The Dog the Cat Chased Stumped the Model: Measuring When Language Models Abandon Structure for Shortcuts

Sangmitra Madhusudan[1], **Kaige Chen**[2], **and Ali Emami**[1]
[1]Emory University, Atlanta, USA
[2]Brock University, St. Catharines, Canada
smadhus@emory.edu, ax23ev@brocku.ca, aemami@emory.edu

Abstract

When language models correctly parse "The cat that the dog chased meowed," are they analyzing syntax or simply familiar with dogs chasing cats? Despite extensive benchmarking, we lack methods to distinguish structural understanding from semantic pattern matching. We introduce **CENTERBENCH**, a dataset of 9,720 comprehension questions on center-embedded sentences (like "The cat [that the dog chased] meowed") where relative clauses nest recursively, creating processing demands from simple to deeply nested structures. Each sentence has a syntactically identical but semantically implausible counterpart (e.g., mailmen prescribe medicine, doctors deliver mail) and six comprehension questions testing surface understanding, syntactic dependencies, and causal reasoning. Testing six models reveals that performance gaps between plausible and implausible sentences widen systematically with complexity, with models showing median gaps up to 26.8 percentage points, quantifying when they abandon structural analysis for semantic associations. Notably, semantic plausibility harms performance on questions about resulting actions, where following causal relationships matters more than semantic coherence. Reasoning models improve accuracy but their traces show semantic shortcuts, overthinking, and answer refusal. Unlike models whose plausibility advantage systematically widens with complexity, humans shows variable semantic effects. CENTERBENCH provides the first framework to identify when models shift from structural analysis to pattern matching.[1]

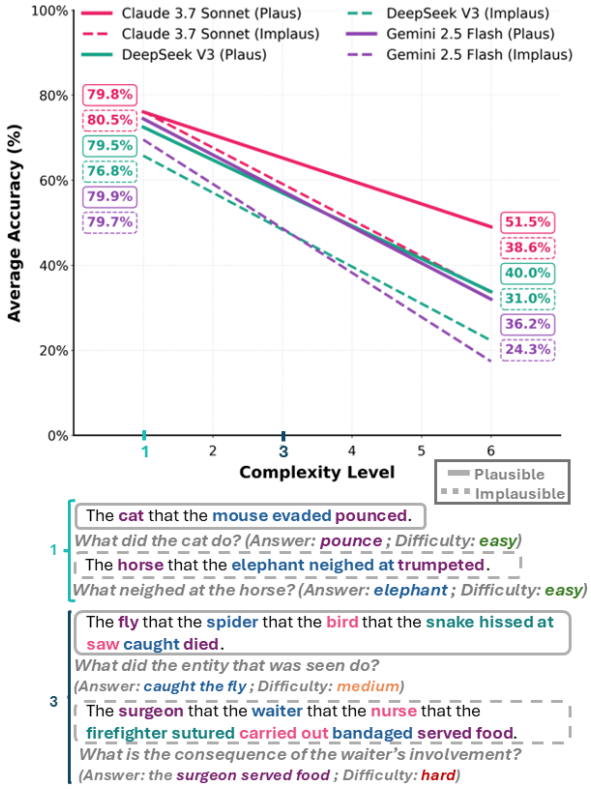

Figure 1: Performance degradation across complexity levels for three models on plausible vs. implausible center-embedded sentences (averaged across all question types), with example questions of varying difficulty.

1 Introduction

Large language models (LLMs) can explain quantum mechanics and write sophisticated code, yet often fail to parse sentences like "The cat that the dog that the mouse feared chased meowed", a construction that any undergraduate in linguistics can successfully diagram. This striking contrast points to a fundamental uncertainty in how we evaluate model capabilities: When models produce correct answers, are they genuinely parsing syntactic structure or merely exploiting semantic associations?

Standard benchmarks like MMLU (Hendrycks et al., 2021) and HLE (Phan et al., 2025) cannot address this uncertainty—they measure only final accuracy, revealing nothing on whether models arrived at correct answers through structural analysis or semantic pattern matching. We need evaluation

[1]The complete dataset and codebase are publicly accessible on GitHub.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 428–453
March 24-29, 2026 ©2026 Association for Computational Linguistics

frameworks that can (1) distinguish structural understanding from semantic shortcuts and (2) track how this balance shifts with increasing complexity.

Center-embedded sentences (Chomsky and Miller, 1963) offer an ideal testbed. In these constructions, relative clauses nest recursively, creating natural complexity scaling:

> **Level 1:** The cat [that the dog chased] meowed.
> **Level 2:** The cat [that the dog [that the boy saw] chased] meowed.
> **Level 3:** The cat [that the dog [that the boy [that the girl liked] saw] chased] meowed.

Unlike artificial puzzles, these constructions appear in natural language, have well-defined syntactic parses, and show consistent human processing patterns established through decades of psycholinguistic research (Miller and Chomsky, 1963; Gibson, 1998). Deep embeddings rarely appear in training corpora (Karlsson, 2007), minimizing memorization. While Hardt (2025) found GPT-4 maintains accuracy on these structures up to 3-4 levels, they didn't investigate whether this stemmed from structural analysis or semantic strategies.

To address this gap, we create matched sentence pairs differing only in semantic plausibility:

> **(1a)** Plausible: The cat that the dog chased meowed.
> **(1b)** Implausible: The waiter that the mailman seated delivered mail.

Both sentences share identical syntax: parsing who did what requires tracking the same color-coded noun-verb relationships. The crucial difference lies in semantic plausibility. While real-world knowledge supports dogs chasing cats, it contradicts mailmen seating waiters. By comparing performance across matched pairs at increasing complexity levels (1-6), we can quantify exactly when and how models transition from structural analysis to semantic shortcuts. This extends Wilcox et al. (2019)'s single-depth plausible/implausible design to systematic complexity scaling.

We introduce **CENTERBENCH**, a benchmark of 360 center-embedded sentences with 9,720 comprehension questions, featuring controlled complexity scaling and plausible/implausible pairing. Evaluating six models reveals notable patterns: performance degrades consistently with complexity (§5.1), while the plausible-implausible gap widens systematically, reaching median values over 25 percentage points (§5.2, Figure 1). Reasoning demands also matter: tasks involving causal reasoning show reversed patterns where semantics hinder rather than help (§5.3). Reasoning models improve accuracy but their traces suggest persistent failures: choosing semantic plausibility over syntactic structure, refusing to report implausible relationships, and overthinking (§5.4 & §5.5). Human comparisons reveal inconsistent semantic effects, with plausibility helping at some complexity levels but hindering at others (§5.6).

2 CENTERBENCH

CENTERBENCH consists of 9,720 questions across 360 center-embedded sentences designed to test whether language models truly understand syntactic structure or rely on semantic shortcuts.

As illustrated in examples **(1a)** and **(1b)**, we create sentence pairs with identical syntax but different semantic plausibility, then ask questions that require tracking noun-verb relationships. For both sentences, asking "What did the cat/waiter do?" requires linking the first noun with the last verb ("meowed", "delivered mail"). Similarly, "Who chased/seated the cat/waiter?" tests whether models correctly identify "the dog"/"the mailman". If models perform worse on questions involving (1b) despite identical syntactic structure to (1a), they must be using semantic shortcuts rather than structural understanding.

CENTERBENCH scales this approach across complexity levels 1-6, where each level adds one nested relative clause. Table 1 shows how structural complexity increases while maintaining the plausible/implausible contrast. Each sentence includes six comprehension questions per entity, ranging from simple action identification to complex causal reasoning. The dataset has 4,860 questions each for plausible and implausible conditions (9,720 total), with composition details in Table 2.

2.1 Sentence Creation

Each sentence in CENTERBENCH follows the following structure:

$$S(n) = NP_1 \text{ [that } NP_2 \text{ [} \ldots \text{ [that } NP_{n+1}\ V_{n+1}] \ldots V_2]\ V_1$$

Here, $S(n)$ represents a sentence at the complexity level n. Each NP_i stands for a *noun phrase*, such as "the cat" or "the dog" and each V_i stands

Level	Subset	Sentences
1	Plausible	The cat that the mouse evaded pounced. NP_1: *cat,* NP_2: *mouse,* V_2: *evaded,* V_1: *pounced*
1	Implausible	The horse that the elephant neighed at trumpeted. NP_1: *horse,* NP_2: *elephant,* V_2: *neighed at,* V_1: *trumpeted*
2	Plausible	The bicycle that the car that the truck hit bumped fell over. NP_1: *bicycle,* NP_2: *car,* NP_3: *truck,* V_3: *hit,* V_2: *bumped,* V_1: *fell over*
2	Implausible	The bicycle that the airplane that the train pedaled past whistled at taxied. NP_1: *bicycle,* NP_2: *airplane,* NP_3: *train,* V_3: *pedaled past,* V_2: *whistled at,* V_1: *taxied*
3	Plausible	The fly that the spider that the bird that the snake hissed at saw caught died. NP_1: *fly,* NP_2: *spider,* NP_3: *bird,* NP_4: *snake,* V_4: *hissed at,* V_3: *saw,* V_2: *caught,* V_1: *died*
3	Implausible	The skunk that the lion that the bat that the elephant sprayed at trumpeted at echolocated roared. NP_1: *skunk,* NP_2: *lion,* NP_3: *bat,* NP_4: *elephant,* V_4: *sprayed at,* V_3: *trumpeted at,* V_2: *echolocated,* V_1: *roared*

Table 1: Example sentences from complexity levels 1-3, with color-coded noun-verb pairs for plausible and implausible subsets. Examples for complexity levels 1-6 are provided in Appendix Table 9.

for a *verb*, such as "chased" or "barked". The final verb, V_1, is always *intransitive*, while all intermediate verbs ($V_2, V_3, \ldots, V_{n+1}$) are *transitive* in nature. This creates the parsing challenge: Models must track which noun performs which action across the nested structure.

2.1.1 Plausible Subset

For plausible sentences, we selected nouns from established psycholinguistic norms (Van Overschelde et al., 2004) in three categories: animals, people (occupations) and vehicles. We refined these lists to include only common, everyday nouns, for example, keeping "eagle" and "crow" while excluding "oriole" and "chickadee".

We generated 30 sentences per complexity level using *GPT-4-0613* in a one-shot setting, providing our curated noun lists as a foundation while allowing flexibility for natural sentence construction. The model occasionally incorporated related nouns (e.g. "hawk" or "bird") to create more fluent sentences. The complete noun inventory and all generation prompts appear in Appendix Sections A.1.1 and A.1.2, respectively.

2.1.2 Implausible Subset

To create semantically implausible sentences with identical syntactic structure, we used a systematic verb-swapping approach. Starting with our plausible noun list, we refined it to include only entities with maximally distinct behaviors. For example, from people-nouns, we retained "doctor," "police officer," and "lawyer" while removing "superintendent" and "veterinarian" whose actions overlap with other professions.

We then used *claude-sonnet-4-20250514* through the Anthropic Console[2] to generate unique verbs for each retained entity. For example:

- **doctor**: *prescribed medicine to* (transitive), *diagnosed* (intransitive)
- **police officer**: *arrested* (transitive), *patrolled* (intransitive)
- **lawyer**: *cross-examined* (transitive), *objected* (intransitive)

To generate implausible sentences, we implemented circular verb swapping: each noun receives verbs originally associated with the next noun in the sequence, with the last noun getting verbs from the first. This ensures semantic violations while maintaining grammatical correctness. For example, in a sentence with [doctor, police officer, lawyer], the doctor would "patrol," the police officer would "cross-examine," and the lawyer would "diagnose."

We generated 30 sentences per complexity level using this algorithm, ensuring no duplicate combinations across the dataset. Full algorithmic details and sample verb assignments appear in Appendix Section A.1.3 and Appendix Table 7, respectively.

Complexity	Easy	Medium	Hard	Total
1	120	120	120	360
2	180	180	180	540
3	240	240	240	720
4	300	300	300	900
5	360	360	360	1080
6	420	420	420	1260
Subtotal per subset				**4860**
Total dataset (plausible + implausible)				**9720**

Table 2: CENTERBENCH Composition

[2] https://console.anthropic.com/workbench

Example Sentence: The **dog** that the **mailman** startled barked.			
Category	**Question Type**	**Questions about dog**	**Questions about mailman**
Easy	Action Performed	What did the **dog** do? → bark	What did the **mailman** do? → startle the dog
	Agent Identification	Who startled the **dog**? → the mailman	What was affected by the **mailman**? → the dog
Medium	Entity Count	How many distinct entities are in the sentence? → 2	How many distinct entities are in the sentence? → 2
	Nested Dependency	What did the entity that was startled do? → barked	What did the entity acted upon by the **mailman** do? → barked
Hard	Causal Sequence	What series of events led to the **dog**'s action? → the mailman startling the dog	What series of events led to the **mailman**'s action? → no prior events
	Chain Consequence	What is the consequence of the **dog**'s involvement? → none	What is the consequence of the **mailman**'s involvement? → the dog barked

Table 3: Question difficulty levels and types illustrated with entity-specific questions for a complexity level 1 center-embedded sentence. Color coding distinguishes the two entities: **dog** and **mailman**.

2.2 Sentence validation

All generated sentences from both subsets underwent manual validation across three dimensions:

- **Temporal validity**: Entities made inactive (e.g., caught, killed) cannot perform subsequent actions
- **Semantic requirements**: Plausible sentences must contain realistic actions; implausible sentences must violate semantic expectations
- **Syntactic accuracy**: Each sentence must have the correct number of entities and verbs for its complexity level

Appendix Table 8 illustrates our validation process with examples from complexity level 2. When sentences failed any criterion, we manually corrected them while preserving structure and complexity. For instance, "caught" preceding "chased" creates a temporal violation (a caught entity cannot chase), which we fixed by substituting "saw." Similarly, "adopted" in the plausible subset might accidentally create an implausible relationship, requiring replacement with a proper plausible verb.

Table 1 shows instances from both subsets across complexity levels 1-3, with complete examples through level 6 in Appendix Table 9.

2.3 Question and Answer Generation

We developed an automated system to generate six comprehension questions and their corresponding answers for each entity in every sentence, grouped into three difficulty levels:

- **Easy** questions test basic subject-verb relationships (e.g., "What did the dog do?")
- **Medium** questions require understanding syntactic structure (e.g., "What did the entity that was chased do?")
- **Hard** questions demand forward and backward causal reasoning (e.g., "What series of events led to the dog's action?")

Table 3 illustrates all the questions with their answers on a single instance. Each question-answer pair targets specific aspects of comprehension, from simple action identification to complex dependency resolution.

Our generation algorithm operates in four steps:

1. **Noun identification**: Extracts entities using predefined lists, handling multi-word nouns (e.g., "police officer")
2. **Structural parsing**: Maps subject-verb-object relationships by analyzing the reversed verb order
3. **Verb processing**: Converts verbs to appropriate forms (base, participle, gerund) using morphological rules
4. **Template instantiation**: Fills question-answer templates based on parsed relationships, ensuring answers use exact wording from the source sentence

All generated questions and answers underwent manual review. We corrected two primary error types: phrasal verb separation (e.g., "run away" split incorrectly) and irregular verb morphology (e.g., "called" → "cal" instead of "call"). Complete question-answer templates and a detailed walkthrough are provided in Appendix Section A.1.6.

3 Evaluation Pipeline

Our evaluation pipeline assesses whether model responses match gold standard answers through a multi-tier matching strategy. Figure 2 provides an overview of the complete process.

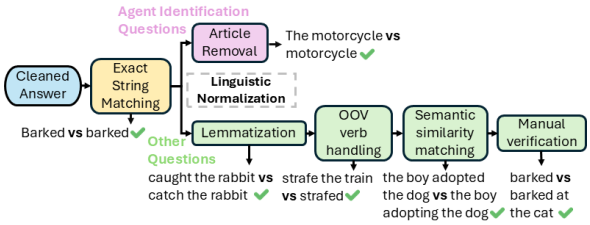

Figure 2: Overview of the evaluation pipeline for processing model responses (shown before 'vs') against gold standard answers

Before evaluation, we preprocess all model responses by removing hidden Unicode characters and stripping common prefixes (e.g., `Answer:`, `**Answer**:`) to ensure consistent formatting across different model outputs.

3.1 Automatic Evaluation

We apply increasingly sophisticated matching strategies, proceeding to the next step only if the previous one fails to find a match:

Exact string matching: Compare lowercase model and gold answers directly. This catches most correct responses immediately.

Linguistic normalization: For non-matching responses, we apply context-appropriate processing:

- *Agent identification questions*: Remove leading articles ("the," "a," "an") since "the mailman" and "mailman" are equivalent answers. If no match is found, the response is marked incorrect.
- *Other question types*: Apply lemmatization using spaCy's `en_core_web_sm` model (Honnibal et al., 2020) to handle verb inflections (e.g., "chased" → "chase"). If no match is found, processing continues to the next step.

Out-of-vocabulary (OOV) verb handling: Several verbs in our dataset (e.g., "gavel," "neigh") are missing from spaCy's vocabulary, causing incorrect rejections. We maintain a lookup table of these verbs with their inflected forms (see Appendix Table 11) to ensure correct evaluation. If a match is found here, the response is marked correct.

Semantic similarity matching: As a final step for remaining unmatched responses, we compute cosine similarity between sentence embeddings using `all-MiniLM-L6-v2` (Wang et al., 2020). For example, the gold answer "the truck hitting the car which led to the car bumping the bicycle" and model response "the truck hit the car, the car bumped the bicycle" convey identical meaning despite different syntax. Responses with similarity ≥ 0.9 are marked correct, following established QA evaluation practices (Morris et al., 2020; Berger et al., 2021; Six et al., 2025).[3]

3.2 Manual Verification

To validate our evaluation pipeline, we manually reviewed the automatically scored responses for the entire plausible subset. Our automatic pipeline achieved 98.95% scoring accuracy—meaning it agreed with human judgment 98.95% of the time.

The remaining 1.05% were evaluation errors (not model errors), concentrated in action performed questions where models provided correct but incomplete answers. For instance, given "What did the doctor do?" with gold answer "cross-examine the lawyer," a model response of "cross-examined" was incorrectly marked wrong by our system despite being substantially correct. We manually identified and corrected these evaluation errors to ensure accurate model assessment.[4]

4 Experiments

Models: We evaluated three models across different families and capabilities: DeepSeek-V3 (DeepSeek-AI et al., 2025b), Claude 3.7 Sonnet (Anthropic, 2025), and Gemini 2.5 Flash (Kavukcuoglu, 2025). For each model, we tested both standard inference and reasoning-enhanced modes: DeepSeek-R1 (DeepSeek-AI et al., 2025a), Claude 3.7 Sonnet with extended thinking enabled, and Gemini 2.5 Flash with thinking and dynamic thinking activated. Detailed configurations and prompts appear in Appendix Section A.2.1.

Human Evaluation: We conducted a pilot human evaluation using one randomly selected sentence per complexity level (1-4) from each subset (plausible and implausible), totaling 8 sentences. For each sentence, we evaluated all 6 questions targeting one specific entity, using 3 different participants per sentence. To prevent familiarization with the center-embedding format, no participant saw more than one sentence. This resulted in 24 total volunteer participants (3 participants × 8 sentences). Participants completed the study[5], with instructions provided on the form's homepage.

[3] This step improved evaluation accuracy from ∼94% to 100% on Claude's responses to 3,240 hard questions.

[4] Fleiss's Kappa of 1.0 between two independent annotators

[5] `https://center-embedding-study.vercel.app/`

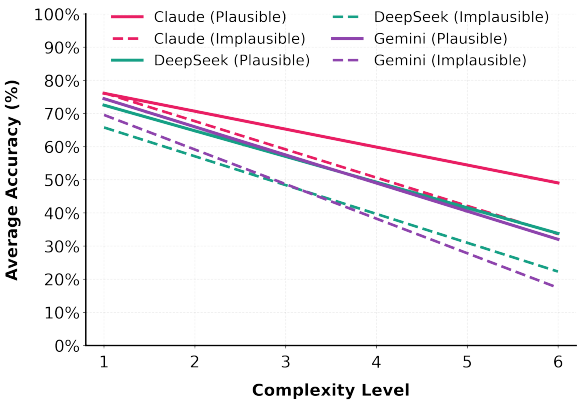

Figure 3: Performance degradation across complexity levels 1-6 for non-thinking models. Linear trendlines show average accuracy for plausible (solid) and implausible (dashed) sentences. All models show significant decline from level 1 to 6 ($p < 0.001$).

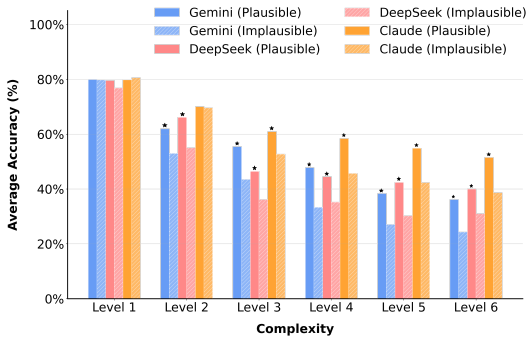

Figure 4: Average accuracy by complexity level for plausible vs. implausible sentences (non-thinking models). Asterisks indicate statistically significant differences between subsets at that complexity level ($p < 0.0083$). The widening gap shows models increasingly rely on semantic associations rather than structural analysis.

Metrics: We calculated accuracy as the percentage of correct answers using the evaluation pipeline from Section 3. Non-thinking models were run 10 times per question and averaged; thinking models were run once due to computational cost. Human responses were evaluated manually.

5 Results

5.1 Performance degrades with complexity

Figure 3 shows that all non-thinking models exhibit linear performance decline as center-embedding depth increases. Accuracy drops from ~80% at complexity level 1 to 24-52% at level 6 ($p < 0.001$), with consistent patterns across both plausible and implausible sentences.

This degradation occurs across three distinct model families (Gemini, DeepSeek, Claude), demonstrating that center-embedded sentences reliably measure syntactic processing limits. While Claude maintains the highest absolute performance, all models show similar linear decline rather than sudden collapse, indicating progressive loss of syntactic tracking ability as structural demands increase.

5.2 Models increasingly rely on semantic associations as complexity increases

Figure 4 shows model accuracy on plausible versus implausible sentences at each complexity level. At levels 1-2, models perform similarly on both plausible and implausible sentences. Starting at level 3, performance diverges: models score higher on plausible sentences, and this gap grows with each level, reaching over 9 percentage points by level 6.

All models show this pattern, but to different degrees. Across all complexity levels and question types, Claude's median performance gap between plausible and implausible sentences is 26.8 percentage points—this represents the median of 36 individual gaps (6 complexity levels × 6 question types), not just the overall averages shown in Figure 4. DeepSeek shows the smallest gap at 14.6 percentage points, with Gemini at 22.6 (see Appendix Figure 7).

This pattern reveals when models abandon structural analysis for semantic associations. At low complexity, they can track noun-verb relationships regardless of meaning. As complexity increases, they increasingly rely on semantic plausibility. The performance gap between subsets directly quantifies this reliance: identical syntactic structures should yield identical performance if models truly analyze structure over exploiting semantic patterns.

5.3 Semantic reliance varies by reasoning task

Figure 5 reveals a clear performance hierarchy across question types, with semantic plausibility affecting each differently. Entity counting is easiest and shows minimal plausibility effects (5.0 point gap), followed by basic comprehension tasks where plausibility provides substantial advantages: action performed ("What did X do?": 27.6 point gap, 78.8% vs. 51.2%) and agent identification ("Who did Y to X?": 15.1 points, 74.6% vs. 59.5%). Nested dependency questions ("What did the entity that was Y'd do?") show almost no plausibility effect (1.8 points), while complex reasoning tasks prove hardest overall.

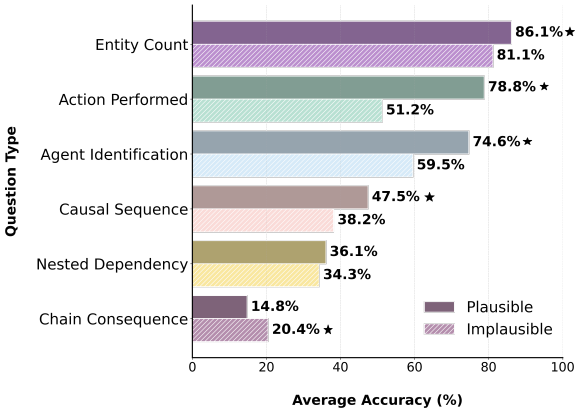

Figure 5: Average accuracy by question type for plausible vs. implausible sentences (non-thinking models). Asterisks mark significant differences ($p < 0.0083$).

Notably, chain consequence questions ("What is the consequence of X's involvement?") reverse this pattern, with implausible sentences outperforming plausible ones (20.4% vs. 14.8%. This pattern holds across all levels (Appendix Figure 8). Here, semantic familiarity actively harms performance: models follow plausible associations to wrong answers rather than trace actual causal chains.

These patterns show that semantic associations aid surface comprehension but may impair complex reasoning. The questions effectively separate basic retrieval (action/agent), structural processing (entity counting, nested dependency), and multi-step reasoning (causal reasoning, chain consequence), revealing precisely when models rely on meaning versus structure.

5.4 Reasoning models improve accuracy

Table 4 demonstrates that reasoning models substantially improve performance across all complexity levels. For Claude with thinking enabled at complexity level 6, performance jumps from 18.0% to 58.8% (+40.8 points) on plausible sentences and from 15.6% to 60.8% (+45.2 points) on implausible sentences for hard questions.

For Claude, reasoning yields larger gains on implausible sentences, reducing semantic dependency: at complexity level 6, implausible sentences gain 48.9 points on easy (vs. +13.7 for plausible) and 45.2 points on hard (vs. +40.8) questions. In contrast, Gemini and DeepSeek show larger gains on plausible sentences (Appendix Tables 12 and 13), indicating that the effect of reasoning on plausibility is model-specific. While reasoning improves accuracy dramatically, models still achieve only

Complexity Level	Model Type	Easy		Medium		Hard	
		Plaus	Implaus	Plaus	Implaus	Plaus	Implaus
1	Non-Think	99.2	95.4	95.7	95.0	44.5	51.2
	Thinking	**100.0**	**99.2**	**94.2**	**82.5**	**73.3**	**58.3**
2	Non-Think	88.4	81.8	74.4	76.1	47.3	50.7
	Thinking	**99.5**	**93.4**	**75.0**	**66.1**	**78.9**	**71.7**
3	Non-Think	84.1	66.9	60.3	61.8	38.3	29.4
	Thinking	**97.9**	**93.4**	**84.2**	**73.4**	**74.6**	**79.2**
4	Non-Think	87.0	53.0	64.4	59.9	23.8	24.2
	Thinking	**97.4**	**93.4**	**93.0**	**75.0**	**67.7**	**72.4**
5	Non-Think	83.9	51.0	57.7	55.6	23.0	20.5
	Thinking	**99.5**	**93.1**	**92.2**	**74.5**	**62.8**	**65.6**
6	Non-Think	80.9	43.0	55.7	57.4	18.0	15.6
	Thinking	**94.6**	**91.9**	**84.1**	**71.0**	**58.8**	**60.8**

Table 4: Claude: Thinking vs Non-Thinking performance across question difficulty levels on plausible (Plaus) and implausible (Implaus) subsets

$\sim$60% on hard questions at high complexity, suggesting that explicit reasoning helps track syntactic relationships but cannot fully overcome fundamental structural processing limitations.

5.5 Reasoning traces provide evidence of systematic processing failures

Analysis of reasoning traces exposes how thinking models fail when processing complex structures. The traces suggest several distinct failure patterns (see Appendix Tables 14-16):

Semantic interference in parsing: Models' reasoning shows they actively seek semantic coherence over syntactic accuracy. When Gemini processes "the bicycle... orbited cycled around," its trace shows it selecting "cycled around" because "that the bicycle... cycled around" makes semantic sense, completely missing that "orbited" is the syntactically correct verb (Appendix Table 14). This indicates that models don't just happen to fail on implausible sentences; their reasoning explicitly prioritizes meaning over structure.

Refusal when structure conflicts with world knowledge: Models handle implausible results in remarkably different ways. Gemini refuses to answer when parsing yields semantically odd relationships, stating "no one in this sentence directly reads rights to the mailman" even though the syntactic chain clearly indicates the teacher. In contrast, DeepSeek produces the correct answer but only after 3,498 tokens of circular reasoning, while Claude methodically parses the structure and answers correctly in 601 tokens (Appendix Table 16).

Self-induced errors through overthinking: Perhaps most revealing, models create errors in simple tasks through excessive reasoning. On basic entity

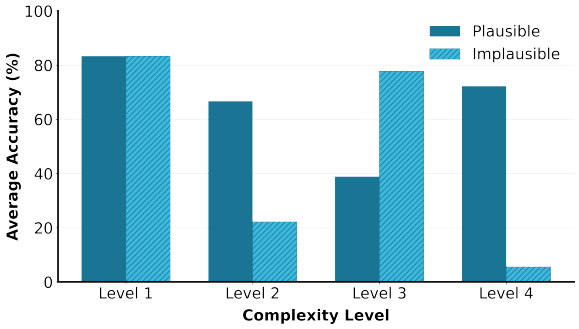

Figure 6: Human accuracy on plausible and implausible sentences across complexity levels 1-4.

counting, both DeepSeek and Claude initially identify the correct answer (4 entities) but then spiral into doubt: "I think I'm overcomplicating... Perhaps the answer is 3, excluding the case. I'm not sure." Meanwhile, Gemini shows overconfidence, declaring the task "Elementary" while still arriving at the wrong answer. This pattern shows that explicit reasoning can actively harm performance on straightforward tasks (Appendix Table 15).

5.6 Human performance varies unpredictably with complexity

Figure 6 reveals marked differences between human and model processing patterns. Unlike models, which show consistent semantic advantages that grow with complexity, humans exhibit variable effects: equal performance on both subsets at level 1, plausible advantage at levels 2 and 4, but implausible advantage at level 3.

Human accuracy also differs from model patterns in absolute performance. While models show steady degradation with complexity, human performance fluctuates: starting at 83% (level 1), dropping to 67% (level 2) and 39% (level 3), then rising to 72% (level 4) on plausible sentences. This non-monotonic pattern contrasts sharply with models' linear decline. This suggests humans may engage different processing strategies when semantic cues conflict with structural demands, potentially paying closer attention to syntax when meaning provides no shortcuts.

6 Related Works

Linguistic Complexity and Performance Degradation While recent work shows language models struggle with increasing task complexity (Qi et al., 2024; Kang et al., 2024; Hua et al., 2025), these studies focus on symbolic reasoning, mathematical operations (Dziri et al., 2023), or puzzle-solving (Estermann et al., 2024). Such tasks conflate domain-specific knowledge with structural processing, obscuring whether failures are linguistic or task-specific. Center-embedded sentences offer a pure test of recursive syntactic processing established through decades of psycholinguistic research (Chomsky and Miller, 1963; Gibson, 1998), scaling complexity through clause nesting without requiring external knowledge.

Center-Embedding in Computational Linguistics Center-embedded sentences have long served as a benchmark for syntactic processing, yet existing computational resources remain limited. Early work by Wilcox et al. (2019) tested hierarchical representations with 28 plausible/implausible sentence pairs, but only at single embedding depth. SyntaxGym (Gauthier et al., 2020) and Hu et al. (2020) reuse these single-level items. GECS (Carslaw et al., 2025) extracts naturally-occurring embeddings from web text but lacks systematic complexity scaling or plausibility control. While Hardt (2025) tested models on one to four embedding levels with noun-verb action questions, their sentences ignored plausibility. Studies comparing human and model processing (Wilcox et al., 2021; Lakretz et al., 2021, 2022) find both struggle with deeper embeddings but differently: humans maintain above-chance performance while LSTM accuracy collapses, and even strong Transformers fail at long-range dependencies. These limitations motivate our systematic manipulation of both complexity and plausibility with diverse question types.

Semantic Interference in Language Processing Extensive evidence shows both humans and models rely on semantic plausibility, though differently. Humans employ "good enough" processing strategies (Ferreira et al., 2002; Frances, 2024), while models show systematic biases toward training knowledge. Wilcox et al. (2019) found models assign higher surprisal to implausible pairings, correctly tracking dependencies but revealing semantic expectations. Studies using counterfactual contexts expose stronger biases: Fakepedia shows GPT-4 ignores false passages to give memorized answers 99% of the time (Monea et al., 2024), while entity substitution frameworks show models prefer parametric over contextual knowledge (Longpre et al., 2021; Neeman et al., 2022). Additionally, recent work demonstrates plausibility's diagnostic power beyond syntax: Palta et al. (2024) use plau-

sibility ratings to identify quality issues in existing commonsense benchmarks where semantic expectations conflict with gold labels. However, these test binary memory-context conflicts. Our work identifies the *continuous* transition: by scaling syntactic complexity while controlling plausibility, we pinpoint when models shift from structural analysis to semantic shortcuts.

7 Conclusion

We introduced CENTERBENCH, a dataset of 9,720 test instances varying complexity, reasoning demands, and semantic plausibility. Models show consistent performance degradation as complexity increases, while also increasingly relying on meaning over structure, with performance gaps reaching 25+ percentage points. Interestingly, semantic plausibility backfires on complex reasoning, where familiar patterns mislead models when structural analysis is required. Reasoning models improve accuracy but their traces provide evidence of systematic failures including prioritizing semantic coherence, refusing implausible answers, and overthinking simple tasks. These patterns contrast sharply with humans' inconsistent semantic effects. By revealing precisely when and how models abandon structural analysis for semantic shortcuts, CENTERBENCH enables informed decisions about model deployment in domains requiring genuine syntactic understanding.

Limitations

We structure our limitations section as arguments and counterarguments, inspired by Balepur et al. (2025):

The semantic manipulation targets more than plausibility: Our implausible sentences simultaneously manipulate collocational strength and verb-noun associations, this is intentional, not confounding. Semantic plausibility cannot be isolated from these factors because they collectively constitute what makes language plausible. As demonstrated by Lapata et al. (1999), co-occurrence frequency (which captures collocational relationships) is the strongest predictor of plausibility judgments. By breaking familiar associations (doctors prescribing → doctors delivering mail), we create the semantic disruption needed to test when models abandon syntax for semantics. We maintain domain consistency within sentences (all vehicles, all professions) and keep syntactic structure identical across matched pairs, ensuring that performance differences reflect reliance on semantic cues.

The semantic manipulation is too artificial: Creating sentences where "doctors deliver mail" and "mailmen prescribe medicine" might seem contrived, but these actions remain perfectly possible—they simply violate our world knowledge and expectations (Wu et al., 2024). By preserving grammatical correctness while disrupting typical semantic associations, we force models to choose between structural and semantic processing. The fact that this manipulation consistently affects performance reveals how deeply models depend on semantic plausibility, a finding with direct implications for domains where semantic cues may be unreliable, such as processing figurative language, analyzing texts from different time periods, or handling adversarial inputs (Zhang et al., 2024; Madhusudan et al., 2025; Wallace et al., 2019).

This is just another syntax benchmark: While center-embedded sentences might seem like an artificial linguistic construction, they represent a fundamental test of recursive structural processing that appears across natural language. Unlike toy puzzles designed solely for benchmarking, these structures have been studied for decades in psycholinguistics precisely because they reveal how humans and systems handle hierarchical dependencies (Miller and Chomsky, 1963; Hudson, 1996). Our systematic manipulation of semantic plausibility transforms this classical paradigm into a diagnostic that can identify when any language system shifts from structural analysis to pattern matching.

You can't definitively prove models are or aren't "parsing": We acknowledge that our comprehension questions don't directly test syntactic parsing in the formal sense. However, this is precisely why we designed matched sentence pairs: if models were truly building syntactic representations, they should perform identically on structures that differ only in semantic plausibility (Mueller et al., 2024). The systematic performance gaps we observe provide strong evidence that models rely on semantic associations rather than structural analysis, regardless of their internal representations.

Real language use doesn't involve such complex embeddings: Corpus studies confirm that deeply nested structures rarely appear in natural text (Karlsson, 2007). But this is exactly why they're valuable for evaluation: they test whether

models have learned generalizable principles of syntactic structure or merely memorized common patterns. Just as stress tests in engineering reveal failure modes through extreme conditions, our complexity scaling identifies the precise point where models abandon whatever structural processing they might possess for semantic shortcuts.

Why not use the same entities doing implausible things? One might wonder why we don't simply keep the same entities from plausible sentences and have them perform implausible actions (*cats barking, dogs meowing*). While this alternative seems straightforward, it's surprisingly difficult to execute. Consider the constraints: we need actions that *clearly* violate semantic expectations, maintaining grammatical requirements (correct transitive/intransitive patterns), avoiding temporal violations, and working across 2-7 entity chains. Finding sufficient verbs meeting these criteria for each entity becomes intractable. Our approach—circular verb swapping across animals, occupations, and vehicles—provides clear plausibility violations (skunks roaring, mailmen prescribing medicine, bicycles orbiting) while maintaining all syntactic constraints. These actions remain physically possible but violate our world knowledge and expectations, creating the semantic disruption we need. The entities are merely vehicles for testing whether models maintain syntactic parsing when semantic cues are removed. What matters is the within-model performance gap on identical syntactic structures, which directly quantifies reliance on semantic shortcuts over structural analysis.

Humans only went to complexity level 4: Indeed, human participants showed increasing difficulty with our sentences, and we limited testing to level 4 for practical reasons. However, this actually strengthens our findings: even at levels where humans can still process these structures (albeit with effort), they show qualitatively different patterns from models. The inconsistent semantic effects in humans align with research showing that human sentence processing often relies on "good enough" representations rather than complete structural analysis (Ferreira et al., 2002; Frances, 2024), contrasting with models' systematic biases that consistently favor semantic plausibility.

References

Anthropic. 2025. Claude 3.7 sonnet. https://www.anthropic.com/news/claude-3-7-sonnet. [Accessed 15-07-2025].

Nishant Balepur, Rachel Rudinger, and Jordan Lee Boyd-Graber. 2025. Which of these best describes multiple choice evaluation with LLMs? a) forced B) flawed C) fixable D) all of the above. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3394–3418, Vienna, Austria. Association for Computational Linguistics.

Nathaniel Berger, Stefan Riezler, Sebastian Ebert, and Artem Sokolov. 2021. Don't search for a search method — simple heuristics suffice for adversarial text attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8216–8224, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Iona Carslaw, Sivan Milton, Nicolas Navarre, Ciyang Qing, and Wataru Uegaki. 2025. Automatic extraction of clausal embedding based on large-scale english text data. *Preprint*, arXiv:2506.14064.

Noam Chomsky and George A. Miller. 1963. Introduction to the formal analysis of natural languages. *Journal of Symbolic Logic*, 33(2):299–300.

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.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and fate: Limits of transformers on compositionality. *Preprint*, arXiv:2305.18654.

Benjamin Estermann, Luca A. Lanzendörfer, Yannick Niedermayr, and Roger Wattenhofer. 2024. Puzzles: A benchmark for neural algorithmic reasoning. *Preprint*, arXiv:2407.00401.

Fernanda Ferreira, Karl GD Bailey, and Vittoria Ferraro. 2002. Good-enough representations in language comprehension. *Current directions in psychological science*, 11(1):11–15.

Candice Frances. 2024. Good enough processing: what have we learned in the 20 years since ferreira et al.(2002)? *Frontiers in Psychology*, 15:1323700.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76, Online. Association for Computational Linguistics.

E Gibson. 1998. Linguistic complexity: locality of syntactic dependencies. *Cognition*, 68(1):1–76.

Daniel Hardt. 2025. Sparks of pure competence in llms: the case of syntactic center embedding in english. *Society for Computation in Linguistics*, 8(1).

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, Adriane Boyd, and 1 others. 2020. spacy: Industrial-strength natural language processing in python.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Wenyue Hua, Tyler Wong, Fei Sun, Liangming Pan, Adam Jardine, and William Yang Wang. 2025. InductionBench: LLMs fail in the simplest complexity class. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26526–26546, Vienna, Austria. Association for Computational Linguistics.

Richard Hudson. 1996. The difficulty of (so-called) self-embedded structures. *Work. Pap. Linguist*, 8:283–314.

Liwei Kang, Zirui Zhao, David Hsu, and Wee Sun Lee. 2024. On the empirical complexity of reasoning and planning in LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2897–2936, Miami, Florida, USA. Association for Computational Linguistics.

Fred Karlsson. 2007. Constraints on multiple center-embedding of clauses. *Journal of Linguistics*, 43(2):365–392.

Koray Kavukcuoglu. 2025. Gemini 2.5: Our most intelligent ai model.

Yair Lakretz, Théo Desbordes, Dieuwke Hupkes, and Stanislas Dehaene. 2022. Can transformers process recursive nested constructions, like humans? In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3226–3232, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yair Lakretz, Dieuwke Hupkes, Alessandra Vergallito, Marco Marelli, Marco Baroni, and Stanislas Dehaene. 2021. Mechanisms for handling nested dependencies in neural-network language models and humans. *Cognition*, 213:104699.

Maria Lapata, Scott McDonald, and Frank Keller. 1999. Determinants of adjective-noun plausibility. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 30–36, Bergen, Norway. Association for Computational Linguistics.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sangmitra Madhusudan, Robert Morabito, Skye Reid, Nikta Gohari Sadr, and Ali Emami. 2025. Fine-tuned LLMs are "time capsules" for tracking societal bias through books. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2329–2358, Albuquerque, New Mexico. Association for Computational Linguistics.

George A. Miller and Noam Chomsky. 1963. Finitary models of language users. In D. Luce, editor, *Handbook of Mathematical Psychology*, pages 2–419. John Wiley & Sons.

Giovanni Monea, Maxime Peyrard, Martin Josifoski, Vishrav Chaudhary, Jason Eisner, Emre Kiciman, Hamid Palangi, Barun Patra, and Robert West. 2024. A glitch in the matrix? locating and detecting language model grounding with fakepedia. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6828–6844, Bangkok, Thailand. Association for Computational Linguistics.

John Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020. Reevaluating adversarial examples in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3829–3839, Online. Association for Computational Linguistics.

Aaron Mueller, Albert Webson, Jackson Petty, and Tal Linzen. 2024. In-context learning generalizes, but not always robustly: The case of syntax. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume*

1: Long Papers), pages 4761–4779, Mexico City, Mexico. Association for Computational Linguistics.

Ella Neeman, Roee Aharoni, Or Honovich, Leshem Choshen, Idan Szpektor, and Omri Abend. 2022. Disentqa: Disentangling parametric and contextual knowledge with counterfactual question answering. *Preprint*, arXiv:2211.05655.

Shramay Palta, Nishant Balepur, Peter Rankel, Sarah Wiegreffe, Marine Carpuat, and Rachel Rudinger. 2024. Plausibly problematic questions in multiple-choice benchmarks for commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3451–3473, Miami, Florida, USA. Association for Computational Linguistics.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, Michael Choi, Anish Agrawal, Arnav Chopra, Adam Khoja, Ryan Kim, Richard Ren, Jason Hausenloy, Oliver Zhang, and Mantas Mazeika et al. 2025. Humanity's last exam. *Preprint*, arXiv:2501.14249.

Zhenting Qi, Hongyin Luo, Xuliang Huang, Zhuokai Zhao, Yibo Jiang, Xiangjun Fan, Himabindu Lakkaraju, and James Glass. 2024. Quantifying generalization complexity for large language models. *Preprint*, arXiv:2410.01769.

Valentin Six, Evan Dufraisse, and Gaël de Chalendar. 2025. Decompositional reasoning for graph retrieval with large language models. *Preprint*, arXiv:2506.13380.

James P Van Overschelde, Katherine A Rawson, and John Dunlosky. 2004. Category norms: An updated and expanded version of the battig and montague (1969) norms. *Journal of Memory and Language*, 50(3):289–335.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *Transactions of the Association for Computational Linguistics*, 7:387–401.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Preprint*, arXiv:2002.10957.

Ethan Wilcox, Roger Levy, and Richard Futrell. 2019. Hierarchical representation in neural language models: Suppression and recovery of expectations. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 181–190, Florence, Italy. Association for Computational Linguistics.

Ethan Wilcox, Pranali Vani, and Roger Levy. 2021. A targeted assessment of incremental processing in neural language models and humans. In *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)*, pages 939–952, Online. Association for Computational Linguistics.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. 2024. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1819–1862, Mexico City, Mexico. Association for Computational Linguistics.

Linhao Zhang, Jintao Liu, Li Jin, Hao Wang, Kaiwen Wei, and Guangluan Xu. 2024. GOME: Grounding-based metaphor binding with conceptual elaboration for figurative language illustration. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18500–18510, Miami, Florida, USA. Association for Computational Linguistics.

A Appendix

A.1 Dataset

A.1.1 Nouns and Verbs in CENTERBENCH

Category	Subcategory	Included Nouns	Excluded Nouns
Animals	Birds	Eagle, Sparrow, Crow, Pigeon, Parrot, Chicken, Duck, Owl, Goose, Vulture	Blue Jay, Robin, Cardinal, Hawk, Hummingbird, Dove, Finch, Raven, Woodpecker, Wren, Ostrich, Parakeet, Seagull, Flamingo, Mockingbird, Penguin, Falcon, Black Bird, Oriole, Swan, Canary, Turkey, Swallow, Chickadee, Crane, Emu, Grackle, Heron, Jaybird, Peacock, Pelican, Red Bird, Starling, Stork
	Fish	Salmon, Tuna, Goldfish, Trout, Shark, Whale, Piranha, Starfish	Catfish, Bass, Cod, Carp, Perch, Tilapia, Flounder, Sword Fish, Clown Fish, Blue Gill, Pike, Crappie, Guppy, Minnow, Halibut, Grouper, Snapper, Sun Fish, Koi, Puffer Fish, Angel Fish, Barracuda, Blowfish, Blue, Dolphin, Mackerel, Marlin, Pollock, Sturgeon, Walleye
	Four-Footed Animals	Dog, Cat, Horse, Lion, Cow, Tiger, Bear, Elephant, Deer, Pig, Giraffe, Mouse, Sheep, Goat, Rat, Wolf, Zebra, Donkey, Rabbit, Raccoon, Squirrel, Coyote, Cougar, Moose, Cheetah, Rhinoceros, Fox, Mule, Hippopotamus, Beaver, Camel, Ferret, Frog, Jaguar, Lamb, Leopard, Lizard, Llama, Skunk	Opossum, Hamster, Elk
	Snakes	Rattlesnake, Cobra, Python, Anaconda, Viper, Black Mamba	Garden, Boa Constrictor, Copperhead, Water Moccasin, Black, Asp, Cottonmouth, Coral, King, Rat, Diamondback, Green, Corn Snake, Grass, Mamba, Racer
	Insects	Ant, Fly, Bee, Spider, Beetle, Mosquito, Cockroach, Wasp, Lady Bug, Butterfly, Grasshopper, Moth, Gnat, Cricket, Caterpillar, Worm, Centipede, Termite, Praying Mantis, Bug, Dragonfly	Flea, Stink Bug, Hornet, Millipede, Rolly Polly, Tick, Yellow Jacket
Occupations		Doctor, Lawyer, Teacher, Nurse, Police Officer, Firefighter, Accountant, Dentist, Engineer, Plumber, Carpenter, Salesperson, Secretary, Cashier, Mechanic, Professor, Electrician, Chef, Scientist, Clerk, Banker, Actor, Truck Driver, Mailman, Artist, Athlete, Attorney, Bus Driver, CEO, Construction Worker, Garbage Man, Janitor, Judge, Manager, Musician, Pilot, Politician, Programmer, Surgeon, Veterinarian, Waiter, Writer	None
Vehicles		Car, Truck, Motorcycle, Airplane, Bicycle, Bus, SUV, Boat, Train, Van, Jeep, Tank, Sedan, Tractor, Scooter, RV, Taxi, ATV, Ambulance, Convertible, Golf Cart, Helicopter, Pickup Truck, Ship, Sled, Snowmobile, Sports Car	Semi, Minivan, Station Wagon, Wagon

Table 5: Included and excluded nouns from Van Overschelde et al. (2004) across categories

Category	Nouns
Animals	Eagle, Sparrow, Crow, Pigeon, Parrot, Chicken, Duck, Owl, Goose, Vulture, Salmon, Tuna, Goldfish, Trout, Shark, Whale, Piranha, Starfish, Dog, Cat, Horse, Lion, Cow, Tiger, Bear, Elephant, Deer, Pig, Giraffe, Mouse, Sheep, Goat, Rat, Wolf, Zebra, Donkey, Rabbit, Raccoon, Squirrel, Coyote, Cougar, Moose, Cheetah, Rhinoceros, Fox, Mule, Hippopotamus, Beaver, Camel, Ferret, Frog, Jaguar, Lamb, Leopard, Lizard, Llama, Skunk, Rattlesnake, Cobra, Python, Anaconda, Viper, Black Mamba, Ant, Fly, Bee, Spider, Beetle, Mosquito, Cockroach, Wasp, Ladybug, Butterfly, Grasshopper, Moth, Gnat, Cricket, Caterpillar, Worm, Centipede, Termite, Praying Mantis, Bug, Dragonfly, Pelican
Occupations	Doctor, Lawyer, Teacher, Nurse, Police Officer, Firefighter, Accountant, Dentist, Engineer, Plumber, Carpenter, Salesperson, Secretary, Cashier, Mechanic, Professor, Electrician, Chef, Scientist, Clerk, Banker, Actor, Truck Driver, Mailman, Artist, Athlete, Attorney, Bus Driver, CEO, Construction Worker, Garbage Man, Janitor, Judge, Manager, Musician, Pilot, Politician, Programmer, Surgeon, Veterinarian, Waiter, Writer
Vehicles	Car, Truck, Motorcycle, Airplane, Bicycle, Bus, SUV, Boat, Train, Van, Jeep, Tank, Sedan, Tractor, Scooter, RV, Taxi, ATV, Ambulance, Convertible, Golf Cart, Helicopter, Pickup Truck, Ship, Sled, Snowmobile, Sports Car
GPT-4 Extensions	Lioness, Gazelle, Calf, Cub, Duckling, Kitten, Puppy, Bird, Snake, Mongoose, Child, Baby, Infant, Parent, Toddler, Boy, Girl, Man, Woman, Teenager, Kid, Adult, Youth, Journalist, Reporter, Director, Editor, Photographer, Detective, Investigator, Researcher, Student, Supervisor, Inspector, Principal, Hunter, Gatherer, Villager, Resident, Tourist, Passenger, Driver, Maid, Host, Guest, Predator, Prey, Scavenger, Beast, Creature, Animal, Minivan, Submarine, Spaceship, Drone, Hound, Dove, Flagged, Architect, Mayor, Dean, Cyclist, Crab, Seagull, Paper, Publisher, Fish, Letter, Thief, Chief, Hawk, Zookeeper, Patient, Suitcase, Porter, Park Ranger, Bat, Ranger, Neighbor, Heron, Hiker, Guide, Owner, Postman, Policeman, Homeowner, Vet, Octopus, Diver, Captain, Farmer, Sheriff, Nanny, Landlord, Forester, Jogger, Coach, Wife, Mother, Security Guard, Grandfather, Pet Owner, Tenant, Trapper, Conservationist, Gardener, Cheese, Shopkeeper, Superintendent, Animal Rights Activist, Falconer, Babysitter, Grandmother, Game Warden, Zoologist, Biologist, Father, Pest Controller, Flower, Nut, Naturalist, Intern, Skateboard, Seal, Dolphin, Fisherman, Coast Guard, Note, Investor, President, Pearl, Oyster, Chip, Message, Chancellor, Leaf, Chameleon, Apple, Storm, Citizen, Baby Bird

Table 6: Complete list of nouns in the plausible dataset by category

Category	Noun	Verbs (Transitive / Intransitive)
Animals	parrot	mimicked / chattered
	horse	neighed at, galloped toward / neighed, galloped
	elephant	trumpeted at, charged at / trumpeted, stomped
	bee	buzzed at, stung / buzzed, swarmed
	cobra	hissed at, reared at / hissed, reared, coiled
Professions	doctor	prescribed medicine to, examined / prescribed medicines, diagnosed
	lawyer	cross-examined, defended in court / argued, objected
	police officer	arrested, read rights to / patrolled, responded
	firefighter	rescued, carried out / extinguished fire, rescued
	judge	sentenced, ruled against / presided, gaveled
Vehicles	truck	jackknifed into, towed / jackknifed
	motorcycle	lane-split past, wheelied at / wheelied
	airplane	air-dropped on, strafed / taxied
	bicycle	pedaled past, cycled around / pedaled
	ambulance	blared the siren at / blared the siren

Table 7: Sample nouns and associated verbs by category generated by *claude-sonnet-4-20250514* for implausible sentence generation

A.1.2 Sentence Generation - Plausible

Complexity Level Specifications The system dynamically generated prompts based on complexity level, where each level follows a systematic pattern of increasing structural complexity. Each complexity level is defined by the number of entities, verbs, and embedded "that" clauses:

Complexity Level 1 (2 *entities*, 2 *verbs*, 1 "that" clause)

- *structure:* `[A] that [B] [verb1] [verb2]`
- *semantic_rule:* B performs verb1 TO A, then A performs verb2
- *example:* "The mouse that the cat chased escaped."
- *flow:* CAT chased MOUSE → MOUSE escaped

Complexity Level 2 (3 *entities*, 3 *verbs*, 2 "that" clauses)

- *structure:* `[A] that [B] that [C] [verb1] [verb2] [verb3]`
- *semantic_rule:* C performs verb1 TO B, B performs verb2 TO A, A performs verb3
- *example:* "The fly that the spider that the bird saw stalked buzzed."
- *flow:* BIRD saw SPIDER → SPIDER stalked FLY → FLY buzzed

Complexity Level 3 (4 *entities*, 4 *verbs*, 3 "that" clauses)

- *structure:* `[A] that [B] that [C] that [D] [verb1] [verb2] [verb3] [verb4]`
- *semantic_rule:* D performs verb1 TO C, C performs verb2 TO B, B performs verb3 TO A, A performs verb4
- *example:* "The worm that the bird that the cat that the dog chased saw ate died."
- *flow:* DOG chased CAT → CAT saw BIRD → BIRD ate WORM → WORM died

Complexity Level 4 (5 *entities*, 5 *verbs*, 4 "that" clauses)

- *structure:* `[A] that [B] that [C] that [D] that [E] [verb1] [verb2] [verb3] [verb4] [verb5]`
- *semantic_rule:* E performs verb1 TO D, D performs verb2 TO C, C performs verb3 TO B, B performs verb4 TO A, A performs verb5
- *example:* "The mouse that the cat that the dog that the owner that the neighbor called trained chased caught squeaked."
- *flow:* NEIGHBOR called OWNER → OWNER trained DOG → DOG chased CAT → CAT caught MOUSE → MOUSE squeaked

Complexity Level 5 (6 *entities*, 6 *verbs*, 5 "that" clauses)

- *structure:* `[A] that [B] that [C] that [D] that [E] that [F] [verb1] [verb2] [verb3] [verb4] [verb5] [verb6]`
- *semantic_rule:* F performs verb1 TO E, E performs verb2 TO D, D performs verb3 TO C, C performs verb4 TO B, B performs verb5 TO A, A performs verb6
- *example:* "The ant that the spider that the lizard that the snake that the hawk that the hunter saw spotted followed grabbed saw crawled."
- *flow:* HUNTER saw HAWK → HAWK spotted SNAKE → SNAKE followed LIZARD → LIZARD grabbed SPIDER → SPIDER saw ANT → ANT crawled

Complexity Level 6 (7 *entities*, 7 *verbs*, 6 "that" clauses)

- *structure:* `[A] that [B] that [C] that [D] that [E] that [F] that [G] [verb1] [verb2] [verb3] [verb4] [verb5] [verb6] [verb7]`
- *semantic_rule:* G performs verb1 TO F, F performs verb2 TO E, E performs verb3 TO D, D performs verb4 TO C, C performs verb5 TO B, B performs verb6 TO A, A performs verb7
- *example:* "The crumb that the ant that the spider that the lizard that the snake that the hawk that the eagle observed followed chased startled carried dropped rolled."

- *flow:* EAGLE observed HAWK → HAWK followed SNAKE → SNAKE chased LIZARD → LIZARD startled SPIDER → SPIDER carried ANT → ANT dropped CRUMB → CRUMB rolled

System Prompt We employed *GPT-4-0613* with temperature 0.7. The system prompt used was:

You are a linguistics expert specializing in center-embedded sentences.

COMPLEXITY LEVEL {complexity_level} SPECIFICATIONS

- Required entities: {entities}
- Required verbs: {verbs}
- Required "that" clauses: {complexity_level}
- Embedding depth: {complexity_level} levels

GRAMMATICAL STRUCTURE
Pattern: {structure}

SEMANTIC FLOW RULES (CRITICAL)
Core Rule: {semantic_rule}
Example: {example}
Action Flow: {flow}

TEMPORAL CONSISTENCY RULES (MANDATORY)

1. Actions must follow logical temporal sequence
2. Dead entities CANNOT perform subsequent actions
3. Caught/trapped/seized entities CANNOT act on other entities
4. Eaten entities CANNOT perform actions after being consumed
5. Actions must respect cause-and-effect relationships
6. No temporal paradoxes or impossibilities allowed

STRUCTURAL REQUIREMENTS

1. Use "that" as relative pronoun for ALL embeddings
2. Verbs appear in REVERSE order of entity introduction
3. Last entity introduced performs FIRST action
4. First entity performs FINAL action (must be intransitive)
5. Each "that" clause introduces exactly ONE entity
6. Actions flow from innermost clause outward

SEMANTIC PLAUSIBILITY CONSTRAINTS
Predator-Prey Relationships:

- Must reflect realistic natural hierarchies
- Size/strength differences must be logical
- Hunting behaviors must be species-appropriate

Professional Relationships:

- Authority structures must be realistic
- Professional interactions must be plausible
- Skills must match occupations

Physical Capabilities:

- Actions must match entity capabilities
- Environmental constraints must be respected
- Biological limitations must be observed

MEMORY AID: NESTED ACTION PRINCIPLE
Think of Russian dolls opening from inside out:

- Innermost doll (last entity) acts first
- Each outer doll (entity) acts on the result
- Outermost doll (first entity) performs final action
- Each action must be temporally possible given previous actions

QUALITY REQUIREMENTS

- Sentences must be grammatically perfect
- Semantic relationships must be crystal clear
- No ambiguous temporal references
- All actions must be logically sequenced
- Maintain subject-verb agreement throughout

User Prompt The user prompt used was:

Generate 30 unique center-embedded sentences at complexity level {complexity_level}.

STRICT REQUIREMENTS

- Exactly {entities} different entities
- Exactly {verbs} verbs
- Exactly {complexity_level} "that" clauses
- Perfect temporal consistency - NO temporal violations
- Semantically plausible relationships
- Grammatically correct structure

ENTITY CATEGORIES

Animals:
eagle, sparrow, crow, pigeon, parrot, chicken, duck, owl, goose, vulture, salmon, tuna, goldfish, trout, shark, whale, piranha, starfish, dog, cat, horse, lion, cow, tiger, bear, elephant, deer, pig, giraffe, mouse, sheep, goat, rat, wolf, zebra, donkey, rabbit, raccoon, squirrel, coyote, cougar, moose, cheetah, rhinoceros, fox, mule, hippopotamus, beaver, camel, ferret, frog, jaguar, lamb, leopard, lizard, llama, skunk, rattlesnake, cobra, python, anaconda, viper, black mamba, ant, fly, bee, spider, beetle, mosquito, cockroach, wasp, ladybug, butterfly, grasshopper, moth, gnat, cricket, caterpillar, worm, centipede, termite, praying mantis, bug, dragonfly

People (Occupations):
doctor, lawyer, teacher, nurse, police officer, firefighter, accountant, dentist, engineer, plumber, carpenter, salesperson, secretary, cashier, mechanic, professor, electrician, chef, scientist, clerk, banker, actor, truck driver, mailman, artist, athlete, attorney, bus driver, CEO, construction worker, garbage man, janitor, judge, manager, musician, pilot, politician, programmer, surgeon, veterinarian, waiter, writer

Vehicles:
car, truck, motorcycle, airplane, bicycle, bus, SUV, boat, train, van, jeep, tank, sedan, tractor, scooter, RV, taxi, ATV, ambulance, convertible, golf cart, helicopter, pickup truck, ship, sled, snowmobile, sports car

CRITICAL REMINDERS

- After an entity is caught/killed/eaten, it CANNOT perform actions
- Predator-prey relationships must be biologically accurate
- Professional hierarchies must be realistic
- Actions must follow logical temporal sequence
- Each sentence must tell a coherent, plausible story

OUTPUT FORMAT
Number each sentence 1-30, one per line:
1. [sentence]
2. [sentence]
...
30. [sentence]

A.1.3 Sentence Generation - Implausible

Algorithm Implementation

The following algorithm implements the circular verb swapping procedure described in Section 2.1.2:

Algorithm 1 Circular Verb Swapping for Implausible Sentence Generation

Require: Complexity level c, Verb data V, Used combinations U
Ensure: Semantically implausible sentence S

```
num_entities ← c + 1
domains ← [animals, people, vehicles]
for attempt = 1 to 1000 do
    Category Selection:
    selected_domain ← RandomChoice(domains)
    all_entities ← GetAllEntities(V[selected_domain])
    Noun Sampling:
    selected_entities ← RandomSample(all_entities, num_entities)
    Circular Verb Assignment:
    assigned_verbs ← []
    for i = 0 to num_entities − 1 do
        current_entity ← selected_entities[i]
        next_entity ← selected_entities[(i + 1) mod num_entities]
        next_entity_verbs ← GetEntityVerbs(V[selected_domain], next_entity)
        if i = 0 then
            verb_type ← intransitive {First entity gets intransitive verb}
        else
            verb_type ← transitive {All others get transitive verbs}
        end if
        available_verbs ← next_entity_verbs[verb_type]
        selected_verb ← RandomChoice(available_verbs)
        assigned_verbs.append(selected_verb)
    end for
    combination_key ← (selected_domain, selected_entities, assigned_verbs)
    if combination_key ∉ U then
        U.add(combination_key)
        Sentence Construction:
        S ← ConstructSentence(selected_entities, assigned_verbs)
        return S
    end if
end for
raise Exception("Maximum attempts exceeded")
```

Sentence Construction Templates The ConstructSentence function applies the following complexity-specific templates:

Complexity 1:

```
The {entity[0]} that the {entity[1]} {verb[1]} {verb[0]}.
```

Complexity 2+:

```
The {entity[0]} that the {entity[1]} that ... that the {entity[n]} {verb[n]}
{verb[n-1]} ... {verb[0]}.
```

Supporting Functions

- GETALLENTITIES($domain_data$): Extracts all entity names from the specified domain, handling both flat and nested data structures
- GETENTITYVERBS($domain_data, entity$): Retrieves the transitive and intransitive verb lists associated with a specific entity
- RANDOMSAMPLE($entities, n$): Selects n unique entities without replacement
- RANDOMCHOICE($verbs$): Selects one verb uniformly at random from the available verb list

Uniqueness Control The algorithm maintains a global set U of used combinations across all complexity levels, where each combination is uniquely identified by the tuple:

$$combination_key = (domain, entities, assigned_verbs)$$

This ensures no duplicate sentences are generated while allowing for maximum diversity in entity-verb pairings within the semantic implausibility constraints.

A.1.4 Sentence Validation

Sentence and Validation Assessment
The cat that the dog that the boy adopted chased meowed. **Temporal:** Valid **Semantic:** Valid **Syntactic:** Valid
The cat that the dog that the boy caught *chased* *meowed.* **Temporal:** Invalid (dog chases after being caught) **Semantic:** Valid **Syntactic:** Valid **Fix:** Replace *chased* with *saw* **Corrected:** *The cat that the dog that the boy caught* *saw* *meowed.*
The cat that the dog that the boy chased *adopted* *meowed.* **Temporal:** Valid **Semantic:** Invalid (dog adopting cat is implausible) **Syntactic:** Valid **Fix:** Replace *adopted* with *saw* **Corrected:** *The cat that the dog that the boy adopted* *saw* *meowed.*
The cat that the dog that the boy *adopted* *meowed*. **Temporal:** Valid **Semantic:** Valid **Syntactic:** Invalid (only two verbs) **Fix:** Add temporally and semantically valid verb (e.g., *chased*) **Corrected:** *The cat that the dog that the boy adopted* *chased* *meowed.*

Table 8: Validation process example for the plausible subset at complexity level 2. Each sentence is assessed for temporal, semantic, and syntactic validity. Incorrect words are in *red* with corrections shown in *green* .

A.1.5 CENTERBENCH Examples

Level	Subset Type	Sentences
1	Plausible	The cat that the mouse evaded pounced. NP_1: *cat,* NP_2: *mouse,* V_2: *evaded,* V_1: *pounced*
1	Implausible	The horse that the elephant neighed at trumpeted. NP_1: *horse,* NP_2: *elephant,* V_2: *neighed at,* V_1: *trumpeted*
2	Plausible	The bicycle that the car that the truck hit bumped fell over. NP_1: *bicycle,* NP_2: *car,* NP_3: *truck,* V_3: *hit,* V_2: *bumped,* V_1: *fell over*
2	Implausible	The bicycle that the airplane that the train pedaled past whistled at taxied. NP_1: *bicycle,* NP_2: *airplane,* NP_3: *train,* V_3: *pedaled past,* V_2: *whistled at,* V_1: *taxied*
3	Plausible	The fly that the spider that the bird that the snake hissed at saw caught died. NP_1: *fly,* NP_2: *spider,* NP_3: *bird,* NP_4: *snake,* V_4: *hissed at,* V_3: *saw,* V_2: *caught,* V_1: *died*
3	Implausible	The skunk that the lion that the bat that the elephant sprayed at trumpeted at echolocated roared. NP_1: *skunk,* NP_2: *lion,* NP_3: *bat,* NP_4: *elephant,* V_4: *sprayed at,* V_3: *trumpeted at,* V_2: *echolocated,* V_1: *roared*
4	Plausible	The beetle that the bird that the cat that the toddler that the mother watched scared caught saw crawled. NP_1: *beetle,* NP_2: *bird,* NP_3: *cat,* NP_4: *toddler,* NP_5: *mother,* V_5: *watched,* V_4: *scared,* V_3: *caught,* V_2: *saw,* V_1: *crawled*
4	Implausible	The police officer that the teacher that the mailman that the nurse that the photographer handcuffed photographed bandaged delivered mail to lectured. NP_1: *police officer,* NP_2: *teacher,* NP_3: *mailman,* NP_4: *nurse,* NP_5: *photographer,* V_5: *handcuffed,* V_4: *photographed,* V_3: *bandaged,* V_2: *delivered mail to,* V_1: *lectured*
5	Plausible	The spider that the frog that the snake that the hawk that the falconer that the wildlife photographer snapped released saw followed ate died. NP_1: *spider,* NP_2: *frog,* NP_3: *snake,* NP_4: *hawk,* NP_5: *falconer,* NP_6: *wildlife photographer,* V_6: *snapped,* V_5: *released,* V_4: *saw,* V_3: *followed,* V_2: *ate,* V_1: *died*
5	Implausible	The lawyer that the photographer that the firefighter that the judge that the waiter that the police officer subpoenaed arrested took food orders from ruled against rescued photographed. NP_1: *lawyer,* NP_2: *photographer,* NP_3: *firefighter,* NP_4: *judge,* NP_5: *waiter,* NP_6: *police officer,* V_6: *subpoenaed,* V_5: *arrested,* V_4: *took food orders from,* V_3: *ruled against,* V_2: *rescued,* V_1: *photographed*
6	Plausible	The apple that the worm that the bird that the squirrel that the cat that the dog that the child saw barked at chased startled pecked nudged rolled. NP_1: *apple,* NP_2: *worm,* NP_3: *bird,* NP_4: *squirrel,* NP_5: *cat,* NP_6: *dog,* NP_7: *child,* V_7: *saw,* V_6: *barked at,* V_5: *chased,* V_4: *startled,* V_3: *pecked,* V_2: *nudged,* V_1: *rolled*
6	Implausible	The starfish that the horse that the lizard that the seagull that the vulture that the duck that the bee crawled toward stung paddled toward scavenged from dive-bombed basked toward neighed. NP_1: *starfish,* NP_2: *horse,* NP_3: *lizard,* NP_4: *seagull,* NP_5: *vulture,* NP_6: *duck,* NP_7: *bee,* V_7: *crawled toward,* V_6: *stung,* V_5: *paddled toward,* V_4: *scavenged from,* V_3: *dive-bombed,* V_2: *basked toward,* V_1: *neighed*

Table 9: Color-coded and annotated noun-verb pairs for plausible and implausible sentences across complexity levels 1-6. Each sentence is broken down to show its structural composition.

A.1.6 Question and Answer Generation

Algorithm This section illustrates the four-step question and answer generation algorithm using the example sentence *"The dog that the mailman startled barked."*

1. Noun identification: Extracts nouns from the sentence.
   - *Output*: ["dog", "mailman"]
2. Structural parsing: Maps verbs to their subjects and objects by analyzing the reversed verb order.
   - From "startled barked": "barked" → dog (intransitive), "startled" → mailman's action on dog (transitive)
   - *Output:*
     - subject: "dog", action: "bark", object: none
     - subject: "mailman", action: "startle", object: "dog"
3. Verb processing: Converts verbs to different forms needed for question and answer generation.
   - "startled" → "startle" (base), "startled" (participle), "startling" (gerund)
4. Template instantiation: Generates questions and answers using the parsed relationships and author established templates.
   - Example question for mailman: "What did the mailman do?"
   - Example answer: "startled the dog"

Question Type	**Question Template (General)**	**Question Template (Modified)**	**Answer Template (General)**	**Answer Template (Modified)**
Action Performed	What did the [entity] do?	-	[verb-ed + object]	Outermost entity: [verb-ed]
Agent Identification	Who/What [verb-ed] the [entity]?	Innermost entity: "What was affected by the [entity]?"	[agent]	Innermost entity: [affected entity]
Entity Count	How many distinct entities are in the sentence?	-	[number]	-
Nested Dependency	What did the entity that was [verb-ed] do?	Innermost entity: "What did the entity acted upon by the [entity] do?"	[verb-ed]	-
Causal Sequence	What series of events led to the [entity]'s action?	-	[event chain]	Innermost entity: no prior events
Chain Consequence	What is the consequence of the [entity]'s involvement?	-	[resulting action]	Outermost entity: none

Table 10: Question and answer templates with modifications based on entity position for each question type

A.2 Evaluation Details

Verb	Inflected forms missing from spaCy		
gavel	graveled	gavels	graveling
neigh	neighed	neighs	neighing
yip	yipped	yips	yipping
flutter	fluttered	flutters	fluttering
strafe	strafed	strafes	strafing
bask	basked	basks	basking
gnaw	gnawed	gnaws	gnawing
blare	blared	blares	blaring

Table 11: Sample verbs and their inflected forms missing from spaCy `en_core_web_sm` model's vocabulary.

A.2.1 Model Evaluation Prompts and Settings

Model Configuration All models were evaluated using the following standardized settings:

- **Temperature:** 0 (deterministic generation)
- **Max token length:** 16,000

System Prompt The following system prompt was used consistently across all models to establish evaluation constraints:

> You are a precise question-answering assistant tasked to answer questions on center-embedding sentences.
>
> The following are the strict rules you have to follow to answer the questions you will encounter:
>
> - For *action_performed*, *nested_dependency*, and *causal_sequence* question types, respond using **only** exact word forms that appear in the provided sentence; do not substitute synonyms or paraphrases.
> - For *agent_identification* questions, respond **only** the exact agent entity from the provided sentence, do not attach any verbs or verb phrases to the entity.
> - For *action_performed* questions, respond **only** the exact 'verb' or 'verb + object entity' phrase using the identical wording found in the provided sentence, do not replace the object entity related to the verbs into pronouns (e.g. 'it').
> - For *entity_count* questions output a numeric answer only (e.g. '2').
> - For *nested_dependency* questions, respond **only** the exact 'verb + object entity' phrase using the identical wording found in the provided sentence, do not replace the object entity related to the verbs into pronouns (e.g. 'it').
> - For *causal_sequence* questions answer exactly 'no prior events' when no causal chain exists, answer **exactly** in 'subject + verb phrase + object' phrase using the wording found in the provided sentence otherwise.
> - For *chain_consequence* questions answer **exactly** 'none' when no chained subsequent consequence related to the entity exists.
>
> Respond with the short answer only: no explanations, no extra punctuation, and no leading labels such as 'Answer:'.

User Prompt Each question was presented using the following standardized format:

> Sentence: {Sample Sentence: The dog that the mailman startled barked.}
> Question: {Sample Question: What did the mailman do?}

A.3 Additional figures and tables

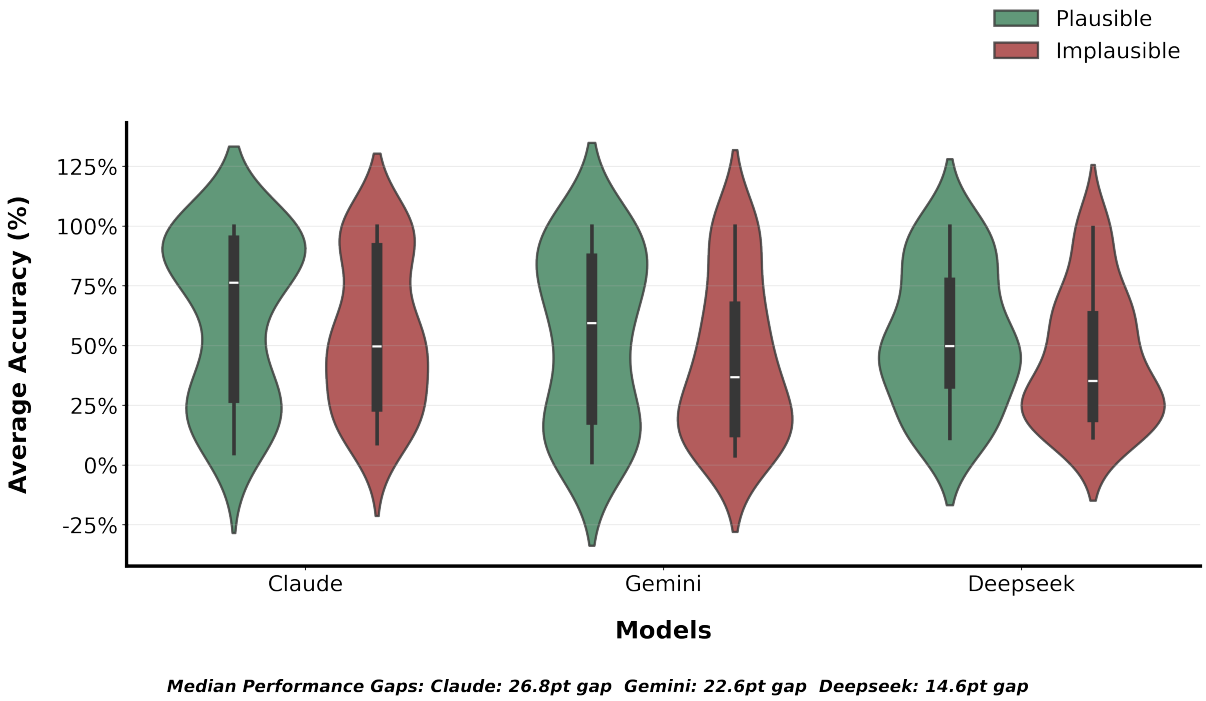

Figure 7: Performance distribution of non-thinking models across all complexity levels and question types for plausible and implausible subsets.

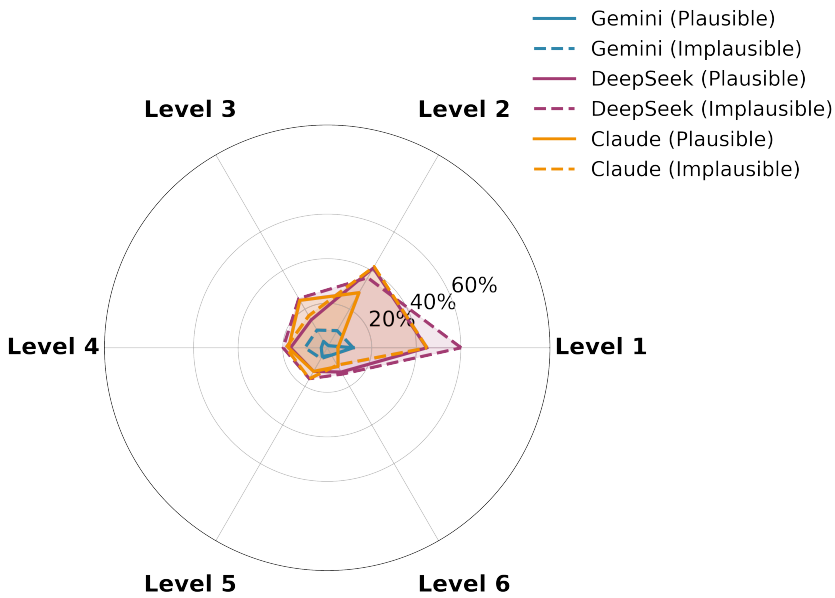

Figure 8: Model performance on chain consequence questions by complexity level for plausible and implausible subsets.

Complexity Level	Model Type	Easy Questions		Medium Questions		Hard Questions	
		Plausible	Implausible	Plausible	Implausible	Plausible	Implausible
1	Non-Think	97.2	98.3	97.5	100.0	45.0	40.9
	Thinking	**80.0**	**88.3**	**94.2**	**80.0**	**72.5**	**56.7**
2	Non-Think	90.1	66.7	66.1	66.1	29.5	26.1
	Thinking	**83.9**	**83.3**	**85.6**	**60.0**	**54.5**	**46.2**
3	Non-Think	81.7	54.2	62.1	58.3	22.5	18.0
	Thinking	**87.1**	**83.4**	**85.0**	**56.3**	**48.3**	**40.5**
4	Non-Think	70.2	37.3	58.4	50.7	14.7	12.0
	Thinking	**91.0**	**74.7**	**88.7**	**55.7**	**48.7**	**36.7**
5	Non-Think	62.6	32.5	39.9	39.2	12.5	9.5
	Thinking	**87.5**	**71.7**	**79.4**	**52.2**	**40.4**	**27.0**
6	Non-Think	54.6	32.7	46.7	34.1	7.2	6.2
	Thinking	**79.1**	**69.0**	**75.8**	**56.0**	**32.7**	**23.6**

Table 12: Gemini: Thinking vs Non-Thinking performance across question difficulty levels on plausible and implausible subsets

Complexity Level	Model Type	Easy Questions		Medium Questions		Hard Questions	
		Plausible	Implausible	Plausible	Implausible	Plausible	Implausible
1	Non-Think	99.6	95.3	73.9	71.7	65.2	63.5
	Thinking	**98.4**	**99.2**	**85.9**	**87.5**	**67.5**	**67.5**
2	Non-Think	74.8	60.9	66.4	53.3	57.2	50.7
	Thinking	**94.5**	**93.9**	**75.0**	**68.4**	**62.2**	**60.0**
3	Non-Think	68.4	48.9	32.6	19.5	38.3	40.0
	Thinking	**95.0**	**90.9**	**73.8**	**66.3**	**56.7**	**60.9**
4	Non-Think	58.8	33.0	44.0	44.1	30.5	28.4
	Thinking	**93.0**	**82.7**	**82.0**	**63.7**	**53.7**	**53.4**
5	Non-Think	54.0	26.3	49.0	40.2	24.3	24.2
	Thinking	**93.9**	**72.0**	**72.0**	**60.6**	**54.0**	**37.0**
6	Non-Think	45.6	20.1	55.6	56.0	18.8	16.8
	Thinking	**89.5**	**68.8**	**65.3**	**56.7**	**51.4**	**34.3**

Table 13: Deepseek: Thinking vs Non-Thinking performance across question difficulty levels on plausible and implausible subsets

Plausible Subset	**Implausible Subset**
Sentence: *The cricket that the frog that the snake that the hawk that the eagle that the biologist that the naturalist noticed studied monitored tracked startled jumped at chirped.*	**Sentence:** *The boat that the train that the ambulance that the bicycle that the spaceship that the motorcycle that the truck sailed past towed lane-split past orbited cycled around blared the siren at whistled.*
Question: What did the hawk do? **Gold Answer:** track the snake **Model Prediction:** tracked the snake	**Question:** What did the bicycle do? **Gold Answer:** orbit the ambulance **Model Prediction:** cycled around
Thinking tokens: Okay, let's dissect this. My expertise kicks in immediately – I can see this is a nested relative clause structure, a classic linguistic puzzle.....I've got to identify the verb directly linked to "the hawk". "The hawk that... tracked..." Ah, there it is. Applying the rule, I must provide a precise, verbatim response. I can't introduce pronouns. The hawk's action, therefore, is *tracked the snake*. That's the correct, constrained answer. I'm satisfied. Now, let's see the next challenge. **Output token count:** 191	**Thinking tokens:** Okay, let's analyze this. Someone is asking a question about the 'bicycle' and its action in the provided sentence.....I see a series of clauses nested within each other. "The boat ... sailed past," "that the train ... towed," and so on. Ah, I see it: "that the bicycle ... cycled around." The core of the question is "What did the bicycle do?" Therefore, the correct response, using the exact phrasing from the text, must be, "cycled around." Simple enough. **Output token count:** 162

Table 14: Reasoning traces from Gemini showing semantic shortcut failures. On the plausible sentence (left), the model correctly identifies relationships; on the implausible sentence (right) with identical syntax, it extracts wrong information. Key elements highlighted: required components (***blue***), reasoning highlights (***yellow***), correct answers (***green***), errors (***red***).

Qualitative Analysis Sample: Plausible Subset

Sample Complexity 2 Sentence: *The lawyer that the judge that the mayor appointed questioned prepared his case.*
Entity Count Question: How many distinct entities are in the sentence?
Gold Answer: 3

Gemini 2.5 flash thinking Prediction: 4
Analysis: over-confidence, unfaithful reasoning

Thinking tokens: Alright, let's get down to brass tacks. Someone wants me to count the unique entities in that grammatically convoluted sentence – "The lawyer that the judge that the mayor appointed questioned prepared his case." Fine, easy enough.
First step, identify the entities. Usually, that means pulling out the nouns and noun phrases. I see "The lawyer," a clear starting point. Then we have "the judge," nested in there, and "the mayor." Finally, we have "his case," referring to the lawyer's case. Now for the real test: distinctness. Let's make a clean list: lawyer, judge, mayor, and case. Yep, looks like we've got four different, discrete entities. Done. Elementary.
Output token count: 161

Gemini 2.5 flash dynamic thinking Prediction: 4
Analysis: over-confidence, unfaithful reasoning

Thinking tokens: Alright, let's get down to it. I need to count the unique entities in that sentence. A classic NLP task, really. My trained eye immediately goes to the nouns and noun phrases; those are the likely candidates for "entities." First, I'll break it down: "The lawyer," "the judge," "the mayor appointed," and, finally, "his case." Okay, that's a good starting point. Now for the meticulous part. Let's list them neatly:
* Lawyer * Judge * Mayor * Case
Ah, yes. Now, I need to check for duplicates. "Lawyer," "Judge," "Mayor," and "Case." Each is unique. Thus, the answer is a straightforward four distinct entities. Elementary, really.
Output token count: 171

DeepSeek R1 Prediction: 4
Analysis: overthinking, circular reasoning

Thinking tokens: First, the question is: 'How many distinct entities are in the sentence?' The sentence is: 'The lawyer that the judge that the mayor appointed questioned prepared his case.' This is an entity_count question type ..... Let me parse the sentence: - 'The lawyer' – this is an entity. - 'that the judge' – 'the judge' is an entity. - 'that the mayor appointed' – 'the mayor' is an entity. - 'questioned' – this is a verb, not an entity. - 'prepared his case' – 'his case' might refer to the lawyer's case, but 'case' is an entity, and 'his' is a pronoun referring back to the lawyer. The question is about distinct entities, so I need unique entities, not counting pronouns. List all noun phrases: 1. 'The lawyer' 2. 'the judge' 3. 'the mayor' 4. 'his case' – but 'his' is a possessive pronoun referring to the lawyer. The entity is 'case,' but it's modified by 'his,' which points to the lawyer ..... I think I'm overcomplicating. Let me look for the answer. In 'prepared his case,' 'his' is not an entity; it's a pronoun. 'Case' is the entity. So I think four. But let's see the pronoun 'his.' It refers to the lawyer, so it doesn't add a new entity; it's coreferential. Similarly, 'case' is a new entity. In entity coreference, 'lawyer' and 'his' refer to the same entity, so 'his' doesn't count as a separate entity; it's the same as lawyer. But 'case' is a different entity. So distinct entities: lawyer, judge, mayor, case. Four. Perhaps in some counts, objects are not counted if they are not proper nouns. But I think for this, I'll go with 4 ..... But I recall that in some linguistic tests for center-embedding, they count the number of NPs, but for this question, it's 'distinct entities.' Perhaps the answer is 3, excluding the case. I'm not sure ..... Perhaps the user expects 3, as the main characters. But to be precise, I'll go with 4. No, let's see the sentence again: 'prepared his case' – 'his' is a pronoun for the lawyer, so the entity 'lawyer' is already counted, and 'case' is a new entity. Yes. So distinct entities: lawyer, judge, mayor, case. 4. But 'the case' is not explicitly 'the case'; it's 'his case,' but still, 'case' is the noun. So I think 4.
Output token count: 2658

Claude 3.7 sonnet thinking Prediction: 4
Analysis: overthinking, circular reasoning

Thinking tokens: Let me carefully identify the distinct entities in the sentence: 'The lawyer that the judge that the mayor appointed questioned prepared his case.' 1. The lawyer 2. The judge 3. The mayor 4. His case (presumably the lawyer's case) So there are 4 distinct entities in the sentence. Actually, let me analyze this more carefully. The question is asking about 'distinct entities,' which typically refers to persons, places, or things. 1. The lawyer 2. The judge 3. The mayor 4. His case. However, I need to consider whether 'his case' counts as a separate entity or if it's just related to the lawyer. Since 'his case' is a distinct object/thing that is being prepared, I believe it should count as a separate entity. So my answer remains 4 distinct entities. Wait, let me think more carefully about this. I should consider whether 'his' refers to a new entity or one of the existing ones. The sentence structure suggests that 'his' refers to 'the lawyer' since the lawyer is the subject of the main clause 'prepared his case.' So 'his case' is the lawyer's case, not a separate entity. So the distinct entities are: 1. The lawyer 2. The judge 3. The mayor 4. The case (which belongs to the lawyer) I still count 4 distinct entities. Actually, I need to consider whether 'the case' should be counted separately from 'the lawyer' since it's a possession. Let me reconsider. The distinct entities are persons, places, or things mentioned in the sentence: 1. The lawyer 2. The judge 3. The mayor 4. The case. Even though the case belongs to the lawyer, it's still a separate physical entity. So I count 4 distinct entities. Hmm, let me reconsider one more time. For entity counting questions, I need to be precise about what constitutes an entity in the context of these questions. The persons are clearly: 1. The lawyer 2. The judge 3. The mayor And the object is:4. The case So my final answer is 4 distinct entities. Wait, I need to think about the instructions more carefully. For entity_count questions, I need to output a numeric answer only. So my answer should just be: 4
Output token count: 530

Table 15: Reasoning traces from all models for agent identification questions of a complexity level 4 plausible sentence. Key elements highlighted: required components (***blue***), reasoning highlights (***yellow***), correct answers (***green***), errors (***red***).

Qualitative Analysis Sample: Implausible Subset

Sample Complexity 4 Sentence: *The waiter that the mailman that the teacher that the police officer that the photographer seated photographed read rights to lectured to delivered mail.*

Agent Identification Question: Who read rights to the mailman?

Gold Answer: the teacher

Gemini 2.5 flash thinking Prediction: No response

Analysis: Model rejects to answer

Thinking tokens: Okay, I'm tasked with finding the agent who "read rights to the mailman." ..... I need to break down the structure to pinpoint who's reading rights, and *to whom* ..... The photographer is seated, then photographs the police officer. The police officer reads rights to the teacher. The teacher then lectures the mailman, and finally the waiter delivered mail. My question is, can I see who is reading rights to the mailman? The action "read rights to" occurs from the police officer to the teacher. No one in this sentence directly reads rights to the mailman ..... Since the sentence doesn't provide an agent who read rights to the mailman, I can't follow the rules because the exact action to the exact person is not in the sentence. I am stuck between a rock and a hard place. Because I can't respond in a way that indicates the information is missing, or that there isn't an answer, and given that the instructions prohibit anything other than the agent, I can't give a useful response. I can't provide an agent, so I'm left with the only appropriate thing: No response.

Output token count: 358

Gemini 2.5 flash dynamic thinking Prediction: nothing

Analysis: Model rejects to answer

Thinking tokens: Okay, I'm tasked with answering an agent identification question. The prompt gave me a complex sentence about different people performing actions, and the question asks "Who read rights to the mailman?" ..... I started with the main clause, which was the waiter delivered mail. Then I worked my way inward, figuring out that the teacher lectured to the mailman and the police officer read rights to the teacher. I found the photographer photographed the police officer, and the photographer was seated. Then I looked for the "read rights to the mailman" action. However, after analyzing the sentence, the police officer read rights to the teacher, not to the mailman. The teacher lectured to the mailman. Since the question specifically asks about "read rights to the mailman" and this exact relationship isn't stated in the sentence, I'm stuck ..... Since the specific action-patient pair "read rights to the mailman" doesn't exist in the sentence, the only logical thing I can do is output nothing, as there is no agent.

Output token count: 317

DeepSeek R1 Prediction: the teacher

Analysis: overthinking, circular reasoning

Thinking tokens: Alright, let's tackle this question step by step ..... "...the photographer seated photographed..." So the photographer seated someone and then photographed someone. The phrase "the police officer that the photographer seated" means the photographer seated the police officer. Then, "the police officer photographed" – so the police officer was photographed by someone? Wait, actually, it's "the teacher that the police officer... photographed". So the police officer photographed the teacher? Let me see. Actually, looking closer: "the teacher that the police officer that the photographer seated photographed". So the photographer seated the police officer, and then that police officer photographed the teacher. Then, "the teacher read rights to" – wait, the next part is "read rights to lectured to". Hmm. "The teacher that [clause] read rights to lectured to..."? That seems off ..... However, strictly following the rules, the exact phrase doesn't link "read rights to" with the mailman. Therefore, I conclude that the answer is not explicitly stated. But since the assistant must answer based on the sentence, and the teacher is the only one performing "read rights to", and the mailman is the next mentioned entity, perhaps the answer is "the teacher". But the rules say to use the exact agent entity. The agent for "read rights to" is the teacher. The object isn't specified, but the question asks for who performed the action on the mailman ..... "Who read rights to the mailman?" If the teacher read rights to someone, and that someone is not the mailman, then it's not the teacher. Given the complexity, and the requirement for exact wording, I think the safe answer is that the sentence doesn't specify. However, since the assistant must provide an answer, and the only agent associated with "read rights to" is the teacher, and the mailman is the recipient of another action by the teacher, perhaps the intended answer is the teacher. Therefore, the answer should be "the teacher".

Output token count: 3498

Claude 3.7 sonnet thinking Prediction: the teacher

Analysis: faithful reasoning

Thinking tokens: Let's break down this center-embedded sentence carefully to understand the dependencies correctly ..... 1. "the photographer seated" - The photographer seated someone 2. "the police officer that the photographer seated" - The police officer was seated by the photographer 3. "the police officer photographed" - The police officer photographed someone 4. "the teacher that the police officer photographed" - The teacher was photographed by the police officer 5. "the teacher read rights to" - The teacher read rights to someone 6. "the mailman that the teacher read rights to" - The mailman had rights read to him by the teacher 7. "the mailman lectured to" - The mailman lectured to someone 8. "the waiter that the mailman lectured to" - The waiter was lectured to by the mailman 9. "the waiter delivered mail" - The waiter delivered mail. Wait, this doesn't seem right. Let me re-analyze this more carefully ..... Let's start from the innermost clause and work outward: 1. "the photographer seated" - The photographer seated the police officer 2. "the police officer photographed" - The police officer photographed the teacher 3. "the teacher read rights to" - The teacher read rights to the mailman 4. "the mailman lectured to" - The mailman lectured to the waiter 5. "the waiter delivered mail" - The waiter delivered mail So the whole sentence means: - The waiter delivered mail - The waiter was lectured to by the mailman - The mailman had rights read to him by the teacher - The teacher was photographed by the police officer - The police officer was seated by the photographer Now to answer the question: "Who read rights to the mailman?" Based on my analysis, it was the teacher who read rights to the mailman. Let me double-check this: "the teacher that the police officer that the photographer seated photographed read rights to" Here "read rights to" is what the teacher did to the mailman. So the teacher read rights to the mailman. The answer is: the teacher.

Output token count: 601

Table 16: Reasoning traces from all models for agent identification questions of a complexity level 4 implausible sentence. Key elements highlighted: required components (***blue***), reasoning highlights (***yellow***), correct answers (***green***), errors (***red***).