Language Models and Logic Programs for Trustworthy Tax Reasoning

William Jurayj[1], **Nils Holzenberger**[2], **Benjamin Van Durme**[1]

[1]Johns Hopkins University
[2]Télécom Paris, Institut Polytechnique de Paris
wjurayj1@jhu.edu

Abstract

According to the United States Internal Revenue Service, "the average American spends $270 and 13 hours filing their taxes". Even beyond the U.S., tax filing requires complex reasoning, combining application of overlapping rules with numerical calculations. Because errors can incur costly penalties, any automated system must deliver high accuracy and auditability, making modern large language models (LLMs) poorly suited for this task. We propose an approach that integrates LLMs with a symbolic solver to calculate tax obligations. We evaluate variants of this system on the challenging StAtutory Reasoning Assessment (SARA) dataset, and include a novel method for estimating the cost of deploying such a system based on real-world penalties for tax errors. We further show how combining up-front translation of plain-text rules into formal logic programs, combined with intelligently retrieved exemplars for formal case representations, can dramatically improve performance on this task and reduce costs to well below real-world averages. Our results demonstrate the effectiveness of applying semantic parsing methods to statutory reasoning, and show promising economic feasibility of neuro-symbolic architectures for increasing access to reliable tax assistance.

Code — https://github.com/wjurayj/legal_logic_programs
Extended version — https://arxiv.org/abs/2508.21051

Introduction

"GPT is not a certified tax professional, nor am I, so you should always check with your tax advisor."
— Greg Brockman, CTO of OpenAI

Of life's two certainties, taxes should be preferred; yet they may well be the more complicated one. Each year, virtually every adult in the world must calculate and pay a fee to some government, in order to reside and earn a living within the state's guardianship. Even for individuals with relatively simple financial situations, the annual filing process demands meticulous reading and following of dozens of form instructions and the copying of values across schedules, worksheets, and eligibility tests. Completing these tasks without professional assistance can take hours. Alternatively, taxpayers may hire a professional preparer, incurring substantial fees depending on the complexity of their return (Internal Revenue Service 2025).

Accuracy in tax filing is essential. Over-reported income or missed deduction opportunities lead to unnecessary overpayment, while under-reporting may result in penalties, interest, and potential legal consequences. In the United States, the costs of inaccuracies affect lower income communities more significantly, in part because these groups offer the Internal Revenue Service (IRS) high audit success rates (Dean 2022; Black et al. 2022; Elzayn et al. 2025). However, these audits deliver a modest return on investment compared to audits of wealthier taxpayers, so there is an opportunity to better align community and institutional interests through improved tax advice to lower income taxpayers (Boning et al. 2024).

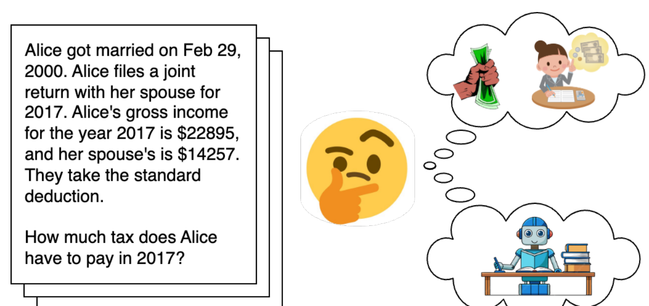

Figure 1: A taxpayer confronted with a tax question might choose between an inexpensive AI preparer and a costlier human professional. The decision considers trade-offs between cost, convenience, and confidence in the result.

However, the concrete costs for errors present a substantial challenge for modern large language models (LLMs). An AI assistant deployed in this domain must meet higher standards than basic accuracy: it should (1) recognize when it lacks sufficient certainty to offer guidance and (2) generate a transparent and faithful trace of logical steps so that taxpayers and auditors can easily verify the derivation of each answer. In this paper we show that symbolic reasoning tools, integrated with LLMs, offer a promising approach to meeting these standards. Our method provides the language model with access to a symbolic solver, enabling it to trans-

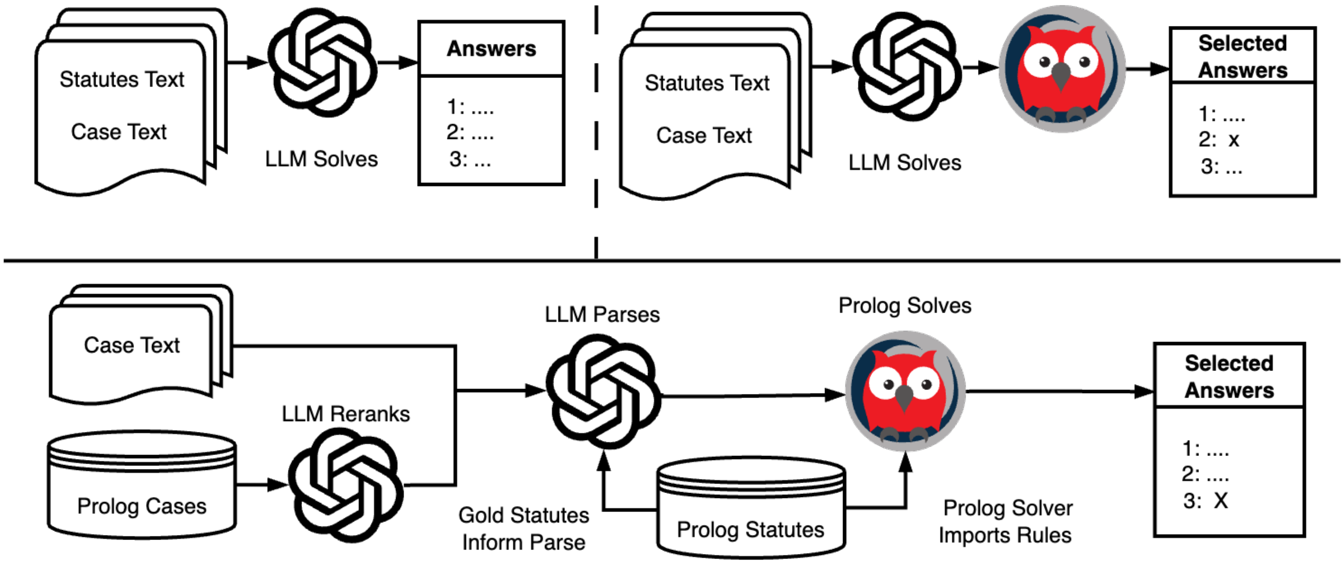

Figure 2: **Methods for solving. Top Left:** Plain-text for statutes and a case is fed into a language model, along with the instruction to calculate a person's tax obligation. **Top Right:** Statutes and a case are fed into the model as before, but it is instructed to convert these into a logic program which calculates a person's tax obligation. If the SWI-Prolog engine fails to execute the program, the case is considered unanswered. **Bottom:** A language model parses a case's facts into Prolog, conditioned on gold parses of the most relevant cases and of the rules contained in the statutes. The symbolic solver imports the gold parses of the statutes before attempting to execute the generated parse of the case. Note that unlike the approaches above it, this requires gold symbolic representations of both the statutes and a representative selection of correctly-decided cases.

late statutory text and taxpayer information into logic programs, which are processed by a trusted execution engine. We evaluate the method on the StAtutory Reasoning Assessment (SARA) dataset, a benchmark of tax scenarios paired with liability calculations carried out through ground-truth representations of rules and facts in formal logic (Holzenberger, Blair-Stanek, and Van Durme 2020).

Our experiments are the first to demonstrate strong performance by applying semantic parsing methods to statutory reasoning. This includes two key findings. First, whereas frontier reasoning models outperform non-reasoning models at both directly solving and at parsing case and statute text into the symbolic solver, non-reasoning models consistently outperform their reasoning counterparts when given gold symbolic representations of statutes and of their application to similar cases. Second, we show that by adding additional refusal criteria through a symbolic solver and self-checking, the expected costs of deploying such a system in the real world could be brought down to less than 20% of the average cost for an American to file their taxes. Our results indicate the promise of neuro-symbolic architectures for expanding access to trustworthy and reliable tax expertise.

Background

Logic Programming for Legal Reasoning

Several programming languages have been designed to represent and facilitate logical reasoning. Prolog is a declarative programming language for representing and reasoning over knowledge, with roots in first order logic. A programmer defines rules using Horn clauses (Horn 1951) and facts by declaring which rules apply to entities, thus populating a knowledge base. Subsequently, this knowledge base can be queried by defining a 'goal', which launches computation in the form of a backward-chaining search attempting to prove that the goal holds a certain value by iteratively unifying sub-goals (Wielemaker et al. 2010). Prolog has been used since the early days of legal AI, where it has formed the backbone of legal expert systems because its declarative syntax keeps knowledge base entries for rules human-readable while powerfully representing the reasoning around which legal questions revolve (Sherman 1989). Efforts in countries like the United Kingdom (Sergot et al. 1986), Canada (Sherman 1987), and the United States (Kant et al. 2025) have leveraged this capacity to encode legal rules in executable formal logic. Related languages have also been used in the legal domain, such as Answer-set Programming (Gelfond and Lifschitz 1988; Morris 2020), Datalog (Ceri, Gottlob, and Tanca 1989), and Catala (Merigoux, Chataing, and Protzenko 2021; Merigoux 2023). More broadly, hierarchical templates are a popular tool for evaluation of legal reasoning (Hou et al. 2024). We focus on the SARA dataset (Holzenberger, Blair-Stanek, and Van Durme 2020), which encodes statutes and cases into Prolog logic programs to show how a symbolic expert system can perfectly solve a task which large language models struggle to complete.

Statutory Tax Reasoning and the SARA Dataset

We focus on the task of statutory reasoning for tax law. Some elements of this task bear similarity to popular math-

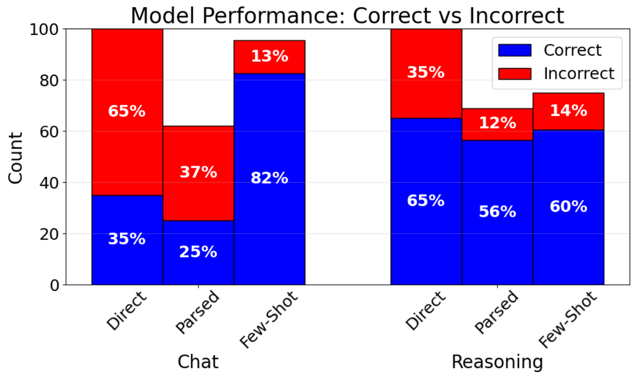

Figure 3: Number of correct and incorrect solutions produced by each solution method, for large chat- and reasoning-optimized models (served by DeepSeek and OpenAI).

ematical reasoning tasks such as GSM-8k (Cobbe et al. 2021) or MATH-500 (Hendrycks et al. 2021), such as chaining together mathematical operations to solve a real-world problem described in words. However, unlike these math datasets which require application of a small set of universal arithmetic rules which models learn during training, statutory reasoning considers a set of contingent rules contained within documents provided to a model in-context at inference time, in addition to these basic arithmetic principles.

We evaluate our methods on the SARA dataset, which tests the ability of language models to do statutory reasoning about the United States Tax Code (Holzenberger, Blair-Stanek, and Van Durme 2020). This dataset is included in the popular aggregate benchmark LegalBench (Guha et al. 2023), and was used in the GPT-4 product launch to highlight the model's superior reasoning capacity (Blair-Stanek, Holzenberger, and Van Durme 2024). The SARA dataset consists of 9 sections from the US federal tax code which have been moderately edited to make them self-contained and unambiguous. These statute sections are accompanied by 376 hand-crafted cases to test understanding of these statutes, each containing a question about a person's tax obligation. Each statute and case has been manually translated into Prolog, which allows them to be trivially solved using the language's powerful execution engine to resolve queries about cases. This Prolog is defined using neo-Davidsonian event semantics (Davidson 1966), categorizing each event as one of 61 possible predicates onto which various arguments are attached. 276 of these cases require binary responses about whether a section applies, and the remaining 100 require calculating a tax obligation; we focus on the 100 cases because of their increased difficulty and because a trivial baseline of always guessing a single answer delivers poor performance at predicting the numerical output.

Methodology

Direct Calculation

We evaluate our methods against the baseline method of direct solving, mirroring the approach used in OpenAI's GPT-4 demonstration (Blair-Stanek, Holzenberger, and Van Durme 2024). This approach treats the tax calculation as a mathematical question answering task with the additional demand that the model must apply entries from a large corpus of rules contained in the statutory text, in addition to the generic arithmetic rules that govern all calculation. For each case in this setting, a model's context is filled with all sections of the statutes concatenated together, the description of the case's facts, and the question about a person in that case's tax liability, all in plain text. It is instructed to calculate the person in question's tax obligation based on the rules outlined in the statutes.

Chat Model	Reasoning Model	Size
Qwen2.5-Coder	R1-Distill Qwen2.5	32 billion
Llama 3.3	R1-Distill Llama 3.3	70 billion
DeepSeek-V3	DeepSeek-R1	671 billion
GPT-4.1	OpenAI o3	$8/m tokens

Table 1: **Chat and Reasoning model pairs.** Each open-weight model pair is fine-tuned from the same base model. Although the exact dimensions and provenance of OpenAI's models are unknown, the two models have identical token pricing structures, suggesting that they incur similar costs for OpenAI to serve.

Zero-Shot Parsing for a Symbolic Solver

To extend the direct solution approach, we augment the language model with a symbolic solver. Here, a model is given the plain text of the statutes as in the direct calculation case. It is instructed to generate a Prolog program which encodes the relevant rules and facts necessary to compute the person in question's tax obligation. The symbolic solver ingests a set of rules and facts in Prolog, and is invoked to execute a query. The execution of this Prolog program offers a straightforward mechanism for refusal: if the program fails to execute into the proper format or hangs beyond a time limit (10 seconds), the system abstains from answering.

Few-Shot Parsing using Gold Statutes

We additionally consider what advantage could be given by offering language model parsers access to gold symbolic representations of the statutes, alongside demonstrations of cases applying the rules from these statutes, building on previous work showing how demonstrations can aid in semantic parsing (Shin and Van Durme 2022; Spiegel et al. 2024). Notably, the gold parsed cases all reference the same manually-translated Prolog representation of the statutes. As such, these manual translations formalize certain types of facts in one particular way where several may be viable, given alternative representations of the rules which are plausible but not implemented in practice. Thus, when using already parsed cases as few-shot examples, the execution engine must have access to this particular formalization of the rules as well. This reduces the per-case reasoning task to tax-relevant event extraction (Holzenberger and Van Durme 2023; Gantt et al. 2024)

Family	Model	Method	Correct	Incorrect	Abstentions	Break-Even Price
Baseline	N/A	Always Abstain	0	0	100	$270 ± 0
	N/A	Always $0	5	95	0	$16227.11 ± 7805.94
Qwen-2.5	Qwen-32b	Direct	13	87	0	$3,051.64 ± 1,828.31
	Qwen-32b	Parsed	2	17	81	$490.34 ± 230.75
	R1-32b	Direct	38	62	0	$505.25 ± 287.67
	R1-32b	Parsed	1	2	97	$278.70 ± 24.33
Llama-3.3	Llama-70b	Direct	9	91	0	$1,065.90 ± 675.07
	Llama-70b	Parsed	1	43	56	$252,027.73 ± 414,049.97
	R1-70b	Direct	43	57	0	$1,257.03 ± 1,620.47
	R1-70b	Parsed	2	1	97	$266.10 ± 6.81
DeepSeek	DeepSeek-V3	Direct	22	78	0	$739.45 ± 474.59
	DeepSeek-V3	Parsed	11	43	46	$2,099.13 ± 1,253.57
	DeepSeek-V3	Direct + Direct	16	15	69	$265.46 ± 63.53
	DeepSeek-V3	Direct + Parsed	7	4	89	$285.53 ± 55.57
	DeepSeek-V3	Parsed + Parsed	5	8	87	$310.47 ± 67.95
	DeepSeek-R1	Direct	74	26	0	$304.29 ± 225.57
	DeepSeek-R1	Parsed	38	10	52	$249.64 ± 84.77
	DeepSeek-R1	Direct + Direct	66	12	22	$94.20 ± 59.76
	DeepSeek-R1	Direct + Parsed	34	3	63	$170.10 ± 21.75
	DeepSeek-R1	Parsed + Parsed	17	4	79	$241.80 ± 29.45
OpenAI GPT-4.1	GPT-4.1	Direct	48	52	0	$532.84 ± 492.99
	GPT-4.1	Parsed	39	31	30	$228.89 ± 151.69
	GPT-4.1	Direct + Direct	42	13	45	$196.92 ± 88.43
	GPT-4.1	Direct + Parsed	27	6	67	$185.10 ± 21.33
	GPT-4.1	Parsed + Parsed	26	5	69	$186.30 ± 20.84
	o3	Direct	56	44	0	$6,431.84 ± 2,637.94
	o3	Parsed	75	15	10	$47.43 ± 22.16
	o3	Direct + Direct	41	17	42	$3,472.29 ± 1,859.32
	o3	Direct + Parsed	52	10	38	$115.90 ± 24.63
	o3	Parsed + Parsed	65	9	26	$77.51 ± 22.41
OpenAI GPT-5	GPT-5	Direct	76	24	0	$299.11 ± 288.41
	GPT-5	Parsed	53	13	34	$122.72 ± 29.21
	GPT-5	Direct + Direct	73	9	18	$218.64 ± 270.19
	GPT-5	Direct + Parsed	46	6	48	$138.30 ± 25.53
	GPT-5	Parsed + Parsed	31	5	64	$180.23 ± 23.42

Table 2: **Results of different methods without gold statutes.** Models in the same family have the same base model (or seem most likely to, in the case of closed-weights models). Note that "break-even price" measures only the costs of failures and abstentions, and does not include inference costs. For each model, the approach that delivered the lowest break-even price is shown in bold. The top two rows show the break-even price of trivial systems, which always defer to an expert or which always tell a person not to pay any taxes. Errors represent a symmetricized 90% confidence interval.

To identify the most salient exemplar cases, we apply information retrieval systems which can follow instructions (Weller et al. 2025a), directing a retrieval system to rank cases based on how similar the logical structure of the case's text is to the case at hand. For each case, we instruct a lightweight reasoning model (OpenAI o4-mini) to rank the other 99 cases, following recent work showing the effectiveness of test-time scaling for reranking documents (Weller et al. 2025b; Yang et al. 2025). As few-shot examples, we provide the 5 most relevant cases and their gold Prolog translations in-context for the language model to condition its parse on. These cases can be analogized to precedents, because our tax calculation agent uses them to understand how terms from the statutes are applied in practice. We note that this use of the term 'precedent' is informal and does not refer to the common-law practice of binding precedent. Rather, these retrieved precedent cases are more analogous to 'persuasive' precedent, which might help a court understand how terms have been applied in the past but is not itself new law which future courts must apply (Kozel 2014).

Experimental Setup

We run experiments across four model families of different sizes, three of which are open-weight models. The bases for these models are: Qwen2.5 32B (Qwen et al. 2025), Llama 3.3 70B (Grattafiori et al. 2024), DeepSeek-V3 671B (DeepSeek-AI et al. 2025b), and OpenAI's GPT-4.1 (OpenAI 2025a,b), each of which has an instruction-tuned version designed for common chat applications, and a reasoning version optimized to expend additional inference-time compute to solve harder problems. The full list of models is included in table 1. We run auxiliary experiments using GPT-5, but we do not conduct the same chat vs. reasoning comparison because this product appears to be an integrated system containing several models, and therefore is not as analogous to these other comparisons (Zhang et al. 2025). The reasoning model for three of these pairings stem from the DeepSeek R1 project (DeepSeek-AI et al. 2025a), where strong base models were fine-tuned to generate long chains-of-thought that help them solve harder quantitative reasoning problems. Although there is no formal documentation stating that OpenAI's GPT-4.1 and o3 models are derived from the same base model, the proximity of the two model's launch dates and their identical per-token pricing schemes suggest that these two models have a similar relationship to the other model pairs we explore.

We consider 'correct' attempts to be those where the output calculated by our system is exactly the same as the actual tax obligation, when rounded to the nearest dollar. All Prolog code is executed using the SWI-Prolog (Wielemaker et al. 2010) implementation of Prolog, and externally halted after 10 seconds of reasoning. We run experiments on the v2 release of the SARA dataset, because its programmatic representations most closely match the natural language surface forms (Holzenberger and Van Durme 2021).

Self-Consistency Tests

We further ask how effectively these methods can serve to improve each other's selectivity, by using comparisons between different solution methods to expend additional compute to help determine whether an answer should be trusted (Wang et al. 2023; Stengel-Eskin and Van Durme 2023; Jurayj, Cheng, and Van Durme 2025). In these settings, an answer is only accepted if it is reached via two independent reasoning processes: two chains-of-thought and answers (either directly calculated obligations or Prolog programs) are sampled from the same model. When self-checking using the same method (for instance, "Parsed + Parsed"), these answers are conditioned on the same prompt and context. In the parsing-based approaches, this can be considered a more stringent version of the existing refusal system which rejects questions for which the parsed solution does not execute, by additionally deferring to a tax professional where a combination of attempts do not reach a consensus answer. We test the effectiveness of each combination of reasoning processes for each model to show where additional selectivity can further improve performance.

Incorporating Costs of Incorrect Judgments

To file a faulty tax return can incur substantial financial costs. These can take the form of government-imposed penalties for understatement, or simply the cost of paying more in tax than one actually owes. Previous evaluations of SARA typically report exact match scores on the tax cases. Recent work has examined how large the errors are between the obligations calculated by a language model, observing

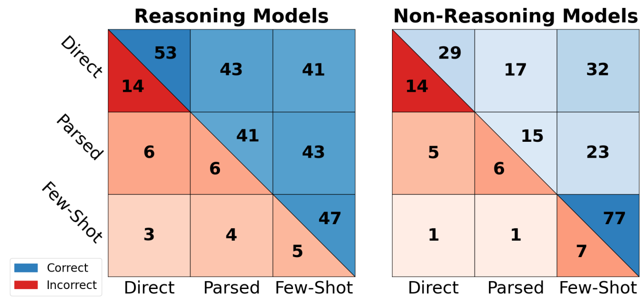

Figure 4: **Success and failure rates of method mixtures:** The top right corner counts the average number of successes yielded by each method combination, and the bottom left corner counts the average number of failures for models over 100 billion parameters optimized for reasoning (DeepSeek R1 and OpenAI o3) and chat (DeepSeek V3 and GPT-4.1)

that smaller errors are more frequent than larger ones (Blair-Stanek, Holzenberger, and Van Durme 2024). We extend this approach by calculating the costs that would be incurred by using these systems to file the taxes for the 100 tax cases in the SARA dataset.

We posit that the deployment of an automated tax advisor system should require some accountability for the organization deploying it. To provide a realistic estimate of the costs of employing the methods we outline above draw from the US Internal Revenue Code (IRC) §6662, which imposes a penalty of 20% of the amount underpaid for "substantial understatement of income tax". The cases in SARA are all personal or household tax cases, so a tax filing is considered a "substantial understatement" if they are less than the maximum of 10% of the actual amount owed or $5,000. We finally impose a penalty of $270 for refusing to answer, to model the cost of the personal or professional time required to complete one's taxes (Internal Revenue Service 2025).

This gives us the following formula: Let N be the number of cases the system ingests, and $\Delta y_i = y_i - \hat{y}_i$ be the difference between the actual obligation y_i and predicted obligation $\hat{y}_i$, such that positive values of Δy_i indicate understatement and negative values indicate overstatement.

$$\text{cost} = \frac{1}{N}\sum_{i=1}^{N}\begin{cases} -(\Delta y_i), & \Delta y_i < 0, \\ 0.2\,(\Delta y_i), & \Delta y_i > \max\bigl(\$5{,}000,\ 0.1 \cdot y_i\bigr), \\ \$270, & \text{if refused}, \\ \$0, & \text{otherwise}. \end{cases}$$

The first line corresponds to the cost of overstatement (i.e. the amount overstated), the second and fourth lines correspond to the respective fees incurred for substantial and non-substantial understatements, while the third line simulates the average cost for an American to file their taxes according to the IRS (Internal Revenue Service 2025).

This cost also corresponds to the **break-even price** of the tax assistant, i.e. the minimum price at which they might offer this service without becoming insolvent. This simulates a real-world scenario in which an organization assumes liability for the costs of the errors their system makes for their users, and offers tax filing with deferral to a tax expert at a fixed price point. Here, the break-even price would inform the minimum price at which they might offer this service without becoming insolvent, such that a more accurate system delivers a lower break-even price. LLM inference costs remain under $1 per-question, and are omitted because of their marginal impact.

We note that the penalty discussed can also be imposed for "negligence or disregard of rules or regulations" while filing taxes. One interpretation might be that using an AI system to complete one's taxes is inherently negligent, but for the purposes of this work we assume this is not the case.

Results

We display the effectiveness of each method which doesn't use the gold statutes in table 2. To extend these, we show the effectiveness of methods which access gold statutes and exemplars in table 3, although we note that these are not directly comparable to the results in table 2 because they require more up-front human effort.

We observe a substantial divergence between the effectiveness of reasoning- and chat-optimized models on different variants of this task. We visualize this in fig. 3, aggregating performance of the more powerful DeepSeek and OpenAI models optimized for Chat and Reasoning. Figure 4 shows how this disparity is further amplified when an answer must be reached via two independent reasoning paths in order to be accepted; whereas reasoning models can effectively mix methods to check their work, chat models deliver exceptional performance when self-checking few-shot solutions, but weaker performance otherwise.

Discussion

Our results indicate the promise of augmenting large language models with a symbolic solver. In both settings with and without access to gold symbolic representations of the statutes, the most effective method combined the strongest model (i.e. the OpenAI offering) with the Prolog symbolic solver. In addition to this performance advantage, the use of the symbolic solver is desirable for this specific task because it means that taxpayers or auditors may inspect and debug the system's reasoning process after the fact, with a guarantee that the decision the system reached was achieved by the path which the logic program articulates. Although one might attempt to audit the chains of thought used to solve the problem in the direct solution cases, the causal relationship between these chains and the answer they help produce is less robust than symbolic program execution, and can be deeply misleading to human readers (Paul et al. 2024; Barez et al. 2025; Li et al. 2025; Skaf et al. 2025).

Interestingly, our experiments aggregated in fig. 3 reveal a notable divergence between models optimized for reasoning and for chat applications. Although reasoning models perform better at direct solving and zero-shot parsing into symbolic representations of rules and facts, the chat models exhibit surprising effectiveness at few-shot parsing of

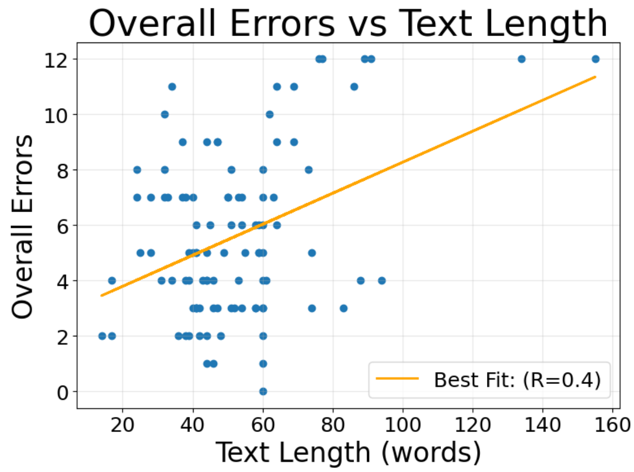

Figure 5: **Cases with more words are more likely to be miscalculated.** We note a moderate correlation between the case length and the failure rate of solution attempts.

case facts. It is possible that reasoning models' post-training focuses their efforts on emulating explicit reasoning steps, such as those Prolog would itself execute, rather than accurately mapping input texts into symbolic form. Whereas long chains of thought are useful for the complicated arithmetic required for directly calculating a tax obligation or for discerning which portions of the statutes should be included in a zero-shot parse, they degrade performance on the relatively simpler task of imitating exemplary conversions from natural language into formal logic. This aligns with concurrent work highlighting cases in which long chains of thought can distract models from tasks where shorter chains would deliver stronger performance (Gema et al. 2025).

In practice, this additionally means that with gold exemplars of case and statute translations, an individual's tax case could be processed much more quickly than it could be without these manual annotations, because chat models take dramatically less time than reasoning models to produce answers. This could help build more interactive tax assistance tools which allow users to quickly iterate to amend and clarify their relevant tax information. This approach notably reduces the expected real-world cost per successful tax filing, highlighting the critical role that intelligent exemplar selection can play in guiding model outputs toward faithful and structured reasoning. The most effective configuration at minimizing the system's break-even price provides models with intelligently retrieved precedential examples, and deferring to a tax expert unless this system yields the same answer from two independently sampled solutions. The break-even price of $15.78 for this system would save the average American taxpayer over 90% of the amount spent on tax filing (Internal Revenue Service 2025). This strong performance could be a first step toward real-world pilot studies.

Of course, to expect that statutory codes would be manually translated into a logic programming language like Prolog is a constraining assumption to deliver this strong performance. However, we note that a handful of existing orga-

Family	Model	Method	Correct	Incorrect	Abstentions	Break-Even Price
Previous Best	o3	Parsed	75	15	10	$47.43 ± 22.16
Qwen-2.5	Qwen-32b	Few-Shot	42	38	20	$4,676.49 ± 4,623.77
	R1-32b	Few-Shot	47	33	20	$7,783.29 ± 6,244.96
Llama-3.3	Llama-70b	Few-Shot	70	27	3	$1,917.32 ± 1,247.50
	R1-70b	Few-Shot	29	57	14	$6,328.48 ± 2,545.70
DeepSeek	DeepSeek-V3	Few-Shot	78	18	4	$468.66 ± 273.47
	DeepSeek-V3	Direct + Few-Shot	18	2	80	$223.43 ± 19.69
	DeepSeek-V3	Parsed + Few-Shot	9	1	90	$250.43 ± 15.31
	DeepSeek-V3	Few-Shot + Few-Shot	73	9	18	$271.45 ± 230.09
	DeepSeek-R1	Few-Shot	40	16	44	$378.73 ± 155.14
	DeepSeek-R1	Direct + Few-Shot	32	2	66	$178.20 ± 21.34
	DeepSeek-R1	Parsed + Few-Shot	19	2	79	$234.65 ± 28.76
	DeepSeek-R1	Few-Shot + Few-Shot	20	3	77	$215.33 ± 20.63
OpenAI GPT-4.1	GPT-4.1	Few-Shot	87	8	5	$247.99 ± 341.76
	GPT-4.1	Direct + Few-Shot	47	0	53	$143.10 ± 22.49
	GPT-4.1	Parsed + Few-Shot	38	1	61	$164.70 ± 21.98
	GPT-4.1	Few-Shot + Few-Shot	81	5	14	$40.08 ± 15.87
	o3	Few-Shot	81	13	6	$60.26 ± 58.93
	o3	Direct + Few-Shot	51	5	44	$126.41 ± 24.54
	o3	Parsed + Few-Shot	68	7	25	$75.11 ± 22.46
	o3	Few-Shot + Few-Shot	74	8	18	$58.13 ± 20.91
OpenAI GPT-5	GPT-5	Few-Shot	86	9	5	$15.78 ± 10.34
	GPT-5	Direct + Few-Shot	71	5	24	$64.98 ± 19.23
	GPT-5	Parsed + Few-Shot	50	2	48	$129.60 ± 22.51
	GPT-5	Few-Shot + Few-Shot	84	6	10	$29.28 ± 13.84

Table 3: **Results of different methods with access to gold statutes and intelligently retrieved parsing exemplars.** Columns have the same meaning as table 2. Note that because these results require the additional work of manually translating all statutes and a set of representative cases, they are not directly comparable to those in table 2. The top row shows the best previous result.

nizations have embarked on this task, such as projects to encode the French tax code (Merigoux, Monat, and Protzenko 2021) or Canadian policy proposals (Morris 2020) into logic programs. In the United States, private companies build similar logic programs into the backend of consumer tax arrangement software (Yu, McCluskey, and Mukherjee 2020). As this practice increases in popularity, it will expand opportunities to implement methods which use gold-standard programmatic representations of rules. Furthermore, we believe the relative strength of o3 at zero-shot parsing (which includes parsing a subset of the statutory rules) suggests promise in semi-automated encoding of these rules; as models improve at this type of complex semantic parsing, entire legal systems could be distilled into programmatic logic.

We look closely at the cases which are most and least commonly solved correctly by the most powerful models (DeepSeek and OpenAI). We see in fig. 5 that cases which have more words in them are more likely to be mistaken across models, perhaps because they contain more intricate logical structures that are harder to parse. Real-world deployment should consider how other metrics for case difficulty can inform the decision to defer to a tax professional.

Finally, we note in table 2 a dramatic jump in performance with scale at parsing cases and statutes without gold translations. Although smaller reasoning models (those derived from Qwen2.5 32B and Llama 3.3 70B) perform solidly at direct solving, and the smaller chat models perform solidly at few-shot parsing, neither category of smaller model solves more than a handful of cases in the zero-shot parsing setting. In contrast, all larger models solve at least 10% of these cases correctly, while for OpenAI o3 this is the most successful setting. We hope that further increases in model scale will further improve performance in this setting, since it combines the low human cost of direct solving with the easy auditing of systems that rely on the symbolic solver.

Conclusion

We frame statutory reasoning as a semantic parsing task, instructing language models to convert statutory and case text into executable code. We show how the integration of symbolic solvers with frontier language models can enhance the capability of AI tax assistance systems, and highlight their potential to improve access to accurate and affordable tax guidance. Although tradeoffs remain between upfront costs of translating rules into formal logic versus ongoing inference-time computational and error costs, leveraging symbolic reasoning reduces overall expenses while improving auditability in both cases. Future research will explore how further scale or specialized smaller models optimized for faithful translation can improve aspects of this task, and develop methods for efficient translation of statutory rules into formal logic to enable effective few-shot learning.

Ethical Statement

Although we believe our results are promising, this paper should not be understood as a recommendation for individuals filing their taxes. Instead, we hope that policymakers and software developers will extend these insights to reduce the individual effort required for tax filing, building systems which can allow a single human-in-the-loop to complete filings more efficiently. Such steps would require careful consideration of liability issues, impacts on vulnerable populations, and non-financial costs associated with tax reviews such as stress or time. Furthermore, the reliance on closed-source models may place private companies at a choke point, so further advances in the open-source domain are essential.

Acknowledgments

This work was funded in part by the U.S. National Science Foundation under grant No. 2204926, and by the Defense Advanced Research Project Agency (DARPA) CODORD program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or DARPA.

References

Barez, F.; Wu, T.-Y.; Arcuschin, I.; Lan, M.; Wang, V.; Siegel, N.; Collignon, N.; Neo, C.; Lee, I.; Paren, A.; Bibi, A.; Trager, R.; Fornasiere, D.; Yan, J.; Elazar, Y.; and Bengio, Y. 2025. Chain-of-Thought Is Not Explainability. *ArXiv*.

Black, E.; Elzayn, H.; Chouldechova, A.; Goldin, J.; and Ho, D. 2022. Algorithmic Fairness and Vertical Equity: Income Fairness with IRS Tax Audit Models. In *2022 ACM Conference on Fairness Accountability and Transparency*, 1479–1503. Seoul Republic of Korea: ACM.

Blair-Stanek, A.; Holzenberger, N.; and Van Durme, B. 2024. OpenAI Cribbed Our Tax Example, But Can GPT-4 Really Do Tax? ArXiv:2309.09992 [cs].

Boning, W. C.; Hendren, N.; Sprung-Keyser, B.; and Stuart, E. 2024. A Welfare Analysis of Tax Audits Across the Income Distribution*. *The Quarterly Journal of Economics*, 140(1): 63–112.

Ceri, S.; Gottlob, G.; and Tanca, L. 1989. What you always wanted to know about Datalog (and never dared to ask). *IEEE Transactions on Knowledge and Data Engineering*, 1(1): 146–166.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. ArXiv:2110.14168 [cs].

Davidson, D. 1966. The Logical Form of Action Sentences. In Davidson, D., ed., *Essays on Actions and Events*, 0. Oxford University Press. ISBN 978-0-19-924627-4.

Dean, S. A. 2022. Filing While Black: The Casual Racism of the Tax Law. *Utah Law Review*, 2022(4): 801–812.

DeepSeek-AI; Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; Zhang, X.; Yu, X.; Wu, Y.; Wu, Z. F.; Gou, Z.; Shao, Z.; Li, Z.; Gao, Z.; Liu, A.; Xue, B.; Wang, B.; Wu, B.; Feng, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; Dai, D.; Chen, D.; Ji, D.; Li, E.; Lin, F.; Dai, F.; Luo, F.; Hao, G.; Chen, G.; Li, G.; Zhang, H.; Bao, H.; Xu, H.; Wang, H.; Ding, H.; Xin, H.; Gao, H.; Qu, H.; Li, H.; Guo, J.; Li, J.; Wang, J.; Chen, J.; Yuan, J.; Qiu, J.; Li, J.; Cai, J. L.; Ni, J.; Liang, J.; Chen, J.; Dong, K.; Hu, K.; Gao, K.; Guan, K.; Huang, K.; Yu, K.; Wang, L.; Zhang, L.; Zhao, L.; Wang, L.; Zhang, L.; Xu, L.; Xia, L.; Zhang, M.; Zhang, M.; Tang, M.; Li, M.; Wang, M.; Li, M.; Tian, N.; Huang, P.; Zhang, P.; Wang, Q.; Chen, Q.; Du, Q.; Ge, R.; Zhang, R.; Pan, R.; Wang, R.; Chen, R. J.; Jin, R. L.; Chen, R.; Lu, S.; Zhou, S.; Chen, S.; Ye, S.; Wang, S.; Yu, S.; Zhou, S.; Pan, S.; Li, S. S.; Zhou, S.; Wu, S.; Ye, S.; Yun, T.; Pei, T.; Sun, T.; Wang, T.; Zeng, W.; Zhao, W.; Liu, W.; Liang, W.; Gao, W.; Yu, W.; Zhang, W.; Xiao, W. L.; An, W.; Liu, X.; Wang, X.; Chen, X.; Nie, X.; Cheng, X.; Liu, X.; Xie, X.; Liu, X.; Yang, X.; Li, X.; Su, X.; Lin, X.; Li, X. Q.; Jin, X.; Shen, X.; Chen, X.; Sun, X.; Wang, X.; Song, X.; Zhou, X.; Wang, X.; Shan, X.; Li, Y. K.; Wang, Y. Q.; Wei, Y. X.; Zhang, Y.; Xu, Y.; Li, Y.; Zhao, Y.; Sun, Y.; Wang, Y.; Yu, Y.; Zhang, Y.; Shi, Y.; Xiong, Y.; He, Y.; Piao, Y.; Wang, Y.; Tan, Y.; Ma, Y.; Liu, Y.; Guo, Y.; Ou, Y.; Wang, Y.; Gong, Y.; Zou, Y.; He, Y.; Xiong, Y.; Luo, Y.; You, Y.; Liu, Y.; Zhou, Y.; Zhu, Y. X.; Xu, Y.; Huang, Y.; Li, Y.; Zheng, Y.; Zhu, Y.; Ma, Y.; Tang, Y.; Zha, Y.; Yan, Y.; Ren, Z. Z.; Ren, Z.; Sha, Z.; Fu, Z.; Xu, Z.; Xie, Z.; Zhang, Z.; Hao, Z.; Ma, Z.; Yan, Z.; Wu, Z.; Gu, Z.; Zhu, Z.; Liu, Z.; Li, Z.; Xie, Z.; Song, Z.; Pan, Z.; Huang, Z.; Xu, Z.; Zhang, Z.; and Zhang, Z. 2025a. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. ArXiv:2501.12948 [cs].

DeepSeek-AI; Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; Dai, D.; Guo, D.; Yang, D.; Chen, D.; Ji, D.; Li, E.; Lin, F.; Dai, F.; Luo, F.; Hao, G.; Chen, G.; Li, G.; Zhang, H.; Bao, H.; Xu, H.; Wang, H.; Zhang, H.; Ding, H.; Xin, H.; Gao, H.; Li, H.; Qu, H.; Cai, J. L.; Liang, J.; Guo, J.; Ni, J.; Li, J.; Wang, J.; Chen, J.; Chen, J.; Yuan, J.; Qiu, J.; Li, J.; Song, J.; Dong, K.; Hu, K.; Gao, K.; Guan, K.; Huang, K.; Yu, K.; Wang, L.; Zhang, L.; Xu, L.; Xia, L.; Zhao, L.; Wang, L.; Zhang, L.; Li, M.; Wang, M.; Zhang, M.; Zhang, M.; Tang, M.; Li, M.; Tian, N.; Huang, P.; Wang, P.; Zhang, P.; Wang, Q.; Zhu, Q.; Chen, Q.; Du, Q.; Chen, R. J.; Jin, R. L.; Ge, R.; Zhang, R.; Pan, R.; Wang, R.; Xu, R.; Zhang, R.; Chen, R.; Li, S. S.; Lu, S.; Zhou, S.; Chen, S.; Wu, S.; Ye, S.; Ye, S.; Ma, S.; Wang, S.; Zhou, S.; Yu, S.; Zhou, S.; Pan, S.; Wang, T.; Yun, T.; Pei, T.; Sun, T.; Xiao, W. L.; Zeng, W.; Zhao, W.; An, W.; Liu, W.; Liang, W.; Gao, W.; Yu, W.; Zhang, W.; Li, X. Q.; Jin, X.; Wang, X.; Bi, X.; Liu, X.; Wang, X.; Shen, X.; Chen, X.; Zhang, X.; Chen, X.; Nie, X.; Sun, X.; Wang, X.; Cheng, X.; Liu, X.; Xie, X.; Liu, X.; Yu, X.; Song, X.; Shan, X.; Zhou, X.; Yang, X.; Li, X.; Su, X.; Lin, X.; Li, Y. K.; Wang, Y. Q.; Wei, Y. X.; Zhu, Y. X.; Zhang, Y.; Xu, Y.; Xu, Y.; Huang, Y.; Li, Y.; Zhao, Y.; Sun, Y.; Li, Y.; Wang, Y.; Yu, Y.; Zheng, Y.; Zhang, Y.; Shi, Y.; Xiong, Y.; He, Y.; Tang, Y.; Piao, Y.; Wang, Y.; Tan, Y.; Ma, Y.; Liu, Y.; Guo, Y.; Wu, Y.; Ou, Y.; Zhu, Y.; Wang, Y.; Gong, Y.; Zou, Y.; He, Y.; Zha, Y.; Xiong, Y.; Ma, Y.; Yan, Y.; Luo, Y.; You, Y.; Liu, Y.; Zhou, Y.; Wu,

Z. F.; Ren, Z. Z.; Ren, Z.; Sha, Z.; Fu, Z.; Xu, Z.; Huang, Z.; Zhang, Z.; Xie, Z.; Zhang, Z.; Hao, Z.; Gou, Z.; Ma, Z.; Yan, Z.; Shao, Z.; Xu, Z.; Wu, Z.; Zhang, Z.; Li, Z.; Gu, Z.; Zhu, Z.; Liu, Z.; Li, Z.; Xie, Z.; Song, Z.; Gao, Z.; and Pan, Z. 2025b. DeepSeek-V3 Technical Report. ArXiv:2412.19437 [cs].

Elzayn, H.; Smith, E.; Hertz, T.; Guage, C.; Ramesh, A.; Fisher, R.; Ho, D. E.; and Goldin, J. 2025. Measuring and Mitigating Racial Disparities in Tax Audits*. *The Quarterly Journal of Economics*, 140(1): 113–163.

Gantt, W.; Martin, A.; Kuchmiichuk, P.; and White, A. S. 2024. Event-Keyed Summarization. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Findings of the Association for Computational Linguistics: EMNLP 2024*, 7333–7345. Miami, Florida, USA: Association for Computational Linguistics.

Gelfond, M.; and Lifschitz, V. 1988. The Stable Model Semantics for Logic Programming. In Kowalski, R.; Bowen; and Kenneth, eds., *Proceedings of International Logic Programming Conference and Symposium*, 1070–1080. MIT Press.

Gema, A. P.; Hägele, A.; Chen, R.; Arditi, A.; Goldman-Wetzler, J.; Fraser-Taliente, K.; Sleight, H.; Petrini, L.; Michael, J.; Alex, B.; Minervini, P.; Chen, Y.; Benton, J.; and Perez, E. 2025. Inverse Scaling in Test-Time Compute. arXiv:2507.14417.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; Yang, A.; Fan, A.; Goyal, A.; Hartshorn, A.; Yang, A.; Mitra, A.; Sravankumar, A.; Korenev, A.; Hinsvark, A.; Rao, A.; Zhang, A.; Rodriguez, A.; Gregerson, A.; Spataru, A.; Roziere, B.; Biron, B.; Tang, B.; Chern, B.; Caucheteux, C.; Nayak, C.; Bi, C.; Marra, C.; McConnell, C.; Keller, C.; Touret, C.; Wu, C.; Wong, C.; Ferrer, C. C.; Nikolaidis, C.; Allonsius, D.; Song, D.; Pintz, D.; Livshits, D.; Wyatt, D.; Esiobu, D.; Choudhary, D.; Mahajan, D.; Garcia-Olano, D.; Perino, D.; Hupkes, D.; Lakomkin, E.; AlBadawy, E.; Lobanova, E.; Dinan, E.; Smith, E. M.; Radenovic, F.; Guzmán, F.; Zhang, F.; Synnaeve, G.; Lee, G.; Anderson, G. L.; Thattai, G.; Nail, G.; Mialon, G.; Pang, G.; Cucurell, G.; Nguyen, H.; Korevaar, H.; Xu, H.; Touvron, H.; Zarov, I.; Ibarra, I. A.; Kloumann, I.; Misra, I.; Evtimov, I.; Zhang, J.; Copet, J.; Lee, J.; Geffert, J.; Vranes, J.; Park, J.; Mahadeokar, J.; Shah, J.; Linde, J. v. d.; Billock, J.; Hong, J.; Lee, J.; Fu, J.; Chi, J.; Huang, J.; Liu, J.; Wang, J.; Yu, J.; Bitton, J.; Spisak, J.; Park, J.; Rocca, J.; Johnstun, J.; Saxe, J.; Jia, J.; Alwala, K. V.; Prasad, K.; Upasani, K.; Plawiak, K.; Li, K.; Heafield, K.; Stone, K.; El-Arini, K.; Iyer, K.; Malik, K.; Chiu, K.; Bhalla, K.; Lakhotia, K.; Rantala-Yeary, L.; Maaten, L. v. d.; Chen, L.; Tan, L.; Jenkins, L.; Martin, L.; Madaan, L.; Malo, L.; Blecher, L.; Landzaat, L.; Oliveira, L. d.; Muzzi, M.; Pasupuleti, M.; Singh, M.; Paluri, M.; Kardas, M.; Tsimpoukelli, M.; Oldham, M.; Rita, M.; Pavlova, M.; Kambadur, M.; Lewis, M.; Si, M.; Singh, M. K.; Hassan, M.; Goyal, N.; Torabi, N.; Bashlykov, N.; Bogoychev, N.; Chatterji, N.; Zhang, N.; Duchenne, O.; Çelebi, O.; Alrassy, P.; Zhang, P.; Li, P.; Vasic, P.; Weng, P.; Bhargava, P.; Dubal, P.; Krishnan, P.; Koura, P. S.; Xu, P.; He, Q.; Dong, Q.; Srinivasan, R.; Ganapathy, R.; Calderer, R.; Cabral, R. S.; Stojnic, R.; Raileanu, R.; Maheswari, R.; Girdhar, R.; Patel, R.; Sauvestre, R.; Polidoro, R.; Sumbaly, R.; Taylor, R.; Silva, R.; Hou, R.; Wang, R.; Hosseini, S.; Chennabasappa, S.; Singh, S.; Bell, S.; Kim, S. S.; Edunov, S.; Nie, S.; Narang, S.; Raparthy, S.; Shen, S.; Wan, S.; Bhosale, S.; Zhang, S.; Vandenhende, S.; Batra, S.; Whitman, S.; Sootla, S.; Collot, S.; Gururangan, S.; Borodinsky, S.; Herman, T.; Fowler, T.; Sheasha, T.; Georgiou, T.; Scialom, T.; Speckbacher, T.; Mihaylov, T.; Xiao, T.; Karn, U.; Goswami, V.; Gupta, V.; Ramanathan, V.; Kerkez, V.; Gonguet, V.; Do, V.; Vogeti, V.; Albiero, V.; Petrovic, V.; Chu, W.; Xiong, W.; Fu, W.; Meers, W.; Martinet, X.; Wang, X.; Wang, X.; Tan, X. E.; Xia, X.; Xie, X.; Jia, X.; Wang, X.; Goldschlag, Y.; Gaur, Y.; Babaei, Y.; Wen, Y.; Song, Y.; Zhang, Y.; Li, Y.; Mao, Y.; Coudert, Z. D.; Yan, Z.; Chen, Z.; Papakipos, Z.; Singh, A.; Srivastava, A.; Jain, A.; Kelsey, A.; Shajnfeld, A.; Gangidi, A.; Victoria, A.; Goldstand, A.; Menon, A.; Sharma, A.; Boesenberg, A.; Baevski, A.; Feinstein, A.; Kallet, A.; Sangani, A.; Teo, A.; Yunus, A.; Lupu, A.; Alvarado, A.; Caples, A.; Gu, A.; Ho, A.; Poulton, A.; Ryan, A.; Ramchandani, A.; Dong, A.; Franco, A.; Goyal, A.; Saraf, A.; Chowdhury, A.; Gabriel, A.; Bharambe, A.; Eisenman, A.; Yazdan, A.; James, B.; Maurer, B.; Leonhardi, B.; Huang, B.; Loyd, B.; Paola, B. D.; Paranjape, B.; Liu, B.; Wu, B.; Ni, B.; Hancock, B.; Wasti, B.; Spence, B.; Stojkovic, B.; Gamido, B.; Montalvo, B.; Parker, C.; Burton, C.; Mejia, C.; Liu, C.; Wang, C.; Kim, C.; Zhou, C.; Hu, C.; Chu, C.-H.; Cai, C.; Tindal, C.; Feichtenhofer, C.; Gao, C.; Civin, D.; Beaty, D.; Kreymer, D.; Li, D.; Adkins, D.; Xu, D.; Testuggine, D.; David, D.; Parikh, D.; Liskovich, D.; Foss, D.; Wang, D.; Le, D.; Holland, D.; Dowling, E.; Jamil, E.; Montgomery, E.; Presani, E.; Hahn, E.; Wood, E.; Le, E.-T.; Brinkman, E.; Arcaute, E.; Dunbar, E.; Smothers, E.; Sun, F.; Kreuk, F.; Tian, F.; Kokkinos, F.; Ozgenel, F.; Caggioni, F.; Kanayet, F.; Seide, F.; Florez, G. M.; Schwarz, G.; Badeer, G.; Swee, G.; Halpern, G.; Herman, G.; Sizov, G.; Guangyi; Zhang; Lakshminarayanan, G.; Inan, H.; Shojanazeri, H.; Zou, H.; Wang, H.; Zha, H.; Habeeb, H.; Rudolph, H.; Suk, H.; Aspegren, H.; Goldman, H.; Zhan, H.; Damlaj, I.; Molybog, I.; Tufanov, I.; Leontiadis, I.; Veliche, I.-E.; Gat, I.; Weissman, J.; Geboski, J.; Kohli, J.; Lam, J.; Asher, J.; Gaya, J.-B.; Marcus, J.; Tang, J.; Chan, J.; Zhen, J.; Reizenstein, J.; Teboul, J.; Zhong, J.; Jin, J.; Yang, J.; Cummings, J.; Carvill, J.; Shepard, J.; McPhie, J.; Torres, J.; Ginsburg, J.; Wang, J.; Wu, K.; U, K. H.; Saxena, K.; Khandelwal, K.; Zand, K.; Matosich, K.; Veeraraghavan, K.; Michelena, K.; Li, K.; Jagadeesh, K.; Huang, K.; Chawla, K.; Huang, K.; Chen, L.; Garg, L.; A, L.; Silva, L.; Bell, L.; Zhang, L.; Guo, L.; Yu, L.; Moshkovich, L.; Wehrstedt, L.; Khabsa, M.; Avalani, M.; Bhatt, M.; Mankus, M.; Hasson, M.; Lennie, M.; Reso, M.; Groshev, M.; Naumov, M.; Lathi, M.; Keneally, M.; Liu, M.; Seltzer, M. L.; Valko, M.; Restrepo, M.; Patel, M.; Vyatskov, M.; Samvelyan, M.; Clark, M.; Macey, M.; Wang, M.; Hermoso, M. J.; Metanat, M.; Rastegari, M.; Bansal, M.; Santhanam, N.; Parks, N.; White, N.; Bawa, N.; Singhal, N.; Egebo, N.; Usunier, N.; Mehta, N.; Laptev, N. P.; Dong, N.; Cheng, N.; Chernoguz, O.; Hart, O.; Salpekar, O.; Kalinli, O.; Kent, P.; Parekh, P.; Saab, P.; Balaji, P.; Rittner, P.; Bontrager, P.; Roux, P.; Dollar, P.; Zvyagina, P.; Ratanchandani, P.; Yuvraj, P.; Liang, Q.; Alao, R.; Rodriguez, R.; Ayub, R.; Murthy, R.; Nayani, R.; Mitra, R.; Parthasarathy, R.; Li, R.; Hogan, R.; Battey, R.; Wang, R.; Howes, R.; Rinott, R.; Mehta, S.; Siby, S.; Bondu, S. J.; Datta, S.; Chugh, S.; Hunt, S.; Dhillon, S.; Sidorov, S.; Pan, S.; Mahajan, S.; Verma, S.; Yamamoto, S.; Ramaswamy, S.; Lindsay, S.; Lindsay, S.; Feng, S.; Lin, S.; Zha, S. C.; Patil, S.; Shankar, S.; Zhang, S.; Zhang, S.; Wang, S.; Agarwal, S.; Sajuyigbe, S.; Chintala, S.; Max, S.; Chen, S.; Kehoe, S.; Satterfield, S.; Govindaprasad, S.; Gupta, S.; Deng, S.; Cho, S.; Virk, S.; Subramanian, S.; Choudhury, S.; Goldman, S.; Remez, T.; Glaser, T.; Best, T.; Koehler, T.; Robinson, T.; Li, T.; Zhang, T.; Matthews, T.; Chou, T.; Shaked, T.; Vontimitta, V.; Ajayi, V.; Montanez, V.; Mohan, V.; Kumar, V. S.; Mangla, V.; Ionescu, V.; Poenaru, V.; Mihailescu, V. T.; Ivanov, V.; Li, W.; Wang, W.; Jiang, W.; Bouaziz, W.; Constable, W.; Tang, X.; Wu, X.; Wang, X.; Wu, X.; Gao, X.; Kleinman, Y.; Chen, Y.; Hu, Y.; Jia, Y.; Qi, Y.; Li, Y.; Zhang, Y.; Zhang, Y.; Adi, Y.; Nam, Y.; Yu; Wang; Zhao, Y.; Hao, Y.; Qian, Y.; Li, Y.; He, Y.; Rait, Z.; DeVito, Z.; Rosnbrick, Z.; Wen, Z.; Yang, Z.; Zhao, Z.; and Ma, Z. 2024. The Llama 3 Herd of Models. ArXiv:2407.21783 [cs].

Guha, N.; Nyarko, J.; Ho, D. E.; Re, C.; Chilton, A.; Narayana, A.; Chohlas-Wood, A.; Peters, A.; Waldon, B.; Rockmore, D.; Zambrano, D.; Talisman, D.; Hoque, E.; Surani, F.; Fagan, F.; Sarfaty, G.; Dickinson, G. M.; Porat, H.; Hegland, J.; Wu, J.; Nudell, J.; Niklaus, J.; Nay, J. J.; Choi, J. H.; Tobia, K.; Hagan, M.; Ma, M.; Livermore, M.; Rasumov-Rahe, N.; Holzenberger, N.; Kolt, N.; Henderson, P.; Rehaag, S.; Goel, S.; Gao, S.; Williams, S.; Gandhi, S.; Zur, T.; Iyer, V.; and Li, Z. 2023. LegalBench: A Collaboratively Built Benchmark for Measuring Legal Reasoning in Large Language Models. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Holzenberger, N.; Blair-Stanek, A.; and Van Durme, B. 2020. A Dataset for Statutory Reasoning in Tax Law Entailment and Question Answering. *San Diego*.

Holzenberger, N.; and Van Durme, B. 2021. Factoring Statutory Reasoning as Language Understanding Challenges. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2742–2758. Online: Association for Computational Linguistics.

Holzenberger, N.; and Van Durme, B. 2023. Connecting Symbolic Statutory Reasoning with Legal Information Extraction. In Preoțiuc-Pietro, D.; Goanta, C.; Chalkidis, I.; Barrett, L.; Spanakis, G.; and Aletras, N., eds., *Proceedings of the Natural Legal Language Processing Workshop 2023*, 113–131. Singapore: Association for Computational Linguistics.

Horn, A. 1951. On Sentences Which are True of Direct Unions of Algebras. *The Journal of Symbolic Logic*, 16(1): 14–21. Publisher: Association for Symbolic Logic.

Hou, A.; Jurayj, W.; Holzenberger, N.; Blair-Stanek, A.; and Van Durme, B. 2024. Gaps or Hallucinations? Scrutinizing Machine-Generated Legal Analysis for Fine-grained Text Evaluations. In Aletras, N.; Chalkidis, I.; Barrett, L.; Goanță, C.; Preoțiuc-Pietro, D.; and Spanakis, G., eds., *Proceedings of the Natural Legal Language Processing Workshop 2024*, 280–302. Miami, FL, USA: Association for Computational Linguistics.

Internal Revenue Service. 2025. U.S. Department of the Treasury, IRS Announce Direct File as Permanent Free Tax Filing Option, All 50 States and D.C. Invited to Join in Filing Season 2025.

Jurayj, W.; Cheng, J.; and Van Durme, B. 2025. Is That Your Final Answer? Test-Time Scaling Improves Selective Question Answering. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 636–644. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-252-7.

Kant, M.; Nabi, S.; Kant, M.; Scharrer, R.; Ma, M.; and Nabi, M. 2025. Towards Robust Legal Reasoning: Harnessing Logical LLMs in Law. ArXiv:2502.17638 [cs] version: 1.

Kozel, R. 2014. The Scope of Precedent. *Michigan Law Review*, 113(2): 179–230.

Li, D.; Cao, S.; Griggs, T.; Liu, S.; Mo, X.; Tang, E.; Hegde, S.; Hakhamaneshi, K.; Patil, S. G.; Zaharia, M.; Gonzalez, J. E.; and Stoica, I. 2025. LLMs Can Easily Learn to Reason from Demonstrations Structure, not content, is what matters! ArXiv:2502.07374 [cs].

Merigoux, D. 2023. Experience report: implementing a real-world, medium-sized program derived from a legislative specification. *Programming Languages and the Law*.

Merigoux, D.; Chataing, N.; and Protzenko, J. 2021. Catala: a programming language for the law. *Proc. ACM Program. Lang.*, 5(ICFP): 77:1–77:29.

Merigoux, D.; Monat, R.; and Protzenko, J. 2021. A modern compiler for the French tax code. In *Proceedings of the 30th ACM SIGPLAN International Conference on Compiler Construction*, CC 2021, 71–82. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-8325-7.

Morris, J. 2020. Blawx: Rules as Code Demonstration. *MIT Computational Law Report*. Https://law.mit.edu/pub/blawxrulesascodedemonstration.

OpenAI. 2025a. Introducing GPT-4.1 in the API.

OpenAI. 2025b. Introducing OpenAI o3 and o4-mini.

Paul, D.; West, R.; Bosselut, A.; and Faltings, B. 2024. Making Reasoning Matter: Measuring and Improving Faithfulness of Chain-of-Thought Reasoning. In Al-Onaizan, Y.;

Bansal, M.; and Chen, Y.-N., eds., *Findings of the Association for Computational Linguistics: EMNLP 2024*, 15012–15032. Miami, Florida, USA: Association for Computational Linguistics.

Qwen; Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; Lin, H.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Lin, J.; Dang, K.; Lu, K.; Bao, K.; Yang, K.; Yu, L.; Li, M.; Xue, M.; Zhang, P.; Zhu, Q.; Men, R.; Lin, R.; Li, T.; Tang, T.; Xia, T.; Ren, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Qiu, Z. 2025. Qwen2.5 Technical Report. ArXiv:2412.15115 [cs].

Sergot, M. J.; Sadri, F.; Kowalski, R. A.; Kriwaczek, F.; Hammond, P.; and Cory, H. T. 1986. The British Nationality Act as a logic program. *Communications of the ACM*, 29(5): 370–386. Publisher: Association for Computing Machinery (ACM).

Sherman, D. M. 1987. A Prolog model of the income tax act of Canada. In *Proceedings of the 1st international conference on Artificial intelligence and law*, ICAIL '87, 127–136. New York, NY, USA: Association for Computing Machinery. ISBN 978-0-89791-230-3.

Sherman, D. M. 1989. Expert systems and ICAI in tax law: killing two birds with one AI stone. In *Proceedings of the 2nd international conference on Artificial intelligence and law*, ICAIL '89, 74–80. New York, NY, USA: Association for Computing Machinery. ISBN 978-0-89791-322-5.

Shin, R.; and Van Durme, B. 2022. Few-Shot Semantic Parsing with Language Models Trained on Code. In Carpuat, M.; de Marneffe, M.-C.; and Meza Ruiz, I. V., eds., *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 5417–5425. Seattle, United States: Association for Computational Linguistics.

Skaf, J.; Ibanez-Lissen, L.; McCarthy, R.; Watts, C.; Georgiv, V.; Whittingham, H.; Gonzalez-Manzano, L.; Lindner, D.; Tice, C.; Young, E. J.; and Radmard, P. 2025. Large language models can learn and generalize steganographic chain-of-thought under process supervision. ArXiv:2506.01926 [cs].

Spiegel, B. A.; Yang, Z.; Jurayj, W.; Bachmann, B.; Tellex, S.; and Konidaris, G. 2024. Informing Reinforcement Learning Agents by Grounding Language to Markov Decision Processes. In *Workshop on Training Agents with Foundation Models at RLC 2024*.

Stengel-Eskin, E.; and Van Durme, B. 2023. Did You Mean...? Confidence-based Trade-offs in Semantic Parsing. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2621–2629. Singapore: Association for Computational Linguistics.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q. V.; Chi, E. H.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*.

Weller, O.; Durme, B. V.; Lawrie, D.; Paranjape, A.; Zhang, Y.; and Hessel, J. 2025a. Promptriever: Instruction-Trained Retrievers Can Be Prompted Like Language Models. In *The Thirteenth International Conference on Learning Representations*.

Weller, O.; Ricci, K.; Yang, E.; Yates, A.; Lawrie, D.; and Durme, B. V. 2025b. Rank1: Test-Time Compute for Reranking in Information Retrieval. In *Second Conference on Language Modeling*.

Wielemaker, J.; Schrijvers, T.; Triska, M.; and Lager, T. 2010. SWI-Prolog. ArXiv:1011.5332 [cs].

Yang, E.; Yates, A.; Ricci, K.; Weller, O.; Chari, V.; Durme, B. V.; and Lawrie, D. 2025. Rank-K: Test-Time Reasoning for Listwise Reranking. ArXiv:2505.14432 [cs] version: 1.

Yu, J.; McCluskey, K.; and Mukherjee, S. 2020. Tax Knowledge Graph for a Smarter and More Personalized TurboTax. ArXiv:2009.06103 [cs].

Zhang, Y.; Li, H.; Chen, J.; Zhang, H.; Ye, P.; Bai, L.; and Hu, S. 2025. Beyond GPT-5: Making LLMs Cheaper and Better via Performance-Efficiency Optimized Routing. ArXiv:2508.12631 [cs].