OpenExempt: A Diagnostic Benchmark for Legal Reasoning and a Framework for Creating Custom Benchmarks on Demand

Sergio Servantez[1†], **Sarah B. Lawsky**[3], **Rajiv Jain**[2],
Daniel W. Linna Jr.[1], **Kristian Hammond**[1]
[1]Northwestern University, [2]Adobe Research, [3]University of Illinois Urbana-Champaign
†Corresponding author: servantez@u.northwestern.edu

Abstract

Reasoning benchmarks have played a crucial role in the progress of language models. Yet rigorous evaluation remains a significant challenge as static question-answer pairs provide only a snapshot of performance, compressing complex behavior into a single accuracy metric. This limitation is especially true in complex, rule-bound domains such as law, where existing benchmarks are costly to build and ill suited for isolating specific failure modes. To address this, we introduce OpenExempt, a framework and benchmark for diagnostic evaluation of legal reasoning. The OpenExempt Framework uses expert-crafted symbolic representations of U.S. Bankruptcy Code statutes to dynamically generate a large space of natural language reasoning tasks and their machine-computable solutions on demand. This gives users fine-grained control over task complexity and scope, allowing individual reasoning skills to be probed in isolation. Using this system, we construct the OpenExempt Benchmark, a diagnostic benchmark for legal reasoning with 9,765 samples across nine evaluation suites designed to carefully probe model capabilities. Experiments on 13 diverse language models reveal sharp performance cliffs that emerge only under longer reasoning paths and in the presence of obfuscating statements. We release the framework and benchmark publicly to support research aimed at understanding and improving the next generation of reasoning systems.

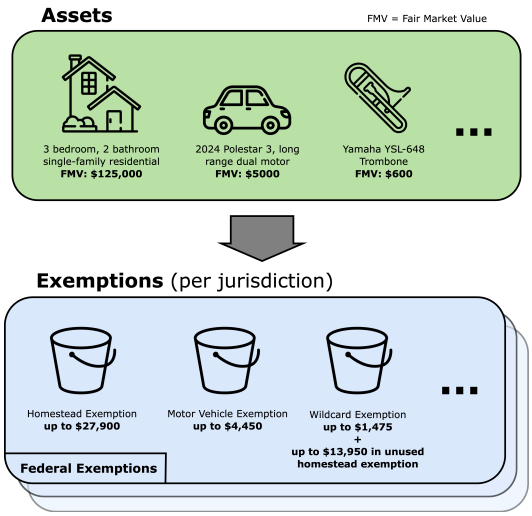

Figure 1: OpenExempt tasks center on U.S. bankruptcy law, primarily asset exemption where assets must be assigned to exemption statutes with dollar limits.

1 Introduction

Language models (LMs) now demonstrate remarkable performance on a wide array of complex tasks, from writing code to passing professional exams. Yet, recent work has begun to question these abilities, probing whether models are truly reasoning or relying on sophisticated forms of memorization and pattern matching (Shojaee et al., 2025; Mirzadeh et al., 2025). This uncertainty has fueled a critical need for new evaluation methodologies that move beyond static evaluation (Hofmann et al., 2025), providing deeper, more diagnostic insights into the competencies and failure modes of these powerful systems.

This challenge is particularly acute in the legal domain where reasoning demands precision, consistency, and an understanding of intricate, interdependent rules (Servantez et al., 2024). Consequently, creating benchmarks for legal tasks has historically relied on expert annotation of solutions, a process that is not only expensive and time-consuming but also results in static datasets of fixed question-answer pairs (Guha et al.). Such datasets struggle to keep pace with rapidly evolving models and make it difficult to disentangle the many reasoning skills that a single legal problem may require. A model's failure on a complex task provides only a single, opaque signal of error. Gaining deeper insight requires a more controlled evaluation approach, one where task complexity and

Findings of the Association for Computational Linguistics: ACL 2026, pages 26644–26669

scope can be systematically adjusted to reveal a model's specific breaking points.

To address these limitations, we introduce OpenExempt, a framework and benchmark constructed through an interdisciplinary collaboration of computer scientists and legal professionals. The **OpenExempt Framework** enables fine-grain control over crafting complex legal reasoning tasks and their solutions on demand. This dynamic approach directly overcomes the constraints of static benchmarks by allowing users to vary many aspects of the task, including case details, jurisdictions, and the scope of the task itself, thereby enabling the isolation of different types of reasoning. This makes it possible to disentangle performance across distinct reasoning processes, avoiding the conflation of errors that can obscure a model's capabilities and limitations.

OpenExempt tasks center on the application of federal and state laws governing the exemption of assets under the United States Bankruptcy Code (U.S. Code Title 11)[1]. This process allows a person filing for bankruptcy to protect certain property from creditors. Inspired by computable contracts where natural language clauses are paired with formal, machine-readable logic (Surden, 2012; Clack et al., 2017), we combine statute text with structured, symbolic representations of their logic and dependencies, making solutions machine-computable. While our approach also requires legal interpretation, OpenExempt does not rely on direct annotation of task solutions. Instead, we use legal knowledge to encode statutes and case assets into structured representations, from which we can construct an immense space of natural language tasks and solutions, removing a key bottleneck in legal reasoning benchmark construction while preserving the accuracy and depth of expert reasoning.

Using this system, we construct the **OpenExempt Benchmark**, a diagnostic benchmark for legal reasoning composed of nine evaluation suites: 6 diagnostic suites and 3 competency suites. Diagnostic suites isolate specific reasoning challenges by varying task complexity across a single axis, such as the number of assets. This allows us to go beyond single points of failure by precisely measuring the performance delta caused by each variation. Competency suites provide a more holistic assessment of a model's legal reasoning capabilities at three levels of difficulty: basic, intermediate, and advanced. We release the OpenExempt Framework[2] and Benchmark[3] to the public under a permissive license (CC BY 4.0). OpenExempt is intended to support further research in both the legal and NLP communities.

2 Related Work

2.1 Legal Reasoning Benchmarks

A large body of prior work has examined the legal reasoning capabilities of language models using static datasets of fixed question-answer pairs. Large scale benchmarks like LegalBench (Guha et al., 2023), LEXTREME (Niklaus et al., 2023), LawBench (Fei et al., 2023), and LexGLUE (Chalkidis et al., 2022) provide broad assessments across diverse sets of legal tasks. Beyond multi-task benchmarks, other works have targeted specific legal skills, including contract review (Hendrycks et al., 2021), legal information retrieval (Zheng et al., 2025; Joshi et al., 2023), legal exam question answering (Fan et al., 2025), case holding identification (Zheng et al., 2021), legal judgment prediction (Chalkidis et al., 2019), as well as legal datasets for domain adaptation through pretraining (Niklaus et al., 2024; Henderson et al., 2022) and instruction tuning (Niklaus et al., 2025). These benchmarks and datasets have significantly advanced legal reasoning evaluation, yet their static design narrows evaluation to a one-size-fits-all assessment that neither accounts for varying model capabilities nor isolates specific failure modes. OpenExempt introduces a new benchmark paradigm where the user is in control of dynamically crafting legal tasks and defining complexity across multiple dimensions based on their specific evaluation goals.

2.2 Computable Statutory Reasoning

Our work is grounded in the field of computational law, where statutes are modeled as executable logic programs. A prominent example is Catala (Huttner and Merigoux, 2020; Lawsky, 2022), a domain-specific programming language designed to encode real-world tax laws in an executable form using prioritized default logic (Lawsky, 2017). Related work has also demonstrated how natural language contracts can be converted into executable

[1] https://uscode.house.gov

[2] Code: https://github.com/servantez/OpenExempt

[3] Data: https://huggingface.co/datasets/SergioServantez/OpenExempt

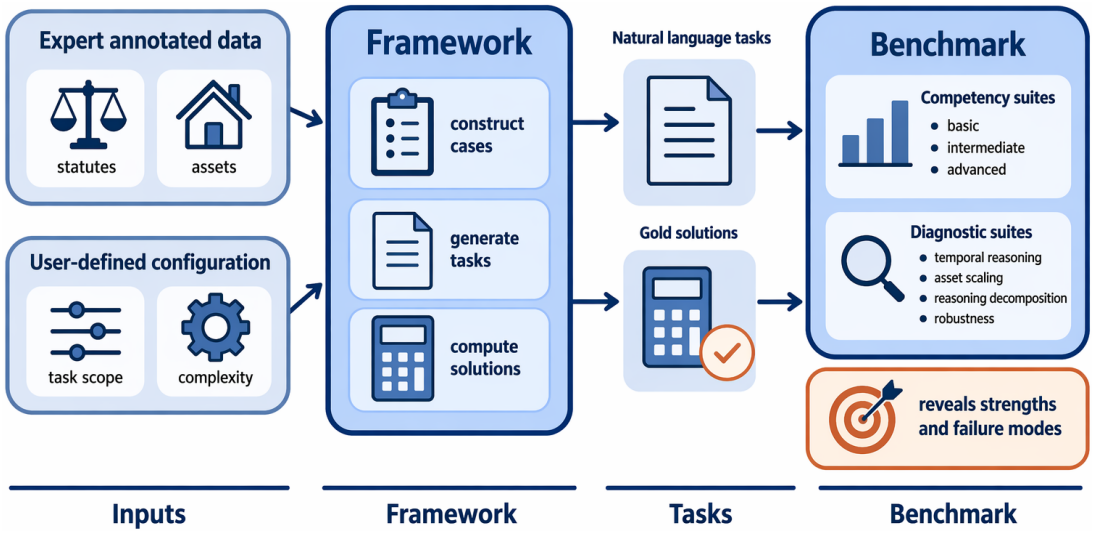

Figure 2: OpenExempt Framework Architecture. Dynamic task generation driven by user-defined configuration and grounded in structured legal knowledge.

programs in the Accord programming language (Roche et al., 2021), using an intermediate layer of symbolic legal representations (Servantez et al., 2023). While these works establish the feasibility of symbolic legal encodings, they primarily function as implementation languages or reasoning architectures rather than evaluation benchmarks. SARA (Holzenberger et al., 2020; Blair-Stanek et al., 2023) is the seminal dataset for evaluating statutory reasoning in language models using Prolog encodings to compute gold solutions for tax problems, yet its approach requires that each scenario be hand-crafted, yielding a fixed dataset of only a few hundred examples. Other prior work has taken an important step toward evaluating hierarchical legal reasoning using case-based analogies, but is also limited by a small, static dataset and does not address statutory reasoning (Zhang et al., 2025). These threads of work point to the need for a benchmark that is simultaneously dynamic, configurable, diagnostic, grounded in legal knowledge, and scalable beyond hand-crafted datasets – a combination realized in OpenExempt.

3 OpenExempt Framework

The OpenExempt framework is a dynamic task generation system that creates complex legal reasoning benchmarks in the domain of U.S. bankruptcy exemptions. This framework consists of three primary components: 1) a knowledge representation layer that encodes expert legal annotations; 2) a task generator that constructs paired representations in both symbolic and natural language forms; and 3) a deterministic solver that computes ground truth solutions using branch and bound optimization. The bankruptcy exemption process is an ideal proving ground for controlled evaluation because it allows incremental adjustments to task complexity. Adding assets to a case increases complexity superlinearly: each new asset must not only be evaluated individually, but also in competition with others for shared statutory limits.

3.1 Asset Exemption in Bankruptcy

A person filing for bankruptcy, called the Debtor, is allowed to protect certain property from seizure by creditors. An exemption defines a category of property which can be protected - for example, up to $4,450 in a motor vehicle (Figure 1). Each state enacts its own exemption statutes which differ considerably in regards to which assets are protected. The debtor may claim state or federal exemptions, unless their state specifically prohibits the use of federal exemptions, known as "opt-out"[4]. Which state exemption laws apply to a given case is determined by where the debtor lived in the 730 days before filing for bankruptcy[5]. Asset exemption is a combinatorial optimization problem, like a legal version of the well known knapsack problem (Cormen et al., 2022), where an asset can only be protected by certain exemptions and the goal is to minimize the dollar value of unprotected assets. This process involves many intermediate tasks and can be challenging even for legal professionals.

[4] 11 U.S.C. §522(b)(2)
[5] Id. §522(b)(3)(A)

3.2 Structured Legal Knowledge

While the dynamic nature of the framework enables controlled variation in task structure and complexity, it also requires that the underlying legal process be represented in a precise and machine computable form. At the core of OpenExempt are two expert annotated datasets:

- **Debtor Assets**. The framework contains over 500 assets (motor vehicles, real estate, household goods) each manually labeled with the complete set of applicable exemptions for every supported jurisdiction. During task generation, the framework samples from this asset collection, providing one of the many factors that enable OpenExempt to construct a vast combinatorial space of possible cases.

- **Exemption Statutes**. We encode federal and state exemption statutes into structured representations that capture the logical rules required for symbolic reasoning, including monetary caps and state opt-out provisions. We also capture a rich set of constraints and relationships that commonly arise in exemption statutes, as discussed in Section A.5. OpenExempt currently supports federal exemption statutes and state exemptions for Arizona, Illinois, Oregon, Pennsylvania, and Wisconsin[6]. See Section A.11 for a list of statute sources.

Both datasets adopt a dual representation design, where each asset or statute exists as a pair: a natural language form used in constructing the task prompt, and a structured representation used in computing gold solutions grounded in legal knowledge. This approach is inspired by prior work on smart contracts for legal documents, most notably Accord (Roche et al., 2021) and Catala (Merigoux et al., 2021).

3.3 Dynamic Task Generation

OpenExempt dynamically generates benchmark tasks from a user-defined configuration file that specifies the structure, scope and complexity of the legal problems to be created (see Table 5 for complete list of parameters). This process is largely driven by two components: `CaseGenerator` and `TaskGenerator` (Figure 2). At runtime, `CaseGenerator` constructs symbolic bankruptcy cases by sampling case attributes within the bounds set by the configuration. The resulting case object captures relevant legal facts, including parties, marital status, petition date, domicile timeline, and the applicable exemption jurisdiction based on that timeline. `TaskGenerator` then renders these structured facts into a natural language prompt. When the user-defined task scope includes intermediate subtask solutions, `TaskGenerator` invokes the OpenExempt task solver to compute the required intermediate outputs and embeds them into the prompt.

To convert structured case data into natural language, OpenExempt uses a hybrid, human-in-the-loop approach rather than relying on direct end-to-end generation by a language model. We first use a language model to produce many candidate phrasings for fixed scenarios, such as asset ownership descriptions, residency histories, and obfuscating facts. A legal professional then screens these candidates and removes any phrasing that introduces ambiguity, alters legal meaning, or misstates a material fact. This step is necessary because fluent model-generated text can still contain subtle but legally consequential errors (Dahl et al., 2024), such as describing a debtor as merely possessing an asset rather than legally owning it. The approved phrasings are converted into parameterized templates with placeholders for variable elements, such as debtor names, dates, and asset descriptions, which are hydrated during prompt construction. This approach strikes a necessary balance between introducing linguistic variation and preserving factual and legal fidelity.

3.4 Computing Gold Solutions

While OpenExempt tasks are dynamically generated, all solutions are grounded in expert knowledge. The annotated assets enumerate the exemption claims permissible for each asset, while the machine-readable statutes encode the constraints that govern how those claims may be applied. Together, these resources enable the solver to validate candidate outputs and thus define the solution space. For asset-level tasks, like Task EC and EV (defined in Section 4.1), the ground truth is directly recovered from the asset annotations for the relevant jurisdiction. For estate-level tasks, like Task NA and OE, which require jointly allocating exemptions across all assets, the framework employs the symbolic solver to perform a branch and bound search over all legally valid exemption assignments.

[6]These states were selected for diversity in both opt-out status and generosity in exemption coverage and limits.

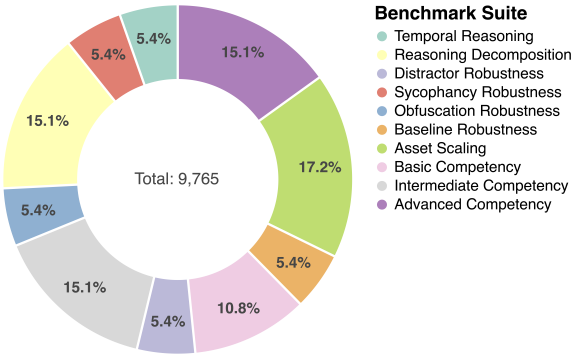

Figure 3: Sample distribution across benchmark suites.

This brute force search is made tractable by pruning partial solutions that cannot surpass the best known allocation, based on remaining exemption capacity and unprocessed assets. Because the solver only explores legally valid allocations, and because all legal rules defining valid claims originate from expert curated encodings, the resulting optimal allocation is both computationally verified and expert grounded.

4 OpenExempt Benchmark

Using the above framework, we construct the OpenExempt benchmark consisting of 9,765 samples across nine evaluation suites (Figure 3 shows the sample distribution across suites).

4.1 Tasks

OpenExempt is composed of five tasks, with a total of 15 task variants, that mirror the sequence of legal reasoning steps a debtor's attorney performs when protecting assets in bankruptcy. For each task, the model receives a fact pattern detailing the debtor's situation, which may include asset disclosures and residential history, along with a corpus of relevant federal and state laws. We provide several example task prompts with solutions in Section A.10, and describe each task below:

Task AE (Allowable Exemptions): Before exemptions can be claimed, the Bankruptcy Code requires first understanding which state or federal exemptions are available to the Debtor. This task involves applying the multi-step "730-day Rule"[7] to the Debtor's residency history to identify the applicable exemption jurisdictions, while accounting for state opt-out provisions[8].

[7] 11 U.S.C. §522(b)(3)(A)
[8] Id. §522(b)(2)

Task EC (Exemption Classification): Once the allowed exemption jurisdictions have been identified, each asset must be matched to the categories of exempt property defined by statute. This task requires rule-based reasoning to determine if a given asset satisfies the exemption antecedent, the specific property category defined by the statute. This is a multi-label classification problem since multiple exemptions can apply to a single asset.

Task EV (Exemption Valuation): Exemptions are typically limited to a fixed dollar amount defined by the statute. This task requires not only identifying applicable exemptions, but also applying these statutory caps to calculate the maximum protectable dollar value for each asset under each of its available exemptions. Tasks EC and EV require asset-level reasoning since each asset-exemption pair is considered independently, without factoring in aggregate limits.

Task NA (Non-exempt Assets): Tasks NA and OE demand estate-level reasoning to strategically allocate exemptions across all the Debtor's assets. This task requires solving this strategic allocation to determine the minimal total dollar value of non-exempt assets after applying all applicable exemptions, for each allowable exemption jurisdiction.

Task OE (Optimal Exemptions): This task requires articulating the complete, optimal strategy to achieve the best outcome from Task NA. This requires selecting the allowable exemption jurisdiction that minimizes non-exempt asset value, and generating the explicit exemption schedule for that optimal jurisdiction. The schedule must produce a complete mapping of which exemptions, and what dollar amounts, are allocated to each specific asset.

Task Variants. Since OpenExempt tasks form a sequential pipeline (Figure 5), solving any given task depends on the successful completion of its predecessors. The OpenExempt framework allows users to configure which earlier steps are already solved and provided in the task prompt. This creates a family of task variants, where a task can be presented in its vanilla form (no prior steps solved) or with some or all preceding steps solved by the framework. Each variant corresponds to a contiguous interval of the pipeline. This design enables a fine-grained assessment of how cumulative complexity and error propagation impact model performance, a capability we later demonstrate with the Reasoning Decomposition evaluation suite.

4.2 Benchmark Suites

The OpenExempt Benchmark organizes its evaluation into nine suites designed to capture both broad and fine-grained assessments of legal reasoning. The three competency suites evaluate a wide range of exemption scenarios to provide a holistic view of a model's reasoning capabilities, mirroring a traditional benchmark. In contrast, the six diagnostic suites isolate and vary one specific dimension of task complexity, such as the density of obfuscating statements. This approach enables targeted, causal analysis, allowing us to isolate and understand precisely which specific reasoning challenges are contributing to performance degradation. See Table 6 for a summary of configuration settings for each suite.

4.2.1 Competency Suites

Basic, Intermediate, and Advanced Competency. These three suites form a structured progression of difficulty where higher tiers contain more complex fact patterns, including larger asset pools, more extensive domicile histories, and exemption statutes drawn from a broader set of state jurisdictions. This tiered design ensures that the benchmark remains informative across a wide range of model capabilities: smaller models can be meaningfully evaluated on the lower tiers without collapsing to failure, while larger models can be challenged at higher tiers without saturating performance. In this way, the competency suites yield reliable, discriminative signals that align with the reasoning capacity of the model being assessed.

4.2.2 Diagnostic Suites

Temporal Reasoning. This suite isolates reasoning about temporal rules that govern exemption eligibility under the Bankruptcy Code. In these tasks, the Debtor's prior residences are spread across multiple states and dates, and the model must determine which exemptions the Debtor is permitted to claim by correctly applying the 730-day Rule under 11 U.S.C. §522(b)(3)(A). This rule is a three part statutory test that requires reasoning about both the duration and location of the Debtor's domicile. By increasing the complexity of the domicile history while holding all other settings constant, we can precisely measure how temporal complexity affects model performance.

Reasoning Decomposition. This diagnostic suite measures the effects of cumulative complexity and error propagation across the OpenExempt task pipeline (Figure 5). It evaluates Tasks EC, EV, NA and OE, by testing against all possible preceding task variants (Task AE has no preceding tasks). This configuration allows for a causal decomposition of total error into two components: stage error, which reflects the model's inability to solve the target task itself (e.g., Task EV), and propagation error, which arises from the model's reliance on its own incorrect conclusions from preceding steps (e.g., Tasks AE and EC).

Distractor, Sycophancy, and Obfuscation Robustness. The OpenExempt benchmark includes three diagnostic suites designed to evaluate how well models maintain legal accuracy when faced with extraneous, misleading, or irrelevant information within the fact pattern. OpenExempt supports two forms of obfuscation: irrelevant facts, which introduce legally immaterial details about assets or prior residences (e.g., property not owned by the Debtor or travel that does not alter domicile), and opinions, which present subjective statements about assets or exemption eligibility that carry no legal force. The Distractor Robustness suite introduces irrelevant facts, testing whether models can ignore seemingly pertinent but legally inconsequential information. The Sycophancy Robustness suite introduces opinion statements, measuring whether models are influenced by subjective assertions rather than statutory requirements. The Obfuscation Robustness suite combines both types, presenting the full range of distracting and misleading content. The effect of each perturbation is isolated by comparison to a baseline configuration with no obfuscating statements.

Asset Scaling. Asset pool size is a significant driver of task complexity. This suite evaluates how model performance changes as the number of assets in the Debtor's estate is incrementally scaled. The primary source of emerging difficulty is the strategic competition between assets for limited statutory exemption values. In cases with one or two assets, the optimal allocation of exemptions can be trivial. However, as the asset pool size grows and multiple assets become eligible for the same finite exemption values, the task transitions into a complex optimization problem. The model is forced to consider alternative exemptions and strategically allocate the available dollar value of each to maximize the total protected value of the estate, thereby demanding a much stricter and more comprehensive legal reasoning process.

Table 1: Model performance (sample-based $F1$) by task across Basic (bc), Intermediate (ic), and Advanced Competency (ac) suites.

Task	AE	EC	EV	NA	OE
Suite	bc / ic / ac	bc / ic / ac	bc / ic / ac	bc / ic / ac	bc / ic / ac
GPT-5	.884 /.743 /.612	**.924** /**.744** /.554	.893 /.635 /.496	.893 /**.733** /**.558**	.933 /**.744** /.404
o3	.917 /.747 /.604	.901 /.743 /**.571**	**.912** /**.651** /.500	**.949** /.728 /.548	**.944** /.724 /.408
o4-mini	.940 /**.757** /.621	.711 /.575 /.418	.691 /.541 /.388	.744 /.539 /.326	.844 /.543 /.265
Claude-Sonnet-4	.940 /.743 /.625	.723 /.502 /.452	.697 /.478 /.353	.718 /.499 /.340	.805 /.476 /.255
Gemini-2.5-Pro	.943 /.753 /.605	.900 /.740 /.549	.877 /.623 /**.502**	.889 /.665 /.540	.938 /.714 /**.518**
DeepSeek-R1	**.957** /.728 /.610	.809 /.612 /.457	.771 /.546 /.364	.860 /.649 /.476	.901 /.607 /.347
GPT-4.1	.955 /.714 /.588	.522 /.276 /.224	.453 /.207 /.163	.689 /.504 /.316	.777 /.538 /.240
Llama-4-Maverick	.865 /.659 /.586	.515 /.317 /.226	.496 /.320 /.193	.554 /.227 /.122	.703 /.260 /.095
DeepSeek-V3	.942 /.673 /**.627**	.598 /.365 /.291	.530 /.366 /.274	.594 /.393 /.196	.802 /.330 /.170
Claude-3.5-Haiku	.707 /.572 /.360	.529 /.411 /.317	.415 /.344 /.256	.502 /.204 /.069	.561 /.147 /.026
Gemma-3	.710 /.539 /.403	.503 /.447 /.331	.373 /.264 /.191	.404 /.224 /.137	.660 /.140 /.007
Gemini-2.5-Flash	.935 /.723 /.574	.835 /.671 /.474	.836 /.562 /.396	.870 /.586 /.441	.935 /.671 /.355
Llama-4-Scout	.596 /.480 /.386	.401 /.360 /.281	.350 /.294 /.172	.422 /.188 /.118	.569 /.136 /.033

5 Results

We summarize our findings here, and provide complete experiment details in the Appendix.

5.1 Experimental Setup

To support few-shot learning, we follow LEXam (Fan et al., 2025) and split 5 samples from each task dataset into a dev set, with the remaining 100 samples in the test set. Evaluation suites contain a collection of these 105-sample datasets, each with its own configuration file. Across all suites, this yields 9,765 samples in total, split into 9,300 test samples and 465 dev samples. Prior work has shown that language models can struggle with in-context demonstrations in the legal domain (Servantez et al., 2024). Therefore, we focus this work on evaluating models in a zero-shot setting to establish baseline performance, but leave exploration of few-shot learning for future work.

5.2 Models

We evaluate 13 language models grouped into three categories: 1) *reasoning models*: GPT-5 (OpenAI, 2025a), Claude-Sonnet-4 (Anthropic, 2025), Gemini-2.5-Pro (Google, 2025b), o3 (OpenAI, 2025c), o4-mini (OpenAI, 2025c), and Deepseek-R1 (DeepSeek-AI et al., 2025a); 2) *large models*: GPT-4.1 (OpenAI, 2025b), Llama-4-Maverick (17B-128E-Instruct) (Meta, 2025), and Deepseek-V3 (DeepSeek-AI et al., 2025b); 3) *efficient models*: Gemini-2.5-Flash (Google, 2025a), Claude-3.5-Haiku (Anthropic, 2024), Llama-4-Scout (17B-16E-Instruct) (Meta, 2025), and Gemma-3-(27b-it) (Team et al., 2025). We use a temperature of 0 for all models that support temperature, except for DeepSeek-R1 which we set to 0.6 based on developer recommended settings (DeepSeek, 2025). We set max token length to 16384, or the maximum token length supported by the model if it is less. We find these extended outputs are necessary to ensure complete answers.

5.3 Evaluation Protocol

Across all tasks, OpenExempt reports precision, recall and F1 scores computed at the sample level and then macro-averaged across samples. For asset-level tasks (EC, EV), the evaluator first computes per asset scores within a case, then averages across assets to determine the sample score, preventing assets with more applicable exemptions from dominating the aggregate. For tasks with multi-label predictions (AE, EC, EV), we evaluate using set overlap across discrete labels (jurisdictions or exemption citations). For tasks involving dollar valued predictions (EV, NA, OE), we additionally compute mean absolute relative error (MARE) between predicted and gold amounts. A numeric prediction is treated as correct if it falls within a 5% absolute relative error tolerance of the corresponding gold value. Because relative error can become unstable for gold amounts near zero, we add a small stabilizing constant ϵ. Formally, for a predicted amount $\hat{y}$ and gold amount y, we define the within-tolerance indicator:

$$\mathbb{I}_\tau(\hat{y}, y) = \mathbf{1}\left[\left|\frac{\hat{y}}{y+\epsilon} - 1\right| < \tau\right]_{\epsilon=1,\ \tau=0.05}.$$

For tasks that require structured outputs (EC, EV, NA, OE), predictions that fail to parse are marked

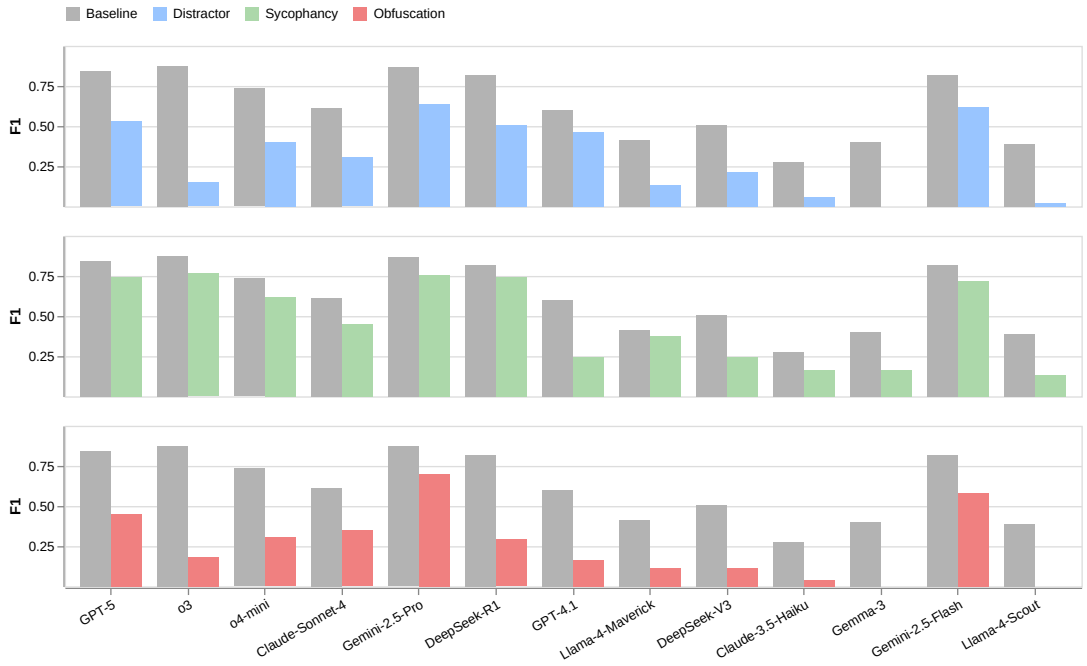

Figure 4: Model performance (F1) on Task OE under Distractor, Sycophancy, and Obfuscation perturbations. Colored bars show performance under each robustness suite, highlighting the degree to which model accuracy degrades relative to baseline.

as *invalid format* and are scored as incorrect, making format compliance a measured component of the benchmark. We observe a low rate of malformed responses across all evaluated models, typically below 1% (Table 2). We discuss LM response validation in detail in Section A.3.

5.4 Competency Evaluation

Basic, Intermediate, and Advanced Competency. **Competency suites produce stable, monotonic performance degradation across increasing difficulty levels, yielding clear and reliable distinctions between model reasoning capacities** (Table 1). Reasoning LMs predominantly outperform efficient and large non-reasoning models, with performance gaps particularly pronounced in more difficult settings. Gemini-2.5-Flash is a clear exception to this trend: despite being an efficient model, it consistently performs closer to reasoning models across our evaluations. GPT-5 and Gemini-2.5-Pro often rank among the best performing models. Yet, model recency alone does not explain performance, as o3 frequently outperforms newer models, even in the advanced tier. Notably, no model achieves an F1 score above 0.63 on any task in the advanced suite, indicating room for improvement in advanced legal reasoning where multi-step inference and complex rule application are prevalent. The Basic Competency suite reveals clear distinctions between efficient models on Task OE that vanish under advanced settings. For example, Gemma-3 outperforms Llama-4-Scout in the basic tier ($F1$ = 0.66 vs. 0.569), but both models collapse to near-zero performance in the advanced tier. Conversely, reasoning models approach saturation on simpler tasks ($F1 > 0.9$ on AE-bc), masking capabilities that only diverge under higher complexity settings. Our tiered evaluation approach mitigates these floor and ceiling effects that would otherwise obscure distinctions between models. To disentangle the failures observed in the competency evaluations, we next turn to the diagnostic suites.

5.5 Diagnostic Evaluation

Temporal Reasoning. **Temporal reasoning fails at a predictable threshold for reasoning models: performance declines modestly, then drops sharply, before leveling off** (Figure 7, Table 7). As temporal complexity increases while holding all other settings constant, reasoning models achieve perfect or near perfect performance with one domicile, and degrade only slightly at two (typically less than 0.03 $F1$). Performance drops become noticeably sharper as complexity reaches three and four domiciles, with four marking the clearest breaking point (at least a 0.145 $F1$ decrease for all reason-

ing models). This trend does not continue at five domiciles, where performance decreases typically return to only a few $F1$ points. We observe a similar pattern across all reasoning models (Figure 7). In contrast, efficient models exhibit greater variability, and tend to degrade earlier and more smoothly as domicile complexity increases.

Asset Scaling. **Asset scaling sharply separates model capacity: a select few frontier reasoning models remain robust under multi-asset exemption optimization, while many other models collapse.** (Figure 7, Tables 14 to 16). Performance declines are comparatively modest on asset-level tasks (EC, EV), even for efficient models, but become markedly sharper on estate-level tasks (NA, OE). For example, when scaling assets from 2 to 8, the $F1$ score for Llama-4-Scout decreases by 0.057 (EC) and 0.053 (EV), compared to 0.177 (NA) and 0.348 (OE). This gap reflects the shift from local exemption decisions to globally constrained allocation across competing assets. However, this degradation pattern is not universal, as three reasoning models (GPT-5, o3, Gemini-2.5-Pro) prove significantly more robust to increases in asset count, even where optimization demands are most acute. For example, GPT-5 declines by only 0.034 $F1$ on Task OE under full asset scaling (2 to 8 assets). These results indicate that the performance drops observed in the Advanced Competency suite cannot be attributed to asset complexity alone, but rather to the interaction of multiple difficulty dimensions.

Distractor, Sycophancy, and Obfuscation Robustness. **Model vulnerability to obfuscation is not uniform: identical statements can produce disparate levels of harm, which compound as legal reasoning becomes more complex.** (Figures 4 and 6, Tables 11 to 13). Model performance typically declines in the presence of obfuscating statements (distractors, opinions), but several models exhibit slight performance increases on simpler tasks (AE, EC), particularly under distractors alone. This pattern suggests extraneous information can encourage more deliberate analysis in less complex settings. Across Tasks AE through NA, the strongest reasoning models (GPT-5, o3, Gemini-2.5-Pro) exhibit little to no degradation under any obfuscation setting (Figure 6). Yet for Task OE, all reasoning models show substantial declines, with distractor and obfuscation perturbations producing the sharpest drops (Figure 4 and 6). This contrast is revealing given that all tasks use identical obfuscation statements. Models demonstrate a clear ability to discount irrelevant facts and opinions in simpler tasks, but stumble when the same statements appear in a more complex setting.

Reasoning Decomposition. **Correct intermediate solutions do not guarantee downstream performance gains and can even degrade it, revealing that reasoning through intermediate steps can be more beneficial than conditioning on partial solutions.** (Figure 8, Tables 8 to 10). Providing gold intermediate solutions typically boosts downstream performance, indicating that error propagation is a dominant factor in multi-step reasoning failures. Yet exceptions to this trend suggest that partial solutions can disrupt the model's reasoning trajectory for estate-level tasks that involve long reasoning paths. For example, efficient models often perform worse on Task OE when provided with gold NA solutions, despite NA supplying the exact non-exempt dollar amount that OE seeks to minimize. Notably, four of six reasoning models perform worse on Task NA when provided with gold EC or EV solutions, despite these steps specifying the exemptions and valuation limits needed to compute the total non-exempt amount. This behavior is particularly surprising for two reasons: (1) it is most pronounced in the reasoning models that otherwise exhibit the strongest reasoning capabilities across our experiments (GPT-5, o3, Gemini-2.5-Pro), and (2) it is largely absent in efficient and large non-reasoning models. We hypothesize that reasoning oriented post-training reinforces end-to-end reasoning trajectories, rather than conditional reasoning from partially solved states. As a result, providing intermediate conclusions can sometimes reduce performance by disrupting these reasoning trajectories.

6 Conclusion

We introduce OpenExempt, a framework for dynamically generating complex legal tasks grounded in structured legal knowledge, and a diagnostic benchmark for evaluating legal reasoning capabilities in language models. We release OpenExempt to the public to support further research and encourage collaboration between the legal and NLP communities.

Limitations

We note several limitations rooted in our design choices. OpenExempt currently: (i) focuses on bankruptcy and state exemption law; (ii) evalu-

ates only U.S. federal law and a small number of selected state jurisdictions; (iii) does not support multilingual tasks; and (iv) focuses on objectively correct tasks, which does not reflect the ambiguity common in legal practice. Given these limitations, OpenExempt should be treated as a complement to current evaluation methods, not a replacement. OpenExempt was designed to be easily extended by either legal or technical skill sets. We believe there is significant potential to build on OpenExempt and view these limitations as natural starting points for future work, including developing and evaluating new approaches to instruction tuning for stepwise legal reasoning.

Acknowledgments

This work was supported by the Center for Advancing the Safety of Machine Intelligence (CASMI).

References

Norah Alzahrani, Hisham Abdullah Alyahya, Yazeed Alnumay, Sultan Alrashed, Shaykhah Alsubaie, Yusef Almushaykeh, Faisal Mirza, Nouf Alotaibi, Nora Altwairesh, Areeb Alowisheq, M Saiful Bari, and Haidar Khan. 2024. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. *Preprint*, arXiv:2402.01781.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. Accessed: 2025-12-25.

Anthropic. 2025. Claude sonnet 4: System card. Accessed: 2025-12-25.

Andrew Blair-Stanek, Nils Holzenberger, and Benjamin Van Durme. 2023. Can gpt-3 perform statutory reasoning? *Preprint*, arXiv:2302.06100.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural legal judgment prediction in English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4317–4323, Florence, Italy. Association for Computational Linguistics.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. 2022. Lexglue: A benchmark dataset for legal language understanding in english. *Preprint*, arXiv:2110.00976.

Christopher D. Clack, Vikram A. Bakshi, and Lee Braine. 2017. Smart contract templates: foundations, design landscape and research directions. *Preprint*, arXiv:1608.00771.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2022. *Introduction to Algorithms*, 4 edition. MIT Press, Cambridge, MA.

Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E Ho. 2024. Large legal fictions: Profiling legal hallucinations in large language models. *Journal of Legal Analysis*, 16(1):64–93.

DeepSeek. 2025. Deepseek r1 model card. Accessed: 2025-12-27.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025b. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Yu Fan, Jingwei Ni, Jakob Merane, Yang Tian, Yoan Hermstrüwer, Yinya Huang, Mubashara Akhtar, Etienne Salimbeni, Florian Geering, Oliver Dreyer, Daniel Brunner, Markus Leippold, Mrinmaya Sachan, Alexander Stremitzer, Christoph Engel, Elliott Ash, and Joel Niklaus. 2025. Lexam: Benchmarking legal reasoning on 340 law exams. *Preprint*, arXiv:2505.12864.

Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Songyang Zhang, Kai Chen, Zongwen Shen, and Jidong Ge. 2023. Lawbench: Benchmarking legal knowledge of large language models. *Preprint*, arXiv:2309.16289.

Google. 2025a. Gemini 2.5 flash model card. Accessed: 2025-12-25.

Google. 2025b. Gemini 2.5 pro model card. Accessed: 2025-12-25.

Neel Guha, Julian Nyarko, Daniel E. Ho, and Christopher Ré. Building genai benchmarks: A case study in legal applications.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, and 21 others. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Preprint*, arXiv:2308.11462.

Peter Henderson, Mark S. Krass, Lucia Zheng, Neel Guha, Christopher D. Manning, Dan Jurafsky, and Daniel E. Ho. 2022. Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset. *Preprint*, arXiv:2207.00220.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. Cuad: An expert-annotated nlp dataset for legal contract review. *Preprint*, arXiv:2103.06268.

Valentin Hofmann, David Heineman, Ian Magnusson, Kyle Lo, Jesse Dodge, Maarten Sap, Pang Wei Koh, Chun Wang, Hannaneh Hajishirzi, and Noah A. Smith. 2025. Fluid language model benchmarking. *Preprint*, arXiv:2509.11106.

Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. 2020. A dataset for statutory reasoning in tax law entailment and question answering. *Preprint*, arXiv:2005.05257.

Liane Huttner and Denis Merigoux. 2020. Catala: Moving towards the future of legal expert systems. *Artificial Intelligence and Law*.

Abhinav Joshi, Akshat Sharma, Sai Kiran Tanikella, and Ashutosh Modi. 2023. U-creat: Unsupervised case retrieval using events extraction. *Preprint*, arXiv:2307.05260.

Sarah Lawsky. 2017. A logic for statutes. *Florida Tax Review*, 21:60–80.

Sarah Lawsky. 2022. Coding the code: Catala and computationally accessible tax law. *SMU Law Review*, 75:535.

Denis Merigoux, Nicolas Chataing, and Jonathan Protzenko. 2021. Catala: a programming language for the law. *Proc. ACM Program. Lang.*, 5(ICFP).

Meta. 2025. Llama 4 model cards and prompt formats. Accessed: 2025-12-25.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2025. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *Preprint*, arXiv:2410.05229.

Joel Niklaus, Veton Matoshi, Pooja Rani, Andrea Galassi, Matthias Stürmer, and Ilias Chalkidis. 2023. Lextreme: A multi-lingual and multi-task benchmark for the legal domain. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, page 3016–3054. Association for Computational Linguistics.

Joel Niklaus, Veton Matoshi, Matthias Stürmer, Ilias Chalkidis, and Daniel E. Ho. 2024. Multilegalpile: A 689gb multilingual legal corpus. *Preprint*, arXiv:2306.02069.

Joel Niklaus, Lucia Zheng, Arya D. McCarthy, Christopher Hahn, Brian M. Rosen, Peter Henderson, Daniel E. Ho, Garrett Honke, Percy Liang, and Christopher Manning. 2025. Lawinstruct: A resource for studying language model adaptation to the legal domain. *Preprint*, arXiv:2404.02127.

OpenAI. 2025a. Gpt-5 system card. Accessed: 2025-12-25.

OpenAI. 2025b. Introducing gpt-4.1 in the api. Accessed: 2025-12-25.

OpenAI. 2025c. Openai o3 and o4-mini system card. Accessed: 2025-12-25.

Pydantic. 2025. Pydantic validation. Accessed: 2025-12-28.

RapidFuzz. 2025. Rapidfuzz documentation. Accessed: 2025-12-28.

Niall Roche, Walter Hernandez, Eason Chen, Jérôme Siméon, and Dan Selman. 2021. Ergo–a programming language for smart legal contracts. *arXiv preprint arXiv:2112.07064*.

Sergio Servantez, Joe Barrow, Kristian Hammond, and Rajiv Jain. 2024. Chain of logic: Rule-based reasoning with large language models. *Preprint*, arXiv:2402.10400.

Sergio Servantez, Nedim Lipka, Alexa Siu, Milan Aggarwal, Balaji Krishnamurthy, Aparna Garimella, Kristian Hammond, and Rajiv Jain. 2023. Computable contracts by extracting obligation logic graphs. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, ICAIL '23, page 267–276, New York, NY, USA. Association for Computing Machinery.

Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. 2025. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. *Preprint*, arXiv:2506.06941.

Harry Surden. 2012. Computable contracts. *UC Davis Law Review*, 46.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Preprint*, arXiv:2406.01574.

Li Zhang, Matthias Grabmair, Morgan Gray, and Kevin Ashley. 2025. Thinking longer, not always smarter: Evaluating llm capabilities in hierarchical legal reasoning. *Preprint*, arXiv:2510.08710.

Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. 2021. When does pretraining help? assessing self-supervised

learning for law and the casehold dataset. *Preprint*, arXiv:2104.08671.

Lucia Zheng, Neel Guha, Javokhir Arifov, Sarah Zhang, Michal Skreta, Christopher D. Manning, Peter Henderson, and Daniel E. Ho. 2025. A reasoning-focused legal retrieval benchmark. In *Proceedings of the Symposium on Computer Science and Law on ZZZ*, CSLAW '25, page 169–193. ACM.

A Appendix

A.1 Modular Task Prompts

We manually write instructions for each task, which are combined with the generated fact patterns and selected exemption statutes to form the task prompts. When the user specifies a task variant (Section 4.1), we also include solved intermediate reasoning steps. All task prompt components (instructions, fact patterns, statutes) are stored separately in the benchmark, allowing us to collapse repeated elements across prompts to substantially reduce storage size. This modular design also aligns with our diagnostic evaluation goals by allowing the community to explore changes to question format and phrasing, which prior work has shown can have unpredictable effects on model performance (Wang et al., 2024; Alzahrani et al., 2024). OpenExempt provides a `TaskDataset` class to handle loading and iterating over examples.

A.2 Response Format Compliance

Table 2: Frequency and Percentage of Model Responses with Malformed JSON

Model	Frequency	Percent
GPT-5	2	0.03
o3	4	0.05
o4-mini	34	0.44
Claude-Sonnet-4	1	0.01
Gemini-2.5-Pro	3	0.04
DeepSeek-R1	34	0.44
GPT-4.1	12	0.15
Llama-4-Maverick	173	2.22
DeepSeek-V3	3	0.04
Claude-3.5-Haiku	22	0.28
Gemma-3	198	2.54
Gemini-2.5-Flash	153	1.96
Llama-4-Scout	117	1.50
Total	756	0.75

A.3 Response Validation

OpenExempt provides an `Evaluator` class to handle task-specific evaluation logic, including response format compliance and validation of predicted claims. For each sample, the evaluator: (i) isolates the final solution by extracting the suffix after the "FINAL ANSWER:" marker; (ii) parses the response with a task-specific Pydantic parser (Pydantic, 2025); and (iii) normalizes exemption citations (case folding, trimming) and aligns asset descriptions using fuzzy string matching (RapidFuzz (RapidFuzz, 2025) partial ratio with a threshold of 90) to ensure stable mapping between model predictions and gold labels. For all tasks except OE, evaluation compares predictions directly against the provided gold targets. Since optimal exemption schedules may not be unique, we evaluate Task OE by first validating predicted claims (e.g., ensuring claims obey exemption caps), before comparing against the known optimal outcome. This validation process is grounded in the same symbolic case objects and machine-readable statutes used during task generation. The predicted solution need not match the gold target to be correct, as long as its legally valid and achieves the same degree of protection.

A.4 Targeted Few-Shot Analysis

While we leave a thorough exploration of few-shot learning for future work, we conduct a targeted analysis on a single model-suite pair and observe no consistent performance improvement across settings.

Table 3: o4-mini Performance (F1) on Temporal Reasoning Suite Under Zero-Shot and 3-Shot Settings.

Setting	One	Two	Three	Four	Five
0-shot	1.00	.978	.888	.722	.670
3-shot	1.00	1.00	.890	.730	.670

A.4.1 Objective Correctness

Legal reasoning benchmarks must navigate the inherent gray area of statutory interpretation. OpenExempt mitigates this challenge by tightly controlling the scope of legal content and assets included in the benchmark, enabling the construction of tasks where solutions are objectively correct to a high degree of confidence. This requires the deliberate exclusion or modification of statutory provisions that introduce subjectivity. The goal of OpenExempt is not to perfectly model the application of the Bankruptcy Code and state exemption laws, but rather to construct complex legal reasoning tasks with objectively correct answers, which closely resemble real-world legal problems. We prioritize objective correctness through the following design choices:

Table 4: Exemption constraints and dependencies represented in OpenExempt, with citations to exemption statutes that exhibit these properties.

Variable Name	Description	Example Exemption
`single_limit`, `married_limit`	Maximum aggregated dollar amount that may be claimed by a single debtor or a married couple filing jointly.	11 U.S.C. § 522(d)(1)
`per_item_limit`	Maximum claimable amount per item, distinct from the overall aggregate limit.	11 U.S.C. § 522(d)(3)
`single_item_claim_count`, `married_item_claim_count`	Restricts the use of an exemption to a single item per claim (e.g., one motor vehicle). Married couples filing jointly may each be entitled to a separate single-item claim (e.g., one motor vehicle each).	735 ILCS 5/12-1001(c)
`fallback_exemption`	Specifies a relationship with another exemption, whose unused aggregate limit may be reallocated to this exemption.	11 U.S.C. § 522(d)(5)
`fallback_single_limit`, `fallback_married_limit`	Maximum amount claimable under the fallback exemption, based on marital status.	11 U.S.C. § 522(d)(5)
`mutual_exclusion`	Defines a mutual exclusion relationship with another exemption, such that claiming either one prohibits the use of the other.	Wis. Stat. § 815.18(3)(b)

- **Controlled Asset and Statute Selection**. We curate the pool of assets and exemptions to exclude provisions that rely on subjective standards, such as those requiring an item to be "reasonably necessary". By focusing primarily on tangible assets with clear statutory definitions and avoiding exemptions that depend on complex debtor attributes (disability status, profession), we ensure that the applicability of an exemption is a binary and deterministic question. The description of each asset contains all necessary predicates for a model to determine its eligibility.

- **Normalized Statutory Text for Self-Contained Reasoning**. The exemption statutes in OpenExempt are normalized to eliminate external references and latent ambiguity. For example, an exemption can incorporate requirements defined outside the current title: "Uniforms and accoutrements as provided by 51 Pa.C.S. § 4103"[9]. In these situations, we omit the reference or inline relevant text if possible. This ensures the model is evaluated on its ability to reason over the task prompt, rather than its ability to recall external legal knowledge not present in that context.

- **Encodable Exemption Logic**. We restrict the benchmark to exemption provisions whose operative logic can be faithfully captured by our formal constraint and dependency representation (Section A.5). While this representation covers many common statutory patterns, not all exemptions can be reduced to these encodings given the infinite variability in natural language. By excluding provisions that cannot be encoded, we ensure that our derived ground truth solutions remain computationally verifiable.

A.5 Exemption Constraints and Dependencies

To capture the structure and logic of legal exemptions, such as those found in the U.S. Bankruptcy Code, we introduce a formal representation of exemption constraints and dependencies. These refer to the various common conditions and relationships that govern how an exemption may be applied in practice. Exemption constraints include quantitative or structural limitations, such as caps on the allowable amount per item, differentiated limits for single versus married filers, or restrictions limiting a claim to a single asset. Exemption dependencies, in contrast, encode logical relationships between exemptions, such as mutual exclusions or fallback provisions. Together, these elements form a layer of semantic structure that is critical to accurately modeling exemption behavior and enabling reasoning over exemption applicability and interaction. See Table 4 for details on exemption constraints and dependencies.

[9] 42 Pa. Cons. Stat. §8124(a)(4)

A.6 Additional Figures

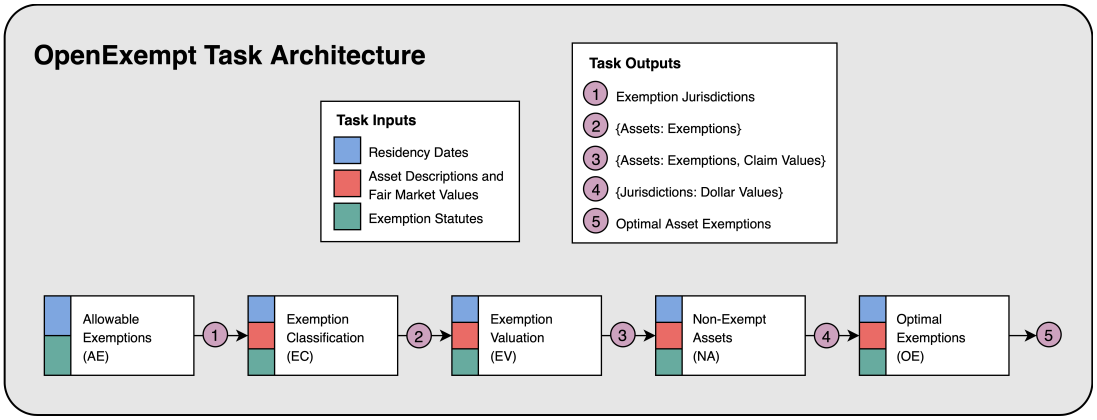

Figure 5: OpenExempt Task Pipeline. Each task depends on the successful completion of its predecessors, forming a composable sequence in which users can select any slice to isolate evaluation of specific reasoning types.

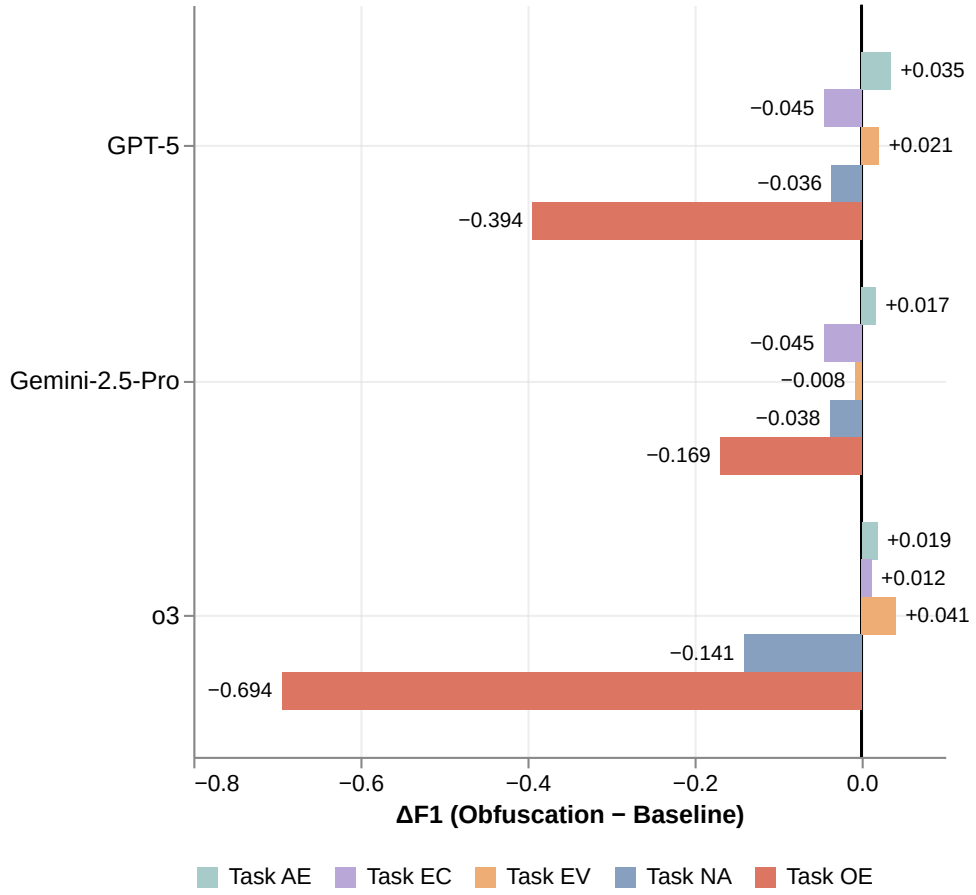

Figure 6: Obfuscation Robustness by task for three top performing reasoning models. Each bar shows the absolute change in F1 (ΔF_1) under obfuscation perturbations, computed as obfuscation minus baseline. Obfuscating statements are identical across tasks. Positive values indicate performance gains, negative values indicate degradation.

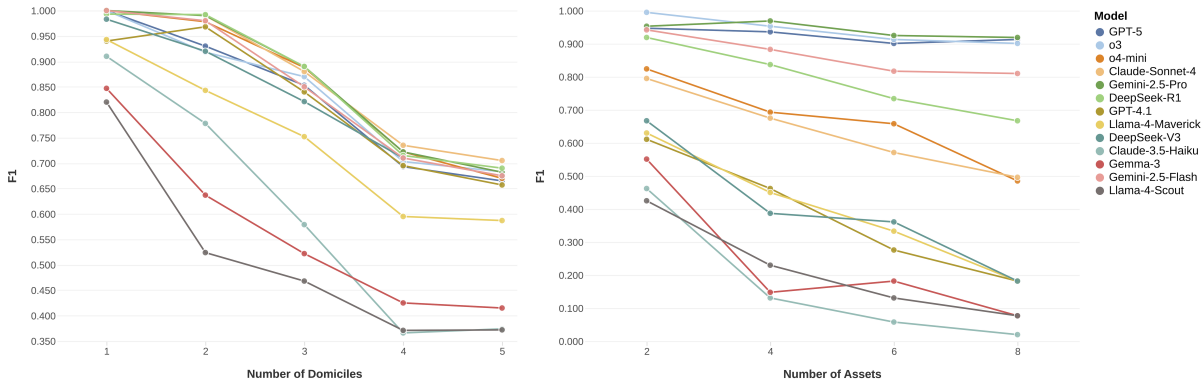

Figure 7: Model performance (F1) on Temporal Reasoning (left) and Asset Scaling (right) suites.

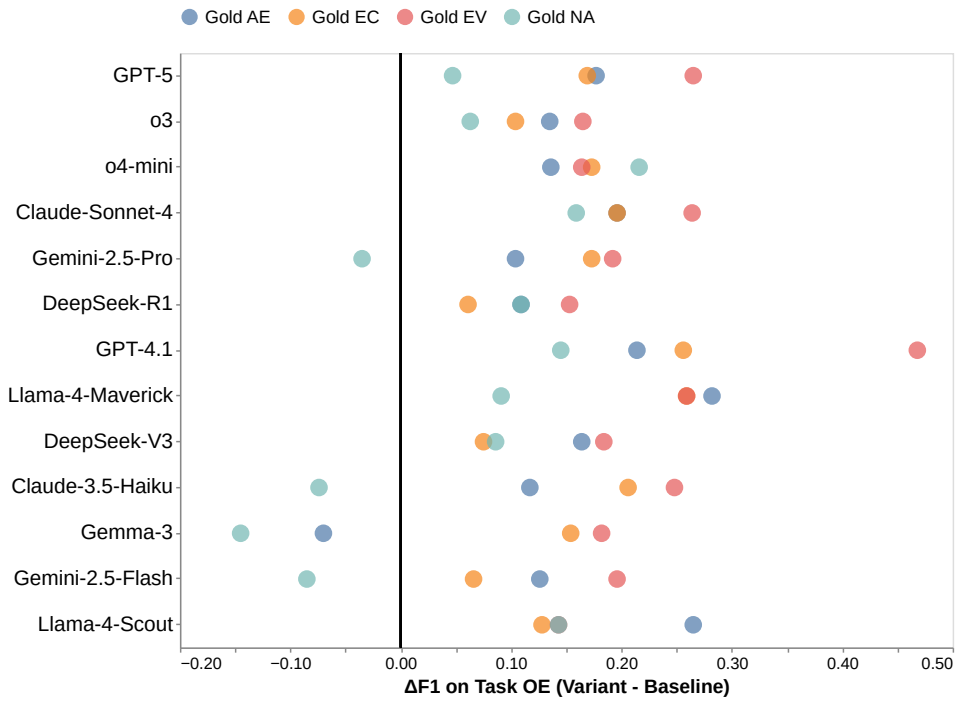

Figure 8: Reasoning Decomposition on Task OE. Each point shows the absolute change in F1 (ΔF_1) when a model is provided with gold solutions to a specific intermediate subtask, computed as variant minus baseline (no solved steps). Positive values indicate performance gains, negative values indicate degradation.

A.7 Configuration Parameters

Table 5: OpenExempt configuration parameters. Each parameter is specified within the configuration file to control task scope and complexity, dataset size, and degree of obfuscation.

Parameter(s)	Description
`start_task_id`, `terminal_task_id`	The process of exempting assets under the Bankruptcy Code proceeds through a fixed sequence of intermediate tasks (see Figure 5). These configuration parameters specify which portion of that sequence the model is responsible for solving. `start_task_id` marks the first task to be solved, and `terminal_task_id` marks the last. When both are set to the same value (e.g., 3–3), the configuration isolates a single reasoning task; when set to the broadest range (e.g., 1–5), it evaluates the entire exemption process. This design enables fine-grained analysis of how performance changes as cumulative reasoning complexity increases.
`dataset_size`	Specifies the number of unique tasks, and their corresponding ground-truth solutions, to generate under the given configuration. Each task is independently sampled using the specified asset ranges, jurisdictions, obfuscation settings, and all other configuration parameters.
`asset_count_min`, `asset_count_max`	Defines the minimum and maximum number of assets to include in each generated task. The actual asset count is sampled uniformly across this range, ensuring an equal distribution of tasks at each asset count. This allows controlled variation in task complexity across the dataset.
`married_ratio`	Specifies the proportion of generated tasks that involve married debtors. Marital status affects applicable exemption limits in many jurisdictions.
`domicile_count_min`, `domicile_count_max`	The minimum and maximum number of prior domiciles to include in each fact pattern, sampled uniformly across the specified range. Domicile history determines which federal and state exemption laws a Debtor is eligible to claim.
`state_jurisdictions`	Specifies the set of U.S. state jurisdictions used for task generation. For each task, one jurisdiction is sampled uniformly from this list to serve as the Debtor's allowable exemption jurisdiction. The exemption statutes for all listed jurisdictions are included in the prompt, requiring the model to identify the correct jurisdiction and apply its exemption laws to the facts.
`irrelevant_asset_facts`, `irrelevant_domicile_facts`, `asset_opinions`, `domicile_opinions`	Boolean parameters that control the inclusion of obfuscating information in the fact pattern. When enabled, the benchmark injects legally immaterial details or subjective statements related to assets or domicile history. These parameters are used to evaluate the model's robustness to distraction, misdirection, and sycophancy by testing its ability to disregard extraneous details while applying the correct legal reasoning.
`data_directory`, `asset_directory`, `statute_directory`, `template_directory`, `output_directory`	File path parameters that specify where the framework loads input resources and saves generated outputs. The input directories point to data dependencies required for task generation (annotated assets, exemption statutes, natural-language templates), while `output_directory` designates where generated tasks and solutions are written.

A.8 Benchmark Suite Composition

Table 6: Summary of configuration settings for each evaluation suite. Each dataset in the benchmark contains a configuration file with the exact construction specification.

Suite	Tasks	Solved Steps	Asset Count	Married Ratio	Domicile Count	Obfusc-ation	States
Temporal Reasoning	AE	No	N/A	0.5	1-5	No	All
Reasoning Decomposition	EC-OE	Yes	6	1.0	4	No	WI, IL, OR
Distractor Robustness	All	No	4	0.5	3	Yes	AZ, PA, WI
Sycophancy Robustness	All	No	4	0.5	3	Yes	AZ, PA, WI
Obfuscation Robustness	All	No	4	0.5	3	Yes	AZ, PA, WI
Asset Scaling	EC-OE	No	2-8	0.0	2	No	IL, OR, PA
Basic Competency	All	No	2	0.0	2-3	No	WI, IL
Intermediate Competency	All	No	3-5	0.5	4	Yes	AZ, PA, OR
Advanced Competency	All	No	6-8	1.0	5	Yes	All

A.9 Diagnostic Suite Results

Table 7: Model Performance (F1) on Temporal Reasoning Suite.

Number of Domiciles	One	Two	Three	Four	Five
GPT-5	1.00	.930	.853	.693	.665
o3	1.00	.918	.870	.703	.683
o4-mini	1.00	.978	.888	.722	.670
Claude-Sonnet-4	1.00	.990	.880	.735	.705
Gemini-2.5-Pro	1.00	.990	.890	.722	.682
DeepSeek-R1	.993	.992	.890	.715	.690
GPT-4.1	.940	.968	.840	.695	.657
Llama-4-Maverick	.943	.843	.752	.595	.587
DeepSeek-V3	.983	.920	.821	.710	.675
Claude-3.5-Haiku	.910	.778	.579	.366	.374
Gemma-3	.847	.637	.522	.425	.415
Gemini-2.5-Flash	1.00	.980	.850	.710	.675
Llama-4-Scout	.820	.524	.468	.371	.372

Table 8: Efficient Model Performance (F1) on Reasoning Decomposition Suite.

Task	Solved Steps	Claude-3.5 Haiku	Gemma-3	Gemini-2.5 Flash	Llama-4 Scout
EC	*None*	.402	.437	.595	.267
	AE	.706	.444	.854	.565
EV	*None*	.326	.262	.544	.223
	AE	.544	.280	.712	.242
	EC	.685	.492	.827	.347
NA	*None*	.137	.248	.513	.167
	AE	.190	.187	.759	.309
	EC	.322	.289	.513	.297
	EV	.372	.224	.462	.290
OE	*None*	.113	.165	.592	.148
	AE	.230	.095	.718	.413
	EC	.319	.319	.658	.276
	EV	.361	.347	.788	.291
	NA	.039	.020	.507	.291

Table 9: Large Model Performance (F1) on Reasoning Decomposition Suite.

Task	Solved Steps	GPT-4.1	Llama-4 Maverick	DeepSeek V3
EC	*None*	.265	.258	.367
	AE	.629	.533	.724
EV	*None*	.271	.219	.396
	AE	.593	.512	.559
	EC	.785	.798	.750
NA	*None*	.481	.197	.288
	AE	.511	.445	.473
	EC	.361	.420	.297
	EV	.441	.503	.397
OE	*None*	.305	.214	.387
	AE	.519	.496	.551
	EC	.561	.473	.462
	EV	.773	.473	.571
	NA	.450	.305	.473

Table 10: Reasoning Model Performance (F1) on Reasoning Decomposition Suite.

Task	Solved Steps	GPT-5	o3	o4-mini	Sonnet-4	Gemini Pro	DeepSeek R1
EC	*None*	.714	.742	.499	.534	.714	.575
	AE	.983	.983	.789	.924	.958	.803
EV	*None*	.671	.700	.549	.402	.668	.552
	AE	.800	.814	.695	.716	.811	.610
	EC	.836	.875	.882	.869	.851	.847
NA	*None*	.650	.681	.427	.414	.662	.582
	AE	.917	.907	.649	.735	.840	.709
	EC	.588	.529	.524	.519	.567	.499
	EV	.557	.539	.407	.531	.563	.458
OE	*None*	.611	.630	.485	.462	.684	.540
	AE	.788	.765	.621	.658	.788	.649
	EC	.780	.734	.658	.658	.857	.601
	EV	.876	.795	.649	.726	.876	.693
	NA	.658	.693	.701	.621	.649	.649

Table 11: Efficient Model Performance (F1) on Baseline, Distractor, Sycophancy, and Obfuscation Robustness Suites.

Task	Suite	Claude-3.5 Haiku	Gemma-3	Gemini-2.5 Flash	Llama-4 Scout
AE	*Baseline*	.618	.528	.745	.577
	Distractor	.645	.575	.831	.510
	Sycophancy	.507	.377	.831	.442
	Obfuscation	.509	.390	.796	.457
EC	*Baseline*	.478	.466	.772	.394
	Distractor	.453	.477	.739	.459
	Sycophancy	.312	.385	.749	.314
	Obfuscation	.319	.359	.719	.390
EV	*Baseline*	.382	.264	.715	.333
	Distractor	.378	.308	.677	.298
	Sycophancy	.236	.199	.701	.271
	Obfuscation	.262	.201	.652	.237
NA	*Baseline*	.297	.356	.763	.316
	Distractor	.154	.189	.753	.194
	Sycophancy	.159	.198	.776	.152
	Obfuscation	.112	.115	.787	.130
OE	*Baseline*	.276	.400	.817	.387
	Distractor	.058	.000	.621	.020
	Sycophancy	.165	.165	.718	.131
	Obfuscation	.039	.000	.582	.000

Table 12: Large Model Performance (F1) on Baseline, Distractor, Sycophancy, and Obfuscation Robustness Suites.

Task	Suite	GPT-4.1	Llama-4 Maverick	DeepSeek V3
AE	*Baseline*	.837	.820	.813
	Distractor	.835	.720	.802
	Sycophancy	.837	.748	.764
	Obfuscation	.788	.755	.710
EC	*Baseline*	.313	.341	.478
	Distractor	.335	.408	.472
	Sycophancy	.143	.303	.219
	Obfuscation	.095	.277	.239
EV	*Baseline*	.285	.369	.407
	Distractor	.273	.355	.401
	Sycophancy	.118	.232	.208
	Obfuscation	.105	.192	.195
NA	*Baseline*	.555	.416	.461
	Distractor	.467	.271	.328
	Sycophancy	.517	.321	.198
	Obfuscation	.450	.189	.186
OE	*Baseline*	.601	.413	.507
	Distractor	.462	.131	.214
	Sycophancy	.246	.374	.246
	Obfuscation	.165	.113	.113

Table 13: Reasoning Model Performance (F1) on Baseline, Distractor, Sycophancy, and Obfuscation Robustness Suites.

Task	Suite	GPT-5	o3	o4-mini	Sonnet-4	Gemini Pro	DeepSeek R1
AE	*Baseline*	.785	.808	.825	.810	.835	.835
	Distractor	.843	.845	.855	.865	.895	.887
	Sycophancy	.792	.805	.823	.830	.843	.827
	Obfuscation	.820	.827	.842	.820	.852	.830
EC	*Baseline*	.824	.816	.725	.597	.840	.755
	Distractor	.837	.832	.636	.535	.839	.738
	Sycophancy	.858	.871	.528	.486	.833	.533
	Obfuscation	.779	.828	.469	.419	.795	.435
EV	*Baseline*	.771	.737	.602	.532	.767	.637
	Distractor	.755	.734	.590	.487	.784	.656
	Sycophancy	.804	.787	.456	.411	.760	.510
	Obfuscation	.792	.778	.468	.336	.759	.432
NA	*Baseline*	.862	.868	.644	.511	.865	.789
	Distractor	.850	.764	.573	.557	.837	.705
	Sycophancy	.867	.835	.560	.539	.832	.818
	Obfuscation	.826	.727	.467	.474	.827	.674
OE	*Baseline*	.844	.876	.734	.611	.870	.817
	Distractor	.529	.148	.400	.305	.639	.507
	Sycophancy	.742	.765	.621	.450	.758	.742
	Obfuscation	.450	.182	.305	.347	.701	.291

Table 14: Efficient Model Performance (F1) on Asset Scaling Suite.

Task	Asset Count	Claude-3.5 Haiku	Gemma-3	Gemini-2.5 Flash	Llama-4 Scout
EC	*2*	.484	.445	.927	.426
	4	.452	.420	.902	.382
	6	.432	.404	.898	.367
	8	.412	.434	.888	.369
EV	*2*	.338	.362	.910	.367
	4	.343	.319	.899	.334
	6	.318	.235	.892	.270
	8	.328	.254	.866	.314
NA	*2*	.411	.370	.868	.416
	4	.314	.276	.848	.354
	6	.191	.242	.832	.290
	8	.140	.266	.845	.239
OE	*2*	.462	.551	.942	.425
	4	.131	.148	.883	.230
	6	.058	.182	.817	.131
	8	.020	.077	.810	.077

Table 15: Large Model Performance (F1) on Asset Scaling Suite.

Task	Asset Count	GPT-4.1	Llama-4 Maverick	DeepSeek V3
EC	*2*	.441	.514	.473
	4	.345	.465	.521
	6	.363	.479	.520
	8	.327	.470	.439
EV	*2*	.437	.447	.426
	4	.416	.438	.477
	6	.349	.437	.457
	8	.364	.447	.429
NA	*2*	.588	.528	.562
	4	.545	.480	.444
	6	.546	.472	.483
	8	.531	.433	.411
OE	*2*	.611	.630	.667
	4	.462	.450	.387
	6	.276	.333	.361
	8	.182	.182	.182

Table 16: Reasoning Model Performance (F1) on Asset Scaling Suite.

Task	Asset Count	GPT-5	o3	o4-mini	Sonnet-4	Gemini Pro	DeepSeek R1
EC	*2*	.964	.968	.794	.807	.951	.809
	4	.981	.961	.720	.770	.950	.801
	6	.941	.965	.754	.712	.958	.802
	8	.941	.945	.701	.706	.935	.804
EV	*2*	.949	.956	.763	.736	.934	.774
	4	.968	.945	.737	.721	.946	.752
	6	.957	.918	.720	.658	.946	.742
	8	.955	.950	.683	.666	.935	.729
NA	*2*	.927	.955	.737	.676	.883	.870
	4	.898	.962	.734	.633	.898	.850
	6	.918	.942	.687	.605	.889	.870
	8	.938	.949	.717	.643	.897	.867
OE	*2*	.947	.995	.824	.795	.953	.919
	4	.936	.953	.693	.675	.969	.837
	6	.901	.913	.658	.571	.925	.734
	8	.913	.901	.485	.496	.919	.667

A.10 Task Prompt Examples

Temporal Reasoning Suite: Task AE

Determine which state or federal exemptions may be claimed by the Debtor(s) under the provided statutes.

Your answer to this task must be based solely on applying the provided Federal and State statutes to the given facts.

Response Format: Your response must end with your final answer in the following template: FINAL ANSWER: [YOUR FINAL ANSWER]. Your final answer must consist of only a comma-separated list of jurisdictions, without any additional text. States should be identified by name. If federal exemptions are allowed, include 'Federal' in the list. Example response format: FINAL ANSWER: Alaska, Federal

Facts: Luis Gonzalez (hereinafter the Debtor) filed for bankruptcy on 14 July 2024. After moving their household to Delta, Pennsylvania on Saturday, March 21st, 2020, Luis Gonzalez eventually relocated to Marana, Arizona on 29th of February 2024.

Statutes:

Solution

Federal, Pennsylvania

Distractor Robustness Suite: Task EC

For each asset in the estate, identify all applicable exemptions under which that asset may be protected.

Your answer to this task must be based solely on applying the provided Federal and State statutes to the given facts. If the task involves a married couple, assume all assets mentioned are jointly owned, with each spouse holding an equal undivided interest, unless explicitly stated otherwise. Assume all assets are held for the personal use of the Debtor(s), unless explicitly stated otherwise.

Response Format: Your response must end with your final answer in the following template: FINAL ANSWER: [YOUR FINAL ANSWER]. Your final answer must consist of only valid JSON in the exact format specified below. Provide your final answer as a JSON object where: each key is the exact asset description provided in the fact pattern, and each value is an array of applicable exemption citations. Example response format: FINAL ANSWER: "1981 DeLorean DMC-12": ["11 U.S.C. § 522(d)(2)", "11 U.S.C. § 522(d)(5)"]

Facts: Megha and Dalia Joshi (hereinafter the Debtors) filed for bankruptcy on March 7th 2024. The Debtors began living in Litchfield Park, Arizona on 19 March 2012, but relocated to Waunakee, Wisconsin on 9.10.2021. For the 28 days following that date, Megha stayed in Patagonia, Arizona to complete a boater safety course and obtain a state-issued boating certificate, a new requirement for their job as a marine biologist. While there, they signed a rental agreement (Contract #R-781) for a boat slip at a local marina to berth the assigned training vessel for the duration of the course. The Joshis chose to relocate their household to Weyauwega, Wisconsin on 12th of September, 2021. Records show that Megha and Dalia Joshi possess a compact Bluetooth speaker with splash-resistant casing with a value of $400.00. Megha and Dalia own a calico cat worth $145.00. The Debtor's name is listed on a UTMA savings account, and the corresponding Form 1099-INT is mailed to their address. The account holds a balance of $8,150, which originated as an irrevocable gift from the Debtor's brother to the account's beneficiary, the Debtor's 14 year-old nephew. The known assets of Megha and Dalia Joshi include a woven tapestry wall hanging with bohemian motif worth $1,305.00 and a Hi-Point C9 9mm pistol appraised at $195.00.

Statutes:

Solution

{"compact Bluetooth speaker with splash-resistant casing": ["11 U.S.C. § 522(d)(3)", "11 U.S.C. § 522(d)(5)", "Wis. Stat. § 815.18(3)(d)"], "calico cat": ["11 U.S.C. § 522(d)(3)", "11 U.S.C. § 522(d)(5)", "Wis. Stat. § 815.18(3)(d)"], "woven tapestry wall hanging with bohemian motif": ["11 U.S.C. § 522(d)(3)", "11 U.S.C. § 522(d)(5)", "Wis. Stat. § 815.18(3)(d)"], "Hi-Point C9 9mm pistol": ["11 U.S.C. § 522(d)(5)", "Wis. Stat. § 815.18(3)(d)"]}

Reasoning Decomposition Suite: Task OE

Determine the optimal set of exemptions to best protect the assets in the estate. The goal is to minimize the total dollar value of non-exempt assets. If property may be exempted under multiple jurisdictions, you must select the jurisdiction that would result in the best solution.

Your answer to this task must be based solely on applying the provided Federal and State statutes to the given facts. If the task involves a married couple, assume all assets mentioned are jointly owned, with each spouse holding an equal undivided interest, unless explicitly stated otherwise. Assume all assets are held for the personal use of the Debtor(s), unless explicitly stated otherwise.

Response Format: Your response must end with your final answer in the following template: FINAL ANSWER: [YOUR FINAL ANSWER]. Your final answer must consist of only valid JSON in the exact format specified below. Provide your response as a JSON object where: each key is the exact asset description provided in the fact pattern, and each value is an array representing the optimal exemptions for this asset. Each exemption in this array is an object containing a citation and a claim value. Claim values must not contain any commas or dollar signs. Example response format: FINAL ANSWER: "1981 DeLorean DMC-12": ["citation": "11 U.S.C. § 522(d)(2)", "claim_value": 3000]

Facts: Tobias and Leon Fischer (hereinafter the Debtors) filed for bankruptcy on 10th day of January, 2024. Ownership of the small mountain cabin used year-round as the principal residence, priced at $49,500.00, is claimed by Tobias and Leon Fischer. The pair of suede ankle boots with zipper closure currently owned by the Fischers carries a value of $225.00. The Debtors assert ownership of an audiologist prescribed custom-fit hearing aids with behind-the-ear receiver and noise filtering with a market value of $1,425.00. A 14-karat gold engagement band with engraving on the inner surface valued at $770.00 is under the ownership of Tobias and Leon. An oxygen concentrator with portable carry cart and backup battery (physician authorized) appraised at $3,250.00 is the property of Tobias and Leon Fischer. A disclosure of assets by the Fischers reports a floor-length curtains with floral embroidery with a current value of $280.00.

Solved Reasoning Steps: The following reasoning steps have already been solved. Use this information to aid you in completing the remainder of the task. All applicable exemptions have been identified below for each asset in the estate. The small mountain cabin used year-round as the principal residence may be exempted under 11 U.S.C. § 522(d)(1), 11 U.S.C. § 522(d)(5), and Or. Rev. Stat. § 18.395(1). The pair of suede ankle boots with zipper closure may be exempted under 11 U.S.C. § 522(d)(3), 11 U.S.C. § 522(d)(5), Or. Rev. Stat. § 18.345(1)(b), and Or. Rev. Stat. § 18.345(1)(p). The audiologist prescribed custom-fit hearing aids with behind-the-ear receiver and noise filtering may be exempted under 11 U.S.C. § 522(d)(5), 11 U.S.C. § 522(d)(9), Or. Rev. Stat. § 18.345(1)(h), and Or. Rev. Stat. § 18.345(1)(p). The 14-karat gold engagement band with engraving on the inner surface may be exempted under 11 U.S.C. § 522(d)(4), 11 U.S.C. § 522(d)(5), Or. Rev. Stat. § 18.345(1)(b), and Or. Rev. Stat. § 18.345(1)(p). The oxygen concentrator with portable carry cart and backup battery (physician authorized) may be exempted under 11 U.S.C. § 522(d)(5), 11 U.S.C. § 522(d)(9), Or. Rev. Stat. § 18.345(1)(h), and Or. Rev. Stat. § 18.345(1)(p). The floor-length curtains with floral embroidery may be exempted under 11 U.S.C. § 522(d)(3), 11 U.S.C. § 522(d)(5), Or. Rev. Stat. § 18.345(1)(f), and Or. Rev. Stat. § 18.345(1)(p).

Statutes:

Solution

{"small mountain cabin used year-round as the principal residence": [{"citation": "11 U.S.C. § 522(d)(5)", "claim_value": 30850}, {"citation": "11 U.S.C. § 522(d)(1)", "claim_value": 18650}], "pair of suede ankle boots with zipper closure": [{"citation": "11 U.S.C. § 522(d)(3)", "claim_value": 225}], "audiologist prescribed custom-fit hearing aids with behind-the-ear receiver and noise filtering": [{"citation": "11 U.S.C. § 522(d)(9)", "claim_value": 1425}], "14-karat gold engagement band with engraving on the inner surface": [{"citation": "11 U.S.C. § 522(d)(4)", "claim_value": 770}], "oxygen concentrator with portable carry cart and backup battery (physician authorized)": [{"citation": "11 U.S.C. § 522(d)(9)", "claim_value": 3250}], "floor-length curtains with floral embroidery": [{"citation": "11 U.S.C. § 522(d)(3)", "claim_value": 280}]}

A.11 Statute Source by Jurisdiction

Jurisdiction	Source
Federal	https://uscode.house.gov
Arizona	https://www.azleg.gov/arstitle/
Illinois	https://www.ilga.gov/legislation/ilcs/ilcs.asp
Oregon	https://www.oregonlegislature.gov/bills_laws/pages/ors.aspx
Pennsylvania	https://www.palegis.us/statutes/consolidated
Wisconsin	https://docs.legis.wisconsin.gov/statutes/statutes/815