
Change objects	On May 21, 2013, defendant Guo went to Duan's home in Anli Village. Guo stole RMB 13,700 of money from the pocket of Duan's cotton sweater, a computer, and a diamond ring in the bedroom. After identification, the computer was worth RMB 3,426, and the diamond ring was worth RMB 12,000
Change objects and values	On May 21, 2013, defendant Guo went to Duan's home in Anli Village. Guo stole RMB 20,550 of money from the pocket of Duan's cotton sweater, a computer, and a diamond ring in the bedroom. After identification, the computer was worth RMB 5,139, and the diamond ring was worth RMB 18,000

4.2.5 Robustness analysis

Robustness is one of the crucial elements for the model to be available in real scenarios. We have demonstrated the effectiveness of our approach through a series of experiments. However, some researchers have found that even a fragile model can achieve a promising performance, which makes the model's performance on the benchmark dataset potentially misleading Nie et al. (2020), Ribeiro et al. (2020), Patel et al. (2021). Inspired by literature Patel et al. (2021), we attempt to illustrate the robustness by comparing the performance of different models on variant datasets. We sampled 100 seed examples of *Theft* from CAIL2018 and manually constructed a variant set named **VarLJP100**. In order to collect more representative seed samples, the sampling principle is to keep the numerals in different cases spread over distinct numerical anchor intervals as much as possible ¹⁰

Table 5 presents an original fact description and a set of variations created manually from it. Each type of variation is described as follows:

1. **Change values** denotes increasing or decreasing each numeral in the criminal fact by 50%. For example, 13,700 can be changed to 20,550 (+50%), or 6,850 (-50%). Note that the modified numeral needs to be greater than the smallest numerical anchor in judicial knowledge. It may be innocent because the theft acquisition is lower than the minimum numerical anchor.

¹⁰ the anchors of *Theft* are 1,000, 3,000, 30,000, 100,000, 300,000, 500,000.

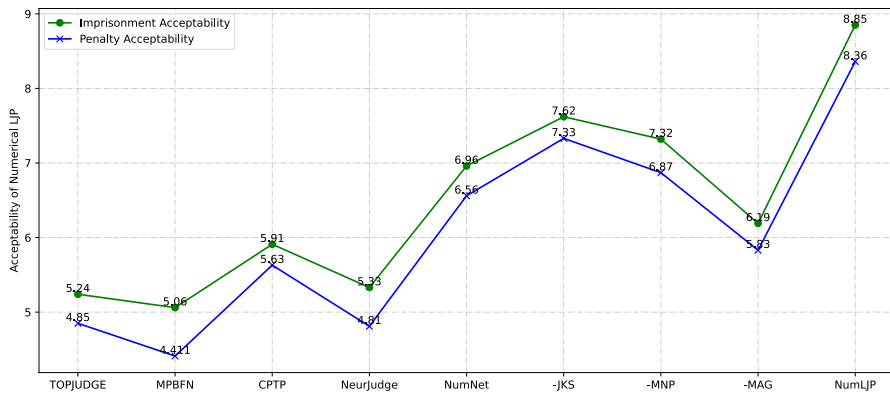


Fig. 9 The average *acceptability* of the model predictions on the numerical LJP

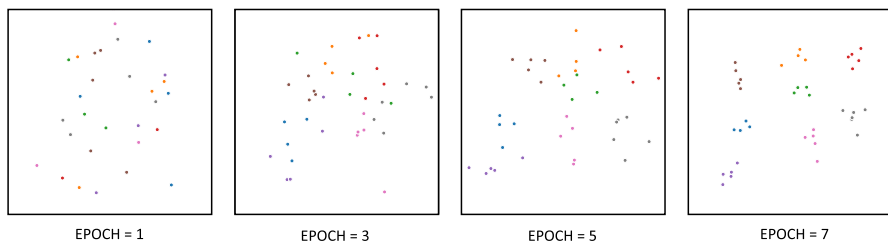


Fig. 10 Dynamic visualization of numerical anchors in judicial knowledge with the training procedure

2. **Change objects** indicates modifying the objects of the stolen property involved in the fact description. The modification principle is that an original object and the modified one are of the same type and comparable value. For example, a *smartphone* can be replaced by a *computer*.
3. **Change objects and values** denotes modifying both the objects and the values.

Based on the above approach, we obtained a total of 400 manual samples. Since there is no groundtruth for the variations, we recruit five legal professional annotators to judge the **acceptability** of the model's predictions manually. Specifically, given the original criminal fact, these annotators were asked to score the model's predictions based on judicial knowledge of variant data. The scoring is performed independently on a Likert scale of 1–10, with 1 being the worst and 10 being the best. To ensure the reliability of the annotators, we conducted a *Fleiss' kappa* test Fleiss (1971) for them using the original data. Specifically, we asked these annotators to predict imprisonment for the original cases manually. The prediction is considered correct if it falls within 15% of the groundtruth, and vice versa. After the annotation is completed, we obtain the confusion matrix for each annotator and

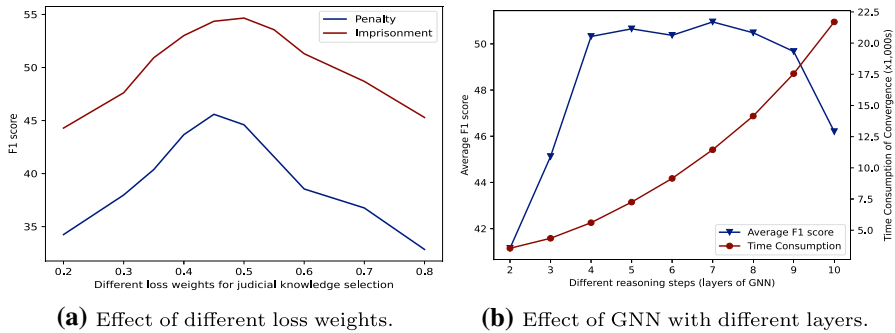


Fig. 11 Hyperparameter sensitivity analysis

calculate the corresponding kappa coefficients with *statsmodels*.¹¹ The annotator who obtained a kappa coefficient greater than 0.8 was selected.

The average *acceptability* of the model predictions on the numerical LJP task is shown in Fig. 9. We found that our full model NumLJP yields the optimal acceptability on the numerical LJP task. The acceptability of imprisonment prediction is higher than penalty prediction due to the uneven distribution of penalty labels. The models with numerical reasoning, such as NumLJP, -JKS, -MNP, and NumNet, outperformed the other baselines, which sufficiently illustrates the significance of numerical reasoning on the numerical LJP. -MAG performs similarly to CPTP and outperforms the other models without numerical reasoning, suggesting that introducing external legal knowledge is essential. We believe that introducing judicial knowledge and acquiring legal numerical commonsense are critical to the success of NumLJP. In particular, MagNet models scales of different judicial knowledge, effectively capturing the distance between numerals in fact description and numerical anchors. On the one hand, NumLJP can avoid learning exponential numerals in fact description. On the other hand, the prediction of NumLJP relies on judicial knowledge, which is consistent with the general observation in practice. It makes the model more robust and interpretable.

4.2.6 Visualization

We selected eight most frequent types of judicial knowledge, and each of them contains about 4–6 numerical anchors. We saved the vectors of corresponding numerical anchors in incremental epochs during the training and visualized these vectors using t-SNE package Van der Maaten and Hinton (2008), as shown in Fig. 10.

Points with the same color indicate numerical anchors of the same type. As the training progresses, the different types of numerical anchors start to move away from each other, while the same kinds of numerical ones gradually cluster together. It confirms that our model can distinguish between distinct types of numerals and numerical reasoning in legal domain requires different reference frames.

¹¹ <https://www.statsmodels.org/>

4.2.7 Sensitivity analysis

The model proposed in this paper involves several hyperparameters. In order to investigate the impact of various hyperparameters on the performance, we perform a sensitivity analysis on the CAIL-small dataset, which facilitates the selection of the optimal parameter combination¹², as shown in Fig. 11c. We highlight the impact of the reasoning step T and the trade-off parameter γ of judicial knowledge selection and masked numeral prediction

Effect of γ . Fig. 11a shows the tendency of the performance on numerical LJP corresponding to a series of γ . We can observe that the model achieves the optimal average performance around $\gamma = 0.45$. From an overall perspective, the model performance tends to increase and then decrease as γ grows. Further investigation reveals that when $\gamma < 0.5$, the main reason for the improvement is the higher contribution of the judicial knowledge selection task, which leads to an increase in accuracy. When $\gamma \geq 0.5$, the recall of those low-frequency criminal charges begins to decrease, as well as the gain in masked numeral prediction is reduced, causing the model performance to decline. This suggests that the significance of judicial knowledge selection and legal numeral commonsense is comparable to numerical LJP.

Effect of T . Fig. 11b demonstrates the effect of different reasoning steps (number of GNN layers) on the model performance and training time, where *Average F1 score* denotes the averaged F1 score of penalty and imprisonment prediction. We can find that when $T < 4$, the model performance rises sharply as the number of reasoning step T increases. The performance starts to fluctuate slightly when $4 \leq T \leq 9$ and drops remarkably when $T > 9$. Intensive analysis reveals that the tendency of the model performance is consistent with the number of nodes in the numerical graph. Intuitively, a GNN with T layers can probe a graph structure of depth T (Hamilton et al. 2017, Guo et al. 2019). That is, if the length of the comparison chain¹³ is K , the number of GNN stacking layers T is ideally close to K . Therefore, we can conclude that the number of reasoning steps depends on the average maximum length of the comparison chain. When T is greater than the maximum length, the model gradually manifests overfitting. Besides, The time consumed for model convergence rises quickly as the number of reasoning steps increases. We employ the ratio of averaged F1 score and time consumption as the criterion for selecting T . Thus we set T equal to 4. The finding can help to efficiently determine the range of optimal T in numerical reasoning tasks.

¹² Among all hyperparameters, the learning rate lr , gradient clipping *clipping*, the weight of contrastive learning loss λ , and the temperature τ are set empirically following previous works, which are not repeated in this paper. N' is the multiplier assigned for interval division, and we detail its setting principle in Section 4.3.1.

¹³ The comparison chain is ordered numerals in a numerical graph.

5 Ethical discussion

Here we will discuss the ethical aspects of the legal judgment prediction. Legal AI techniques have been studied for many years, especially when mass judgments have been published on the Internet. We will discuss the following three parts:

- (1) Data sources. The data used in this work are open access. The original source of these data is the China Judgment Online,¹⁴ a Chinese legal available website for the public.
- (2) Sensitive information. We removed sensitive information, such as name, race, and other information that can be traced to a specific person, for privacy protection.
- (3) Impact on judges. The algorithm is to help judicial staff to be more efficient, not to replace them. At no time should the machine interfere with the judge's independent judgment.

6 Conclusion and future work

We study the numerical LJP tasks in legal practice scenarios, i.e., the prediction of imprisonment and penalty. We focus on comparing and magnitude relationships between the numerals of fact description and judicial knowledge. The core novelty of our approach is that we avoid directly learning endless numerical representations in criminal fact but rather a fixed number of numerical anchors in judicial knowledge. To this end, we proposed a judicial knowledge-enhanced magnitude-aware reasoning architecture, NumLJP, which achieves SOTA performance on both tasks.

We can derive two inspirations from the experimental results:

- (1) The comparison and magnitude relationships and how they are modeled are critical to numerical reasoning, which effectively learns the similarity of numerals within the same interval. (2) For sophisticated numerical reasoning in the domain tasks, constructing an appropriate reference frame using reasonable numerals and understanding it could be an effective solution.
- (2) In the future, the numerical LJP still has many directions to explore. For example, complex criminal cases indicate a case involving multiple defendants and multiple facts, which is common in practice scenarios. Complex criminal cases need to consider the complicated coreference and the relationships among multiple defendants, such as principal, accessory, and abettor. What is more challenging is to correctly match numerous defendants and multiple criminal facts because not everyone is involved in the entire crime.

¹⁴ <https://wenshu.court.gov.cn>.

References

- Amini A, Gabriel S, Lin S, Koncel-Kedziorski R, Choi Y, Hajishirzi H (2019) Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In: NAACL, pp. 2357–2367
- Bakalov A, Fuxman A, Talukdar PP, Chakrabarti S (2011) Scad: Collective discovery of attribute values. In: WWW, pp. 447–456
- Baly R, Karadzhov G, Saleh A, Glass JR, Nakov P (2019) Multi-task ordinal regression for jointly predicting the trustworthiness and the leading political ideology of news media. In: NAACL-HLT, pp. 2109–2116
- Banerjee S, Chakrabarti S, Ramakrishnan G (2009) Learning to rank for quantity consensus queries. In: SIGIR, pp. 243–250
- Bi S, Huang Y, Cheng X, Wang M, Qi G (2019) Building chinese legal hybrid knowledge network. KSEM 11775:628–639
- Bi S, Cheng X, Chen J, Qi G, Wang M, Zhou Y, Wang L (2019) Dispute generation in law documents via joint context and topic attention. In: JIST, pp. 116–129
- Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A et al (2020) Language models are few-shot learners. *Neural Inf Process Syst* 33:1877–1901
- Cao W, Mirjalili V, Raschka S (2020) Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognit Lett* 140:325–331
- Chalkidis I, Androustopoulos I, Aletas N (2019) Neural legal judgment prediction in English. In: ACL, pp. 4317–4323
- Chen H, Cai D, Dai W, Dai Z, Ding Y (2019) Charge-based prison term prediction with deep gating network. In: EMNLP, pp. 6361–6366
- Chen K, Xu W, Cheng X, Xiaochuan Z, Zhang Y, Song L, Wang T, Qi Y, Chu W (2020) Question directed graph attention network for numerical reasoning over text. In: EMNLP, pp. 6759–6768
- Cheng X, Bi S, Qi G, Wang Y (2020) Knowledge-aware method for confusing charge prediction. *NLPCC* 12430:667–679
- Devlin J, Chang M, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL, pp. 4171–4186
- Diaz R, Marathe A (2019) Soft labels for ordinal regression. In: CVPR, pp. 4738–4747
- Dong Q, Niu S (2021) Legal judgment prediction via relational learning. In: SIGIR, pp. 983–992
- Dua D, Wang Y, Dasigi P, Stanovsky G, Singh S, Gardner M (2019) DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In: NAACL, pp. 2368–2378
- Fleiss JL (1971) Measuring nominal scale agreement among many raters. *Psychol Bull* 76(5):378
- Ge J, Huang Y, Shen X, Li C, Hu W (2021) Learning fine-grained fact-article correspondence in legal cases. *TASLP* 29:3694–3706
- George TE, Epstein L (1992) On the nature of supreme court decision making. *APSR* 86(2):323–337
- Geva M, Gupta A, Berant J (2020) Injecting numerical reasoning skills into language models. In: ACL, pp. 946–958
- Gunel B, Du J, Conneau A, Stoyanov V (2021) Supervised contrastive learning for pre-trained language model fine-tuning. In: ICLR
- Guo Z, Zhang Y, Teng Z, Lu W (2019) Densely connected graph convolutional networks for graph-to-sequence learning. *TACL* 7:297–312
- Gutmann M, Hyvärinen A (2010) Noise-contrastive estimation: a new estimation principle for unnormalized statistical models. *AISTATS* 9:297–304
- Hamilton WL, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. In: NeurIPS, pp. 1024–1034
- Hu Z, Li X, Tu C, Liu Z, Sun M (2018) Few-shot charge prediction with discriminative legal attributes. In: COLING, pp. 487–498
- Huang D, Shi S, Lin C, Yin J, Ma W (2016) How well do computers solve math word problems? large-scale dataset construction and evaluation. In: ACL
- Huber PJ (1992) Robust estimation of a location parameter. In: *Breakthroughs in Statistics*, pp. 492–518
- Hénaff OJ (2020) Data-efficient image recognition with contrastive predictive coding. *ICML* 119:4182–4192
- Jaiswal A, Babu AR, Zadeh MZ, Banerjee D, Makedon F (2021) A survey on contrastive self-supervised learning. *Technologies* 9(1):2


- Jiang C, Nian Z, Guo K, Chu S, Zhao Y, Shen L, Tu K (2019) Learning numeral embeddings. arXiv pre-print [arXiv:2001.00003](https://arxiv.org/abs/2001.00003)
- Khosla P, Teterwak P, Wang C, Sarna A, Tian Y, Isola P, Maschinot A, Liu C, Krishnan D (2020) Supervised contrastive learning. *Neural Inf Process Syst*, **33**
- Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. In: ICLR
- Kort F (1957) Predicting supreme court decisions mathematically: a quantitative analysis of the “right to counsel” cases. *APSR* 51(1):1–12
- Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, Stoyanov V, Zettlemoyer L (2020) BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: *ACL*, pp. 7871–7880
- Li S, Zhang H, Ye L, Su S, Guo X, Yu H, Fang B (2020) Prison term prediction on criminal case description with deep learning. *Comput Mater Contin* 62(3):1217–1231
- Lin BY, Lee S, Khanna R, Ren X (2020) Birds have four legs?! numersense: Probing numerical common-sense knowledge of pre-trained language models. In: *EMNLP*, pp. 6862–6868
- Liu YH, Chen YL, Ho WL (2015) Predicting associated statutes for legal problems. *IPM* 51(1):194–211
- Liu C-L, Chang C-T, Ho J-H (2004) Case instance generation and refinement for case-based criminal summary judgments in chinese. *JISE*, 783–800
- Liu CL, Liao TM (2005) Classifying criminal charges in chinese for web-based legal services. In: *APCCMI*
- Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: A robustly optimized BERT pretraining approach. *CoRR* **abs/1907.11692**
- Luo B, Feng Y, Xu J, Zhang X, Zhao D (2017) Learning to predict charges for criminal cases with legal basis. In: *EMNLP*, pp. 2727–2736
- Nie Y, Williams A, Dinan E, Bansal M, Weston J, Kiela D (2020) Adversarial NLI: A new benchmark for natural language understanding. In: *ACL*, pp. 4885–4901
- Niu Z, Zhou M, Wang L, Gao X, Hua G (2016) Ordinal regression with multiple output CNN for age estimation. In: *CVPR*, pp. 4920–4928
- Parikh N, Boyd SP (2014) Proximal algorithms. *Found. Trends Optim.* 1(3):127–239
- Patel A, Bhattamishra S, Goyal N (2021) Are NLP models really able to solve simple math word problems? In: *NAACL*, pp. 2080–2094
- Qin J, Lin L, Liang X, Zhang R, Lin L (2020) Semantically-aligned universal tree-structured solver for math word problems. In: *EMNLP*, pp. 3780–3789
- Ran Q, Lin Y, Li P, Zhou J, Liu Z (2019) Numnet: Machine reading comprehension with numerical reasoning. In: *EMNLP*, pp. 2474–2484
- Ribeiro MT, Wu T, Guestrin C, Singh S (2020) Beyond accuracy: Behavioral testing of NLP models with checklist. In: *ACL*, pp. 4902–4912
- Robinson J.D, Chuang C, Sra S, Jegelka S (2021) Contrastive learning with hard negative samples. In: *ICLR*
- Saha A, Joty SR, Hoi SCH (2021) Weakly supervised neuro-symbolic module networks for numerical reasoning. *CoRR* **abs/2101.11802**
- Sanh V, Debut L, Chaumond J, Wolf T (2019) Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR* **abs/1910.01108**
- Segal JA (1984) Predicting supreme court cases probabilistically: The search and seizure cases, 1962–1981. *APSA* 78
- Sermanet P, Lynch C, Chebotar Y, Hsu J, Jang E, Schaal S, Levine S (2018) Time-contrastive networks: Self-supervised learning from video. In: *ICRA*, pp. 1134–1141
- Shi X, Cao W, Raschka S (2021) Deep neural networks for rank-consistent ordinal regression based on conditional probabilities. *CoRR* **abs/2111.08851**
- Shorten C, Khoshgoftaar TM, Furht B (2021) Text data augmentation for deep learning. *J Big Data* 8(1):101
- Spithourakis GP, Riedel S (2018) Numeracy for language models: Evaluating and improving their ability to predict numbers. In: *ACL*, pp. 2104–2115
- Thawani A, Pujara J, Ilievski F, Szekely PA (2021) Representing numbers in NLP: a survey and a vision. In: *NAACL*, pp. 644–656
- Tian Y, Krishnan D, Isola P (2020) Contrastive multiview coding. In: *ECCV*, vol. 12356, pp. 776–794. Springer
- Van der Maaten L, Hinton G (2008) Visualizing data using t-sne. *JMLR* 9(11)

- van den Oord A, Li Y, Vinyals O (2018) Representation learning with contrastive predictive coding. *CoRR* **abs/1807.03748**
- Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. *Neural Inf Process Syst*, pp. 5998–6008
- Wu Z, Xiong Y, Yu SX, Lin D (2018) Unsupervised feature learning via non-parametric instance discrimination. In: *CVPR*, pp. 3733–3742
- Xiao C, Zhong H, Guo Z, Tu C, Liu Z, Sun M, Feng Y, Han X, Hu Z, Wang H, Xu J (2018) CAIL2018: A large-scale legal dataset for judgment prediction. *CoRR* **abs/1807.02478**
- Xu N, Wang P, Chen L, Pan L, Wang X, Zhao J (2020) Distinguish confusing law articles for legal judgment prediction. In: *ACL*, pp. 3086–3095
- Yang W, Jia W, Zhou X, Luo Y (2019) Legal judgment prediction via multi-perspective bi-feedback network. In: *IJCAI*, pp. 4085–4091
- Yoran O, Talmor A, Berant J (2022) Turning tables: Generating examples from semi-structured tables for endowing language models with reasoning skills. In: *ACL*, pp. 6016–6031
- Yue L, Liu Q, Jin B, Wu H, Zhang K, An Y, Cheng M, Yin B, Wu D (2021) Neurjudge: A circumstance-aware neural framework for legal judgment prediction. In: *SIGIR*, pp. 973–982
- Zhong H, Guo Z, Tu C, Xiao C, Liu Z, Sun M (2018) Legal judgment prediction via topological learning. In: *EMNLP*, pp. 3540–3549
- Zhong H, Xiao C, Tu C, Zhang T, Liu Z, Sun M (2020) How does NLP benefit legal system: A summary of legal artificial intelligence. In: *ACL*, pp. 5218–5230

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Sheng Bi^{1,2} · Zhiyao Zhou¹ · Lu Pan³ · Guilin Qi¹ 

Sheng Bi
bisheng@seu.edu.cn

Zhiyao Zhou
seuzzy@seu.edu.cn

Lu Pan
lukepan@tencent.com

¹ School of Computer Science and Engineering, Southeast University, Nanjing 211189, Jiangsu, China

² Judicial Big Data Research Centre, School of Law, Southeast University, Nanjing 211189, Jiangsu, China

³ Tencent Technology (Shenzhen) Co., Ltd., Shenzhen 518057, Guangdong, China