

---

# LawShift: Benchmarking Legal Judgment Prediction Under Statute Shifts

---

Zhuo Han<sup>1\*</sup>, Yi Yang<sup>1\*</sup>, Yi Feng<sup>1†</sup>, Wanhong Huang<sup>1</sup>, Xuxing Ding<sup>1</sup>  
Chuanyi Li<sup>1</sup>, Jidong Ge<sup>1</sup>, Vincent Ng<sup>2</sup>

<sup>1</sup>State Key Laboratory for Novel Software Technology, Nanjing University, China

<sup>2</sup>Human Language Technology Research Institute, University of Texas at Dallas, USA  
fy@nju.edu.cn

## Abstract

Legal Judgment Prediction (LJP) seeks to predict case outcomes given available case information, offering practical value for both legal professionals and laypersons. However, a key limitation of existing LJP models is their limited adaptability to statutory revisions. Current SOTA models are neither designed nor evaluated for statutory revisions. To bridge this gap, we introduce LawShift, a benchmark dataset for evaluating LJP under statutory revisions. Covering 31 fine-grained change types, LawShift enables systematic assessment of SOTA models’ ability to handle legal changes. We evaluate five representative SOTA models on LawShift, uncovering significant limitations in their response to legal updates. Our findings show that model architecture plays a critical role in adaptability, offering actionable insights and guiding future research on LJP in dynamic legal contexts.

## 1 Introduction

Legal Judgment Prediction (LJP) aims to predict judicial outcomes based on case information such as facts and claims [1, 2], and has been studied across multiple legal jurisdictions. In the U.S. and India, LJP typically centers on predicting court decisions, i.e., whether claims will be upheld, given the case facts and arguments [3, 4]. In contrast, in other jurisdictions, LJP emphasizes aligning case facts with applicable law articles [5–11]. For instance, LJP in the EU primarily focuses on predicting applicable law articles [12–14]. Similarly, Chinese LJP targets the prediction of law articles, as well as charges, and prison terms based on case facts [15, 1, 16, 17]. LJP provides timely insights into case results, supporting legal professionals and laypersons in decision-making while minimizing time consumption and the need for legal consultations.

A significant challenge in LJP is the dynamic nature of legal systems, where ongoing statutory revisions in both common and civil law continually reshape legal norms to align with evolving societal demands [18–20]. Table 1 illustrates how statutory revisions can alter case outcomes. In the example, the defendant was acquitted under the prior statute despite discharging radioactive substances into public waters; after the revision, the same act resulted in conviction. This highlights the significant impact that changes in law articles have on judicial decisions, affecting both individuals and society. Consequently, LJP models must swiftly adapt to updated statutes, revise their legal reasoning, and produce judgments grounded in current legal standards.

However, existing LJP research overlooks this challenge. First, LJP models are not explicitly designed to accommodate legal updates, leading to potential judgment errors. While off-the-shelf techniques, such as model fine-tuning [21] and model editing [22, 23], can incorporate updated legal knowledge,

---

\*These authors contributed equally to this work.

†Corresponding author.

Table 1: An example of different judgment outcomes before and after revisions.

<b>Fact Description:</b> The defendant stole Tritium, a radioactive substance, from a local chemistry lab and deliberately introduced it into the public drinking water supply in his neighborhood, resulting in severe consequences, including ...
<b>Old Article:</b> ... commits arson, breaches dikes, cause explosion, <b>spreads poisonous substances</b> shall be sentenced ...
<b>Judgment:</b> <b>Non-Violation</b> ( <i>obsolete</i> )
<b>Revised Article:</b> ... commits arson, breaches dikes, cause explosion, <b>spreads poisonous or radioactive substances</b> shall be sentenced ...
<b>Judgment:</b> <b>Violation</b> ( <i>up-to-date</i> )

they do not ensure consistent adherence to the latest statutes. Second, existing LJP evaluation typically relies on static, held-out datasets that treat law articles as unchanging, ignoring evolving legal norms [16, 17]. These datasets merge different versions of statutes under the same label [15], despite significant variations in the underlying judgment logic across versions. This conflation introduces considerable noise from judgment logic shifts during model training.

This paper investigates the ability of state-of-the-art (SOTA) LJP models to adapt to judgment logic shifts caused by statutory revisions, highlighting their limitations and guiding future research in dynamic legal contexts. To rigorously evaluate model performance amid such changes, we introduce LawShift, a Chinese statutory revision-oriented LJP benchmark featuring a metamorphic testing framework. Note that diverse statutory revisions, such as adding exceptions, redefining terms, or modifying conditions, differentially affect judgment logic, LawShift encompasses 31 fine-grained revision types to capture this complexity.

To evaluate whether LJP models capture shifts in legal reasoning caused by statutory revisions, we employ a metamorphic testing (MT) approach [24]. Traditional evaluations using metrics like F1 depend on ground-truth labels, which are often missing or ambiguous in the context of evolving laws—especially when newly revised statutes lack corresponding judged cases and labels. MT solve this limitation by defining metamorphic relations, which are logically grounded expectations about how model predictions should change following specific statute revisions.

We evaluate five SOTA LJP models using LawShift and find that they exhibit notable vulnerabilities under evolving legal norms. Most models struggle to align their reasoning with updated statutes, defaulting to outdated legal logic learned during training. Models equipped with explicit legal knowledge encoding mechanisms or fine-grained attention to statute semantics show slightly improved adaptability, while those primarily relying on statistical correlations between facts and labels perform significantly worse. In a nutshell, our contributions are three-fold.

First, we conduct the first evaluation of LJP models under statutory shifts by introducing a comprehensive dataset tailored to test model adaptability to statutory revisions. Covering 31 revision types, our dataset serves as a foundational resource for advancing LJP and legal reasoning in language models.

Second, we adopt a metamorphic testing strategy to assess whether model predictions align with revised legal reasoning, without relying on extensive manual annotations. This approach is broadly applicable to other legal tasks where ground-truth labels are unavailable or ambiguous [25–27].

Third, we conduct an in-depth evaluation of SOTA LJP models on our dataset, uncovering key strengths and limitations under statutory revisions. Our analysis of model mechanisms offers actionable insights for improving legal reasoning in dynamic legal contexts.

## 2 Related Works

Current SOTA methods primarily focus on improving performance on existing benchmarks (such as ECtHR and CAIL-2018) [12, 15] with static law labels, employing deep learning models to address LJP subtasks. Some approaches propose neural network-based methods that capture the inherent dependencies among LJP subtasks, thereby enhancing model performance [1, 28]. Others make predictions by reasoning over legal knowledge expressed through the semantics of subtask labels [28, 16]. With the advancement of Pretrained Language Models (PLMs), researchers have developed PLMs specifically pretrained on extensive legal texts [29, 30]. These tailored models are designed to better handle the nuances of legal language and can be fine-tuned for specific LJP subtasks. The rise of powerful Large Language Models (LLMs) has led to their rapid adoption in LJP, with several LLM-based approaches achieving notable success [31, 32].

To the best of our knowledge, no prior work has explored LJP under statutory revisions, a crucial yet overlooked aspect for real-world applicability. Existing datasets conflate different versions of law articles under static labels, introducing legal reasoning noise during training. Moreover, current SOTA models lack specifically designed mechanisms to accommodate evolving statutes. While strategies like model fine-tuning and editing can inject updated legal knowledge, they do not ensure LJP models consistently align with revised statutes, resulting in prediction errors [33, 34]. A related work by Wang et al. [35] benchmarks LLMs’ ability to handle edited legal knowledge, focusing primarily on evaluating updating mechanisms. They apply various updating mechanisms (e.g., RAG [36], ROME [33], LoRA [37]) to multiple LLMs and evaluate their performance. In contrast, our work focuses on evaluating LJP models’ ability to adapt to shifts in underlying judgment logic induced by statutory revisions. Our dataset encompasses 31 distinct types of statutory changes, far exceeding the three covered by prior work. Additionally, their dataset’s fact descriptions are highly condensed and simplified, lacking the complexity of authentic case narratives. This oversimplification risks biasing evaluations and undermining objectivity, as LJP critically depends on nuanced factual details to align with statutory elements. Our dataset preserves the richness and structural complexity of real-world cases, enabling a more realistic and robust assessment of models’ true capability in statute shift scenarios. Another related study [25] examines LJP models’ capacity to extract key elements and ensure fairness across sensitive attributes like race. By contrast, our work probes models’ sensitivity and adaptability to judgment logic shifts induced by statutory revisions, further analyzing how architectural differences affect their legal reasoning adaptability.

### 3 LawShift

This study targets the Chinese Criminal Code<sup>3</sup>. We detail the annotation process of LawShift.

#### 3.1 Identifying Statutory Revision Types

A law article consists of two core components: the *constitutive element*, which specifies the factual conditions required for the law to apply, and the *legal consequence*, which defines the resulting outcomes or penalties when those conditions are met. In practice, statutory revisions modify these components through diverse changes. Note that real-world revisions are often coarse-grained, with multiple component changes occurring simultaneously. However, our aim is to evaluate LJP models on as fine-grained revision types as possible.

To systematically identify fine-grained statutory revisions, we first introduce a **Law Article Template** outlining key legal article components may change. Next, we define a comprehensive set of **Amendment Dimensions** that capture all possible ways these components can be changed (i.e., how components change). We analyze all 12 historical editions of the Chinese Criminal Code Amendments to derive empirically grounded revision types, which define the final statutory revision taxonomy of LawShift.

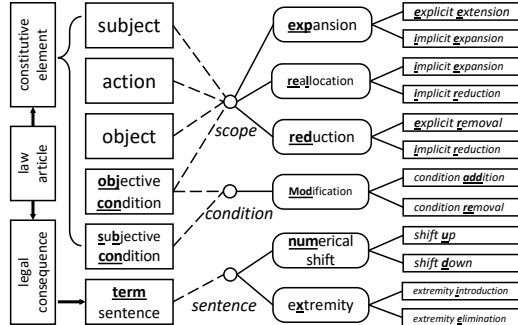


Figure 1: Law article template (left) and amendment dimensions (right).

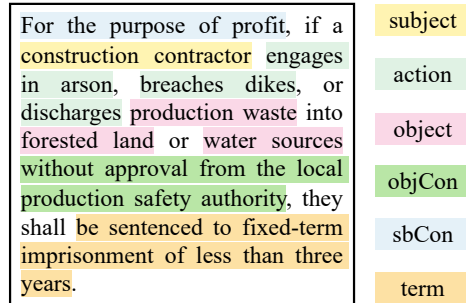


Figure 2: A fictional law article crafted to include all components within a single example.

<sup>3</sup>We emphasize that while our dataset is constructed from cases and laws in China, the methodology is applicable across other legal jurisdictions. Further discussions of generalization can be found in Appendix B

Table 2: Details of the three scope changes.  $S$  and  $S'$  denote the scopes before and after revision

TYPE	DEFINITION	DESCRIPTION	EXAMPLE
expansion	$S \subsetneq S'$	the scope of a component is broadened	<i>adults</i> $\rightarrow$ <i>anyone above 16</i>
reallocation	$S \cap S' \neq \emptyset$ and $S \triangle S' \neq \emptyset$	the new scope added and removed certain entities or actions previously covered	<i>state-owned enterprise employees</i> $\rightarrow$ <i>executives of any corporation</i>
reduction	$S' \subsetneq S$	the scope of a component is narrowed	<i>consume and traffic drugs</i> $\rightarrow$ <i>consume drugs</i>

**Law article template** Under guidance from legal experts, we decompose law articles into multiple sub-components, as illustrated in Figure 1, with the *constitutive elements* further divided into five sub-components: (1) **subject**, which defines the scope of the individuals or groups responsible for the crime; (2) **action**, which specifies the scope of the criminal actions by the *subject*; (3) **object**, which identifies the scope of entities involved in or affected by the *action*; (4) **objective condition**, which outlines situational constraints under which the action becomes criminal; (5) **subject condition**, which captures the required intent or mental state of the *subject*. The *legal consequences* is represented by a single sub-component: the *term sentence*<sup>4</sup>. Figure 2 presents an example of law articles.

**Amendment dimensions** We identify possible changes and organize them into a hierarchical structure, as illustrated in Figure 1.

There are three primary revision types: changes in *scope* definitions, *conditions*, and *sentences*. The *scope* definition changes involve components that specify the defined boundaries of entities or actions subject to regulation under the corresponding articles. This type of change involves four sub-components, i.e., *subject*, *actions*, *object*, and *objective condition*. Recall the example in Table 1, after the *scope* of the *object* changes from *poisonous substances* to *poisonous or radioactive substances*, the case’s outcome shifted from *Non-Violation* to *Violation*. The *scope* can be further classified into three types: expansion, reduction, and reallocation, as shown in Table 2. We can further categorize expansion, reallocation and reduction changes into two subtypes, i.e. *explicit extension/removal* and *implicit rephrasing*. Explicit scope changes occur through enumerative or parallel modifications, whereas implicit rephrased scope changes involve a semantic shift marked by a complete rewording of the component’s definition, as illustrated in Table 3.

Similarly, the *condition* changes involve *subjective condition* and *objective condition*. These revisions typically manifest either through the addition of a condition or the removal of an existing one. The addition of a condition may introduce new criteria or prerequisites that must be met for the law to apply, potentially making the legal requirements more strict.

The *sentence* change involves the *term sentence* component in an article’s *legal consequences*. This type of change can be further divided into two categories, namely *numerical shift* and *extremity*. *Numerical shift* refers to adjustments in penalty ranges, specifically *shift up* and *shift down*. *Extremity* represents non-numerical modifications, including the *introduction* or *elimination* of discrete sentencing terms, such as *life imprisonment* or the *death penalty*.

Table 3: Examples of explicit and implicit scope changes

TYPE	EXAMPLE
<i>explicit extension</i>	<i>poisonous substances</i> $\rightarrow$ <i>poisonous or radioactive substances</i>
<i>implicit expansion</i>	<i>adults</i> $\rightarrow$ <i>anyone above 16</i>
<i>explicit removal</i>	<i>consume and traffic drugs</i> $\rightarrow$ <i>consume drugs</i>
<i>implicit reduction</i>	<i>corporation employees</i> $\rightarrow$ <i>corporation executives</i>

Table 4: Examples of metamorphic testing in LawShift.

TYPE	LAW RVSN.	FACT ALT.	EXPT
T1.1	<i>Article 271: ... personnel of state-owned companies, ...</i> $\rightarrow$ <i>... personnel of private enterprises, state-owned companies, ...</i>	<i>... the defendant while serving as person in charge of company A, which is a state-owned company, ...</i> $\rightarrow$ <i>... the defendant while serving ... a private enterprise B ...</i>	V
T6.1	<i>Article 232: ... the sentence shall be 3 to 10 years imprisonment. <math>\rightarrow</math> ... the sentence shall be more than 10 years imprisonment.</i>	N/A	T $\uparrow$

<sup>4</sup>Other penalties, such as fines, are inconsistently specified across statutes and often ignored by existing LJP models, thus excluded from our study.

Table 5: Revision types, corresponding testing methods, and the number of test instances in LawShift.

TYPE	BRIEF DESCRIPTION	TASK	EXPT.	#
<b>T1.1:</b> subject-exp-ee	identify subject in the explicitly extended scope	Article	V	261
<b>T1.2:</b> subject-exp-ie	identify subjects in the implicitly expanded scope	Article	V	261
<b>T1.3:</b> subject-rel-ie	identify subjects in the expanded scope of the reallocation	Article	V	261
<b>T1.4:</b> subject-rel-ir	identify subjects in the reduced scope of the reallocation	Article	NV	261
<b>T1.5:</b> subject-red-er	identify subjects in the explicitly removed scope	Article	NV	261
<b>T1.6:</b> subject-red-ir	identify subjects in the implicitly reduced scope	Article	NV	261
<b>T2.1:</b> action-exp-ee	identify actions falls in the explicitly extended scope	Article	V	102
<b>T2.2:</b> action-exp-ie	identify actions falls in the implicitly expanded scope	Article	V	102
<b>T2.3:</b> action-rel-ie	identify actions falls in the expanded scope of the reallocation	Article	V	102
<b>T2.4:</b> action-rel-ir	identify actions falls in the reduced scope of the reallocation	Article	NV	102
<b>T2.5:</b> action-red-er	identify actions falls in the explicitly removed scope	Article	NV	102
<b>T2.6:</b> action-red-ir	identify actions falls in the implicitly reduced scope	Article	NV	102
<b>T3.1:</b> object-exp-ee	identify objects in the explicitly extended scope	Article	V	459
<b>T3.2:</b> object-exp-ie	identify objects in the implicitly expanded scope	Article	V	459
<b>T3.3:</b> object-rel-ie	identify objects in the expanded scope of the reallocation	Article	V	459
<b>T3.4:</b> object-rel-ir	identify objects in the reduced scope of the reallocation	Article	NV	459
<b>T3.5:</b> object-red-er	identify objects in the explicitly removed scope	Article	NV	459
<b>T3.6:</b> object-red-ir	identify objects in the implicitly reduced scope	Article	NV	459
<b>T4.1:</b> objCon-exp-ee	identify entities in the explicitly extended scope and determine non-compliance	Article	NV	150
<b>T4.2:</b> objCon-exp-ie	identify entities in the implicitly expanded scope and determine non-compliance	Article	NV	150
<b>T4.3:</b> objCon-rel-ie	identify entities in the expanded scope of the reallocation and determine non-compliance	Article	NV	150
<b>T4.4:</b> objCon-rel-ir	identify entities in the reduced scope of the reallocation and determine compliance	Article	V	150
<b>T4.5:</b> objCon-red-ir	identify entities in the implicitly reduced scope and determine compliance	Article	V	150
<b>T4.6:</b> objCon-mod-add	determine non-compliance with the added objective condition	Article	NV	409
<b>T4.7:</b> objCon-mod-re	determine compliance with the removed objective condition	Article	V	149
<b>T5.1:</b> sbCon-mod-add	determine non-compliance with the added subjective condition	Article	NV	470
<b>T5.2:</b> sbCon-mod-re	determine compliance with the removed subjective condition	Article	V	115
<b>T6.1:</b> term-num-up	sentence longer terms	Term	T↑	215
<b>T6.2:</b> term-num-dn	sentence shorter terms	Term	T↓	215
<b>T6.3:</b> term-x-in	sentence extreme terms	Term	XT	215
<b>T6.4:</b> term-x-el	no longer sentence extreme terms	Term	NX	99

**Determining the final revision types** Following these steps, we define all potential statutory revision types as component-amendment pairs (Figure 1). However, not all theoretically possible types occur in practice. After reviewing all 12 historical editions of the Chinese Criminal Code Amendments, LawShift incorporates only those revision types empirically observed in real-world. The finalized types and their detailed descriptions are provided in Table 5.

### 3.2 Testing Methodology

In LJP, ground-truth labels are often unavailable or ambiguous under evolving statutes as newly revised statutes may lack corresponding real-world cases and no golden label exist. While expert annotation is an option, these labels are annotator-generated rather than court-issued, raising concerns about judicial authenticity and undermining the objectivity and fairness of evaluation. Thus, traditional evaluation methods (e.g., computing the F1 score for law article prediction task) are not suitable for statutory revisions, as they rely on ground-truth labels that may be missing under shifting statutes. Metamorphic testing allows for the testing of situations where it may not be feasible to define ground-truth labels [24]. Specifically, metamorphic testing assesses whether LJP models produce expected outputs consistent with the judgment logic introduced by statutory revisions. We employ two kinds of metamorphic testing. One is comparing the outcomes of LJP models before and after revisions (**T6.1-T6.4**). As shown in Table 4, for revision type **T6.1** (i.e., changes to statutory sentencing ranges), we evaluate whether LJP models adjust sentencing predictions in line with updated judgment logic. For instance, if Article 232’s range shifts from "3–10 years" to "more than 10 years", a model pass the test if cases previously predicted within 3–10 years now yield predictions exceeding 10 years. Here, ground-truth labels are unnecessary—we only assess whether the model outputs a value exceeding 10 years. The other kind of metamorphic testing constructs minimal test instances targeting a single capability, e.g., recognizing a legal condition change (**T1.1-T5.2**). For example, in Article 271 in Table 4, originally applied only to state-owned enterprise personnel but was extended to include private enterprise individuals. We create a minimal test by modifying real cases judged under the pre-revision statute, i.e., replacing "state-owned enterprise" with "private enterprise" while

keeping other facts unchanged. If the LJP model predicts a violation, it reveals that the model internalizes logic changes (e.g., statutory revisions). This strategy avoids crafting entirely new cases and relies on targeted text edits to reflect statutory changes <sup>5</sup>.

### 3.3 Generating Test Instances

Next, we detail how we generate test instances.

**Collecting original cases** We construct LawShift test instances based on real-world judicial documents from China Judgments Online (CJO)<sup>6</sup>. Specifically, we select cases issued between 2017/11/4 and 2021/2/28 as these cases share the same version of laws and judged by the same judgment logic. By controlling the time span, we ensure that the original judgments are consistent with the legal framework in effect at the time, allowing us to attribute changes in model behavior solely to the introduced statutory revisions and avoid interference from other legal contexts or temporal discrepancies [38] <sup>7</sup>.

We then construct LawShift using the original legal cases collected above. Each test instance consists of three elements: a pair of law articles (before and after the revision), a corresponding pair of fact descriptions (from the original case and its edited version), and an expected outcome aligned with the revised law article. During the annotation process, we engaged one senior annotator (a law school professor) and two annotators (law school PhD candidates) to ensure the legal expertise of LawShift.

**Revising law articles** Next, we introduce revisions to law articles. For each revision type in Table 5, annotators collaboratively identify suitable law articles and manually construct corresponding revised versions that reflect the intended changes. For instance, for revision type **T4.7** (removal of an *objective condition*), only law articles containing this component are eligible for revision. In practice, one annotator independently drafts each revised law article based on the target revision type, which is then reviewed by a second annotator for legal validity, coherence, and alignment. Conflicts are resolved through discussions led by a senior annotator, who iteratively refines the drafts to ensure consensus.<sup>8</sup>

To ensure fair comparison of LJP model performance across different revision types affecting the same component, some revision types share identical selected law articles and original cases. For example, **T6.1** and **T6.2**, which involve revisions to sentencing terms, share the same set of selected law articles. This design ensures that, for each component, the impact of its different revision types can be compared fairly without being affected by doctrinal variations, or data distribution differences.

**Editing facts** For each revision type, we retrieve original cases citing the corresponding law article from the previously selected set. We then edit the retrieved case facts, when necessary, to align with the corresponding revised law article. For instance, fact editing is required when testing revision types involving *constitutive elements*, as we adopt a minimal test strategy (Section 3.2). For example, if a statute’s *subject* expands from "poisonous substances" to "poisonous or radioactive substances", we edit the original case facts to replace the poisonous substance with a radioactive one. This minimal change isolates the effect of the revision, enabling us to assess whether the model correctly recognizes the inclusion of radioactive substances under the revised statute. As for **T6.1-T6.4**, we use the original facts as we adopt the comparison strategy (Section 3.2).

There are three approaches we’ve used to revise the case facts: (1) perturbing existing cases, (2) employing LLMs to synthesize new facts, and (3) using real case facts.

For revisions targeting *subject*, *object*, *objective condition*, and *subjective condition* ( **T1.1–1.6**, **T3.1–3.6**, **T4.1–4.7**, **T5.1–5.2**), we apply the perturbation strategy. This involves (1) identifying relevant fact spans using regular expressions, and (2) inserting or replacing them with phrases from predefined sets curated by our three expert annotators during the law revision process. For instance, when editing facts for **T5.1**, where the *subjective condition* may be revised to "out of revenge or to vent frustration", we build a phrase list describing alternative motives unrelated to this condition to prompt non-violation predictions. Using regular expressions, we locate the defendant span in

<sup>5</sup>See Table 13 (Appendix H) for details of revision types with their testing methods, examples.

<sup>6</sup><http://wenshu.court.gov.cn/>

<sup>7</sup>See Appendix C for pre-processing details.

<sup>8</sup>The annotators were paid an hourly wage of 200 RMB.



the case facts and prepend a randomly selected phrase from the list. This perturbation approach ensures minimal, targeted edits specific to the revision type, while preserving the overall coherence and integrity of the case facts.

LLMs are employed to edit case facts for revision types **T2.1–T2.6**, which involve changes to *action scopes*. Unlike entities—typically isolated noun phrases—actions are embedded within complex narrative structures, and revisions to their scope often alter the semantics of the entire case. Such complexity makes simple perturbation insufficient. To address this, we leverage GPT-4o [39] to generate coherent, legally sound case facts that reflect the revised statutory definitions. GPT-4o is guided via one-shot prompting, using exemplar pairs crafted by annotators to ensure alignment with the revision intent and judicial style. Specifically, two annotators independently draft one exemplar for the revised law articles of each revision type. Each exemplar consists of a pair of case facts (before-and-after revision) aligned with the corresponding revised law article. Subsequently, the senior annotator reviews and refines these exemplars when necessary <sup>9</sup>.

Revision types **T6.1–T6.4**, which involve changes to *legal consequences*, do not require modification of case facts. As illustrated in Table 4, these revisions can be evaluated by comparing a model’s sentencing predictions on unchanged facts under the original and revised statutes. Thus, the original case facts are directly reused as the edited ones <sup>10</sup>.

**Expected Output** An LJP model is considered to pass a test instance in LawShift if it generates the expected output given the edited fact. Expected outputs for each revision type are defined in Table 5. For example, in **T1.1** (subject extension), if a model correctly identifies a subject within the newly extended scope and predicts **Violation (V)**, it passes. Other expected outputs include **Non-Violation (NV)**, **Increased Term (T↑)**, **Decreased Term (T↓)**, **Extreme Term (XT)**, and **Non-Extreme Term (NX)**. Different LJP tasks are used depending on the nature of each revision type.

**LawShift statistics** LawShift encompasses a total of 7,569 test instances<sup>11</sup>. The statistics of test instances for each revision type is shown in Table 5.

## 4 Experiments

### 4.1 Evaluation Setup

**Base dataset** Before evaluating LJP models on LawShift, we construct a base dataset for model training. This dataset, like LawShift, is sourced from China Judgment Online and processed using the same pipeline (see Appendix C). To ensure consistent judgment logic, we select cases from the same time span as LawShift. The final dataset includes 253,936 training and 63,484 validation cases.

**Baselines and metrics** We evaluate nine SOTA (TopJudge [1], D-LADAN [16], NeurJudge [40], Lawformer [30], Qwen2.5-7B, Llama3-Instruct-8B, ChatGLM3-6B, Llama3-8B-Law, and Lawmodel-7B) on LawShift and report the *Pass Rate* for each revision type—where a case is considered a *pass* if the model generates the expected outcome given the edited fact. Each baseline represents a distinct family of LJP modeling approaches, some of which are inherently capable of adapting to statutory revisions. For example, NeurJudge and Lawformer include mechanisms for matching with law article texts; during evaluation, we provide them with the revised statutes as input. Similarly, LLM-based models are evaluated using prompts that include the revised law text to assess their ability to adapt to changes in judgment logic. We conduct two evaluations (before and after revision) per revision type on LawShift: (1) assessing model performance on original judgments using original facts, and (2) assessing performance on expected outcomes using the edited facts. This can help us fairly compare LJP models’ performance across different components. Specifically, we compute normalized relative pass rate shift  $\mathcal{R}_c$  (of component  $c$ ), computed as  $\mathcal{R}_c = \sum_{i=1}^{n_c} 2 * (r_{f_i} - r_{ori_i}) / (r_{f_i} + r_{ori_i})$  where  $r_{f_i}$  denotes the pass rate under revision and  $r_{ori_i}$  represents the pass rate before revision for the  $i$ -th revision type of component  $c$ .  $n_c$  is the number of different revision types of component  $c$ . Data

<sup>9</sup>See Appendix H.2 for prompt and exemplar details.

<sup>10</sup>Examples of how we revise law articles and corresponding facts can be found in Appendix H.

<sup>11</sup>Our dataset and code are available on <https://huggingface.co/datasets/triangularPeach/LawShift/tree/main> and <https://github.com/triangularPeach/LawShift>.

distributions vary across components, e.g., the original cases for **T1.1** and **T2.1** differ.  $\mathcal{R}_c$  mitigates these distributional differences, enabling fair comparisons between components (e.g., *subject* vs. *action*). To compare models across different revision types of the same component, we analyze  $r_{fi}$ , since these revision types share the same selected law articles and original cases (as mentioned in Section 3.3). Details on baselines, metrics, and experimental settings are provided in Appendix D.

## 4.2 Performance Analysis

We report experimental results in Figure 3 (See Table 11 in Appendix for detailed values).

**Overall results** In Figure 3, the revision types indicated by the gray background (e.g., **T1.2**, **T4.5**) represent types where predictions are expected to remain consistent, whereas those highlighted with a beige background (e.g., **T1.4** and **T5.1**) correspond to revisions expected to lead to changes in predictions. We can find that, most neural-based LJP models perform reliably when predictions are expected to remain consistent but struggle to adjust when legal revisions require different outcomes, highlighting their limited adaptability to statutory changes, while LLM solutions of LJP exhibit higher level of performance on legal revisions require altered sentence.

Notably, **T2.2** and **T2.3** are often misclassified under an unrelated law article, specifically, *drug possession*, which is a pattern not seen before revision. This may be due to revised cases including phrases like *harboring individuals for drug transactions*, which semantically overlap with possession offenses. The pattern suggests Lawformer relies more on surface-level keywords than on accurate legal interpretation in context.

Revision types related to prison terms are highlighted with a pink background. We observe that models achieves higher pass rates on revisions that reduce sentence severity. Specifically, models perform better on **T6.2** than **T6.1** and on **T6.4** than **T6.3**. This trend indicates a bias toward lighter penalties, possibly due to the training data’s skew toward lenient outcomes.

**Normalized Relative Pass Rate Shift** We further examine model sensitivity to changes to different components using the metric  $\mathcal{R}_c$ . In Figure 4, all neural-based models demonstrate similar level of performance on *subject* and *object* revisions, suggesting these components require similar reasoning process. In contrast, neural-based models perform relatively worse on objective condition revisions, which demand additional reasoning over scope and conditional logic, while performing better on subjective condition revisions that mainly involve adding or removing single conditions.

General purpose LLMs show clear variation in their sensitivity across reasoning components. As shown in Figure 4, LLMs exhibit larger shifts on *object* and *subjective condition* components, indicating that revisions involving semantic role changes or the addition and removal of qualifying conditions are the most likely to affect their predictions. These two components require the model to reinterpret local contextual dependencies while preserving global consistency, which increases the reasoning complexity compared with static entities or structural terms. In contrast, smaller shifts are observed for *subject* and *objective condition* components, suggesting that identifying core entities or scope boundaries is less challenging for models trained on broad text corpora.

Across all models, *term* revisions and *action* revisions remain the most challenging, with traditional baselines showing the lowest normalized pass rate and only marginal improvement from large models. For *term* revisions, the results imply that sentence-term reasoning still poses a persistent limitation even for architectures with large-scale pretraining, as it requires quantitative alignment between case outcomes and statutory boundaries rather than purely linguistic understanding. *Action* revisions also pose a greatest challenge for models, likely because understanding revisions to cases’ actions requires deeper contextual comprehension<sup>12</sup>.

<sup>12</sup>See Appendix F for details on *action* revision error analysis.

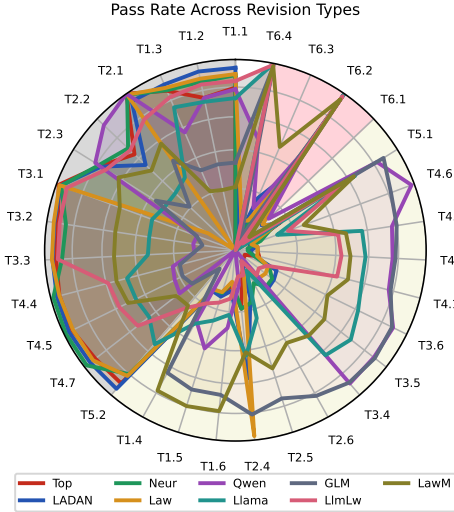


Figure 3: Pass rate across revision types.



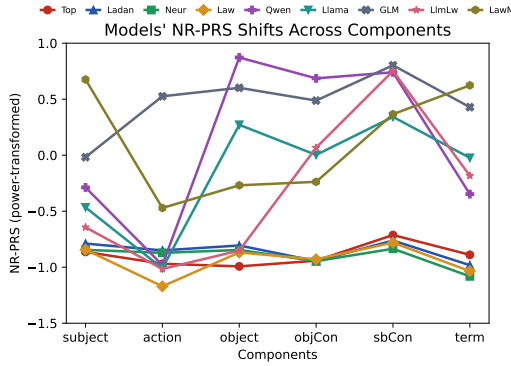


Figure 4: Relative pass rate shifts across components.

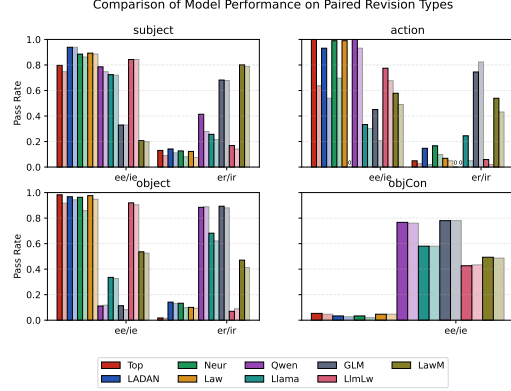


Figure 5: Pass rate differences between explicit and implicit changes. Lighter shades represent implicit changes (ie/ir).

**Pass Rates between Formulations** As described in Section 3.1, we define two formulation types for scope revisions: *explicit extension/removal* and *implicit rephrasing*. Figure 5 presents model pass rates on revision types involving such distinctions, specifically **T1.1-1.2; 1.5-1.6, T2.1-2.2; 2.5-2.6, T3.1-3.2; 3.5-3.6, and T4.1-4.2**. It is evident that all models perform better on explicit changes than on their implicit counterparts. This supports our hypothesis that *implicit rephrasing* poses greater challenges, likely due to its higher demand for semantic understanding of legal language.

### 4.3 Discussions

Among the nine models, TopJudge performs worst due to its lack of law article awareness, underscoring that incorporating legal text boosts adaptability. Models like D-LADAN and NeurJudge, which explicitly reference law articles for legal differentiation or key fact extraction, show better robustness but still struggle with revisions requiring deeper semantics understanding, such as *implicit rephrasing* or *action* revisions. Meanwhile, the general-domain LLMs adapt flexibly via in-context learning, but their outputs are generally inconsistent due to prompt sensitivity and pretraining bias. For general purpose LLMs, their pretraining enables stronger generalization to context-dependent changes, particularly when modifications involve nominal or semantic role adjustments, but also makes them more sensitive to surface-level variations in conditional structures. Among them, Chat-GLM maintains relatively stable shifts across all components, indicating balanced sensitivity to both entity and logical changes, whereas Llama shows more fluctuation between object and term components, implying partial over-reliance on lexical co-occurrence learned from open-domain text. On the other hand, law-oriented models demonstrate a distinct bias towards condition-level reasoning. They perform consistently on *subject* and *object* components but show higher positive shifts on *objective* and *subjective condition* components, suggesting an increased capacity to encode legal relational dependencies once pretraining includes domain statutes or fine-tuning on legal corpora.

To improve LJP models under statute shifts, future research should focus on several directions. First, **better utilization of law content** is needed. Models should move beyond keyword matching and learn context-aware representations that capture legal hierarchies and fact-law matching. Second, **handling implicit rephrasings and action changes** remains a key challenges. Current models struggle with semantic abstraction, calling for contrastive learning and event-level representations tailored to legal text beyond surface-level understanding. Third, **balancing stability and adaptability** is difficult. LLMs like Qwen adapt well in context but lack consistency. Hybrid methods such as adaptive RAG may help by grounding predictions in dynamic legal context while maintaining structured reasoning. Finally, **temporal legal learning is essential**. Models should support timestamp-aware pretraining and continual learning to track legal amendments without full retraining. Moreover, considering that in real-world settings, post-amendment data is scarce, making full retraining impractical. This calls for training frameworks that treat time as part of supervision, guiding models to align predictions with the valid law at the relevant time, even under low-resource conditions.

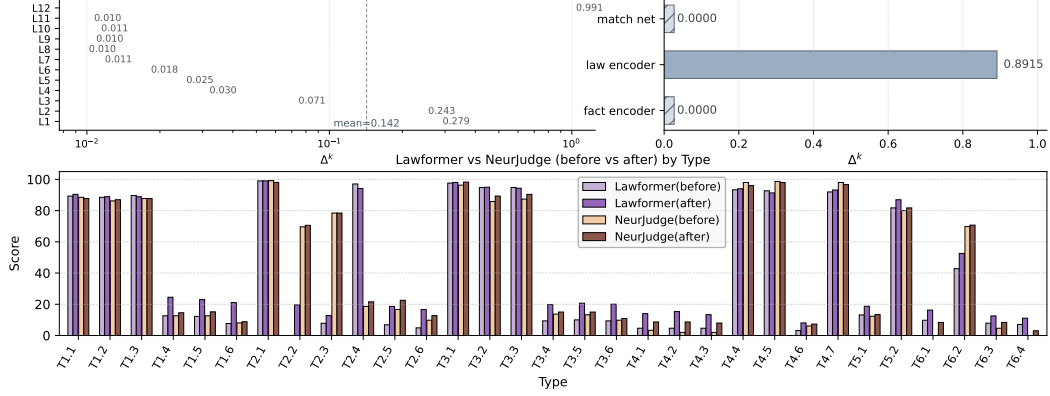


Figure 6: (top) Layer-wise and module-wise hidden-state shifts ( $\Delta^k$ ) with updated statutes of Lawformer and NeurJudge, respectively. (bottom) Post-editing performance across revision types after applying localized model weight updates.

#### 4.4 Broader Evaluation: Knowledge Tracing and Editing

To further explore how model responds to statutory revisions and whether post-hoc updates can effectively adapt models, we conduct broader evaluation experiments using knowledge tracing [41, 42] and model editing [33] on Lawformer and NeurJudge.

**Knowledge Tracing** We trace hidden representations of both models before and after statutory revisions to reveal how internal layers encode legal changes. Figure 6 (top-left) shows that Lawformer exhibits the largest change at its final encoder layer (Layer 12), indicating that statute revisions mainly influence high-level semantic representations. In contrast, NeurJudge displays concentrated variation in its law-encoder component, while its fact encoder and matching module remain nearly stable, as illustrated in Figure 6 (top-right). These findings identify the most revision-sensitive layers, providing concrete anchor points for subsequent lightweight editing<sup>13</sup>.

**Knowledge Editing** Building on the tracing analysis, we took inspirations from the model editing method ROME [33] and applied localized rank-one updates to the most sensitive position in the respective models. As illustrated in Figure 6 (bottom), both backbones benefit from editing, as both models show improvements across most revision types. However, the gains are still rather small, and the results suggest that localized parameter updates can alleviate certain surface-level mismatches but fall short of capturing deeper judgment logic shifts. More principled approaches, such as structure-aware continual learning or representation-level adaptation, are still required to achieve robust responsiveness under statutory revisions.

## 5 Conclusion

We propose LawShift, a benchmark for evaluating LJP models under statutory revisions, covering 31 revision types across 6 categories, each aligned with specific expected model behaviors. We assess five representative SOTA LJP models, revealing their strengths, limitations, and design implications for improving adaptability to legal changes.

**Limitations** Despite LawShift’s jurisdictional and linguistic constraints, its core testing methodology (decomposing legal revisions, aligning them with model behavior, and evaluating reasoning consistency) is broadly generalizable across languages and legal systems. We evaluate baselines with limited revision-handling capabilities, e.g., law-aware attention or prompt injection. Nonetheless, our focus is identifying how models fail under statute shifts, rather than solving the update process.

<sup>13</sup>Details of these experiments are shown in Appendix G.1.

## Acknowledgments

We thank the reviewers for their valuable comments on an earlier draft of this paper. This work was supported by National Natural Science Foundation of China (No. 62406139), State Key Laboratory for Novel Software Technology at Nanjing University (KFKT2025A15, ZZKT2025B14, KFKT2024A07, ZZKT2024B02).

## References

- [1] Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. Legal judgment prediction via topological learning. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3540–3549. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-1390. URL <https://doi.org/10.18653/v1/d18-1390>.
- [2] Jeffrey A Segal. Predicting supreme court cases probabilistically: The search and seizure cases, 1962-1981. *American Political Science Review*, 78(4):891–900, 1984.
- [3] Vijit Malik, Rishabh Sanjay, Shubham Kumar Nigam, Kripabandhu Ghosh, Shouvik Kumar Guha, Arnab Bhattacharya, and Ashutosh Modi. ILDC for CJPE: indian legal documents corpus for court judgment prediction and explanation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4046–4062. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.313. URL <https://doi.org/10.18653/v1/2021.acl-long.313>.
- [4] Gil Semo, Dor Bernsohn, Ben Hagag, Gila Hayat, and Joel Niklaus. Classactionprediction: A challenging benchmark for legal judgment prediction of class action cases in the US. In Nikolaos Aletras, Ilias Chalkidis, Leslie Barrett, Catalina Goanta, and Daniel Preotiuc-Pietro, editors, *Proceedings of the Natural Legal Language Processing Workshop, NLLP@EMNLP 2022, Abu Dhabi, United Arab Emirates (Hybrid), December 8, 2022*, pages 31–46. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.NLLP-1.3. URL <https://doi.org/10.18653/v1/2022.nllp-1.3>.
- [5] Kankawin Kowsrihawatt, Peerapon Vateekul, and Prachya Boonkwan. Predicting judicial decisions of criminal cases from thai supreme court using bi-directional gru with attention mechanism. In *2018 5th Asian Conference on Defense Technology (ACDT)*, pages 50–55. IEEE, 2018.
- [6] Lang Cao, Zifeng Wang, Cao Xiao, and Jimeng Sun. PILOT: legal case outcome prediction with case law. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 609–621. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.NAACL-LONG.34. URL <https://doi.org/10.18653/v1/2024.naacl-long.34>.
- [7] Benjamin Strickson and Beatriz de la Iglesia. Legal judgement prediction for UK courts. In *ICISS 2020: The 3rd International Conference on Information Science and System, Cambridge, UK, March 19-22, 2020*, pages 204–209. ACM, 2020. doi: 10.1145/3388176.3388183. URL <https://doi.org/10.1145/3388176.3388183>.
- [8] Octavia-Maria Sulea, Marcos Zampieri, Mihaela Vela, and Josef van Genabith. Predicting the law area and decisions of french supreme court cases. In Ruslan Mitkov and Galia Angelova, editors, *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, Varna, Bulgaria, September 2 - 8, 2017*, pages 716–722. INCOMA Ltd., 2017. doi: 10.26615/978-954-452-049-6\_092. URL [https://doi.org/10.26615/978-954-452-049-6\\_092](https://doi.org/10.26615/978-954-452-049-6_092).

- [9] Elias Jacob de Menezes-Neto and Marco Bruno Miranda Clementino. Using deep learning to predict outcomes of legal appeals better than human experts: A study with data from brazilian federal courts. *PloS one*, 17(7):e0272287, 2022.
- [10] Alice Saebom Kwak, Cheonkam Jeong, Ji Weon Lim, and Byeongcheol Min. A korean legal judgment prediction dataset for insurance disputes. *CoRR*, abs/2401.14654, 2024. doi: 10.48550/ARXIV.2401.14654. URL <https://doi.org/10.48550/arXiv.2401.14654>.
- [11] Hiroaki Yamada, Takenobu Tokunaga, Ryutaro Ohara, Akira Tokutsu, Keisuke Takeshita, and Mihoko Sumida. Japanese tort-case dataset for rationale-supported legal judgment prediction. *CoRR*, abs/2312.00480, 2023. doi: 10.48550/ARXIV.2312.00480. URL <https://doi.org/10.48550/arXiv.2312.00480>.
- [12] Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. Neural legal judgment prediction in english. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4317–4323. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1424. URL <https://doi.org/10.18653/v1/p19-1424>.
- [13] Ilias Chalkidis, Nicolas Garneau, Catalina Goanta, Daniel Martin Katz, and Anders Søgaard. Lexfiles and legallama: Facilitating english multinational legal language model development. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 15513–15535. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.865. URL <https://doi.org/10.18653/v1/2023.acl-long.865>.
- [14] Joel Niklaus, Veton Matoshi, Pooja Rani, Andrea Galassi, Matthias Stürmer, and Ilias Chalkidis. LEXTREME: A multi-lingual and multi-task benchmark for the legal domain. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 3016–3054. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EMNLP.200. URL <https://doi.org/10.18653/v1/2023.findings-emnlp.200>.
- [15] Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, and Jianfeng Xu. CAIL2018: A large-scale legal dataset for judgment prediction. *CoRR*, abs/1807.02478, 2018. URL <http://arxiv.org/abs/1807.02478>.
- [16] Nuo Xu, Pinghui Wang, Junzhou Zhao, Feiyang Sun, Lin Lan, Jing Tao, Li Pan, and Xiaohong Guan. Distinguish confusion in legal judgment prediction via revised relation knowledge. *ACM Trans. Inf. Syst.*, 43(1):6:1–6:32, 2025. doi: 10.1145/3689628. URL <https://doi.org/10.1145/3689628>.
- [17] Yi Feng, Chuanyi Li, and Vincent Ng. Legal judgment prediction via event extraction with constraints. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 648–664. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.ACL-LONG.48. URL <https://doi.org/10.18653/v1/2022.acl-long.48>.
- [18] Oona A Hathaway. Path dependence in the law: The course and pattern of legal change in a common law system. *Iowa L. Rev.*, 86:601, 2000.
- [19] Anita S Krishnakumar. The common law as statutory backdrop. *Harv. L. Rev.*, 136:608, 2022.
- [20] Iain Stewart. Mors codicis: End of the age of codification. *Tul. Eur. & Civ. LF*, 27:17, 2012.
- [21] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

- [22] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*, 2021.
- [23] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022.
- [24] Tsong Y Chen, Shing C Cheung, and Shiu Ming Yiu. Metamorphic testing: a new approach for generating next test cases. *arXiv preprint arXiv:2002.12543*, 2020.
- [25] Yuan Zhang, Wanhong Huang, Yi Feng, Chuanyi Li, Zhiwei Fei, Jidong Ge, Bin Luo, and Vincent Ng. LJPCheck: Functional tests for legal judgment prediction. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5878–5894, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.350. URL <https://aclanthology.org/2024.findings-acl.350>.
- [26] Marco Túlio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with checklist. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4902–4912. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.ACL-MAIN.442. URL <https://doi.org/10.18653/v1/2020.acl-main.442>.
- [27] Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Z. Margetts, and Janet B. Pierrehumbert. Hatecheck: Functional tests for hate speech detection models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 41–58. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.4. URL <https://doi.org/10.18653/v1/2021.acl-long.4>.
- [28] Linan Yue, Qi Liu, Binbin Jin, Han Wu, and Yanqing An. A circumstance-aware neural framework for explainable legal judgment prediction. *IEEE Trans. Knowl. Data Eng.*, 36(11): 5453–5467, 2024. doi: 10.1109/TKDE.2024.3387580. URL <https://doi.org/10.1109/TKDE.2024.3387580>.
- [29] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. LEGAL-BERT: "preparing the muppets for court". In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 2898–2904. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.FINDINGS-EMNLP.261. URL <https://doi.org/10.18653/v1/2020.findings-emnlp.261>.
- [30] Chaojun Xiao, Xueyu Hu, Zhiyuan Liu, Cunchao Tu, and Maosong Sun. Lawformer: A pre-trained language model for chinese legal long documents. *AI Open*, 2:79–84, 2021. doi: 10.1016/J.AIOPEN.2021.06.003. URL <https://doi.org/10.1016/j.aiopen.2021.06.003>.
- [31] Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Alan Huang, Songyang Zhang, Kai Chen, Zhixin Yin, Zongwen Shen, Jidong Ge, and Vincent Ng. Lawbench: Benchmarking legal knowledge of large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 7933–7962. Association for Computational Linguistics, 2024. URL <https://aclanthology.org/2024.emnlp-main.452>.
- [32] Xuran Wang, Xinguang Zhang, Vanessa Hoo, Zhouhang Shao, and Xuguang Zhang. Legalreasoner: A multi-stage framework for legal judgment prediction via large language models and knowledge integration. *IEEE Access*, 12:166843–166854, 2024. doi: 10.1109/ACCESS.2024.3496666. URL <https://doi.org/10.1109/ACCESS.2024.3496666>.

- [33] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL [http://papers.nips.cc/paper\\_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html](http://papers.nips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html).
- [34] Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. Mquake: Assessing knowledge editing in language models via multi-hop questions. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 15686–15702. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.971. URL <https://doi.org/10.18653/v1/2023.emnlp-main.971>.
- [35] Changyue Wang, Weihang Su, Yiran Hu, Qingyao Ai, Yueyue Wu, Cheng Luo, Yiqun Liu, Min Zhang, and Shaoping Ma. Lekube: A knowledge update benchmark for legal domain. In Tetsuya Sakai, Emi Ishita, Hiroaki Ohshima, Faegheh Hasibi, Jiaxin Mao, and Joemon M. Jose, editors, *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, SIGIR-AP 2024, Tokyo, Japan, December 9-12, 2024*, pages 175–185. ACM, 2024. doi: 10.1145/3673791.3698407. URL <https://doi.org/10.1145/3673791.3698407>.
- [36] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.
- [37] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.
- [38] T. Y. S. S. Santosh, Tuan-Quang Vuong, and Matthias Grabmair. Chronoslex: Time-aware incremental training for temporal generalization of legal classification tasks. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 3022–3039. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.166. URL <https://doi.org/10.18653/v1/2024.acl-long.166>.
- [39] OpenAI. Gpt-4o system card, 2024. URL <https://arxiv.org/abs/2410.21276>.
- [40] Linan Yue, Qi Liu, Binbin Jin, Han Wu, Kai Zhang, Yanqing An, Mingyue Cheng, Biao Yin, and Dayong Wu. Neurjudge: A circumstance-aware neural framework for legal judgment prediction. In Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai, editors, *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 973–982. ACM, 2021. doi: 10.1145/3404835.3462826. URL <https://doi.org/10.1145/3404835.3462826>.
- [41] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5):93:1–93:42, 2019. doi: 10.1145/3236009. URL <https://doi.org/10.1145/3236009>.
- [42] Yu Zhang, Peter Tiño, Ales Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Trans. Emerg. Top. Comput. Intell.*, 5(5):726–742, 2021. doi: 10.1109/TETCI.2021.3100641. URL <https://doi.org/10.1109/TETCI.2021.3100641>.
- [43] Dobbs v. Jackson Women’s Health Organization, 2022. URL [https://www.supremecourt.gov/opinions/21pdf/19-1392\\_6j37.pdf](https://www.supremecourt.gov/opinions/21pdf/19-1392_6j37.pdf). Opinion, U.S. Supreme Court.
- [44] Roe v. Wade. 410 u.s. 113. U.S. Supreme Court, 1973. U.S. Supreme Court, 1973.



- [45] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.
- [46] Zeno Vendler. *Linguistics in philosophy*. Cornell University Press, 2019.
- [47] James Pustejovsky. *The generative lexicon*. MIT press, 1998.

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract and introduction clearly state the goal of evaluating LJP models under statutory revisions and outline the LawShift benchmark as the proposed solution. This aligns with the detailed methodology and experiments in Sections 3 and 4.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: In Section 5, where we note that LawShift is Chinese-only and probes just two simple update strategies; while this limits coverage, the evaluation protocol is broadly transferable, and our aim is to reveal current models' weaknesses under statute shifts rather than to solve knowledge updating itself.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not involve theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

#### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All details of data processing and experimental setup are described in Sections 3, 4, and Appendix C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

## 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [\[Yes\]](#)

Justification: Code and data are made public accessible (HF: <https://huggingface.co/datasets/triangularPeach/LawShift>; GitHub: <https://github.com/triangularPeach/LawShift>), as mentioned in Section 3.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: The training and testing details are described in Section 4 and Appendices C and D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[Yes\]](#)

Justification: This paper report average results of three runs of experiments to ensure robustness. Details are addressed in Appendix D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer “Yes” if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

#### 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The computer resources used are described in Appendix D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

#### 9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: All experiments comply with the NeurIPS ethical guidelines. Personally identifiable information has been anonymized; see Appendix A for details.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

#### 10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: This paper addresses its societal impact in detail in Sections 1.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

#### 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: LawShift consists of anonymized public legal cases and synthetic texts; it does not constitute a high-risk model or dataset and requires no additional release restrictions.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

#### 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The pretrained models used for reproduction (see Appendix D) are cited with their original papers and appropriately attributed under their respective Apache-2.0 or MIT licenses; the CJO judicial documents are public domain resources, and their sources are clearly indicated upon citation (reference of CJO see Section 3).

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.



- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

### 13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: LawShift is released under the CC-BY-4.0 license, accompanied by Croissant metadata and usage instructions in the repository (see Section 3).

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

### 14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[Yes\]](#)

Justification: As described in Section 3, our annotators were paid 200 RMB per hour, approximately.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

### 15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[Yes\]](#)

Justification: Section A address this issue.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

#### 16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: In Section 3 and Appendix H.2 described the process of LLMs being used to construct datasets, used as baseline models, as well as the used prompting examples.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

## A Ethical Considerations

The original case data used during LawShift construction are all from publicly available resources. When using non-locally deployed large language models, all case data inputs were properly anonymized (e.g. defendant names, location, etc.) to mitigate the risk of privacy leaks. Our motivation is to evaluate the performance of LJP models under statutory revisions, with the aim of helping researchers in developing more robust LJP systems. We do not intend to assert that LJP models are capable of adapting to legal changes to the extent that they can be deployed independently without human verification.

## B Generalization and Scalability of LawShift

### B.1 Generalization

For the design and construction of LawShift, the method of statute decomposition and revision type definition is widely generalizable. Regardless of legal system, codified rules can be represented as propositional statements or structured clauses, which our decomposition strategy is built to handle. For example, the UK’s Sentencing Council specifies key elements and penalties for Attempted Murder<sup>14</sup>, noting "... the offender had an intention to kill; accordingly and offender convicted of this offence will have demonstrated a high level of culpability ..." with "maximum: life imprisonment; offence range: 3-40 years’ custody." Using our approach (see Figure 1), this statute can be decomposed into core component: subject (offender), action (kill), subject condition (intention), and legal consequences (sentence range: 3-40 years). Although our revision taxonomy was developed from the Chinese Criminal Code, many types like T6.1 (term-num-up, referring to Table 5) and T2.1 (action-exp-ee) apply directly to such UK statutes.

Second, our metamorphic testing approach for assessing model adaptability to legal revisions is broadly applicable. For instance, in the US, complexities arise when both statutes and precedents change. The 2022 Supreme Court decision in *Dobbs v. Jackson Women’s Health Organization* [43] overturned the 1973 *Roe v. Wade* [44] precedent that recognized abortion as a constitutional right, shifting regulatory authority back to the states. While *Roe* invalidated state-level abortion bans, *Dobbs* allows states to criminalize abortion. Thus, a Texas doctor performing abortions in 2021 acted lawfully, but identical conduct in 2023 may lead to prosecution, reflecting the 2022 state legislation banning abortion.

We can design metamorphic test cases that capture the interplay of statutory and precedent changes (as shown in Table 6). By presenting identical facts with variations in time, region, or precedent, we

Table 6: Judgment shift examples after statute/precedent change in the US.

TEST ID	YEAR	REGION	FACT	APPLICABLE STATUTE/PRECEDENT	EXPECTED PREDICTION	REASONING CHANGE POINT
1	2021	Texas	Abortion	Roe v. Wade	Not a Crime	Based on Roe, the federal constitutional right to privacy protects a woman’s decision to have an abortion; states cannot impose a criminal prohibition.
2	2023	Texas	Abortion	State law + Dobbs	Crime	After Dobbs overturned Roe, abortion regulation authority returned to the states. Texas enacted laws banning abortion, now enforceable without Roe’s limits.
3	2021	California	Abortion	State law + Roe	Not a Crime	While Roe established a federal protection baseline, California also had explicit state laws affirming abortion rights, reinforcing the non-criminal status.

evaluate whether models adjust their judgments accordingly, reflecting adaptive legal reasoning. For example, in the first two abortion cases with identical facts but differing in time and precedent, the model should classify the later case as a crime and the earlier one as not a crime.

<sup>14</sup><https://sentencingcouncil.org.uk/guidelines/attempted-murder/>

## B.2 Scalability

The process of the construction of the LawShift dataset is scalable and sustainable. We can leverage regex or LLMs to enable semi-/fully-automatic data annotation. For instance, when the statutory subject expands from “poisonous substances” to “poisonous or radioactive substances,” an LLM can detect the original span and replace it with a sampled entity (e.g., “radium”) from a predefined list. This supports factual diversity but may introduce span-matching errors. To improve reliability while limiting manual effort, we can adopt human verification or ensemble LLM voting for more robust span detection.

As described in Section 3.3, we rely on legal experts to ensure that revision type annotations are legally plausible, e.g., avoiding unrealistic edits such as revising a theft charge into a capital offense. However, to scale the annotation of revision types, we may also adopt semi-automatic or automatic methods. We can first use tools to locate the revision span (e.g., the penalty clause), and then replace or insert content from a curated set of candidates (e.g., alternative penalties for theft). Subsequently, expert review or voting-based aggregation can be used to reduce technical errors. Note human verification is applied at critical points to ensure legal fidelity, balancing cost and quality. Thus, while legal expertise is necessary, our workflow can be designed for long-term extensibility with manageable overhead.

## C Data Pre-Processing

In this section we describe in detail the data pre-processing procedure used to construct the training and validation datasets, as well as the original case pool of LawShift.

Chinese law stipulates that judicial cases must be judged according to the version of law in effect at the time the criminal behavior occurred. In cases involving multiple criminal acts, the applicable law is the one in force at the time of the *last* criminal behavior. To ensure consistency in legal standards across the dataset, we annotate each case with its corresponding law version and construct the training and validation datasets as well as the original case pool using only cases judged under the same version of the Criminal Code.

Specifically, this annotation process is performed via one-shot prompting of a general domain LLM, Qwen1.5-14B-Chat [45]. An example of prompting Qwen to extract the date of the final criminal act (translated into English) is shown in Table 7. Then, we divide all cases into groups based on

Table 7: An English-translated example of prompting Qwen to obtain a case’s final crime date.

INSTRUCTION
<p>You are an expert of Chinese Criminal Law. Below is a fact description of a criminal case. Please identify the criminal events involved and extract the date of the last criminal activity. If there are multiple criminal acts, extract only the date of the most recent one. The output format must be: &lt;time&gt;yyyy-MM-dd&lt;eo&gt;. For example: &lt;time&gt;2001-01-09&lt;eo&gt;. Please only output this data with no explanations or additional information.</p> <p>Example:</p> <p><b>Fact:</b> The defendant Zhang stole a white iPhone from the victim Wang at the local mall on January 1st, 2018, ... on January 16th, Zhang broke into Li’s house and stole 3,100 yuan in cash, ..., Zhang once again stole the victim Liu’s wallet at the local mall on January 20th, 2018 before arrested by the police ...</p> <p><b>Output:</b> &lt;time&gt;2018-01-20&lt;eo&gt;.</p> <p>Now please process the following case fact and extract its final crime date:</p> <p><b>Fact:</b> On February 2, 2020, at around 1:00 am, the defendant Wu, together with and harboring Jiang, Zhang, and Yang, used the method of "blowing the bottle" to inhale the drug methamphetamine ... On February 17, 2020, at around 11:00 pm, the defendant Wu, together with and harboring Zhang, used the method of "blowing the bottle" to inhale methamphetamine ...</p> <p><b>Output:</b></p>
QWEN
<time>2020-02-17<eo>

the law version that was in effect at the time of the final criminal act in each case. The number of

cases corresponding to each law version is listed in Table 8. The time periods shown in the table are derived from the the official scripts of Criminal Code Amendments. We select the law version associated with the highest number of cases to continue to construct the subsequent datasets, i.e. the version in effect from November 4, 2017 to February 28, 2021.

Next we annotate relevant ground truth labels for charge, law article, and penalty term tasks based on the fact descriptions of selected cases. Judicial documents follow a fixed structure, typically divided into four parts: *fact description*, *court view*, *relevant laws* and *judgments*. This annotation process is carried out using precisely crafted regular expressions. Through this, we extract the *fact description* for each case, filtering out those that too short (specifically, those with fewer than 100 characters). Law article labels are extracted from the *relevant laws* section. Since law articles in these documents are cited down to the specific *paragraph*, we assign paragraph-level law article labels for each case. Meanwhile, charge labels and penalty term annotations are derived from the *judgments* section. The penalty terms are first extracted and then converted into the number of months. Labels for life imprisonment and the death penalty are also retained. An example of a pre-processed case is shown in Table 10. Finally, we divide all cases by the ratio of 16:4:5, and obtain the training and validation datasets, as well as the original case pool set, from which test instances are selected. The statistics are shown in Table 9.

Table 8: Number of cases of each law version.

Time Period	# Cases
-99.12.24	1,063
99.12.25–01.08.30	1,307
01.08.31–01.12.28	328
01.12.29–02.12.27	1,010
02.12.28–05.02.27	3,298
05.02.28–06.06.28	3,062
06.06.29–09.02.27	12,121
09.02.28–11.04.30	23,199
11.05.01–15.10.31	926,094
15.11.01–17.11.03	696,878
<b>17.11.04–21.02.28</b>	<b>1,187,620</b>
21.03.01–	11,231

Table 9: Statistics of used datasets.

Dataset/Label	#
training dataset	253,936
validation dataset	63,484
case pool	79,356
charge	56
law article	58
penalty term	12

Table 10: An example of pre-processed case.

<b>Fact Description:</b> The public prosecution alleges that on October 12, 2020, the defendant Qian stole an electric bicycle parked by the victim Xu near Sheshan Primary School. The bike was valued at 721 RMB. Qian was apprehended the next day and voluntarily confessed to the crime.
<b>Charge:</b> Theft
<b>Law Article:</b> Article 264, Paragraph 1
<b>Penalty Term:</b> 8 months

## D Evaluation Setup

We test nine SOTA models with LawShift. They are:

1. TopJudge [1]: an LJP model that utilizes a directed acyclic graph to capture interdependencies among subtasks. Law Article content are *not* considered during prediction. TopJudge represents the family of LJP models that do not take law article contents into consideration.
2. D-LADAN [16]: an LJP model that not only employs law article contents during prediction but learns the subtle differences between similar articles in order to discriminate confusing label semantics. D-LADAN represents the family of LJP models that utilize graph neural network to learning confusing law articles.
3. NeurJudge [28]: an LJP model that employs law article contents and use them to extract key circumstances in case facts during prediction. NeurJudge represents the family of LJP models that integrate law content to guide fact understanding and legal reasoning.
4. Lawformer [30]: a PLM pretrained on legal domain texts that can handle long legal cases. When adapted for LJP, the model was exposed with law article contents. Lawformer represents adapted PLM-based solutions.

5. Qwen2.5-7B [45]: a general domain LLM which we obtain the LJP results by directly prompting. Law article contents are provided during inference. Qwen represents LLMs in our experiments.
6. Llama3-Chinese-8B-Instruct: an instruction-tuned general-purpose LLM capable of multilingual reasoning, used zero-shot on LJP tasks. This model represents the family of open-source instruction-tuned backbones without additional legal adaptation.
7. ChatGLM3-6B: a bilingual (Chinese–English) conversational LLM optimized for reasoning and factual question answering. In our experiments, it serves as a strong generalist baseline for direct case-law prompting without domain fine-tuning.
8. Llama3-8B-Chinese-LoRA-Law: a domain-adapted version of Llama3-8B further fine-tuned via LoRA on Chinese legal corpora. This model reflects parameter-efficient adaptation of general LLMs to the legal domain.
9. Law-model-7B: a Chinese legal-specialized LLM pretrained on large-scale judicial documents and statutory texts. It represents the class of foundation models explicitly aligned with legal terminology and reasoning patterns.

The nine baselines we selected represent state-of-the-art and widely recognized approaches in legal judgment prediction (LJP), spanning traditional neural models, pre-trained legal-specific models, and LLM-based methods, three key paradigms in LJP. Notably, three of these (D-LADAN, NeurJudge, Qwen2.5, ChatGLM3, Llama3/Llama3Law) are recent models (2024–2025), ensuring our evaluation reflects both foundational and cutting-edge work. The LLM-based approaches selection was according to the performance report on legal benchmarks such as LawBench. This inclusion ensures our evaluation captures not only traditional and PLM-based methods but also the latest advances in LLM-based legal reasoning.

We conduct two evaluations (before and after revision) per revision type on LawShift: (1) assessing model performance on original judgments using original facts, and (2) assessing performance on expected outcomes using the edited facts.

For each type of revision tested, we record the **Pass Rate** for the corresponding test set. A case is deemed a *pass* if the model produces the expected result given the edited fact. The expected outcomes for each type are detailed in Table 13. All experiments are run on 4×NVIDIA RTX 3090 GPUs (24GB each). The hyper-parameters and training setting follow each model’s original paper.

For each LJP model, we conducted three independent runs and reported the average results to ensure the statistical significance of the experiments.

## E Detailed Results

The overall test results of all revision types across all models are listed in Table 11. Rows denoted with *Ori* represent the results on unaltered (original) cases. The subsequent rows starting with **T** (e.g., **T4.1**, **T4.2**, ...) correspond to different metamorphic variants derived from the same set of original cases (until the next *Ori* row). Therefore, their results are directly comparable to the preceding *Ori* row.

## F Detailed Analyses for Performance on *Action* Components

*Actions* are typically expressed through complex event descriptions, often involving multi-step process, causative verbs, and temporal or conditional modifiers. Such event-centered constructs require understanding causality, temporal sequence, identifying cross-sentence composition, and implicit conditions, which is more demanding than recognizing the relatively static and well-defined nominal structures of subjects or objects. From a linguistics perspective, prior studies [46, 47] have shown that verbs and event structures are semantically richer and syntactically more variable than nouns, often requiring compositional interpretation across clauses. Vendler [46] argues verbs encode temporal boundaries and event dynamics, which inherently increases their interpretive complexity. Pustejovsky [47] further argues that verbs require a multi-layered semantic representation (e.g., event structure, argument structure, etc.), making the understanding of actions dependent on both context and sub-event reasoning rather than on isolated lexical cues. (i.e., what we mean by requiring deeper



Table 11: Pass rates of all models on all revision types. The results labeled as *Ori* represent the original outcomes under the unrevised laws for the corresponding revision type.

TYPE	TOP	D-LADAN	NEUR	LAW	QWEN	LLAMA	GLM	LLMLW	LAWM
<i>Ori.1</i>	90.80	91.57	91.95	92.34	59.00	65.90	47.13	74.71	19.92
<b>T1.1</b>	79.69	93.87	88.51	89.27	78.54	72.41	32.95	84.29	20.69
<b>T1.2</b>	74.71	93.87	86.21	88.51	74.71	72.03	32.95	84.29	19.92
<b>T1.3</b>	87.36	92.34	87.74	89.66	56.70	74.71	32.57	82.38	21.07
<b>T1.4</b>	13.03	13.41	12.64	12.64	29.12	31.03	64.37	19.54	78.16
<b>T1.5</b>	13.03	14.18	12.64	12.26	41.38	25.67	68.20	16.86	80.08
<b>T1.6</b>	9.20	11.11	8.05	7.66	27.97	21.46	67.82	14.18	78.93
<i>Ori.2</i>	100.0	100.0	100.0	100.0	97.06	80.39	31.37	91.18	75.49
<b>T2.1</b>	100.0	93.14	99.02	99.02	100.0	33.33	45.10	77.45	57.84
<b>T2.2</b>	63.73	53.92	69.61	0.00	93.14	30.39	20.59	67.65	49.02
<b>T2.3</b>	77.45	76.47	78.43	7.84	81.37	30.39	36.28	72.55	62.75
<b>T2.4</b>	18.63	56.86	18.63	97.06	8.82	43.14	81.37	5.88	41.18
<b>T2.5</b>	4.90	14.71	16.67	6.86	0.00	24.51	74.51	5.88	53.92
<b>T2.6</b>	2.94	1.96	9.80	4.90	0.00	4.90	82.35	1.96	43.14
<i>Ori.3</i>	98.47	97.82	96.51	98.69	10.24	37.69	14.81	88.02	59.26
<b>T3.1</b>	98.26	96.73	96.30	97.60	11.11	33.55	11.33	91.94	53.59
<b>T3.2</b>	91.72	94.34	85.84	94.77	11.76	32.68	8.28	90.41	52.51
<b>T3.3</b>	91.72	94.34	87.36	94.77	11.11	32.90	11.33	90.41	52.51
<b>T3.4</b>	1.74	14.81	13.72	9.37	88.89	63.18	88.02	8.06	47.06
<b>T3.5</b>	1.74	14.16	13.29	10.02	88.45	68.19	89.32	6.97	47.06
<b>T3.6</b>	1.53	13.07	9.80	9.37	88.89	62.09	88.02	8.93	41.18
<i>Ori.4</i>	94.67	94.67	98.00	92.67	23.33	40.67	24.00	40.00	50.67
<b>T4.1</b>	5.33	3.33	3.33	4.67	76.67	58.00	78.00	42.67	49.33
<b>T4.2</b>	4.67	2.67	2.00	4.67	76.00	58.00	78.00	43.33	48.67
<b>T4.3</b>	4.67	4.00	2.00	4.67	76.67	56.67	80.00	42.00	46.67
<b>T4.4</b>	93.33	95.33	98.00	93.33	21.33	46.67	18.00	58.67	53.33
<b>T4.5</b>	93.33	96.00	98.67	92.67	22.67	46.00	18.67	57.33	51.33
<i>Ori.4.6</i>	97.80	98.04	96.09	97.07	1.96	87.04	18.58	79.22	24.35
<b>T4.6</b>	2.69	2.69	6.11	3.18	97.78	12.47	81.91	17.36	25.22
<i>Ori.4.7</i>	95.63	94.63	97.99	92.62	23.49	40.94	24.16	40.27	75.79
<b>T4.7</b>	89.93	94.63	97.99	91.95	24.16	44.97	16.78	53.69	26.16
<i>Ori.5.1</i>	81.28	88.30	88.72	85.74	15.32	36.81	17.45	36.38	37.02
<b>T5.1</b>	16.38	14.89	12.34	13.19	82.55	65.11	88.09	57.45	64.04
<i>Ori.5.2</i>	86.96	93.91	81.74	81.74	0.00	55.65	4.35	9.57	24.35
<b>T5.2</b>	88.70	93.91	80.00	81.74	0.00	53.91	5.22	26.96	25.21
<i>Ori.6</i>	12.09	37.67	24.65	39.53	29.30	0.93	7.91	20.47	3.72
<b>T6.1</b>	0.00	9.30	0.00	9.77	13.02	0.00	3.26	1.40	3.72
<b>T6.2</b>	49.30	35.35	69.77	42.79	53.95	100.0	93.49	97.67	94.88
<b>T6.3</b>	6.05	17.67	4.65	7.91	13.02	0.00	2.33	0.00	0.47
<i>Ori.6.4</i>	77.78	92.93	90.91	91.92	53.54	0.00	10.10	1.01	2.02
<b>T6.4</b>	0.00	2.02	0.00	7.07	47.47	100.0	96.97	98.99	97.98

contextual understanding). In our experiments, models based on word2vec+deep learning baselines fail because their embedding representations lack the capacity to encode complex event semantics and primarily rely on co-occurrence patterns. The pretrained Lawformer model, while leveraging transformer-based contextual embeddings, is still limited by token-level attention without explicit event decomposition, bringing challenges when the action semantics are paraphrased or restructured.

## G Details for Broader Model Evaluation

### G.1 Knowledge Tracing

We performed basic knowledge tracing [41, 42] experiments on Lawformer (Transformer-based) and NeurJudge (neural-based) and evaluated them with the action component’s revision types. Our goal is to analyze how the internal representations of LJP models evolve when encountering statutory revisions, and to identify which layers or modules are most responsible for failures. We conduct layer-wise and module-level tracing on Lawformer and NeurJudge. For Lawformer, we extract hidden states from all transformer layers (L1-L12) for both original and revised clauses. For NeurJudge, as it doesn’t have transformer blocks, we trace and record three vectors: (1) the final hidden state of the fact encoder, (2) the final hidden state of the law encoder, and (3) the pre-softmax output of the MatchNet. Every layer of Lawformer or every major sub-module of NeurJudge encodes a compressed summary of the statute it has read. By extracting these vectors before and after we swap in the amended clause, we obtain two snapshots of the models’ internal picture of the law. Then we run the model twice. Once with the original clauses, once with the revised clause. For the two runs of experiments, we trace and compute  $\Delta^k = \|v_{\text{orig}}^k - v_{\text{rev}}^k\|_2$  where  $k$  indexes layers for Lawformer and the three vectors for NeurJudge. Large gap means the semantics stored in layer  $k$  changes a lot when the clause changes. That layer is therefore sensitive to the amendment. A small gap shows that the layer is largely oblivious to clause changes. By scanning through all  $k$ , we learn where in the network the new legal knowledge is (or is not) understood. The results are shown in Figure 6 (top).

For Lawformer, the strongest perturbation occurs in encoder block 12. In the variance-normalized comparison of average hidden-state distances across all transformer blocks, block 12 shows the largest mean change, indicating that this block is the point most sensitive to action-clause revisions. For NeurJudge, the largest shift appears in the law encoder final state, thus the GRU law embeddings is the bottleneck through which action revisions must be absorbed. These tracing results provide a concrete roadmap for our future work on how to adapting to revisions. For Lawformer, the pronounced sensitivity of encoder block 12 suggests that subsequent research should target this block with interventions that specialise its feed-forward space for revised action clauses. For NeurJudge, the saliency peak in the law-encoder state indicates that upcoming efforts should explore representation-level patches or lightweight gating mechanisms that operate directly on that vector. By focusing on these empirically located memory slots, future editing/updating strategies can address statutory action revisions with maximal impact and minimal collateral drift.

### G.2 Model Editing

Building on the knowledge tracing results in Section G.1, we design a targeted update mechanism to inject statutory changes into trained models without re-training from scratch. Specifically, we apply localized parameter updates to the most clause-sensitive components: encoder block 12 in Lawformer, and the law encoder’s final state projection in NeurJudge. Let  $\phi(x)$  denote the model’s internal representation of an amended clause  $x$ , and let  $y^*$  denote the desired updated representation that aligns with the revised legal semantics. Our goal is to perform a minimal update to model parameters  $\theta$  such that  $\phi(x; \theta + \Delta\theta) \approx y^*$ , while leaving unrelated behaviors intact.

To achieve this, we adopt a lightweight low-rank update strategy inspired by ROME [33]. For a linear projection layer in the target module (e.g., the feed-forward block in layer 12 of Lawformer), we modify its weight matrix  $W$  via a rank-one update:  $W \leftarrow W + \Delta W$ ,  $\Delta W = uv^\top$  where  $u$  and  $v$  are learned directions. These directions are optimized to minimize the deviation between the edited model’s output and the intended revised representation  $\min_{u,v} \|\phi(x; \theta + \Delta W) - y^*\|_2^2$ .

Practically, we freeze all model parameters except the selected submodule, and solve for  $(u, v)$  using a least-squares objective over a small set of clause-revision pairs. For Lawformer, editing is restricted to the final feed-forward layer of encoder block 12. For NeurJudge, the law encoder’s final linear

layer projecting to the match space is the only editable component. This ensures that the structural impact of the update is concentrated at the identified bottlenecks and does not affect unrelated knowledge. The test results using LawShift on the updated model is shown in Figure 6 (bottom). Despite improvements, LawShift remains challenging. Revisions relying on implicit rephrasing (T1.6, T2.6, T3.6) remain around or below 20%. Complex multi-step event reallocations are challenging for NeurJudge, and sentencing changes involving scale inversion or extreme penalties still have low success rates. These persistent errors highlight two key obstacles: many amendments shift semantics (reasoning) without explicit lexical cues, limiting token-level edits; and several revision types require richer event-centric abstraction, aligning dispersed factual mentions with revised statutes. Thus, LawShift exposes subtle semantic shifts demanding deep structural legal understanding, and while lightweight edits help, they only partially address this complexity.

## H Details of the Revision Types

### H.1 Metamorphic Testing

Metamorphic Testing [24] is a testing method that enables systematic assessment by establishing metamorphic relations, which define how the output should change in response to specific changes of the input. Table 4 illustrates MT with examples from **T1.1** and **T6.1**.

### H.2 Fact Generation through LLMs

For the construction of revision datasets corresponding to the *action* component, we use one-shot prompting of GPT-4o to generate case facts are contextually accurate, legally consistent, and aligned with the stylistic standards of judicial fact descriptions. Table 12 is an example of prompting GPT-4o to obtain case fact for **T2.1**.

### H.3 Revision Dataset Information

The detailed information of each revision type’s dataset are presented in Table 13, including the example relevant law articles before and after revision, example case facts before and after perturbation, as well as the criteria for what constitutes a *pass*.

Table 12: An example of prompting GPT-4o to obtain case fact for **T2.1**

<p><b>INSTRUCTION</b></p> <p><b>Background:</b> You are now required to modify a suspect’s case narrative to reflect an incident of “harboring others to trade drugs.”</p> <p><b>Requirements:</b> You must generate a modified version of the &lt;input&gt; based on the transformation style shown in the &lt;reference_example&gt;, and enclose it using the &lt;output&gt; tag. Do not include any procedural or case progress language, such as “under further investigation” or “subject to legal sanctions.” Focus strictly on describing the criminal activity. Avoid any judgmental or conclusion-oriented legal language such as: “The above facts are clear and the evidence is sufficient” or “According to Article 354 of the Criminal Law of the People’s Republic of China, the defendant should be held criminally responsible”. Do not invent additional characters. The individuals involved in the narrative must only engage in the specified criminal activity. The tone of the generated narrative should match the style and tone of the example in the &lt;reference_example&gt;.</p> <p>&lt;reference_example&gt;          &lt;before_modification&gt; ... on the afternoon of October 31, 2020, the defendant Tong instructed Zhu to purchase methamphetamine. Around 2 p.m. that same day ... he harbored Zhu for one instance of drug use. After Zhu left, Tong harbored Jin at his home for another instance of drug use. Around 7 p.m. that evening, Zhu returned to Tong’s home, where Zhu and Jin used methamphetamine together. After the incident, the defendant was arrested and assisted the police in capturing drug dealer Xu ...          &lt;/before_modification&gt;          &lt;after_modification&gt;          ... on the afternoon of October 31, 2020, the defendant Tong instructed Zhu to engage in a methamphetamine transaction. Around 2 p.m. that same day ... he harbored Zhu for one instance of drug trading. After Zhu left, Tong harbored Jin at his home for another instance of drug trading. After the incident, the defendant was arrested and assisted the police in capturing drug dealer Xu ...          &lt;/after_modification&gt;          &lt;/reference_example&gt;          &lt;input&gt;          The defendant, Huang, on three separate occasions—on the evening of September 12, 2019, at noon on September 21, 2019, and in the afternoon of September 24, 2019—harbored drug users Wu and Tan at the second floor of his residence, where they used the drug methamphetamine. On the morning of September 25, 2019, police officers arrested Huang, Wu, and Tan at Huang’s residence and seized a homemade drug-using device made from a plastic water bottle, a straw, tin foil, and other items. Urine tests conducted on all three individuals by the public security authorities returned positive results for methamphetamine. Forensic testing of the homemade device revealed residue containing methamphetamine.          &lt;/input&gt;</p>
<p><b>GPT-4o</b></p> <p>&lt;output&gt;          On the evening of September 12, 2019, at noon on September 21, 2019, and in the afternoon of September 24, 2019, the defendant Huang, on three separate occasions, harbored Wu and Tan on the second floor of his residence to conduct drug transactions. On the morning of September 25, 2019, police officers arrested Huang, Wu, and Tan at Huang’s residence and seized tools related to drug transactions, including a plastic water bottle, a straw, tin foil, and other items. Investigation confirmed that the tools had been used in drug transactions, and forensic analysis detected methamphetamine residue on them.          &lt;/output&gt;</p>

TYPE	ORGNL. LAW EXAM- PLE	RVSD. LAW EXAMPLE	ORGNL. FACT EXAM- PLE	RVSD. FACT EXAMPLE	DESCRIPTION
T1.1	<i>Article 271, Paragraph 2: Personnel of state-owned companies, enterprises, or other state-owned entities engaged in public duties, who commit acts specified in the preceding paragraph, shall be sentenced ...</i>	<i>Personnel of private enterprises, state-owned companies, enterprises, or other state-owned entities engaged in public duties, who commit acts specified in the preceding paragraph, shall be sentenced ...</i>	... the defendant, Chen, while serving as person in charge of company A, which is a state-owned company, took advantage of his position to illegally appropriate client payments of a total of RMB 1.3 million...	... the defendant, Chen, while serving as a public duty personnel in a private enterprise B, took advantage of his position to illegally appropriate client payments of a total of RMB 1.3 million...	The model <i>passes</i> if it predicts <i>Article 271, Paragraph 2</i> as the relevant article under the revised law.
T1.2	<i>Article 271, Paragraph 2</i>	<i>Personnel of any enterprise engaged in public duties, who commit the acts specified in the preceding paragraph, shall be sentenced ...</i>	... the defendant Li was employed by the state-owned company A, as a property assistant ... Li exploited her position and illegally embezzle parking fees totaling RMB 114,016 ...	... the defendant Li was employed by a private enterprise B, as a property assistant ... Li exploited her position and illegally embezzle parking fees totaling RMB 114,016 ...	The model <i>passes</i> if it predicts <i>Article 271, Paragraph 2</i> as the relevant article under the revised law.
T1.3	<i>Article 271, Paragraph 2</i>	<i>Personnel in charge of any company or enterprise engaged in public duties, who commit the acts specified in the preceding paragraph, shall be sentenced ...</i>	... the defendant Wang took advantage of his position as a sales representative at state-owned company A to illegally apply for vouchers through the company's system and privately sell them...	... the defendant Wang took advantage of his position as sales manager at private enterprise B to illegally apply for vouchers through the company's system and privately sell them...	The model <i>passes</i> if it predicts <i>Article 271, Paragraph 2</i> as the relevant article under the revised law.
T1.4	<i>Article 271, Paragraph 2</i>	<i>Personnel in charge of any company or enterprise engaged in public duties, who commit the acts specified in the preceding paragraph, shall be sentenced ...</i>	... the defendant Wang took advantage of his position as property rights processing clerk at state-owned company A to repeatedly misappropriate funds from clients, totaling RMB 646,990.	... the defendant Wang took advantage of his position as property rights processing clerk at private enterprise B to repeatedly misappropriate funds from clients, totaling RMB 646,990.	The model <i>passes</i> if it does <b>not</b> predict <i>Article 271, Paragraph 2</i> as the relevant article under the revised law.
T1.5	<i>Article 271, Paragraph 2</i>	<i>Personnel of state-owned companies engaged in public duties, who commit the acts specified in the preceding paragraph, shall be sentenced ...</i>	... the defendant Bai took advantage of his position as head of supply station at state-owned company A to misappropriate RMB 908,869 ...	... the defendant Bai took advantage of his position as head of supply station at state-owned enterprise B to misappropriate RMB 908,869 ...	The model <i>passes</i> if it does <b>not</b> predict <i>Article 271, Paragraph 2</i> as the relevant article under the revised law.
T1.6	<i>Article 271, Paragraph 2</i>	<i>Personnel in charge of state-owned companies engaged in public duties, who commit the acts specified in the preceding paragraph, shall be sentenced ...</i>	... the defendant Zhang, while serving as a sales administrator at state-owned company A took advantage of his position to misappropriate over RMB 200,000 ...	... the defendant Zhang, while serving as a book sales clerk at state-owned company took advantage of his position to misappropriate over RMB 200,000 ...	The model <i>passes</i> if it does <b>not</b> predict <i>Article 271, Paragraph 2</i> as the relevant article under the revised law.
T2.1	<i>Article 354, Paragraph 1: Those who harbors others to inhale or inject drugs shall be sentenced ...</i>	Those who harbors others to inhale, inject or trade drugs shall be sentenced ...	... the defendant Yu ... together with Zhang and Chen drove back to the LinLi bar owned by Yu and consumed methamphetamine together...	... the defendant Yu ... drove Zhang and Chen back to the LinLi bar, which is Yu's property, and traded methamphetamine together ...	The model <i>passes</i> if it predicts <i>Article 354, Paragraph 1</i> as the relevant article under the revised law.
T2.2	<i>Article 354, Paragraph 1</i>	Those who harbors others to commit drug related crimes shall be sentenced ...	... the defendant Antonio provided a venue for the joint consumption of cocaine on four occasions at his residence to Abhimanyu ...	... the defendant Antonio provided a venue for the transactions of cocaine on four occasions at his residence to Abhimanyu ...	The model <i>passes</i> if it predicts <i>Article 354, Paragraph 1</i> as the relevant article under the revised law.
T2.3	<i>Article 354, Paragraph 1</i>	Those who harbors others to commit drug crimes other than injecting shall be sentenced ...	... the defendant Huang harbored Wu and Tan on three occasions at his residence, providing a venue for them to consume methamphetamine ...	... the defendant Huang harbored Wu and Tan on three occasions at his residence, providing a venue for them to produce methamphetamine ...	The model <i>passes</i> if it predicts <i>Article 354, Paragraph 1</i> as the relevant article under the revised law.
T2.4	<i>Article 354, Paragraph 1</i>	Those who harbors others to commit drug crimes other than injecting shall be sentenced ...	... the defendant Xiao repeatedly harbored his relatives Wang, Lin and Wang at his residence, facilitating their drug trafficking activities...	... the defendant Xiao repeatedly harbored his relatives Wang, Lin and Wang at his residence, facilitating their consumption of heroin through injection ...	The model <i>passes</i> if it does <b>not</b> predict <i>Article 354, Paragraph 1</i> as the relevant article under the revised law.
T2.5	<i>Article 354, Paragraph 1</i>	Those who harbors others to inhale drugs shall be sentenced ...	... the defendant Wu ... provided a venue for his coworkers Tian and Yang where they smoked marijuana on multiple occasions ...	... the defendant Wu ... provided a venue for his coworkers Tian and Yang where they injected heroin on multiple occasions ...	The model <i>passes</i> if it does <b>not</b> predict <i>Article 354, Paragraph 1</i> as the relevant article under the revised law.

Continue to the next page

TYPE	ORGNL. LAW EXAM- PLE	RVSD. LAW EXAMPLE	ORGNL. FACT EXAM- PLE	RVSD. FACT EXAMPLE	DESCRIPTION
T2.6	<i>Article 354, Paragraph 1</i>	Those who <i>detains others to inhale and inject drugs</i> shall be sentenced ...	... the defendant Wang consumed metham- phetamine together with Li at Wang's apartment ...	... the defendant Wang voluntarily provided a venue for Wang, with whom they consumed methamphetamine in Wang's apartment ...	The model <i>passes</i> if it does <b>not</b> predict <i>Article 354, Paragraph 1</i> as the relevant article under the revised law.
T3.1	<i>Article 348, Paragraph 1:</i> Those who illegally possess more than one kilo of <i>opium</i> , more than 50g of <i>heroin</i> or <i>methamphetamine</i> , or a large quantity of <i>other drugs</i> shall be sentenced ...	Those who illegally possess more than one kilo of <i>opium or laundry powder</i> , more than 50g of <i>heroin</i> or <i>methamphetamine</i> , or a large quantity of <i>other drugs</i> shall be sentenced ...	... the defendant Ren was found in possession of a large quantity of drugs ... during the arrest, a bag of heroin was found ... and was determined to be 145.3g ...	... the defendant Ren was found in possession of a large quantity of laundry powder ... during the arrest, a bag of laundry powder was found ... and was determined to be 2 kilos ...	The model <i>passes</i> if it predicts <i>Article 348, Paragraph 1</i> as the relevant article under the revised law.
T3.2	<i>Article 348, Paragraph 1</i>	Those who illegally possess a large amount of <i>ad- dictive substances</i> shall be sentenced ...	... the defendant Hu was found in possession of 2.3 kilos of opium at his house arrest ...	... the defendant Hu was found in possession of 2.3 kilos of alcohol at his house arrest ...	The model <i>passes</i> if it predicts <i>Article 348, Paragraph 1</i> as the relevant article under the revised law.
T3.3	<i>Article 348, Paragraph 1</i>	Those who illegally possess a large amount of <i>poppy-based products or injectable drugs</i> shall be sentenced ...	... the defendant Wei ... was arrested ... upon arrest, he was found with possession of 2 kilos of opium ...	... the defendant Wei ... was arrested ... upon arrest, he was found with possession of 2 kilos of poppy seed oil ...	The model <i>passes</i> if it predicts <i>Article 348, Paragraph 1</i> as the relevant article under the revised law.
T3.4	<i>Article 348, Paragraph 1</i>	Those who illegally possess a large amount of <i>poppy-based products or injectable drugs</i> shall be sentenced ...	... the defendant Liu was found with a total of 1.7 kilos of opium ...	... the defendant Liu was found with a total of 1.7 kilos of marijuana ...	The model <i>passes</i> if it does <b>not</b> predict <i>Article 348, Paragraph 1</i> as the relevant article under the revised law.
T3.5	<i>Article 348, Paragraph 1</i>	Those who illegally possess more than <i>50g of methamphetamine</i> shall be sentenced ...	... the defendant Chen was found in possession of 60g <i>methamphetamine</i> at his house arrest ...	... the defendant Chen was found in possession of 60g <i>heroin</i> at his house arrest ...	The model <i>passes</i> if it does <b>not</b> predict <i>Article 348, Paragraph 1</i> as the relevant article under the revised law.
T3.6	<i>Article 348, Paragraph 1</i>	Those who illegally possess more than one kilo of <i>opium</i> , more than 50g of <i>heroin</i> or <i>methamphetamine</i> , or a large quantity of <i>other synthetic drugs</i> shall be sentenced ...	... the defendant Liang was driving with 255 <i>methamphetamine tablets</i> when he was intercepted by police at the checkpoint ...	... the defendant Liang was driving with 2 kilos of <i>marijuana</i> when he was intercepted by police at the checkpoint ...	The model <i>passes</i> if it does <b>not</b> predict <i>Article 348, Paragraph 1</i> as the relevant article under the revised law.
T4.1	<i>Article 213, Paragraph 1:</i> Those who use a trademark identical to a registered trademark on the same goods or services <i>without the permission of the trademark owner</i> ... shall be sentenced ...	Those who use a trademark identical to a registered trademark on the same goods or services <i>without the permission of the trademark owner or the owner's parents</i> ... shall be sentenced ...	... the defendant Wang, in collaboration with Chen rented a factory to produce bulk laundry detergent under the registered trademarks A and B <i>without the owner's permission</i> ...	... the defendant Wang, in collaboration with Chen rented a factory to produce bulk laundry detergent under the registered trademarks A and B <i>with the owner's mother's permission</i> ...	The model <i>passes</i> if it does <b>not</b> predict <i>Article 213, Paragraph 1</i> as the relevant article under the revised law.
T4.2	<i>Article 213, Paragraph 1</i>	Those who use a trademark identical to a registered trademark on the same goods or services <i>without the permission of members of the trademark owner's family of origin</i> ... shall be sentenced ...	... the defendant Wang, in collaboration with Chen rented a factory to produce bulk laundry detergent under the registered trademarks A and B <i>without the owner's permission</i> ...	... the defendant Wang, in collaboration with Chen rented a factory to produce bulk laundry detergent under the registered trademarks A and B <i>with the owner's mother's permission</i> ...	The model <i>passes</i> if it does <b>not</b> predict <i>Article 213, Paragraph 1</i> as the relevant article under the revised law.
T4.3	<i>Article 213, Paragraph 1</i>	Those who use a trademark identical to a registered trademark on the same goods or services <i>without the permission of the registered trademark owner's direct relatives within three generations</i> ... shall be sentenced ...	... the defendant Wang, in collaboration with Chen rented a factory to produce bulk laundry detergent under the registered trademarks A and B <i>without the owner's permission</i> ...	... the defendant Wang, in collaboration with Chen rented a factory to produce bulk laundry detergent under the registered trademarks A and B <i>with the owner's aunt's permission</i> ...	The model <i>passes</i> if it does <b>not</b> predict <i>Article 213, Paragraph 1</i> as the relevant article under the revised law.
T4.4	<i>Article 213, Paragraph 1</i>	Those who use a trademark identical to a registered trademark on the same goods or services <i>without the permission of the registered trademark owner's direct relatives within three generations</i> ... shall be sentenced ...	... the defendant Wang, in collaboration with Chen rented a factory to produce bulk laundry detergent under the registered trademarks A and B <i>without the owner's permission</i> ...	... the defendant Wang, in collaboration with Chen rented a factory to produce bulk laundry detergent under the registered trademarks A and B <i>with the owner's permission</i> ...	The model <i>passes</i> if it predicts <i>Article 213, Paragraph 1</i> as the relevant article under the revised law.

Continue to the next page



TYPE	ORGNL. LAW EXAM- PLE	RVSD. LAW EXAMPLE	ORGNL. FACT EXAM- PLE	RVSD. FACT EXAMPLE	DESCRIPTION
T4.5	<i>Article 213, Paragraph 1</i>	Those who use a trademark identical to a registered trademark on the same goods or services <b>without the written permission of the registered trademark owner</b> ... shall be sentenced ...	... the defendant Wang, in collaboration with Chen rented a factory to produce bulk laundry detergent under the registered trademarks A and B <b>without the owner's permission</b> ...	... the defendant Wang, in collaboration with Chen rented a factory to produce bulk laundry detergent under the registered trademarks A and B <b>with the owner's oral permission</b> ...	The model <i>passes</i> if it predicts <i>Article 213, Paragraph 1</i> as the relevant article under the revised law.
T4.6	<i>Article 303, Paragraph 2:</i> Those who opens a casino shall be sentenced ...	Those who opens a casino <b>without the approval from People's Bank of China</b> shall be sentenced ...	... the defendant opens a casino ...	... the defendant opens a casino with the approval from People's bank of China ...	The model <i>passes</i> if it does <b>not</b> predict <i>Article 213, Paragraph 1</i> as the relevant article under the revised law.
T4.7	<i>Article 213, Paragraph 1</i>	Those who use a trademark identical to a registered trademark on the same goods or services ... shall be sentenced ...	... the defendant Wang, in collaboration with Chen rented a factory to produce bulk laundry detergent under the registered trademarks A and B <b>without the owner's permission</b> ...	... the defendant Wang, in collaboration with Chen rented a factory to produce bulk laundry detergent under the registered trademarks A and B <b>with the owner's permission</b> ...	The model <i>passes</i> if it predicts <i>Article 213, Paragraph 1</i> as the relevant article under the revised law.
T5.1	<i>Article 274, Paragraph 1:</i> Those who extort public or private property ... shall be sentenced ...	Those who extort public or private property <b>for the purpose of revenge</b> ... shall be sentenced ...	... the victim Li built a house, contracted to Xia, and was forced to pay the defendnat Zhang 9,000 yuan after Zhang obstructed the construction ...	... the victim Li built a house, contracted to Xia, and was forced to pay Zhang 9,000 yuan after Zhang obstructed the construction <b>for the purpose of causing chaos</b> ...	The model <i>passes</i> if it predicts <b>not</b> predict <i>Article 274, Paragraph 1</i> as the relevant article under the revised law.
T5.2	<i>Article 224, Paragraph 1:</i> Whoever, with the intent of illegal possession, defrauds the other party of property during the signing or performance of a contract under any of the following circumstances, shall be sentenced ...	Whoever, defrauds the other party of property during the signing or performance of a contract under any of the following circumstances, shall be sentenced ...	... the defendant Qi, with the intent of illegal possession, engaged in a fraudulent scheme ... through this method, he fraudulently obtained steel worth RMB 721,375.41 ...	... the defendant Qi, with the intent of revenge, engaged in a fraudulent scheme ... through this method, he fraudulently obtained steel worth RMB 721,375.41 ...	The model <i>passes</i> if it predicts <i>Article 274, Paragraph 1</i> as the relevant article under the revised law.
T6.1	<i>Article 232, Paragraph 1:</i> Whoever intentionally commits homicide shall be sentenced to death, life imprisonment, or at least 10 years' imprisonment. For relatively minor cases, the sentence shall be 3 to 10 years' imprisonment.	Whoever intentionally commits homicide shall be sentenced to death, life imprisonment, or at least 10 years' imprisonment. For relatively minor cases, the sentence shall be <b>more than 10 years' imprisonment</b> .	-	-	If, after the law revision, the model predicts at least 10 years of imprisonment for cases originally sentenced to 3–10 years, it is considered a <i>pass</i> .
T6.2	<i>Article 232, Paragraph 1</i>	Whoever intentionally commits homicide shall be sentenced to death, life imprisonment, or at least 10 years' imprisonment. For relatively minor cases, the sentence shall be <b>less than 3 years' imprisonment</b> .	-	-	If, after the law revision, the model predicts no more than 3 years of imprisonment for cases originally sentenced to 3–10 years, it is considered a <i>pass</i> .
T6.3	<i>Article 232, Paragraph 1</i>	Whoever intentionally commits homicide shall be sentenced to death, life imprisonment, or at least 10 years' imprisonment. For relatively minor cases, the sentence shall be <b>death or life imprisonment</b> .	-	-	If, after the law revision, the model predicts death or life imprisonment for cases originally sentenced to 3–10 years, it is considered a <i>pass</i> .
T6.4	<i>Article 232, Paragraph 1</i>	Whoever intentionally commits homicide shall be sentenced to <b>at least 10 years' imprisonment</b> . For relatively minor cases, the sentence shall be 3 to 10 years' imprisonment.	-	-	If, after the law revision, the model predicts more than 10 years of imprisonment for cases originally sentenced to death or life imprisonment, it is considered a <i>pass</i> .

Table 13: Detailed examples and descriptions of every revision type.