LCSL: A SYMBOLIC RULE LANGUAGE AND REASONING FRAMEWORK FOR ENHANCING LEGAL COMPLIANCE IN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Addressing the difficulties of Large Language Models (LLMs) in handling non-monotonic logic and opaque reasoning in legal compliance, we propose the LCSL-Framework, based on a Legal Compliance Symbolic Language. The framework combines Prolog-style predicate conditions with defeasible reasoning to decouple legal rules into prohibitive rules and defeaters. Its central design connects contextual condition interpretation to targeted conclusion revision: `ATTACKS` links an exception to its prohibition, and `OVERRIDES` withdraws that inference while other prohibitions remain subject to judgment. An LLM constructs reusable rule blocks and interprets a three-phase protocol for prohibition assessment, exception verification, and conflict adjudication. Experiments on Chinese Advertising Law, Copyright Law, and Hearsay tasks show the highest F_1 among the evaluated baselines, including SymbCoT. On Hearsay and Abs-Real, LCSL exceeds SePO by 5.8 and 7.6 percentage points. Abs-Real ablations reduce F_1 by 7.1 points without the rule representation and 4.2 without staged reasoning, supporting both aspects of the design. Case analysis illustrates the connection between conditions, exceptions, and judgments; OPP-115-derived classification provides auxiliary privacy evaluation.

1 INTRODUCTION

The primary challenge in automated legal compliance determination lies in resolving non-monotonic conflicts between general statutory prohibitions and exceptional exemptions (Prakken, 2013). Although Large Language Models (LLMs) exhibit strong semantic comprehension, navigating complex legal conflicts frequently triggers logical hallucinations and erratic reasoning pathways (Ji et al., 2023). Standard First-Order Logic (FOL) formulations express statutory conditions as rigorous predicates, but classical monotonic entailment cannot withdraw a derived conclusion when an exception is introduced (Sergot et al., 1986). Conversely, relying on naive prompt engineering leaves the intricate dependencies between prohibitions and exceptions implicit (Huang & Chang, 2023). In practice, this leads to three recurring failure modes in LLM-based legal reasoning: (1) *coarse condition matching*, where the model triggers a rule solely by identifying isolated keywords while ignoring nuanced circumstances; (2) *global immunity fallacy*, where verifying an exception to one prohibition mistakenly leads the model to dismiss all other independent violations; and (3) *premature termination*, where identifying a single applicable defense causes the model to halt before comprehensively examining other triggered rules.

To resolve these challenges without imposing an intractable formalization burden, we propose a neuro-symbolic reasoning framework centered on the **Legal Compliance Symbolic Language (LCSL)**. Rather than formulating a novel logical calculus, LCSL is engineered as an LLM-oriented intermediate rule representation and structured adjudication protocol grounded in established principles of defeasible logic (Nute, 2001) and formal argumentation (Dung, 1995). LCSL addresses the interface gap between natural language cases and non-monotonic compliance logic through two core designs:

- **Context-Grounded Defeasible Representation**: Rather than leaving predicate evaluation unconstrained, LCSL organizes statutory conditions into explicit constituent slots (*Subject,*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

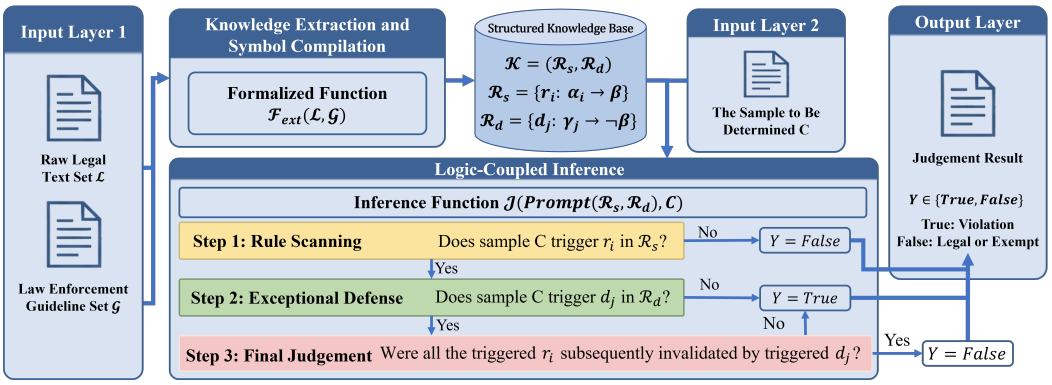

Figure 1: The compliance determination process. Exceptions are checked against their targets; a violation remains if any triggered rule is undefeated.

Action, Object, Circumstance) coupled with contextual semantic anchors, preventing over-generalized keyword matching. Crucially, exemptions are formalized as defeaters carrying explicit, directional bindings (`ATTACKS` and `OVERRIDES`) that designate exactly which prohibitive rule they can nullify (Dung, 1995).

- **Staged Adjudication Protocol**: Instead of delegating multi-hop conflict resolution to an unconstrained generation step, the framework decomposes reasoning into a three-phase protocol inspired by structured legal pleadings (Gordon, 2013): *preliminary presumption* (identifying all candidate prohibitions), *exception verification* (evaluating candidate defeaters strictly against their assigned targets), and *conflict adjudication* (aggregating undefeated prohibitions).

Compared to richer standards (e.g., LegalRuleML, RuleML) (Athan et al., 2015), LCSL focuses on contextual predicate conditions and explicit exception targets. Its design carries this structure from source extraction into reusable rule blocks and then into three separate judgment steps. This connects semantic condition interpretation to defeasible conclusion revision: an exception changes the status of its target, while an independent violation remains subject to its own rule. The resulting workflow combines guidance about how legal knowledge is represented with guidance about how it is applied.

Our main contributions are summarized as follows:

- Predicate–Defeasible Integration: We introduce a code-like legal rule representation that connects first-order-style conditions with targeted defeaters. Subject, action, object, and circumstance descriptions guide condition interpretation; explicit `ATTACKS` links identify which prohibition an accepted exception can override. The representation separates whether a rule matches from whether its conclusion survives.
- Dual-Stage Enhancement Architecture: We design a two-stage framework that segregates rule compilation from logical judgment. The same condition descriptions and exception targets pass from JSON extraction to reusable LCSL rules and a three-phase judgment protocol, connecting source organization, case matching, and conflict adjudication.
- Empirical Support for Representation and Protocol: Experiments across five primary datasets compare LCSL with prompting baselines, SePO, CONTRAPROMPT, and SymbCoT. On Abs-Real, removing initialization, the rule representation, or staged reasoning reduces F_1 by 12.3, 7.1, or 4.2 points, respectively. These ablations and case analysis examine the components supporting the integrated design.

2 Related Work

Existing research on legal compliance determination spans three main paradigms: FOL, DDL and Neuro-Symbolic Deduction.

FOL provides explicit predicate-based descriptions of legal conditions, while defeasible logics model the withdrawal of conclusions under exceptions. Recent work studies LLM-based formalization into Defeasible Deontic Logic (DDL) (Horner et al., 2025) and combines LLMs with SMT solvers for legal analysis (Hsia et al., 2025). To combine neural flexibility with logical structure, SymbCoT (Xu et al., 2024) uses symbolic translation, planning, reasoning, and verification within an LLM-based workflow. Logic-LM (Pan et al., 2023) and VERUS-LM (Callewaert et al., 2025) instead integrate LLMs with external symbolic reasoning systems.

Building on these approaches, LCSL focuses on two design requirements: (1) *Condition Interpretation*: present legal predicates together with the contextual descriptions needed for case matching; (2) *Targeted Exception Handling*: connect each defeater to the prohibition it can block and carry this relation through final judgment.

LCSL builds on established defeasible mechanisms and specializes how they are presented to and applied by an LLM. Its contextual predicate descriptions, named prohibition–exception links, and undefeated-rule criterion form a common structure for rule construction and judgment. SymbCoT provides a general symbolic reasoning workflow; LCSL organizes that reasoning around the legal roles of prohibitions and targeted exceptions. The experiments compare these workflows under shared source information and a common backbone.

3 Problem Formulation and Methodology

3.1 Problem Formalization and Preliminaries

Problem Formulation. Given raw legal texts $\mathcal{L}$, supplementary guidelines $\mathcal{G}$, and a target case description C, the goal of legal compliance determination is to derive a binary judgment $Y \in \{\text{True}, \text{False}\}$. Here, $Y = \text{True}$ indicates a violation of the specified task rules, whereas $Y = \text{False}$ indicates that no such violation remains, including when a statutory exemption applies. For the Hearsay and privacy classification tasks, positive labels instead indicate membership in the specified category (Section 4.1.1).

Legal Knowledge Formalization. We model legal knowledge as a non-monotonic structured base $\mathcal{K} = (\mathcal{R}_s, \mathcal{R}_d)$, comprising: (1) *Prohibitive Rule Set* $\mathcal{R}_s = \{r_i : \alpha_i \Rightarrow \beta_i\}$, where satisfying condition α_i supports a provisional non-compliance conclusion β_i; (2) *Defeater Set* $\mathcal{R}_d = \{d_j : (\gamma_j, T_j)\}$, where $T_j \subseteq \mathcal{R}_s$ identifies the rules whose inferences can be blocked by exemption condition γ_j. The label `STRICT_RULE`, also used in the figures and logs, denotes the first rule type in LCSL; its logical force is defeasible. A defeater blocks its targeted inference rather than deriving a general permission. In Figure 1, the arrow to $\neg\beta$ abbreviates this withdrawal, and "legal or exempt" refers to the selected task rules.

Task Decomposition. The reasoning pipeline is formalized as a two-stage mapping: 1. *Knowledge Compilation*: An extraction function $\mathcal{F}_{ext}$ converts unstructured sources into symbolic logic:

$$\mathcal{F}_{ext}(\mathcal{L}, \mathcal{G}) \Rightarrow \mathcal{K} = (\mathcal{R}_s, \mathcal{R}_d)$$

2. *Logic-Coupled Inference*: An inference function $\mathcal{I}$ evaluates case C against formal knowledge $\mathcal{K}$ via non-monotonic deduction:

$$Y = \mathcal{I}(Prompt(\mathcal{R}_s, \mathcal{R}_d), C)$$

The intended criterion is $Y = \text{True}$ iff case C triggers a prohibition in $\mathcal{R}_s$ that is not defeated by an applicable exception targeting that rule. The complete workflow is illustrated in Figure 1.

3.2 Legal Compliance Symbolic Language (LCSL)

To bridge non-monotonic legal logic with LLMs' code-comprehension strengths, we introduce LCSL, synthesizing Prolog-style predicate conditions with defeasible conflict resolution.

Lexicon and Grammar. The lexicon of LCSL consists of: (1) *Modal Directives*: Define attributes and logical force, including declarators (`[STRICT_RULE]`, `[DEFEATER]`), relational operators

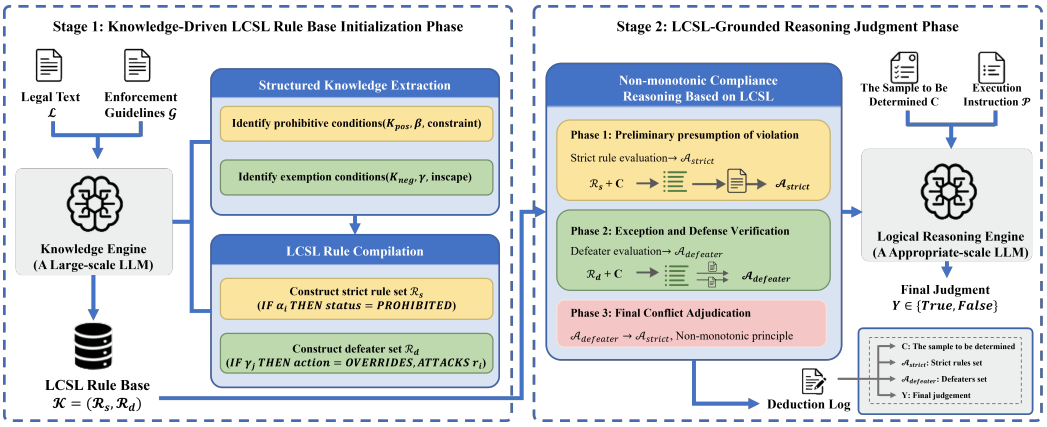

Figure 2: Overview of the LCSL-enhanced legal compliance reasoning framework. The knowledge and reasoning engines are LLM-based; the three phases specify the requested judgment procedure.

(`ATTACKS`, `OVERRIDES`), and decision states (`PROHIBITED`, `PERMITTED`); (2) *Logical Operators*: Standard predicate logic operators (`AND`, `OR`, `NOT`, `ALL`, `EXISTS`, =, !=, >, <); (3) *Semantic Predicates*: Formulated as `Predicate(arg, ...)`, using LLM semantic capability to assess contextual matching (e.g., `Match_Concept(Text, Concept)`) over unstructured facts.

LCSL uses modular rule templates to organize non-monotonic reasoning via deferred conflict resolution:

Prohibitive Rule: [STRICT_RULE_ID]
IF ⟨Condition⟩ THEN status = PROHIBITED
Defeater: [DEFEATER_ID]
TARGET: ATTACKS[Target_ID]
IF ⟨Condition⟩ THEN action = OVERRIDES

Predicate conditions can organize the subject, action, object, and circumstance of a legal provision. A `Semantic_Anchor` describes the relevant concept in context; a defeater's `Boundary_Condition` describes its applicability. Thus, a product-quality assertion and an expression of intent can be distinguished even when they share an absolute adjective. The target relation then identifies which provisional conclusion an accepted exception can withdraw. This separation is central to the design: a prohibition's condition may remain satisfied even after its inference is defeated. Predicate matching therefore supplies the preliminary inference, while the linked defeater governs its revision. The three-phase protocol asks the model to assess these two decisions separately before aggregating the remaining violations.

3.3 LCSL-Grounded Reasoning Architecture

Our architecture decouples legal reasoning into two independent stages: a knowledge-driven rule initialization phase and a symbolic-grounded reasoning judgment phase.

Knowledge-Driven LCSL Rule Base Initialization. As detailed in Algorithm 1, an LLM acts as a logic compiler to parse provisions $\mathcal{L}$ and guidelines $\mathcal{G}$. It extracts prohibitive elements (mapped to $\mathcal{R}_s$) and exemption defenses (mapped to $\mathcal{R}_d$), generating a JSON Intermediate Representation (IR) that is compiled into Prolog-style LCSL rules with explicit `ATTACKS` declarations.

LCSL-Grounded Reasoning Judgment. Algorithm 2 (Appendix A) specifies the three-phase protocol supplied to the judgment LLM. In the implementation, software builds a prompt from $\mathcal{K}$, the case, and demonstrations, calls the model, and parses its predicted label and phase descriptions. The loops and `LLMs_Evaluate` in the pseudocode describe the requested reasoning within that response. Thus, rule matching and conflict adjudication are model-interpreted steps; prompt construction and response parsing are program operations.

Algorithm 1 Legal Knowledge Compilation $\mathcal{F}_{\text{ext}}$

Input: Legal texts $\mathcal{L}$, Enforcement guidelines $\mathcal{G}$
Output: LCSL Rule Base $\mathcal{K} = (\mathcal{R}_s, \mathcal{R}_d)$

1: Extract$(\mathcal{L}, \mathcal{G}) \xrightarrow{\text{LLM_parse}}$ Structured_JSON
2: Identify prohibitive requisites: keywords K_{pos}, predicate β, contextual constraints.
3: Identify exemptions: defense logic γ, clues K_{neg}, constituent elements.
4: Compile Structured_JSON $\xrightarrow{\text{LLM_parse}}$ LCSL_Rules
5: Construct $\mathcal{R}_s = \{r_i \mid \text{IF } \alpha_i \text{ THEN status = PROHIBITED}\}$.
6: Construct $\mathcal{R}_d = \{d_j \mid \text{TARGET: ATTACKS } r_i, \text{ IF } \gamma_j \text{ THEN OVERRIDES}\}$.
7: **return** $\mathcal{K} = (\mathcal{R}_s, \mathcal{R}_d)$

Table 1: Statistics and key characteristics of the evaluation datasets. Size is shown as positive/negative counts. OPP totals are summed task instances.

Dataset	Pos./Neg.	Task structure	Domain and source
Abs-Syn	85/320	Non-monotonic	Chinese advertising; synthetic
Abs-Real	126/129	Non-monotonic	Chinese advertising; collected
Med-Real	192/100	Prohibitive	Chinese advertising; collected
Copyright	251/111	Non-monotonic	Chinese copyright; judicial and synthetic
Hearsay	41/53	Definition	U.S. evidence; LegalBench
OPP-115	4,629/4,629	Classification	Privacy; nine LegalBench tasks

During judgment, the compiled rule base $\mathcal{K}$ is injected into the prompt space as explicit constraints. Algorithm 2 simulates DDL conflict adjudication via three steps: (1) *Presumption*: Evaluate $\mathcal{R}_s$ to identify suspected violations, establishing $\mathcal{A}_{\text{strict}}$; (2) *Defense Verification*: For each triggered rule in $\mathcal{A}_{\text{strict}}$, evaluate corresponding defeaters in $\mathcal{R}_d$ to populate $\mathcal{A}_{\text{defeater}}$; (3) *Conflict Adjudication*: $Y \rightarrow$ True iff there exists at least one violation in $\mathcal{A}_{\text{strict}}$ that is not overridden by an applicable defeater targeting it; otherwise $Y \rightarrow$ False. For fixed applicability decisions a_i for prohibitions and b_{ji} for defeaters applied to their targets, this criterion is

$$Y^\star = \bigvee_i \left(a_i \wedge \neg \bigvee_{j : r_i \in T_j} b_{ji} \right). \tag{1}$$

An empty disjunction is false. If two prohibitions apply and only one is defeated, the case remains positive. $Y^\star$ specifies the protocol's intended result, against which the model's reported judgment can be inspected.

4 Experiments and Empirical Analysis

4.1 Experimental Setup

4.1.1 Dataset Construction and Statistics

To evaluate the effectiveness and generalizability of LCSL, we construct and benchmark across five primary datasets covering both monotonic and non-monotonic structures, with auxiliary OPP-115-derived tasks, as summarized in Table 1:

- Absolute and Exaggerated Language (Abs-Law): Comprises synthetic cases alongside real-world noisy data extracted from online monitoring systems, focusing on non-compliance determinations and statutory exemption scenarios.
- Medical & Copyright (Med & Copyright): Med emphasizes pure prohibitive logic, whereas Copyright encompasses complex infringement determinations alongside fair-use exemptions.

- Hearsay (Guha et al., 2023) & OPP-115 (Wilson et al., 2016): Incorporates publicly available legal benchmark datasets. Hearsay tests whether evidence satisfies the hearsay definition and explicitly excludes exceptions (LegalBench, 2023a). The nine OPP-115-derived LegalBench tasks classify privacy practices described in policy clauses (LegalBench, 2023b).

The dataset construction integrates human annotation with LLM-driven synthetic augmentation. For Abs-Syn and the synthetic portion of Copyright, Qwen3-32B receives several real examples and adds the lawful or unlawful cues required by the target scenario. The resulting text is then submitted to Qwen3-32B again for polishing into a realistic case. Labels are assigned through manual review informed by domain-expert opinions.

4.1.2 Comparison of Benchmarks and Evaluation Indicators

We compare the LCSL reasoning framework against three categories of baselines:

- **Vanilla Prompting Baselines**: Zero-shot, Few-shot, and standard Chain-of-Thought (CoT) prompting.
- **Advanced Prompting Methods**: On Abs-Real and Hearsay, we re-implement SePO (Tao et al., 2026) and CONTRAPROMPT (CP) (Rishav et al., 2026) as competitive baselines.
- **Symbolic Reasoning with LLMs**: SymbCoT (Xu et al., 2024) is evaluated on Abs-Real, Med-Real, and Hearsay using the same base model and source information.

OPP-115 results and historical task-specific results are reported separately in Appendix B.

All LLM-based baselines and LCSL in the main comparison use Qwen3-32B (Yang et al., 2025) as the unified base model. Within each dataset, all methods receive the same legal texts, enforcement guidelines, and case information. All methods except zero-shot receive the same demonstration examples; zero-shot omits examples. Generation temperature is fixed to $T = 0$, and results are reported from a single evaluation pass. All experiments are conducted on a server equipped with dual Intel Xeon Gold 6138 CPUs (80 vCPUs) and four NVIDIA L40 GPUs. Performance is measured using standard classification metrics: Accuracy (A), Precision (P), Recall (R), and F_1 Score (F).

4.2 Results and Analysis

As shown in Table 2, LCSL achieves the highest F_1 scores among the evaluated methods across all five primary datasets, reaching 90.0% and 91.1% on Abs-Real and Hearsay. Unlike prompt-optimization baselines (e.g., SePO, CONTRAPROMPT), LCSL organizes case judgment around explicit rule conditions and exception targets. Several baselines exhibit a precision–recall imbalance. For instance, Zero-shot prompting yields high recall (94.1%) but low precision (58.4%) on Abs-Syn. By contrast, LCSL raises precision to 81.3% while retaining 91.8% recall, consistent with a more selective treatment of apparent prohibitions.

LCSL also performs well across the evaluated domains. It achieves $F_1 = 97.4\%$ on Med-Real and exceeds SePO by 5.8 percentage points on Hearsay. On Abs-Real, which includes OCR errors and colloquialisms, LCSL maintains $F_1 = 90.0\%$, 7.6 points above SePO. On Copyright, F_1 increases from CoT's 95.1% to 96.1%. These results support the utility of combining LLM-based semantic interpretation with an explicit rule representation in the evaluated settings.

Comparison with SymbCoT. Under the shared-input Qwen3-32B setting, LCSL exceeds SymbCoT in all four reported metrics on the three overlapping datasets. Its F_1 is 90.0 versus 50.0 on Abs-Real, 97.4 versus 75.8 on Med-Real, and 91.1 versus 57.1 on Hearsay: differences of 40.0, 21.6, and 34.0 percentage points. SymbCoT organizes reasoning through a general symbolic workflow; LCSL explicitly separates prohibitive conditions from linked exceptions and carries these links into final judgment. The comparison evaluates the task-specific representation and protocol as a package under the recorded settings. The Med-Real and Hearsay results additionally show utility in prohibition and definition application. Together with the component removals below, these results connect the empirical evaluation to both the explicit representation and the prescribed reasoning order.

Table 2: Classification results (percent) using Qwen3-32B. Bold indicates the highest value within each dataset, including ties.

Datasets	Method	A(%)	P(%)	R(%)	F_1(%)
Abs-Real	Zero-shot	71.8	73.3	67.5	70.2
	Few-shot	71.0	87.1	48.4	62.2
	CoT	77.3	84.7	65.9	74.1
	SePO	82.4	81.4	83.3	82.4
	CP	65.1	60.5	84.9	70.6
	SymbCoT	58.4	61.6	42.1	50.0
	LCSL	**89.8**	**87.3**	**92.9**	**90.0**
Hearsay	CoT	78.7	72.3	82.9	77.3
	SePO	88.3	94.1	78.0	85.3
	CP	74.5	74.3	63.4	68.4
	SymbCoT	61.7	55.8	58.5	57.1
	LCSL	**92.6**	**94.7**	**87.8**	**91.1**
Abs-Syn	Zero-shot	84.7	58.4	**94.1**	72.1
	Few-shot	84.2	57.8	91.8	70.9
	CoT	79.3	50.4	82.4	62.5
	LCSL	**93.8**	**81.3**	91.8	**86.2**
Med-Real	Zero-shot	83.9	82.2	96.4	88.7
	Few-shot	90.1	89.8	95.8	92.7
	CoT	80.8	78.1	**98.4**	87.1
	SymbCoT	68.5	76.6	75.0	75.8
	LCSL	**96.6**	**97.9**	96.9	**97.4**
Copyright	Zero-shot	93.4	99.1	91.2	95.0
	Few-shot	90.1	99.5	86.1	92.3
	CoT	93.4	97.5	**92.8**	95.1
	LCSL	**94.8**	**99.6**	**92.8**	**96.1**

Table 3: Ablation study on the key components of LCSL on Abs-Real (percent).

Variant	A (%)	P (%)	R (%)	F_1 (%)
w/o Rule Initialization	78.8	81.0	74.6	77.7
w/o LCSL Rule Base	82.7	81.1	84.9	82.9
w/o Chain-of-Thought	85.9	85.2	86.5	85.8
Full LCSL	89.8	87.3	92.9	90.0

4.3 Ablation Study

Table 3 summarizes the ablation results on Abs-Real, supporting the contribution of LCSL's core components. Removing the rule initialization phase causes the largest degradation: F_1 drops from 90.0% to 77.7%, alongside a recall decrease of 18.3 percentage points. This supports organizing provisions and extracting non-compliance elements before judgment. Replacing the symbolic representation with unconstrained natural language reduces F_1 by 7.1 points, supporting the code-like rule representation. Finally, omitting the step-by-step Chain-of-Thought (CoT) constraint drops F_1 to 85.8%, indicating the utility of guiding the transition from rule scanning to conflict adjudication. The representation and protocol removals test complementary parts of LCSL: the former changes how legal knowledge is expressed, while the latter changes the requested order of applying it.

4.4 Evaluation Across Model Backbones

To assess performance across model backbones, we evaluate LCSL with three base LLMs on Abs-Real (Table 4). LCSL achieves F_1 scores of 87.6%–90.0% across Qwen3-32B, Qwen3.5-397B-

Table 4: Performance across underlying LLM backbones on Abs-Real (percent).

Model Backbone	A (%)	P (%)	R (%)	F_1 (%)
Qwen3-32B	89.8	87.3	92.9	90.0
Qwen3.5-397B-A17B	87.8	88.0	87.3	87.6
DeepSeek-v4	88.6	87.0	90.5	88.7

Case 1: Correctly Predicted Sample (Successful Exception Override)


```
{
 "original_text": "宝子们！最近打算入手粉底液的干皮姐妹们一定要看过来呀~我们一直致力于精心打造顶级品质的粉底液，现在来我们的直播间，还能享受拍 1 发 4 的超值福利哦！#粉底液 #底妆 #种草",
 "Phase 1": "触发STRICT_RULE_003（使用'顶级'属于行政级别/质量等级词汇），触发关键词：顶级",
 "Phase 2": "DEFEATER_001成立（检测到'致力于'作为意愿性动词修饰'顶级品质'，属于未来目标表达），废止STRICT_RULE_003",
 "Phase 3": "所有触发规则均被豁免",
 "illegal_judgement": false
}
```


Case 2: Misclassified Sample (Failure Analysis Case)


```
{
 "original_text": "大幅度降温啦，这种穿在里面超实用的儿童打底衫可不能少！它的面料柔软又舒适，小中领的设计保暖性超棒。我们要做最好的儿童打底衫，孩子们都很爱穿，妈妈们赶快安排起来吧！#儿童打底衫 #保暖又好看 #母婴好物 #高品质童装",
 "Phase 1": "触发STRICT_RULE_001（最字类绝对化用语禁止规则），关键词为'最好的'。未触发其他严格规则",
 "Phase 2": "未发现可废止STRICT_RULE_001的例外规则。DEFEATER_001未成立（'我们要做最好的'中'最好'仍直接修饰商品质量属性），DEFEATER_002/003/004均不适用",
 "Phase 3": "STRICT_RULE_001未被废止，维持原判",
 "illegal_judgement": true
}
```


Figure 3: A correctly predicted example and a false positive. The original Chinese logs show the model's three phases; the decisive expressions are explained in Section 4.5.

A17B, and DeepSeek-v4. The framework therefore retains useful performance across these three settings on the same dataset.

4.5 Case Analysis

Figure 3 presents two advertising examples and their original reasoning logs. Both have negative reference labels in the Abs-Syn source collection. They illustrate a successful exception override and a false positive.

Case 1: successful exception override. The first expression translates as "we have always been committed to carefully creating top-quality foundation." Phase 1 identifies "top-quality" under `STRICT_RULE_003`. Phase 2 accepts `DEFEATER_001`, treating "committed to" as an aspirational qualification, and Phase 3 withdraws the preliminary violation. The negative prediction matches the reference label. The log connects a particular trigger, the contextual interpretation of an exception, and the resulting withdrawal.

Case 2: rejected exception and false positive. The second expression includes "we want to make the best children's base-layer shirts." The model triggers `STRICT_RULE_001`, but rejects the aspirational exception because "best" directly modifies product quality. With no accepted defeater, it

predicts a violation, contrary to the negative reference label. The reported aggregation follows its stated applicability decisions, localizing the disagreement to exception interpretation. Together, the two cases show why evaluating semantic conditions and associating exceptions with their targets are distinct steps: similar goal-oriented language can receive different applicability judgments even under the same protocol.

5 CONCLUSION

This paper presents LCSL, a neuro-symbolic framework addressing LLMs' limitations in resolving non-monotonic conflicts during legal compliance determination. By unifying Prolog-style predicate conditions with defeasible conflict resolution, LCSL decouples general principles from exceptional exemptions, representing the defeasible nature of legal reasoning through explicit `ATTACKS` and `OVERRIDES` instructions. Evaluations across five primary datasets—spanning Chinese advertising and copyright law and U.S. evidence classification—show that LCSL achieves higher F_1 than the evaluated prompting and symbolic-prompting baselines. OPP-115-derived tasks provide an auxiliary privacy-policy classification evaluation. The explicit rule representation and staged judgment protocol offer an inspectable workflow for legal technology.

Despite its efficacy, LCSL has operational boundaries: (1) rule base initialization still incurs engineering overhead, calling for fully automated statute-to-symbolic compilation; (2) legal grey areas requiring judicial discretion demand deeper semantic modeling; and (3) LLM stochasticity induces minor execution fluctuations. Future work will integrate fuzzy logic to model non-binary legal reasoning and explore dynamic self-evolving rule bases to adapt to evolving statutory landscapes.

AI USE STATEMENT

In this work, we used generative AI tools for synthetic dataset generation. We have not used generative AI tools for research ideation and execution, drafting specific sections of the paper, or proving mathematical derivations, and other tasks with required disclosure are not applicable to this work. Additionally, we used generative AI tools for assisting with literature search and improving the academic tone and phrasing of the writing. We have reviewed all AI-assisted work. Specifically, all AI-polished text was thoroughly proofread and verified by the authors; all references retrieved using AI were manually cross-checked to ensure they genuinely exist and are accurately cited; and the detailed methodology for dataset synthesis is explicitly documented in the main text. We take full responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

In this work, all datasets used and constructed are derived exclusively from publicly available sources and established open-source benchmarks. The data collection and processing procedures strictly comply with the respective data licenses, platform terms of service, and applicable legal regulations. We have ensured that the dataset does not contain any personally identifiable information (PII) or sensitive private data, thereby preserving privacy and security. Furthermore, this study does not involve human participants, and no Institutional Review Board (IRB) approval was required. We foresee no direct negative societal impacts, discrimination, or legal concerns arising from this work.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our results, the complete source code, including implementation details, experimental configurations, and dependencies required to replicate the findings, has been made available in the supplementary material. Detailed instructions for running the code and reproducing the main experimental outcomes are provided in the accompanying documentation.

REFERENCES

Tara Athan, Guido Governatori, Monica Palmirani, Adrian Paschke, and Adam Wyner. Legalruleml: Design principles and foundations. In *Reasoning Web International Summer School*, pp. 151–188. Springer, 2015.

Benjamin Callewaert, Simon Vandevelde, and Joost Vennekens. Verus-lm: a versatile framework for combining llms with symbolic reasoning. *arXiv preprint arXiv:2501.14540*, 2025.

Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, 1995.

Thomas F Gordon. *The pleadings game: An artificial intelligence model of procedural justice*. Springer Science & Business Media, 2013.

Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, et al. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in neural information processing systems*, 36:44123–44279, 2023.

Elias Horner, Cristinel Mateis, Guido Governatori, and Agata Ciabattoni. Toward robust legal text formalization into defeasible deontic logic using llms. *arXiv preprint arXiv:2506.08899*, 2025.

Yung-Shen Hsia, Fang Yu, and Jie-Hong Roland Jiang. Neuro-symbolic compliance: Integrating llms and smt solvers for automated financial legal analysis. In *2025 2nd IEEE/ACM International Conference on AI-powered Software (AIware)*, pp. 01–10. IEEE, 2025.

Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. In *Findings of the association for computational linguistics: ACL 2023*, pp. 1049–1065, 2023.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023.

LegalBench. Hearsay task description, 2023a. URL `https://hazyresearch.stanford.edu/legalbench/tasks/hearsay.html`. Accessed September 25, 2026.

LegalBench. OPP-115 data security task description, 2023b. URL `https://hazyresearch.stanford.edu/legalbench/tasks/opp115_data_security.html`. Accessed September 25, 2026.

Donald Nute. Defeasible logic. In *International Conference on Applications of Prolog*, pp. 151–169. Springer, 2001.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. *arXiv preprint arXiv:2305.12295*, 2023. URL `https://arxiv.org/abs/2305.12295`.

Henry Prakken. *Logical tools for modelling legal argument: a study of defeasible reasoning in law*, volume 32. Springer Science & Business Media, 2013.

Rishav Rishav, Pushpak Pujari, and Pushpendre Rastogi. Contraprompt: Contrastive prompt optimization via dyadic reasoning trace analysis. *arXiv preprint arXiv:2604.17937*, 2026.

Marek J. Sergot, Fariba Sadri, Robert A. Kowalski, Frank Kriwaczek, Peter Hammond, and H Terese Cory. The british nationality act as a logic program. *Communications of the ACM*, 29(5):370–386, 1986.

Wangcheng Tao, Han Wu, and Weng-Fai Wong. Sepo: Self-evolving prompt agent for system prompt optimization. *arXiv preprint arXiv:2606.04465*, 2026.

Shomir Wilson, Florian Schaub, Aswarth Abhilash Dara, Frederick Liu, Sushain Cherivirala, Pedro Giovanni Leon, Mads Schaarup Andersen, Sebastian Zimmeck, Kanthashree Mysore Sathyendra, N Cameron Russell, et al. The creation and analysis of a website privacy policy corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1330–1340, 2016.

Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. Faithful logical reasoning via symbolic chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13326–13365, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

A Detailed Judgment Protocol

Algorithm 2 gives the three-phase protocol introduced in Section 3. It records prohibition triggers, checks linked defeaters, and retains a positive judgment when any triggered prohibition remains undefeated. The pseudocode specifies the procedure requested of the judgment LLM; the model/controller boundary is described in Section 3.

Algorithm 2 LCSL-grounded three-phase judgment protocol

Input: Rule base $\mathcal{K}$, target case $\mathcal{C}$, instruction prompt P
Output: Judgment $Y \in \{\text{True}, \text{False}\}$, deduction log $\mathcal{D}$

// Phase 1: Preliminary Violation Presumption
$A_{\text{strict}} \leftarrow \emptyset$
for each rule $r_i : \alpha_i \Rightarrow$ PROHIBITED **in** $\mathcal{R}_s$ **do**
  if LLMs_Evaluate$(\mathcal{C}, \text{MATCH } \alpha_i)$ **is** True **then**
    $A_{\text{strict}} \leftarrow A_{\text{strict}} \cup \{r_i\}$
  end if
end for
if $A_{\text{strict}} = \emptyset$ **then**
  return $Y = \text{False}, \mathcal{D} = \{P_1 : \text{“No rules triggered”}\}$
end if
// Phase 2: Exemption Defense Verification
$A_{\text{defeater}} \leftarrow \emptyset$
for each $r_i \in A_{\text{strict}}$ **do**
  Find defeaters $D_i = \{d_{ij} \in \mathcal{R}_d \mid d_{ij} \text{ ATTACKS } r_i\}$
  for each $d_{ij} \in D_i$ **do**
    if LLMs_Evaluate$(\mathcal{C} \cup \text{Context}, \text{MATCH } \gamma_{ij})$ **is** True **then**
      $A_{\text{defeater}} \leftarrow A_{\text{defeater}} \cup \{d_{ij}\}$
    end if
  end for
end for
// Phase 3: Conflict Adjudication
$Y \leftarrow \text{False}$
for each $r_i \in A_{\text{strict}}$ **do**
  if $\nexists d_{ij} \in A_{\text{defeater}}$ **such that** d_{ij} ATTACKS r_i **then**
    $Y \leftarrow \text{True}$
    break
  end if
end for
return $Y, \mathcal{D} = \text{Format_Output}(\mathcal{C}, A_{\text{strict}}, A_{\text{defeater}}, Y)$

B Auxiliary Privacy-Policy Classification

The nine OPP-115-derived LegalBench tasks ask whether a privacy-policy clause describes a specified data practice. Table 5 reports LCSL results and per-category supports. The 9,258 instances are summed across balanced tasks, rather than a count of distinct policies. The F_1 scores range from 73.8% for Data Retention to 98.2% for Do Not Track.

Table 5: LCSL results on the nine balanced OPP-115-derived tasks (percent), with per-category supports.

Task	Pos.	Neg.	P (%)	R (%)	F_1 (%)
International/specific audiences	347	347	98.7	86.7	92.3
User choice/control	627	627	87.4	73.0	79.6
Access, edit, deletion	226	226	89.4	85.4	87.3
Do Not Track	28	28	100.0	96.4	98.2
First-party collection/use	1,517	1,517	76.7	81.9	79.2
Policy change	181	181	100.0	87.3	93.2
Data retention	152	152	93.0	61.2	73.8
Data security	369	369	79.5	90.2	84.5
Third-party sharing/collection	1,182	1,182	83.0	85.9	84.4

Table 6 provides historical task-specific scores attributed to Wilson et al. (2016), separately from the LCSL evaluation. The historical tasks concern the original corpus and its annotation setup; LCSL uses the binary LegalBench derivatives described above.

Table 6: Historical task-specific results attributed to Wilson et al. (2016), reported under the original study setup (percent).

Historical task category	P (%)	R (%)	F_1 (%)
International/specific audiences	70.0	70.0	70.0
User choice/control	65.0	58.0	61.0
Access, edit, deletion	67.0	56.0	61.0
Do Not Track	100.0	100.0	100.0
First-party collection/use	76.0	73.0	75.0
Policy change	66.0	88.0	75.0
Data retention	10.0	35.0	16.0
Data security	66.0	67.0	67.0
Third-party sharing/collection	67.0	73.0	70.0

C Input and Output Structure

Rule construction. The input consists of legal provisions and supplementary enforcement guidelines. The extraction step requests prohibitive keywords, predicates, and contextual constraints, together with exemption conditions, clue words, and constituent elements. These are rendered as LCSL rule blocks with identifiers, condition descriptions, and explicit `ATTACKS` targets.

Case judgment. The judgment prompt contains the rule base, case description, demonstration examples, and instructions for preliminary presumption, exception verification, and conflict adjudication. The requested output records the case, descriptions of the three phases, and a Boolean `illegal_judgement`. The software parses these fields for evaluation. Figure 3 reproduces two such outputs.

Shared source information. All main-table methods receive the same underlying legal texts, guidelines, and test cases. Demonstration examples are shared among all methods except zero-shot. Method-specific representations and processing follow each method's workflow. The separate backbone study changes the judgment model while assessing LCSL on Abs-Real.