

When Condition Trees Help LLMs Apply Legal Rules: Three-Model Evidence from LegalBench

Anonymous Authors¹

Abstract

Large language models can recall legal rules but consistently misapply them on multi-condition fact patterns. We characterize this failure as condition-evaluation shortcutting: the model produces a conclusion before all necessary conditions are independently verified. We propose **Structured Rule Application (SRA)**, which encodes each doctrine as a condition tree of binary checks with a hard-coded boolean aggregator, evaluates each condition in an isolated LLM call, and combines them deterministically. We evaluate SRA against five baselines (zero-shot, few-shot, chain-of-thought, self-consistency, dynamic-extraction SRA, and FOL grounding) on 14 LegalBench rule-application tasks (n=2,602) across three models (GPT-4o, GPT-4o-mini, GPT-5.5). SRA improves overall accuracy by **+11.0 pp on GPT-4o** (87.3% vs. 76.4%) and **+22.2 pp on GPT-5.5** (88.6% vs. 66.4%), but yields no net gain on GPT-4o-mini (−0.5 pp). The cross-model pattern is not monotonic in model size: GPT-5.5’s baseline is worse than GPT-4o’s on diversity-jurisdiction tasks (66% vs. 76%), yet SRA closes the gap to $\geq 91\%$ on every variant. A controlled ablation against runtime decomposition shows that expert-encoded condition trees matter on tasks requiring precise categorical distinctions, while runtime decomposition is sufficient for clean binary rules. First order logic (FOL) grounding underperforms natural-language condition evaluation by 9–34 pp on StAtutory Reasoning Assessment (SARA), because qualitative statutory conditions resist typed predicate encoding. We synthesize these findings into three exclusion criteria that identify when SRA should not be adopted, covering every observed regression in our data.

¹Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

1. Introduction

A practitioner asked whether evidence is admissible under FRE 801 does not pattern-match. They check three conditions in sequence—is it a statement, is it out-of-court, is it offered for truth—and aggregate by conjunction. The aggregation is part of the doctrine, not part of the reasoning. Each condition is logically independent and each must be checked.

LLMs, by contrast, treat rule application as generation. Given a fact pattern and a question, they produce an answer in a single forward pass, optionally scaffolded by chain-of-thought (Wei et al., 2022). On LegalBench (Guha et al., 2023), this is sufficient for rule recall but degrades sharply on rule application: GPT-4o, with few-shot prompting, scores only 56.4% on hearsay and 71–85% across the six diversity-jurisdiction variants, well below what a first-year law student would produce.

We attribute this gap to a task-framing issue. The LLM is asked to perform two operations simultaneously: identify which facts bear on which conditions, and combine condition outcomes into a final answer. The combination step is purely logical and is fully specified by the doctrine. Asking the model to perform it is unnecessary; it is also where most errors occur, because conditions that share surface features with the facts dominate the model’s attention and the rule’s required conjunction collapses into a single “looks like hearsay” judgment.

Structured Rule Application (SRA) eliminates the collapse. Each legal doctrine is encoded once as a condition tree: an ordered list of binary conditions, optional dependency edges, and a boolean aggregation function (Section 3). At inference time, the LLM is queried independently per condition; the answer is the value of the aggregator, not a generated token. The LLM is restricted to what it does well (extracting and evaluating facts against a single, scoped question), and the logical structure of the rule is enforced by the harness.

This decomposition is conceptually adjacent to prior work on prompt scaffolds (least-to-most (Zhou et al., 2023), Tree-of-Thoughts (Yao et al., 2023), Chain-of-Logic (Servantez et al., 2024), syllogism prompting (Blair-Stanek et al.,

2023)) and to symbolic-solver pipelines (PAL (Gao et al., 2023), LOGIC-LM (Pan et al., 2023)). It differs in two ways: the conditions are pre-encoded from doctrine rather than runtime-extracted, and the aggregator is hard-coded rather than re-generated by the model. Our ablation shows that both choices matter, but their importance is task-specific.

The contribution of this paper is empirical and analytical: when does SRA work, and why. We evaluate SRA against five baselines on 14 LegalBench tasks (2,602 examples) and across three OpenAI models with markedly different capabilities. The results show that SRA’s gain is not a constant “boost” but follows a predictable pattern that depends jointly on the structure of the rule and the model’s baseline ability to internalize it. We surface three findings that we believe matter for the broader Rules-as-Code program (Witt et al., 2024; Mérigoux et al., 2021):

1. **SRA gain tracks baseline rule-internalization failure.** On tasks where GPT-5.5’s baseline is below 50%, SRA gains exceed +40 pp; on tasks where the baseline already exceeds 90%, SRA either matches or regresses.
2. **Model capability is required but not in the obvious direction.** GPT-4o-mini, the weakest model, derives negative net value from pre-encoded SRA but positive value from runtime-extracted decomposition (+4.8 pp). Strong instruction-following is needed for SRA; expert encoding piles structure onto a model that cannot exploit it.
3. **Natural language beats formal predicates for statutory entailment.** A LOGIC-LM-style FOL grounding pipeline underperforms SRA on SARA by 9 pp (GPT-4o), 20 pp (GPT-4o-mini after sample weighting), and 34 pp (GPT-5.5). Statutory conditions like “engaged in a trade or business” resist typed predicates; treating them as binary NL questions preserves the model’s strength.

We synthesize these into three exclusion criteria (Section 6.1) that identify when SRA should not be applied. All code, rule trees, and per-task outputs are released openly.

2. Diagnosis: Three Failure Modes

From qualitative inspection of GPT-4o baseline errors (478 total incorrect predictions across 14 tasks), we identify three recurring failure modes.

Condition shortcutting. Models produce a plausible conclusion without independently verifying each required condition. On hearsay, GPT-4o answers “Yes” whenever it sees the lexical pattern “I heard X say,” regardless of whether the statement was made in court. The middle condition (out-of-court) is dropped from the chain.

Fact: At the suppression hearing, Officer Diaz

```
testified: "I heard the defendant say
he was driving 80 mph."
Gold: No (statement made during this proceeding,
so not out-of-court)
GPT-4o (baseline): "Yes -- statement reports
what defendant said, offered for truth."
SRA c2 (out-of-court): False -> answer No.
```

This is not a knowledge gap: prompted with “was Officer Diaz testifying at this proceeding,” GPT-4o correctly answers yes. The failure is that the baseline does not pose this sub-question to itself.

Threshold conflation. On diversity jurisdiction, the statute requires the amount in controversy to exceed \$75,000 strictly. The baseline accepts the boundary value. The condition exists in the model’s training data; it is not weighted heavily enough in the single-pass answer.

```
Fact: Plaintiff (NY) sues Defendant (CA);
amount in controversy = $75,000 exactly.
GPT-4o (baseline): "Yes -- parties diverse and
amount meets the $75,000 threshold."
SRA c2 ("EXCEED $75,000; equal is insufficient"):
False -> answer No.
```

Precedence collapse. Disjunctive rules with hierarchical structure are evaluated flatly. Personal jurisdiction is satisfied by domicile or consent or (minimum contacts $\wedge$ relatedness $\wedge$ reasonableness). If the defendant is domiciled in the forum, the specific-jurisdiction prong is irrelevant. The baseline evaluates it anyway, occasionally returning “No” because the contacts are weak, even though general jurisdiction obviously applies. SRA’s dependency edges short-circuit: $c_1 \text{ true} \Rightarrow c_3, c_4, c_5 \text{ skipped}$.

These three failure modes have a common cause: the rule’s logical structure is collapsed into a single generation step. Once we restore the structure outside the LLM, the failures disappear—except where the rule itself does not have the structure SRA imposes, which we analyze in Section 6.

3. Structured Rule Application

Definition 1 (Condition tree). A condition tree $R = (C, \text{deps}, \text{agg})$ consists of conditions $C = \{c_1, \dots, c_k\}$ each with a natural-language description d_i ; a dependency relation $\text{deps}(c_i) \subseteq C$; and an aggregation function $\text{agg} : \{0, 1\}^k \rightarrow A$ mapping a vector of condition truth values to an answer in the task’s answer set A .

Given facts F and tree R , SRA evaluates conditions in topological order. For each c_i : if any dependency evaluated to false, c_i is short-circuited to false; otherwise $\text{LLM}(d_i, F)$ returns a JSON object $\{\text{holds: bool, evidence: str}\}$. The final answer is $\text{agg}(v_1, \dots, v_k)$. The LLM never sees the aggregation function and is never asked to combine conditions. Each call sees only d_i and F .

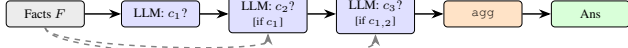


Figure 1. The SRA pipeline for hearsay (FRE 801). Each blue node is an independent LLM call scoped to a single condition; facts are shared (dashed). The orange node is a hard-coded boolean function—never an LLM call.

Table 1. Condition-tree summaries. $\wedge$ = conjunction; *first* = first true condition in priority order; *if/else* = binary branch. Full prompts in Appendix A.

Task	Conditions	Agg.
Hearsay (FRE 801)	(1) Is a “statement”? (2) Out-of-court? [c_1] (3) Offered for truth? [$c_{1,2}$]	$\wedge$
Diversity juris. (§1332)	(1) Complete diversity. (2) Amount > \$75k.	$\wedge$
Pers. juris. (<i>Int’l Shoe</i>)	(1) Domicile/presence. (2) Consent. (3) Min. contacts $\wedge$ fair play $\wedge$ sub. justice.	$\vee$
Abercrombie	(1) Generic. (2) Descriptive. (3) Suggestive. (4) Fanciful. (5) Arbitrary.	<i>first</i>
Telemarketing SR	(1) Is a telemarketer? (2) Violated a TSR provision? [c_1]	$\wedge$
UCC vs. common law	(1) Transaction primarily goods?	<i>if/else</i>
J.Crew blocker	(1) Material IP? (2) Transfer restricted?	$\wedge$

3.1. Pre-Encoded Rule Trees

We encode 12 of the 14 evaluated LegalBench tasks as condition trees from primary doctrinal sources. Representative trees are shown in Table 1; the full appendix gives every condition’s prompt.

3.2. Dynamic Condition Extraction

For SARA (Holzenberger et al., 2020), where the governing statute is provided at runtime, pre-encoded trees are inapplicable. We implement SRA-dynamic: the LLM first decomposes the statute into a JSON list of conditions following an extraction prompt with two worked examples (hearsay, diversity), then evaluates and aggregates as in SRA. A verifier prompt re-asks for the final answer when more than 25% of conditions ground to null (a fact-coverage proxy for “the model is uncertain”). We also apply SRA-dynamic to all 14 tasks as a controlled ablation: any gap between pre-encoded SRA and SRA-dynamic isolates the value of expert encoding from the value of decomposition.

3.3. FOL Grounding (SARA only)

For comparison with LOGIC-LM (Pan et al., 2023), we implement a FOL grounding pipeline that extracts a typed predicate schema from the statute, grounds the predicates

from the facts, and evaluates a Python boolean expression. Threshold checks are normalized into derived boolean predicates. The pipeline reverts to SRA when schema extraction fails.

4. Experimental Setup

Tasks. 14 LegalBench rule-application tasks: hearsay; six diversity-jurisdiction variants; personal jurisdiction; Abercrombie; telemarketing; UCC vs. common law; J.Crew blocker; SARA entailment; SARA numeric. Full test splits, $n = 2,602$ in total.

Models. GPT-4o (gpt-4o-2024-08-06), GPT-4o-mini (gpt-4o-mini-2024-07-18), and GPT-5.5 (gpt-5.5-2026-03), accessed via the OpenAI API.

Baselines. Zero-shot (Kojima et al., 2022); few-shot (4 demos from the train split); chain-of-thought (Wei et al., 2022); self-consistency (Wang et al., 2023) ($k=5$ samples, majority vote); SRA-dynamic; FOL grounding (SARA only).

Metric. Exact-match accuracy on normalized labels (e.g. “Yes/No” regexed, numeric within \$1). 95% confidence intervals from 10,000 bootstrap resamples.

Decoding. Temperature 0, top- $p = 1$ for all calls (baselines and SRA). Self-consistency uses temperature 0.7 with $k=5$ samples. Per-condition calls use max_tokens=256; baseline and CoT calls use 256 and 512 respectively. All runs use a single fixed seed.

Few-shot selection. The 4 in-context demonstrations are the first 4 examples from each task’s train split, fixed across all runs and models. No selection or optimization was performed.

5. Main Results

On GPT-4o, SRA improves overall accuracy by +11.0 pp and beats every baseline on 9 of 14 tasks (Table 2). The largest gains are on tasks with multiple, individually-deterministic binary conditions: hearsay (+21.3 pp) and the six diversity variants (+5 to +21 pp), where SRA reaches 97–100% on four of six. The two regressions are interpretable and motivate Section 6: Abercrombie (−25.3 pp), where conditions must be evaluated comparatively rather than in isolation, and telemarketing (−6.4 pp), where the second condition (“violated a TSR provision”) bundles many specific provisions and loses the granularity that makes SRA work.

5.1. Cross-Model Evaluation

The pattern transfers across models but not uniformly.

Three observations from Table 3.

Table 2. Accuracy (%) on 14 LegalBench rule-application tasks (GPT-4o). Best per row in **bold**. Δ = SRA – few-shot. Overall is sample-weighted.

Task	n	0-shot	CoT	SC ($k=5$)	Few-shot	SRA	Δ
Hearsay	94	53.2	47.9	56.4	56.4	77.7	+21.3
Diversity 1	300	56.3	75.3	85.3	85.3	100.0	+14.7
Diversity 2	300	41.7	70.0	77.7	77.7	98.7	+21.0
Diversity 3	300	78.7	79.3	84.3	84.3	97.0	+12.7
Diversity 4	300	61.7	91.3	91.7	91.7	97.0	+5.3
Diversity 5	300	62.3	90.0	71.0	71.0	84.0	+13.0
Diversity 6	300	61.3	86.0	71.7	71.7	85.0	+13.3
Pers. juris.	50	84.0	94.0	88.0	88.0	88.0	0.0
Abercrombie	95	62.1	20.0	74.7	73.7	48.4	−25.3
Telemarketing	47	80.9	83.0	89.4	89.4	83.0	−6.4
UCC v. common	94	94.7	42.6	95.7	95.7	96.8	+1.1
J.Crew	54	20.4	64.8	88.9	88.9	90.7	+1.9
SARA ent.	272	83.1	86.8	83.1	71.7	86.8	+15.1
SARA num.	96	0.0	0.0	0.0	0.0	9.4	+9.4
Overall	2602	61.5	74.4	77.6	76.4	87.3	+11.0

Table 3. Overall accuracy by method across three models (sample-weighted, $n=2,602$). Δ_{SRA} relative to few-shot baseline.

Model	0-sh	CoT	SC	FS	SRA	Δ
4o-mini	50.1	67.8	68.8	66.2	65.7	−0.5
4o	61.5	74.4	77.6	76.4	87.3	+11.0
5.5	68.5	69.5	67.0	66.4	88.6	+22.2

(1) **SRA’s gain doubles from GPT-4o to GPT-5.5.** On GPT-5.5, diversity 1–4 reach 100% and diversity 5–6 reach 91–98% (vs. baselines of 43–84%); hearsay reaches 84% (vs. 60%). The overall gain is +22.2 pp.

(2) **The stronger model has the weaker baseline on rule-application tasks.** GPT-5.5’s few-shot baseline (66.4%) is below GPT-4o’s (76.4%). The gap is concentrated on diversity-jurisdiction tasks 3, 5, and 6, where GPT-5.5 scores 43–52% to GPT-4o’s 71–84%. The same model with SRA scores $\geq 91\%$ on every diversity variant. We do not have a mechanistic explanation for this inversion, but note that model improvements between generations are not uniformly distributed across reasoning styles.

(3) **GPT-4o-mini does not benefit from SRA in aggregate (−0.5 pp).** The drop is concentrated on diversity tasks 3–6, where SRA falls to 44–60% (vs. a baseline of 40–89%). The model fails to reliably evaluate individual conditions: spot-checks show false positives on “complete diversity” when one defendant is listed before all plaintiffs. Crucially, the same model gains +4.8 pp from SRA-dynamic (70.98% vs. 66.22%). The decomposition framing helps; the specific expert wording of the pre-encoded conditions does not, because the model cannot reliably parse the qualifiers (“strictly greater than”, “NO plaintiff shares a state with ANY defendant”). This is not a capability threshold per se; it is a capacity ceiling for instruction granularity.

Table 4. Pre-encoded SRA vs. runtime-extracted SRA-dynamic. Read Δ_{exp} as the marginal value of expert encoding, holding decomposition fixed.

Task	Model	SRA-dyn	SRA	Δ_{exp}
Hearsay	GPT-4o	80.9	77.7	−3.2
Diversity 1–6 (avg)	GPT-4o	79.1	92.7	+13.6
Personal juris.	GPT-4o	82.0	88.0	+6.0
Abercrombie	GPT-4o	16.8	48.4	+31.6
Telemarketing	GPT-4o	61.7	83.0	+21.3
J.Crew	GPT-4o	31.5	90.7	+59.2
SARA ent.	GPT-4o	87.5	86.8	−0.7
Overall	GPT-4o-mini	71.0	65.7	−5.3
Overall	GPT-4o	75.9	87.3	+11.4
Overall	GPT-5.5	83.9	88.6	+4.7

5.2. Ablation: Expert Encoding vs. Decomposition

To isolate whether SRA’s gains come from the structured-decomposition procedure or from the expert-encoded content, we run SRA-dynamic (Section 3) on all 14 tasks. Both pipelines use independent per-condition evaluation and deterministic aggregation; only the source of conditions differs.

Table 4 resolves the question. Expert encoding contributes the bulk of SRA’s value on tasks with precise categorical distinctions: J.Crew (+59.2 pp), Abercrombie (+31.6 pp), telemarketing (+21.3 pp). On these tasks the LLM extracts verbose, overlapping conditions that do not normalize cleanly to the answer set (e.g. “Does the mark merely describe?” yields free-text rather than the five-class label). Expert encoding constrains the answer surface.

On simple binary doctrines—hearsay, diversity 1, SARA entailment—runtime extraction matches or slightly exceeds pre-encoded SRA. The LLM’s spontaneous decomposition

Table 5. SARA entailment accuracy across methods and models. FOL grounding extracts a typed predicate schema and evaluates a Python boolean. SRA evaluates each condition as a binary NL question.

Method	GPT-4o-mini	GPT-4o	GPT-5.5
Few-shot baseline	73.2	71.7	85.3
Zero-shot	70.6	83.1	79.4
CoT	80.1	86.8	88.6
FOL grounding	65.1	77.9	64.3
SRA-dynamic	76.8	87.5	74.3
SRA	82.4	86.8	61.0

of “hearsay = statement + out-of-court + offered for truth” is essentially what an expert would write.

The cross-model row contains the most surprising finding: expert encoding is harmful for the weakest model. On GPT-4o-mini, SRA-dynamic outperforms pre-encoded SRA by 5.3 pp overall. The mini model cannot reliably parse the strict-inequality qualifier or the “every pair” quantification baked into the diversity conditions and answers them incorrectly, whereas the looser, model-generated decomposition lands within the model’s instruction-following capacity. This means the optimal pipeline depends on the model: practitioners deploying small models should prefer dynamic decomposition, even if expert trees are available.

5.3. FOL Grounding Underperforms NL Conditions

Table 5 shows that FOL grounding underperforms SRA on every model. On GPT-4o the gap is 9 pp; on GPT-5.5 it is 24 pp behind CoT. The reason is structural: SARA statutes contain qualitative conditions (“engaged in a trade or business,” “provided through an employer”) that resist typed-predicate encoding. When the schema-extraction step compiles such conditions into bool predicates, downstream grounding sets them to null (no explicit fact in the description), and the conservative-False default flips the verdict to “Contradiction.” SRA’s natural-language conditions tolerate this fuzziness because the LLM does not need a typed object—it just answers the question.

The GPT-5.5 SARA entailment regression is its own anomaly. Pre-encoded SRA’s verifier step (added to recover from extraction errors on weaker models) systematically downgrades GPT-5.5’s already-strong baseline answer to “Contradiction” when conditions are over-extracted. SRA-dynamic on GPT-5.5 retains 74.3%, indicating the issue is the verifier interaction, not the decomposition; we discuss this in Section 6.

6. Analysis

The empirical results admit a consistent explanation. We state it, then derive exclusion criteria for when SRA should

not be applied.

Two factors gate the gain. The rule must have the structure SRA imposes (binary, independently-evaluable conditions with a deterministic aggregator). The model must lack the ability to apply that structure internally.

Empirically, the relationship between baseline accuracy and SRA gain is monotone: tasks where the baseline already exceeds 90% see zero or negative gains; tasks below 60% see gains of 20–55 pp. The few exceptions correspond to tasks whose structure mismatches SRA’s encoding (Abercrombie, telemarketing).

When SRA helps. Multi-condition binary doctrines where conditions are independently evaluable from facts, the aggregator is purely logical, and the baseline model does not internalize the structure. Hearsay, diversity, J.Crew (on GPT-4o), and SARA (on GPT-4o) all fit. Gains scale with the baseline gap: GPT-5.5 on diversity 5 (baseline 43%) gains +55 pp; GPT-4o on the same task (baseline 71%) gains +13 pp.

When SRA hurts: structural mismatch. Ordinal/comparative classification. Abercrombie requires placing a mark on a 5-point spectrum where adjacent categories (descriptive vs. suggestive) are distinguished by relative degree of imagination required. Per-condition evaluation removes the cross-category context that few-shot baselines use to calibrate. The encoding loses the contrast that the rule is fundamentally about.

Vague conditions. Telemarketing’s “violated a TSR provision” compresses dozens of specific provisions into a single binary. The baseline’s holistic read catches the relevant provision; SRA’s binarized wrapper does not. The encoding is too coarse, not too fine.

Single-condition rules with strong baselines. UCC, J.Crew on GPT-5.5, and personal jurisdiction often need only a single qualitative judgment that LLMs perform well in one pass. Decomposition adds noise. The most striking case: GPT-5.5’s baseline on J.Crew is 100%; SRA drops to 61% because the pre-encoded “material IP” condition fires false on contracts where materiality is implied but not stated explicitly.

When SRA hurts: capacity mismatch. GPT-4o-mini’s failure on diversity 3–6 under SRA is qualitatively different from the structural-mismatch failures. The rule’s structure is fine; the model cannot reliably evaluate the precisely-worded individual conditions. Symptoms: the model marks “complete diversity” as true on fact patterns containing both an in-state plaintiff and an in-state defendant when they happen not to face each other directly. Looser, model-generated conditions perform better because they match

the model’s coarser internal representation. Practical implication: small models prefer SRA-dynamic to expert SRA; the dependence cuts the opposite way from the conventional wisdom that “smaller models need more scaffolding.”

When SRA hurts: verifier overcorrection. GPT-5.5’s SARA-entailment regression (-24.3 pp) is a third failure mode. The pipeline’s verifier-on-null-fraction trigger (Section 3) was tuned for weaker models that over-extract conditions; on GPT-5.5, the verifier interferes with the model’s already-correct holistic judgment. SRA-dynamic, which does not use this verifier, regresses by only -11 pp. The lesson is general: harness logic tuned for one model may be miscalibrated for another.

6.1. Exclusion Criteria

Combining these patterns, we offer three exclusion criteria. If a (model, task) pair passes all three, pre-encoded SRA is expected to help; the gain scales with the baseline gap.

1. Does the rule admit a binary condition tree with a deterministic aggregator?
No (comparative, ordinal, or vague): do not use SRA. (Abercrombie, telemarketing.)
2. Is the baseline model’s accuracy already $\geq 90\%$?
Yes: do not use SRA—no headroom, and any encoding mismatch will regress. (UCC on most models, J.Crew on GPT-5.5, personal jurisdiction.)
3. Is the model strong enough to reliably evaluate scoped binary instructions?
No (e.g. GPT-4o-mini on diversity 3–6): prefer SRA-dynamic to pre-encoded SRA. Looser conditions match looser instruction-following.

These criteria are consistent with every cell in Tables 2 and 6.

Cost. SRA multiplies API calls by the number of conditions ($2-5\times$), but per-condition completions are short (256 tokens); on diversity, 2 calls suffice for perfect accuracy on the two stronger models.

7. Related Work

Prompted reasoning structures. Chain-of-thought (Wei et al., 2022), Tree-of-Thoughts (Yao et al., 2023), and least-to-most prompting (Zhou et al., 2023) improve multi-step reasoning by introducing intermediate generation steps but leave the structure of those steps to the model. Chain-of-Logic (Servantez et al., 2024) and legal syllogism prompting (Blair-Stanek et al., 2023) apply this idea to legal reasoning. SRA differs by externalizing the logical aggregator and by pre-encoding the conditions from doctrine; our ablation in

Table 4 shows that both choices matter on rules with precise categorical distinctions, while clean binary rules are well served by runtime decomposition.

Symbolic-solver pipelines. PAL (Gao et al., 2023) delegates arithmetic to Python; LOGIC-LM (Pan et al., 2023) translates natural-language problems to first-order logic for SMT solvers. SRA shares the spirit of offloading inference from the LLM but keeps conditions in natural language. Our FOL ablation shows the cost of full formalization on statutory text: qualitative predicates resist typed encoding, and accuracy drops 9–34 pp relative to NL SRA.

Rules as Code. Catala (Mérigoux et al., 2021) compiles statutes to executable code; the OECD’s “Rules as Code” initiative (Mohun & Roberts, 2020) advocates for machine-readable legislation at drafting time; Witt et al. (2024) and Surden (2024) survey the broader program. SRA is a much lighter point on this design space: rule structure is imposed at inference time, in 2–5 conditions per doctrine, with no statutory compilation.

Legal LLM evaluation. LegalBench (Guha et al., 2023) is the canonical benchmark for legal reasoning in LLMs. Prior work focuses on either coverage (more tasks) or strategy (one prompting style). We instead use 14 tasks as a controlled cross-product against three models and six methods, surfacing interactions that single-model studies cannot.

8. Discussion

Implications for deployment. A practitioner can encode a doctrine once as a condition tree and run it across any API-accessible model with no training. The encoding effort is modest (an afternoon per doctrine; we release ours). The decision rules in Section 6.1 are sharper than “use SRA when in doubt”—some tasks regress, and the criteria for those regressions are predictable. Practitioners should apply SRA selectively, not universally.

Future work. Three directions matter. First, condition calibration: replace boolean condition outputs with calibrated probabilities and propagate them through the aggregator, enabling uncertainty quantification. Second, automatic tree induction: combine SRA-dynamic’s extraction with a verifier that checks against doctrinal sources, closing the gap between automatic and expert encoding. Third, cross-doctrine composition: real legal questions often require multiple interacting rules (hearsay plus exceptions plus exclusions); composing condition trees compositionally is the natural next step.

9. Conclusion

LLMs fail at legal rule application because they collapse the rule’s logical structure into a single generation step. Restoring that structure outside the LLM—through condition trees with hard-coded aggregators—closes most of the gap on the doctrines that have such structure. Across 14 LegalBench tasks and three models, SRA improves overall accuracy by +11 pp on GPT-4o and +22 pp on GPT-5.5, but the gain is not uniform: it vanishes when the baseline saturates and reverses when the rule’s structure resists binary decomposition or the model cannot follow scoped instructions reliably. Each regression is informative: the Abercrombie collapse identifies ordinal classification as outside SRA’s scope; the GPT-4o-mini result shows that small models need looser scaffolding, not more; the GPT-5.5 SARA regression flags harness logic miscalibration. We see the central contribution as empirical and practical: not that SRA works, but that we can identify in advance when it will work.

10. Limitations

Coverage. 14 of 162 LegalBench tasks, targeting rule-application specifically. **Small task sizes.** Several tasks have small n (personal jurisdiction: 50, telemarketing: 47, J.Crew: 54, hearsay: 94), limiting statistical power; confidence intervals are wide. **Models.** OpenAI’s GPT-4o, GPT-4o-mini, GPT-5.5 only; we do not test open-weight models. The capability-threshold finding may not transfer. **Encoding effort.** Each condition tree requires legal expertise. Our trees were authored by the authors and not independently validated by practicing lawyers. **Information asymmetry.** SRA condition prompts include expert-authored legal guidance (e.g., “exactly \$75,000 is insufficient”) not present in baselines. The comparison is therefore not purely “structure vs. no structure” but also “structure plus expert mini-rubrics vs. unguided prompting.” The ablation partially addresses this (SRA-dynamic gains on binary tasks without expert conditions), but a single-call baseline receiving equivalent guidance would more cleanly isolate decomposition’s effect. **Common-law evolution.** Doctrines that evolve through case law resist condition-tree encoding; our taxonomy does not address these. **Independence assumption.** SRA evaluates conditions independently, which breaks down for semantically entangled rules; our Abercrombie regression makes this concrete. **Benchmark contamination.** LegalBench is publicly available and may be partially memorized by frontier models; our results are best interpreted as a controlled benchmark study. **Error analysis.** Our failure-mode categories (§2) are qualitative; a systematic annotation study with labeled error counts and inter-annotator agreement would strengthen these claims. **SARA numeric.** SRA achieves only 9.4% (GPT-4o) and 22.9% (GPT-5.5), showing that the method does not solve numeric statutory

reasoning.

References

- Blair-Stanek, A., Holzenberger, N., and Van Durme, B. Can GPT-3 perform statutory reasoning? In *Proceedings of the International Conference on Artificial Intelligence and Law (ICAIL)*. ACM, 2023.
- Gao, L., Madaan, A., Zhou, S., Alon, U., Liu, P., Yang, Y., Callan, J., and Neubig, G. PAL: Program-aided language models. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2023.
- Guha, N., Nyarko, J., Ho, D. E., Ré, C., Chilton, A., Narayana, A., Chohlas-Wood, A., Peters, A., Waldon, B., Rockmore, D., et al. LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2023.
- Holzenberger, N., Blair-Stanek, A., and Van Durme, B. A dataset for statutory reasoning in tax law entailment and question answering. In *Proceedings of the Natural Language Processing Workshop (NLLP)*, 2020.
- Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, 2022.
- Mérigoux, D., Chataing, N., and Protzenko, J. Catala: A programming language for the law. In *Proceedings of the ACM on Programming Languages (ICFP)*. ACM, 2021.
- Mohun, J. and Roberts, A. Cracking the code: Rulemaking for humans and machines. OECD Working Papers on Public Governance 42, Organisation for Economic Co-operation and Development, 2020.
- Pan, L., Albalak, A., Wang, X., and Wang, W. Y. LOGIC-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, 2023.
- Servantez, S., Barrow, J., Hammond, K., and Jain, R. Chain of logic: Rule-based reasoning with large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2721–2733, 2024.
- Surden, H. Computable law and artificial intelligence. In *Cambridge Handbook of Private Law and Artificial Intelligence*. Cambridge University Press, 2024. Available at SSRN.
- Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in lan-

guage models. In *International Conference on Learning Representations*, 2023.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

Witt, A., Huggins, A., Governatori, G., and Buckley, J. Encoding legislation: A methodology for enhancing technical validation, legal alignment and interdisciplinarity. *Artificial Intelligence and Law*, 32:293–324, 2024.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., and Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, 2023.

Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q. V., and Chi, E. Least-to-most prompting enables complex reasoning in large language models. In *International Conference on Learning Representations*, 2023.

A. Complete Condition Tree Encodings

The following lists the production condition-tree specifications used for the pre-encoded SRA results. Each condition includes parenthetical guidance that the LLM sees inside the per-condition prompt.

```
Rule: hearsay (FRE 801)
  c1: Is this a "statement"?
      (oral assertion, written assertion, or
       nonverbal conduct intended as assertion)
  c2: Was the statement made outside of the
      current trial or hearing?
      [depends_on: c1]
  c3: Is it offered to prove the truth of the
      matter asserted? (distinguish: effect on
      listener, verbal act, operative words,
      notice)
      [depends_on: c1, c2]

agg: Yes iff (c1 AND c2 AND c3)
```

```
Rule: diversity_jurisdiction (28 U.S.C. 1332)
  c1: Are all plaintiffs citizens of different
      states from all defendants?
      (complete diversity: NO plaintiff shares
      a state with ANY defendant)
  c2: Does the amount in controversy EXCEED
      $75,000? (strictly greater than; exactly
      $75,000 is insufficient)

agg: Yes iff (c1 AND c2)
```

```
Rule: personal_jurisdiction (Int'l Shoe)
  c1: Domiciled or physically present in the
      forum state? (general jurisdiction)
  c2: Consented to jurisdiction? (forum
      selection, registration, appearance)
  c3: Minimum contacts with the forum state?
      (purposeful availment)
  c4: Claim arises out of or relates to those
```

```
contacts? [depends_on: c3]
c5: Exercise of jurisdiction reasonable?
    [depends_on: c3]

agg: Yes iff c1 OR c2 OR (c3 AND c4 AND c5)
```

```
Rule: abercrombie_spectrum
  c1: Generic? Refers to product class itself
  c2: Descriptive? Directly describes a quality
  c3: Suggestive? Requires imagination
  c4: No dictionary meaning relative to product?
      (arbitrary or fanciful)
  c5: Coined/invented word? [depends_on: c4]
      (distinguishes fanciful from arbitrary)

agg: priority cascade c1 -> c2 -> c3 ->
     (c4 ? (c5 ? fanciful : arbitrary) :
     previous category)
```

```
Rule: telemarketing_sales_rule (16 CFR 310)
  c1: Is the entity engaged in telemarketing?
      (initiating telephone calls to consumers
      to induce purchase of goods/services)
  c2: Did conduct violate a TSR provision?
      (misrepresentation, unauthorized billing,
      do-not-call, identity/purpose disclosure,
      threats, prize misrepresentation, etc.)
      [depends_on: c1]

agg: Yes iff (c1 AND c2)
```

```
Rule: ucc_v_common_law
  c1: Does the transaction primarily involve
      sale of goods (movable, tangible
      personal property)?
      (predominant purpose test for mixed
      contracts)

agg: "UCC" if c1 else "Common Law"
```

```
Rule: jcrew_blocker
  c1: Is the IP classified as material to the
      borrower and restricted subsidiaries
      taken as a whole?
  c2: Does the clause restrict transfer of
      this IP to unrestricted subsidiaries?

agg: Yes iff (c1 AND c2)
```

B. Per-Task Cross-Model Table

Table 6 reports the per-task accuracy for SRA and few-shot baseline across all three models.

Table 6. Per-task accuracy by (model, method). FS = few-shot baseline. Δ = SRA – FS. All percentages.

Task	GPT-4o-mini			FS	GPT-4o			FS	GPT-5.5		
	FS	SRA	Δ		FS	SRA	Δ		FS	SRA	Δ
Hearsay	60.6	57.4	−3.2	56.4	77.7	+21.3	59.6	84.0	+24.4		
Div. 1	73.7	99.3	+25.7	85.3	100	+14.7	78.3	100	+21.7		
Div. 2	65.7	86.7	+21.0	77.7	98.7	+21.0	74.0	99.7	+25.7		
Div. 3	82.3	59.7	−22.7	84.3	97.0	+12.7	51.7	100	+48.3		
Div. 4	89.3	52.3	−37.0	91.7	97.0	+5.3	83.7	100	+16.3		
Div. 5	61.3	44.0	−17.3	71.0	84.0	+13.0	43.0	97.7	+54.7		
Div. 6	40.3	47.0	+6.7	71.7	85.0	+13.3	47.0	91.0	+44.0		
Pers. Jur.	76.0	68.0	−8.0	88.0	88.0	0.0	96.0	94.0	−2.0		
Abercrombie	38.9	52.6	+13.7	73.7	48.4	−25.3	73.7	67.4	−6.3		
Telem. SR	59.6	78.7	+19.1	89.4	83.0	−6.4	89.4	78.7	−10.7		
UCC	94.7	97.9	+3.2	95.7	96.8	+1.1	97.9	96.8	−1.1		
J.Crew	68.5	88.9	+20.4	88.9	90.7	+1.9	100	61.1	−38.9		
SARA ent.	73.2	82.4	+9.2	71.7	86.8	+15.1	85.3	61.0	−24.3		
SARA num.	0.0	3.1	+3.1	0.0	9.4	+9.4	0.0	22.9	+22.9		
Overall	66.2	65.7	−0.5	76.4	87.3	+11.0	66.4	88.6	+22.2		