Multi-Constraint State Tracking with Negation: A Diagnostic Benchmark for LLM World Modeling

Ayan Sar[1], Pranav Singh Puri[1], Sumit Aich[1], Anurag Kaushish[1], Tanupriya Choudhury[1], Ajith Abraham[2,3]

[1]School of Computer Science, University of Petroleum and Energy Studies, Dehradun, 248007, Uttarakhand, India.
[2]School of Artificial Intelligence, Sai University, Chennai, Tamil Nadu, 603105, India.
[3]Center for Artificial Intelligence, Innopolis University, Innopolis, Russia.

Correspondence: tanupriya@ddn.upes.ac.in.

Abstract

Large Language Models (LLMs) achieve strong performance on a wide range of reasoning benchmarks, yet it remains unclear whether they can reliably maintain and update internal representations of an evolving world described in natural language. In particular, existing evaluations inadequately probe state tracking under multiple interacting constraints and largely overlook the role of negated actions, despite their ubiquity in real-world language. We address this gap by introducing MCST, a diagnostic benchmark for multi-constraint state tracking that evaluates an LLM's ability to maintain consistent world models across sequences of actions involving inventory changes, spatial movement, temporal ordering, and systematic negation. MCST comprises $100,847$ questions spanning 12 real-world domains, with five calibrated difficulty levels, nine question types, and controlled integration of negated actions. The benchmark further incorporates culturally diverse entity names to enable analysis of cross-cultural robustness. We evaluate 14 SOTA LLMs across multiple model families using a unified evaluation protocol. Our results reveal substantial limitations: even the strongest models exhibit sharp performance degradation as difficulty increases, with accuracy dropping below 35% at the highest level. Most notably, we identify negation as a dominant failure mode, causing accuracy reductions of $23-32\%$ across models. We release MCST and the full evaluation framework to support future research on state tracking and reasoning in language models and is available at GitHub.

1 Introduction

Large Language Models (LLMs) have demonstrated impressive performance across a broad range of natural language processing tasks, including question answering, mathematical reasoning, and code generation (Brown et al., 2020) (Wei et al., 2023) (Cobbe et al., 2021) (Chen et al., 2021). These successes have fueled growing interest in deploying LLMs as reasoning agents in settings that require understanding and acting upon descriptions of the world, such as dialogue systems, planning assistants, and embodied or simulated agents (Yao et al., 2023) (Shinn et al., 2023). A fundamental requirement for such applications is the ability to **maintain an accurate representation of an evolving world state** as events unfold through natural language descriptions.

Consider the following scenario: *"Emma picks up the red book from the desk. Later, she does not place it on the shelf."* To answer a subsequent question - *"Who has the book now?"* - a system must correctly update its belief after the first action and, critically, refrain from updating it after the negated action. While this form of reasoning is trivial for humans, recent evidence suggests that even state-of-the-art LLMs fail on such cases with surprising frequency (Valmeekam et al., 2023) (Dziri et al., 2023). These failures point to a deeper limitation: ***Despite strong surface-level reasoning abilities, LLMs often lack robust mechanisms for belief maintenance and revision over action sequences, especially when reasoning requires tracking what explicitly did not happen.***

While **state tracking** - maintaining a consistent mental model of entities and their relationships throughout a sequence — is foundational to reading comprehension and agentic reasoning, existing benchmarks fail to adequately probe this capability (Weston et al., 2015) (Richardson et al., 2013) (Rajpurkar et al., 2016) (Jiang et al., 2024). Current evaluation paradigms are limited by a lack of complex constraint integration (such as simultaneous spatial and inventory tracking), a neglect of the outsized impact of negation on comprehension, and a reliance on coarse difficulty scales and culturally homogeneous data. As a result, proficiency in static reasoning does not necessarily translate to a large language model (LLM) being able to update

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 1317–1350

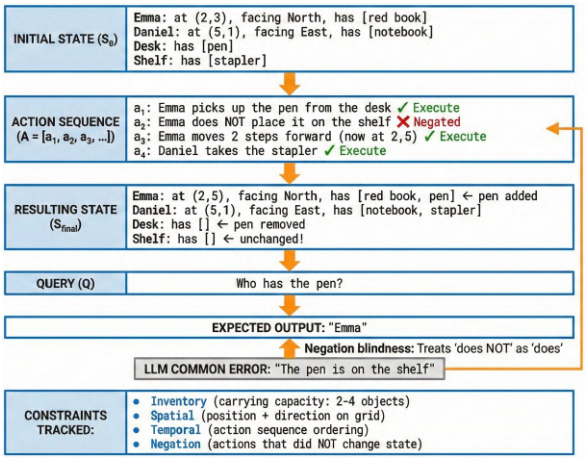

Figure 1: Overview of Multi-Constraint State Tracking with Negation. Given an initial state S_0, a sequence of actions A (including negated actions that do NOT modify the state), the task is to answer queries Q about the resulting state. The example shows a common LLM failure: treating negated action a_2 (Emma does NOT place pen on shelf) as executed, leading to incorrect state inference. MCST measures this skill across 12 domains, varying the difficulty and the number of negatives.

its world model effectively in realistic scenarios (Srivastava et al., 2023) (Bubeck et al., 2023).

Towards this end, we propose **MCST** (**M**ulti-**C**onstraint **S**tate **T**racking), a diagnostic benchmark that systematically assesses LLMs' capacity to consistently update the world with plausible explanations under realistic, difficult conditions. MCST combines the constraints of **inventory management, spatial and temporal ordering,** and **systematic negation** into integrated scenarios, directly addressing interactions between these constraints. Our benchmark includes $100,847$ **questions** across **12 domains of the real world**, with **five levels of difficulty**, and **nine question types** that examine complementary abilities in state tracking (see Figure 1). To support broader robustness analysis, MCST incorporates **culturally diverse entity names** drawn from over 25 global regions. In summary, this work makes the following contributions:

1. We introduce MCST, a large-scale benchmark for evaluating multi-constraint state tracking with systematic negation;

2. We provide a fine-grained difficulty framework and question taxonomy enabling detailed capability analysis; and

3. We present an extensive empirical study demonstrating fundamental limitations of current LLMs in belief maintenance and revision.

We release MCST and the accompanying evaluation framework to facilitate future research on state tracking, world modeling, and reasoning in language models.

2 Related Work

2.1 State Tracking and World Modeling

Early state tracking evaluation relied on synthetic settings like the bAbI tasks, which modern LLMs solve using templated patterns rather than genuine belief maintenance. Later benchmarks like CLUTRR and StepGame added relational and spatial complexity but still isolated individual constraints, such as grid navigation or kinship. Interactive environments such as TextWorld (Côté et al., 2019) and ALFWorld (Shridhar et al., 2021) evaluate agents that must execute winning strategies in a simulated world, which inherently couples three capabilities: comprehending the narrative, maintaining an internal world model, and selecting actions to advance toward a goal. A failure in any one of these is observed only as a downstream task-success signal, making it structurally difficult to isolate where belief maintenance breaks down. MCST is deliberately noninteractive: The model neither generates actions nor receives environmental feedback, only reads a fixed action sequence and answers a query about the resulting state. This disentangles language-based state tracking from action planning and execution, while simultaneously integrating inventory, spatial, temporal, and logical constraints — yielding a controlled, diagnostic evaluation of belief maintenance that text-game benchmarks cannot provide.

2.2 Reasoning and Instruction-Following Benchmarks

Popular reasoning tasks such as GSM8K for arithmetic or LogiQA (Liu et al., 2020) for logic reasoning mostly test static inference where the state of the world remains constant or is not relevant. Likewise, datasets for following written instructions, such as WizardLM or ComplexBench, do not explicitly involve dynamic worlds. In contrast, MCST requires models to dynamically update and maintain a world state over action sequences, with success dependent on maintaining belief consistency over an evolving narrative.

Benchmark	Primary Focus	Dynamic State Tracking	Multi-Constraint Integration	Systematic Negation	Difficulty Calibration	Diagnostic Analysis
bAbI (Weston et al., 2015)	Synthetic reasoning	✓(limited)	×	×	×	×
CLUTRR (Sinha et al., 2019)	Relational chains	✓	×	×	Partial	×
StepGame (Shi et al., 2022)	Spatial tracking	✓	×	×	Partial	×
GSM8K (Cobbe et al., 2021) / MATH (Hendrycks et al., 2021)	Math reasoning	×	×	×	Continuous	×
WizardLM (Xu et al., 2025) / ComplexBench (Wen et al., 2024)	Instruction following	×	Partial	×	✓	×
LegalBench (Guha et al., 2023) / ContractNLI (Koreeda and Manning, 2021)	Policy reasoning	×	Partial	×	×	×
RuleArena (Zhou et al., 2025b)	Rule-guided reasoning	×	Partial	×	✓	✓
MCST (ours)	**State tracking**	✓	✓	✓	**✓(5 levels)**	✓

Table 1: Comparing MCST to other benchmarks. MCST is the first and only benchmark that combines several types of constraints, systematic negation handling and several levels of difficulty.

2.3 Rule-Guided Reasoning and Policy Benchmarks

Existing benchmarks like RuleBERT (Saeed et al., 2021) and RuleArena (Zhou et al., 2025a) assess LLM application of explicit rules from domains such as tax or regulation. These offer detailed diagnostics for policymaking, but are generally based on logical inference in a static world state. MCST is complementary; rather than understanding rules, it involves evolving state. This is important because while a model may comprehend the rule, it may not be able to maintain beliefs about the final state following a series of complex events.

2.4 Negation in Natural Language Understanding

Negation is a known issue in NLP (Naik et al., 2018) that affects LLM performance in a number of reasoning tasks. While previous research has focused on the use of negation cues in individual tasks, it has not addressed the impact on world modelling. MCST is the first benchmark to systematically integrate negated actions into state tracking at scale. This allows for a quantitative analysis of how negation disrupts belief maintenance, exposing it as a dominant failure mode in model reasoning. We note that natural-language negation has multiple semantic functions. MCST scopes negation to its most under-studied form in state-tracking evaluation: the explicit non-occurrence of an action (e.g., "Emma does not place the book on the shelf"), which requires the model to leave the world state unchanged. We do not address implicit state-changing negation (e.g., "he did not arrive on time," which entails a change in attendance status), since this conflates negation handling with pragmatic inference over implicit state schemas. We discuss this in the Limitations and treat it as a complementary direction for future work.

Positioning. Prior benchmarks evaluate model reasoning or rule application. MCST addresses a foundational, complementary question: whether models can maintain a coherent world model over time, specifically when they must account for events that did not occur. Seeing Table 1, it can be understood that MCST uniquely targets dynamic state tracking under multiple interacting constraints with systematic negation and calibrated difficulty, enabling fine-grained diagnosis of belief maintenance failures that are not addressed by prior reasoning or rule-following benchmarks.

3 The MCST Benchmark

MCST (Multi-Constraint State Tracking) is designed to evaluate whether large language models can maintain and update a coherent internal representation of an evolving world state described entirely in natural language. Unlike prior benchmarks that test isolated reasoning skills or static inference, MCST explicitly targets belief maintenance and revision under realistic conditions, where multiple constraints interact and where actions may be explicitly negated.

3.1 Task Formulation

Each MCST instance describes a short narrative consisting of an initial world state, a sequence of actions, and a query about the resulting state. Formally, each instance is defined as a tuple

$$(S_0, A, Q, y),$$

where S_0 denotes the initial state, $A = [a_1, \ldots, a_T]$ is a sequence of natural language actions, Q is a question about the final state, and y is the ground-truth answer.

The world state comprises a set of **entities**, **objects**, and **locations**, together with their relations (e.g., possession, spatial position, orientation). Each action a_t either **modifies the state** (e.g., an entity picks up an object or moves to a new location) or is explicitly **negated** (e.g., "Emma does not place the book on the shelf"), in which

case the state must remain unchanged. Correctly answering Q requires the model to sequentially process all actions, update the state when appropriate, and crucially, **avoid updating the state for negated actions.** MCST evaluates state tracking purely through **language understanding**: models are not required to generate actions, interact with an environment, or execute code. This design isolates comprehension and belief maintenance from action planning or control.

3.2 Design Principles

We constructed MCST around four guiding principles intended to reflect real-world language understanding demands.

Multi-Constraint Integration. Realistic state tracking requires simultaneously managing multiple interacting constraints. MCST integrates **inventory constraints** (who possesses which objects), **spatial constraints** (entity positions and orientations on a grid), **temporal constraints** (the order and persistence of events), and **logical constraints** (negated or conditional actions). Unlike benchmarks that test these aspects in isolation, MCST requires models to reason about their interaction.

Systematic Negation. Negated actions are pervasive in natural language yet underrepresented in existing benchmarks. MCST systematically incorporates negation by explicitly marking a controlled proportion of actions as non-occurring. These negated actions are indistinguishable from executed actions at the surface level except for linguistic cues, forcing models to correctly interpret and apply negation semantics during state updates.

Calibrated Difficulty. Rather than coarse easy-hard splits, MCST employs a five-level difficulty framework with carefully controlled complexity parameters. This enables fine-grained analysis of how model performance degrades as the demands of state tracking increase.

Cultural Inclusivity. Prior work has documented systematic name- and identity-based disparities in LLM behaviour on QA, sentiment, and toxicity tasks, but it remains unclear whether such disparities also propagate to lower-level belief-update behaviour — i.e., whether the same action sequence is tracked equally well when the protagonist's name is Emma versus Aanya or Chinedu. To enable this finer-grained robustness probe, MCST incorporates culturally diverse entity names spanning more than 25 global regions while holding scenario structure, action templates, and ground-truth state transitions identical. This isolates name-conditioned variance in state-tracking performance from variance in task content, surfacing biases that are otherwise confounded in standard QA benchmarks.

Domain	Typical Objects	Locations
Office	stapler, badge, monitor, keyboard	conference room
Home	remote, pillow, spatula, keys	living room
Healthcare	stethoscope, chart, syringe, thermometer	patient room
Education	textbook, microscope, chalk, eraser	laboratory
Travel	passport, boarding pass, luggage	departure gate
Retail	receipt, shopping cart, price tag	checkout
Restaurant	menu, tray, napkin, receipt	dining area
Hospitality	key card, towel, menu, luggage	guest room
Recreation	frisbee, fishing rod, ball, racket	playground
Cultural	audio guide, ticket, painting, catalog	gallery
Manufacturing	safety goggles, wrench, blueprint	assembly line
Technology	USB drive, prototype, server rack	testing lab

Table 2: MCST Domain Coverage: Representative objects and locations across 12 real-world contexts.

Level	Entities	Objects	Actions	Negation Rate	Spatial
1	2	3	3-5	0%	None
2	3	4	6-8	∼15%	None
3	4	5	9-12	∼25%	5×5 grid
4	3-7	4-8	10-20	∼30%	7×7 grid
5	4-8	5-10	12-25	∼40%	10×10 grid

Table 3: MCST Difficulty Calibration Framework: Complexity parameters across five levels.

3.3 Domain Coverage

We designed MCST as it spans 12 real-world domains chosen to balance familiarity, diversity of interactions, and practical relevance. These domains include office environments, homes, healthcare settings, educational spaces, travel scenarios, retail contexts, restaurants, hospitality, recreation, cultural venues, manufacturing, and technology workplaces (see Table 2). Each domain defines a domain-specific ontology of valid objects, locations, and actions (e.g., a stethoscope in healthcare, a boarding pass in travel). We followed three criteria for the domain selection: ❶ everyday plausibility, ensuring minimal reliance on specialized background knowledge; ❷ natural interactions between entities, objects, and locations; and ❸ diversity across personal and professional contexts. This breadth ensures that MCST does not overfit to a narrow narrative style or domain-specific heuristic.

3.4 Difficulty Framework

MCST defines five difficulty levels, each controlling the number of entities, objects, actions, spatial complexity, and negation rate (see Figure 2 and Table 3). Each level introduces successive chal-

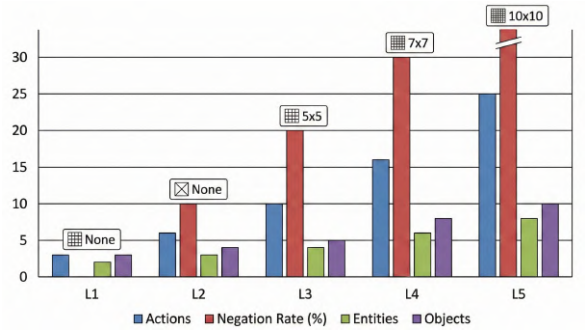

Figure 2: Systematic difficulty calibration across five levels. **L** denotes different levels. Complexity increases along multiple dimensions: number of entities $(2 \rightarrow 8)$, objects $(3 \rightarrow 10)$, actions $(3 \rightarrow 25)$, negation rate $(0\% \rightarrow 40\%)$, and spatial grid size (none$\rightarrow 10 \times 10$). This multi-dimensional scaling reflects the natural co-occurrence of complexity factors in real-world language. Precise single-factor attribution is provided by the controlled ablation studies, where each dimension is varied independently.

lenges in a monotonic way to guarantee gradual performance improvements.

- **Level 1 (Foundational):** Few entities and objects, brief action sequences, no negation and no spatial constraints. This provides basic state maintenance ability.
- **Level 2 (Negation Introduction):** Moderate-frequency negated actions, no spatial considerations, and focuses on the impact of negation on maintaining beliefs.
- **Level 3 (Spatial Reasoning):** Introduces spatial location and orientation with a discrete grid, which must be tracked together with possession.
- **Level 4 (Variable Complexity):** Increases the number of entities and actions with spatial reasoning, to test for variability.
- **Level 5 (Expert Challenge):** Maximizes all complexity parameters, including high negation rates and large spatial grids, representing the upper bound of difficulty evaluated.

While difficulty levels simultaneously scale multiple parameters to reflect naturalistic complexity growth, precise attribution of failures to individual factors is enabled by our controlled single-factor ablations in Section 4.4, where each parameter is vaired independently while all others are held constant.

Empirical Validation of Calibration. Although our levels are constructed by jointly scaling complexity parameters rather than fitted to a target item-difficulty curve, two post-hoc signals support that the resulting gradient is well-calibrated. First, human accuracy degrades monotonically from 97.3% at L1 to 86.7% at L5 (Appendix Table 26), confirming that human-perceived difficulty tracks our parameter scaling. Second, every evaluated model exhibits a monotonic $L1 \rightarrow 5$ decline of consistent magnitude ($\approx 43\%$ mean drop, $\sigma \approx 0.4$ across families; see Table 4), indicating that the levels separate models in a stable, ordered way rather than producing rank-flips or plateaus. We treat fitting per-item difficulty parameters in the IRT sense as future work.

3.5 Question Taxonomy and Construction Methodology

We included nine question types in MCST, mainly designed to probe complementary aspects of state tracking. These include direct state queries (e.g., possession or location), verification questions, counting queries, backward and logical inference, comparisons, spatial orientation questions, and temporal sequence queries. The distribution of question types is calibrated to reflect realistic usage, with state queries and verification questions forming the majority, while inference-heavy queries test deeper belief consistency.

We constructed MCST using a Human-Expert-in-the-Loop (HEIL) process comprising four stages:

Expert Design. A team of NLP researchers defined domain ontologies, action templates, negation patterns, and difficulty parameters. Pilot studies informed the calibration of complexity and negation rates.

Controlled Procedural Generation. Scenarios were generated using expert-defined templates and stochastic simulation, ensuring physically and logically consistent state transitions. Natural language realizations were produced using multiple syntactic and lexical variants to avoid surface-level shortcuts.

Multi-Stage Validation. Generated instances underwent automated filtering (grammar checks, semantic coherence scoring, duplicate detection) followed by crowdsourced verification, with each instance reviewed by multiple annotators. Ambiguous cases were escalated to expert review.

Adversarial Verification. A final adversarial pass targeted negation-heavy and high-difficulty scenarios to ensure robustness against annotation errors and unintended heuristics.

3.6 Quality Assurance

We included multiple quality control measures in MCST to ensure quality. Automatic simulation ensures that ground-truth answers are internally consistent. Inter-annotator agreement exceeds standard thresholds, and lexical diversity metrics confirm variation in surface norms. Random baselines and heuristic probes fail to achieve above-chance performance, indicating resistance to shallow shortcuts. Remaining edge cases with multiple valid interpretations are explicitly annotated with all acceptable answers.

4 Experiments

Models. We evaluated 14 state-of-the-art LLMs spanning multiple model families, including proprietary and open-weight models with diverse training paradigms and parameter scales. This selection enables comparison across architectural choices and training regimes, rather than optimizing for a single family.

Inference Protocol. All models are evaluated under a uniform inference setting to ensure compatibility. We use greedy decoding (temperature = 0) to minimize stochastic variation and focus on intrinsic model capability. The maximum generation length is fixed across models and exceeds the longest expected answer length, preventing truncation effects. No model receives additional task-specific finetuning or external tools.

Prompting Format. Each question is posed in a consistent prompt format of: ❶ a narrative that describes the initial state and a sequence of actions, followed by ❷ a single question. We do not include chain-of-thought demonstrations or intermediate steps. This decision is standard practice for diagnostic benchmarks, and prevents confounding state tracking with prompting.

State-Tracking Prompting (STP). In addition to traditional Direct and Chain-of-Thought (CoT) prompting, we propose a simple instructional prompt, State-Tracking Prompting (STP), which aims to address the specific vulnerabilities of the benchmark. STP adds three constraints to the standard prompt that guide the model to attend to the processing of negation and incremental state updates. The full STP template is:

Listing 1: STP Template

```
Scenario: [Context]
Track the state carefully:
  - Note which actions are EXECUTED versus NOT
      executed
  - Update the world state ONLY for executed
      actions
  - For any action containing "does not", "did
      not", or "NOT:,
    treat the state as UNCHANGED after that
       action
Question: [Q]
Answer:
```

STP differs from CoT in a critical respect: CoT asks the model to reason step-by-step but imposes no constraint on how negation is handled. STP provides explicit logical constraints that define the correct state-update rule. All three prompting strategies are shown in the Appendix (Figure 7).

Evaluation Metric. Performance is measured using exact-match accuracy against the ground-truth answer. For questions with multiple valid answers (e.g., equivalent descriptions of direction), all acceptable variants are considered correct. Accuracy is reported per difficulty level, domain, question type, and negation condition.

4.1 Human Performance Baseline

The full evaluation protocol, participant Instructions and demographic details are provided in Appendix E.7. Humans achieve an overall accuracy of 94.3% (sigma = 4.8%), degrading from 97.3% at Level 1 to 86.7% at Level 5 (Table 26). Critically, human error patterns differ fundamentally from model error patterns: humans primarily commit quantitative errors such as off-by-one counting (38%) and coordinate arithmetic slips (25%), while negation oversight accounts for only 11% of human errors — attributed to attention lapses rather than comprehension failure. In contrast, as shown in Section 4.3, Negation Blindness accounts for 47% of model errors. This inverse pattern between human and model failure modes is one of the central diagnostic findings of this paper.

4.2 Main Results

Table 4 presents the comprehensive performance of 14 state-of-the-art LLMs on the MCST benchmark. We observe that **State-Tracking Prompting (STP)**, described in the subsection above, consistently outperforms Direct and Chain-of-Thought (CoT) strategies, yielding an average improvement of $3-5\%$ across model families. Nonetheless, the

Model	Overall Accuracy (%)				Accuracy by Difficulty Level (Best Strategy)					
	Direct	CoT	STP	**Best**	L1	L2	L3	L4	L5	Δ_{L1-L5}
OpenAI Models										
GPT-4o	54.2	58.1	**59.3**	59.3	**78.4**	**67.2**	**56.8**	**47.3**	**34.7**	-43.7
GPT-4-Turbo	51.8	55.7	57.2	57.2	-	-	-	-	-	-
GPT-3.5-Turbo	38.4	41.2	42.8	42.8	-	-	-	-	-	-
Anthropic Models										
Claude-3.5-Sonnet	52.1	56.8	58.4	58.4	76.9	65.8	55.4	46.1	33.2	-43.7
Claude-3-Opus	49.3	53.4	55.1	55.1	-	-	-	-	-	-
Claude-3-Haiku	36.7	39.5	41.2	41.2	-	-	-	-	-	-
Meta Models										
LLaMA-3.1-405B	47.2	51.8	53.6	53.6	-	-	-	-	-	-
LLaMA-3.1-70B	42.6	46.3	48.1	48.1	68.3	55.2	44.7	36.8	25.4	-42.9
LLaMA-3.1-8B	31.4	33.9	35.2	35.2	-	-	-	-	-	-
Qwen Models										
Qwen-2.5-72B	44.8	48.6	50.3	50.3	70.1	57.6	47.2	38.4	27.1	-43.0
Qwen-2.5-32B	40.2	43.7	45.4	45.4	-	-	-	-	-	-
Qwen-2.5-7B	29.6	32.1	33.7	33.7	-	-	-	-	-	-
Google Models										
Gemini-1.5-Pro	50.4	54.2	56.1	56.1	74.2	62.4	52.8	43.6	31.5	-42.7
Gemini-1.5-Flash	43.1	46.8	48.5	48.5	-	-	-	-	-	-

Table 4: **Main Results on MCST.** Left: Overall accuracy (%) across three prompting strategies (Direct, CoT, State-Tracking Prompt). Right: Performance breakdown by difficulty level (L1−−L5) for representative models using the best prompting strategy. Note the consistent degradation (∼43%) from L1 to L5.

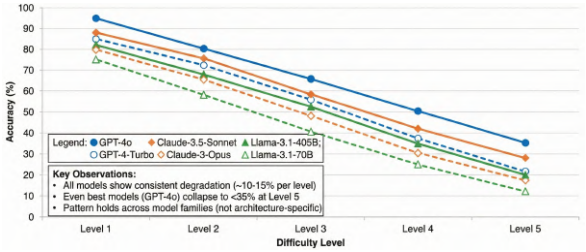

Figure 3: Systematic performance degradation across difficulty levels for six state-of-the-art LLMs.

best model (GPT-4o) only reaches **59.3%** accuracy, leaving a gap with human performance (∼94%). More importantly, the performance decreases as the tasks get harder. For the models where fine-grained analysis was performed (right side of Table 4), accuracy drops by approximately **43%** from Level 1 to Level 5 (see Figure 3). A notable "performance cliff" of ∼10% occurs at Level 3, corresponding to the introduction of spatial constraints.

Single-Factor Attribution. We acknowledge that the multi-dimensional scaling of our difficulty levels reflects real-world complexity co-occurrence but limits direct attribution from level-to-level comparisons alone. To enable precise single-factor analysis, Section 4.4 presents three controlled ablation studies.

4.2.1 Diagnostic Analysis

Table 5 provides a fine-grained analysis of the best-performing model (GPT-4o). The results suggest models are relatively good at basic verification, but struggle with **Backward Inference** (+41.6% increase in error) and spatial reasoning. Additionally, familiar domains like Office and Home yield higher accuracy compared to specialized domains like Manufacturing and Culture.

4.3 Negation Failure Analysis

Table 6 highlights the critical impact of negation. On matched scenario pairs, models exhibit a relative performance drop of 31−−37%. Qualitatively, the dominant failure mode is *Negation Blindness* (47%), where models simply hallucinate the execution of a negated action.

Question Type	Acc. (%)	Domain	Acc. (%)
Foundational Skills		*Top Performing*	
Verification (T/F)	**71.2**	Office	**62.3**
State Query (Possession)	62.4	Home	61.8
State Query (Location)	58.7	Education	60.4
Complex Reasoning		*Mid Performing*	
Comparison	55.4	Healthcare	58.9
Direction Query	52.8	Retail	58.2
Counting (Entity)	48.3	Restaurant	57.8
High Failure Rate		*Lowest Performing*	
Counting (Location)	45.6	Travel	56.4
Position (Coordinates)	43.2	Manufacturing	54.1
Backward Inference	41.6	Cultural	52.4

Table 5: **Diagnostic Analysis (GPT-4o, STP).** Breakdown of performance by Question Type (left) and Domain (right). Error Rate refers to the relative drop compared to the easiest category (Verification).

Model	No Neg.	With Neg.	Δ	Rel. Drop
GPT-4o	72.4	49.3	-23.1	-31.9%
Claude-3.5-Sonnet	71.8	48.1	-23.7	-33.0%
LLaMA-3.1-70B	63.2	39.8	-23.4	-37.0%
Qwen-2.5-72B	65.4	41.2	-24.2	-37.0%
Gemini-1.5-Pro	69.1	45.7	-23.4	-33.9%

Table 6: **Negation Impact Analysis.** Accuracy (%) on matched scenario pairs (identical contexts with vs. without negated actions). All models suffer a catastrophic drop.

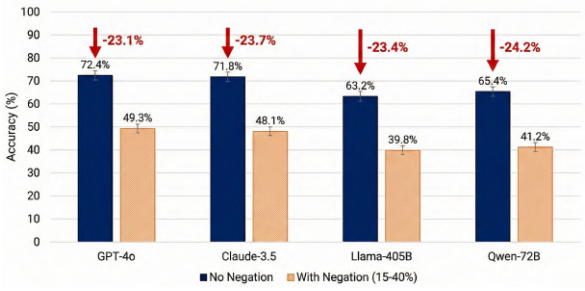

Figure 4: Catastrophic impact of negated actions on state tracking accuracy.

4.4 Diagnostics: Scaling, Bias, and CoT

While Subsection 4.2.1 reports multi-factor difficulty scaling that mirrors real-world complexity, the analyses below isolate individual complexity dimensions to provide causally interpretable findings. We group our diagnostic findings in Table 7. ❶ **Sequence Length:** Accuracy degrades by ∼5% per 3 actions, indicating context window utilization issues for state tracking. ❷ **Cultural Bias:** A systematic bias exists favoring Western names (61.2%) over African names (55.2%), persisting even after controlling for phonological complexity. ❸ **CoT Analysis:** While CoT improves general performance, it remains flawed; 21% of error traces

(a) Sequence Length

Actions	GPT-4o	Claude
3-5	74.2%	72.8%
6-8	63.7%	62.1%
9-12	52.4%	51.2%
13-17	43.8%	42.5%
18-25	35.1%	33.7%

(b) Spatial Limits

Component	Acc.
Grid boundary	67.3%
Turn direction	58.4%
Move position	51.2%
Relative pos.	43.7%

(c) Name Bias (GPT-4o)

Region	Acc.	Δ West
Western	61.2%	–
East. Europe	59.7%	-1.5%
Latin Amer.	58.9%	-2.3%
East Asian	58.4%	-2.8%
South Asian	57.9%	-3.3%
Middle East	56.8%	-4.4%
African	55.2%	-6.0%

(d) Counting Errors

Error Type	Freq.
Off-by-one	38%
Ignore capacity	24%
Include dropped	19%
Double count	11%
Zero confusion	8%

Table 7: **Diagnostic Analysis.** Top Left: Impact of sequence length on accuracy. Top Right: Spatial reasoning accuracy by component. Bottom Left: Cultural bias analysis (GPT-4o). Bottom Right: Distribution of counting errors.

involve explicit mishandling of negation logic.

5 Analysis & Insights

5.1 What Limits State Tracking Performance?

Cognitive complexity is monotonic with performance, which drops an average of 43% from Level 1 to Level 5. This suggests a general instability of world representations as complexity increases. The fact that we observe a pronounced "performance cliff" of $\sim$ 10% at Level 3 (where spatial constraints are introduced) suggests that spatial reasoning and inventory tracking are incompatible, driving models to fall back on local and tenuous patterns in the world.

5.2 Why State-Tracking Prompting Helps?—But Not Enough

While State-Tracking Prompting (STP) yields consistent gains of $3-5\%$ over Direct and Chain-of-Thought (CoT) methods, it does not alter the overall performance trajectory; even GPT-4o caps at 59.3%. The persistence of core belief-update errors—evidenced by 21% of CoT traces explicitly mishandling negation—demonstrates that improved attention mechanisms cannot compensate for the lack of a reliable, persistent memory structure for state variables. Inspecting the generated CoT traces themselves serves as a direct probe of the model's implicit world model: when the model verbalizes its belief updates, 21% of the failing

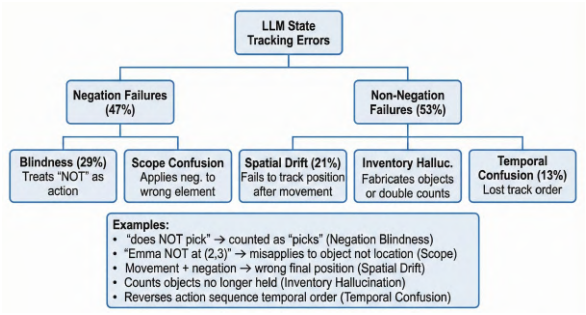

Figure 5: Taxonomy of state tracking failures across 500 manually analyzed errors.

traces explicitly write out an updated state for an action they have just identified as negated, indicating that the failure is not a tokenization or attention artifact but a genuine inconsistency between the model's stated reasoning and its committed belief.

5.3 Negation as a Dominant Failure Mode

Negation emerges as the most destructive factor, causing a consistent $31-37\%$ accuracy drop across all model families on matched scenarios. Qualitative analysis identifies Negation Blindness (hallucinating negated actions) as the dominant failure mode (47%), leading to cascading errors (see Figure 5). This implies that negations are treated as soft qualifiers, which can be readily overruled by action scripts, instead of hard constraints on state transitions.

5.4 Inference Failures Reveal Fragile Internal States

LLMs have difficulty preserving consistent states with interacting constraints, with disastrous effects of negation showing that models treat logical "non-events" as flimsy surface-level heuristics rather than constraints. These errors are perilous for agentic applications, where tracking errors accumulate. While techniques such as State-Tracking Prompting have modest benefits, they cannot address the lack of state representations. In conclusion, our results suggest that scaling is not enough; to progress in understanding the world, models need architectures that include memory and training strategies that guarantee dynamic consistency.

5.5 Scaling Effects: Sequence Length and Error Accumulation

Performance suffers a linear drop of around 5% for each three extra actions, well before context windows are reached. This "state drift" suggests poor context use, rather than capacity. Deficits arise steadily without recovery, affirming that models do not have mechanisms to correct prior belief states after an invalid update (notably related to negation or movement) has been processed.

6 Discussions

MCST reveals a fundamental gap between static reasoning fluency and dynamic world modeling. LLMs have difficulty preserving consistent states with interacting constraints, with disastrous effects of negation showing that models treat logical "non-events" as flimsy surface-level heuristics rather than constraints. These errors are perilous for agentic applications, where tracking errors accumulate. While techniques such as State-Tracking Prompting have modest benefits, they cannot address the lack of state representations. In conclusion, our results suggest that scaling is not enough; to progress in understanding the world, models need architectures that include memory and training strategies that guarantee dynamic consistency.

7 Conclusion

This work introduced MCST, a large-scale benchmark for evaluating multi-constraint state tracking in large language models, with a particular focus on belief maintenance under negated and interacting actions. Through extensive experiments and fine-grained analysis, we showed that even state-of-the-art models suffer substantial performance degradation as complexity increases, with negation emerging as a dominant and previously under-measured failure mode. Our findings demonstrate that current LLMs lack robust mechanisms for maintaining coherent world models over time, limiting their reliability in applications that require dynamic understanding. By releasing MCST and its evaluation framework, we aim to support future research on world modelling, belief revision, and more reliable reasoning in language models.

Future Work. Extensions to MCST include: (i) Few-shot adaptation: testing if demonstrations mitigate Negation Blindness; (ii) Multi-turn evaluation: assessing belief coherence in interactive dialogues; (iii) Implicit negation: covering negations entailing state changes (e.g., "did not arrive" $\implies$ absent); and (iv) IRT calibration: applying Item-Response Theory for instance-level difficulty analysis.

Limitations

While MCST is designed to provide a rigorous and diagnostic evaluation of state tracking in large language models, it has several limitations that should be acknowledged. ❶ First, MCST focuses exclusively on text-based world descriptions and evaluates state tracking through language understanding alone. Although this isolation is intentional to enable controlled analysis, it does not capture challenges arising in multimodal or interactive environments, where perception, action execution, and feedback may further influence belief maintenance. Consequently, performance on MCST should not be interpreted as a complete measure of an agent's world modelling ability in embodied settings. ❷ Second, all scenarios and annotations in MCST are English-only. While we incorporate culturally diverse entity names to probe robustness, linguistic diversity in syntax, morphology, and negation structures across languages is not represented. As a result, our findings may not fully generalise to multilingual or low-resource language contexts, where negation and temporal structure can differ substantially. ❸ Third, MCST evaluates comprehension and belief updating, not learning or adaptation. Models are assessed in a zero-shot or prompt-based setting without interaction, memory externalisation, or tool use. Systems equipped with explicit memory modules, symbolic state representations, or environment interaction may exhibit different behaviours that are not captured by this benchmark. ❹ Fourth, although MCST is large-scale and systematically constructed, it relies on synthetic yet realistic narratives generated under controlled templates and ontologies. While extensive validation mitigates artefacts, these scenarios may still lack some of the ambiguity, underspecification, and noise present in naturally occurring text. ❺ Fifth, MCST scopes negation to the explicit non-occurrence of actions, where the correct behaviour is to leave the world state unchanged. It does not cover implicit state-changing negation, where a negated surface form pragmatically entails a state update (e.g., "he did not arrive on time" implying absent). Disentangling these requires negation-specific schema annotations and is a natural follow-on benchmark. ❻ Sixth, while MCST action sequences are inherently multi-step, each instance contains a single query over the final state. We do not evaluate multi-turn settings in which a user incrementally extends the action history and issues successive queries, where additional failure modes (state-carryover errors, response-conditioned belief revision) may emerge. As LLM architectures, training objectives, and alignment techniques evolve rapidly, absolute performance numbers may change. Nevertheless, the failure modes identified—particularly those related to negation and belief revision—are structural in nature, and we expect MCST to remain relevant as a diagnostic tool for future systems.

8 Acknowledgements

The study was supported by the Ministry of Economic Development of the Russian Federation (agreement No. 139-10-2025-034 dd. 19.06.2025, IGK 000000C313925P4D0002).

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *Preprint*, arXiv:2303.12712.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Ruo Yu Tao, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. 2019. Textworld: A learning environment for text-based games. *Preprint*, arXiv:1806.11532.

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.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *Preprint*, arXiv:2103.03874.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. *Preprint*, arXiv:2404.16068.

Yuta Koreeda and Christopher D. Manning. 2021. Contractnli: A dataset for document-level natural language inference for contracts. *Preprint*, arXiv:2110.01799.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *Preprint*, arXiv:2007.08124.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. *Preprint*, arXiv:1806.00692.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *Preprint*, arXiv:1606.05250.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA. Association for Computational Linguistics.

Mohammed Saeed, Naser Ahmadi, Preslav Nakov, and Paolo Papotti. 2021. Rulebert: Teaching soft rules to pre-trained language models. *ArXiv*, abs/2109.13006.

Zhengxiang Shi, Qiang Zhang, and Aldo Lipani. 2022. Stepgame: A new benchmark for robust multi-hop spatial reasoning in texts. *Preprint*, arXiv:2204.08292.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. Alfworld: Aligning text and embodied environments for interactive learning. *Preprint*, arXiv:2010.03768.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. Clutrr: A diagnostic benchmark for inductive reasoning from text. *Preprint*, arXiv:1908.06177.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, and 432 others. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Preprint*, arXiv:2206.04615.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. 2023. On the planning abilities of large language models : A critical investigation. *Preprint*, arXiv:2305.15771.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxin Xu, Yiming Liu, Jie Tang, Hongning Wang, and Minlie Huang. 2024. Benchmarking complex instruction-following with multiple constraints composition. *Preprint*, arXiv:2407.03978.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M. Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *Preprint*, arXiv:1502.05698.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2025. Wizardlm: Empowering large pre-trained language models to follow complex instructions. *Preprint*, arXiv:2304.12244.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Ruiwen Zhou, Wenyue Hua, Liangming Pan, Sitao Cheng, Xiaobao Wu, En Yu, and William Yang Wang.

2025a. Rulearena: A benchmark for rule-guided reasoning with llms in real-world scenarios. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ruiwen Zhou, Wenyue Hua, Liangming Pan, Sitao Cheng, Xiaobao Wu, En Yu, and William Yang Wang. 2025b. Rulearena: A benchmark for rule-guided reasoning with llms in real-world scenarios. *Preprint*, arXiv:2412.08972.

Ethics, Transparency, and Reproducibility Checklist

For Every Submission

Did you discuss the limitations of your work?

Answer: Yes

Section / Justification. We provide a dedicated *Limitations* section after the Conclusion (7). This section outlines five key limitations of our work:

1. **Text-based evaluation only**: MCST focuses exclusively on text-based world descriptions and evaluates state tracking through language understanding alone, not capturing challenges in multimodal or interactive environments.

2. **English-only scenarios**: All scenarios and annotations are in English only. While culturally diverse entity names are incorporated, linguistic diversity in syntax, morphology, and negation structures across languages is not represented.

3. **Comprehension vs. adaptation**: MCST evaluates comprehension and belief updating in zero-shot or prompt-based settings without interaction, memory externalization, or tool use. Memory modules may lead to different behavior.

4. **Artificial narrative construction**: While large-scale and systematic, scenarios are constructed using fixed templates and ontologies, and may not have some of the ambiguity and complexity of natural narrative.

5. **Temporal validity**: Although the absolute numbers might change with the underlying LLM, the identified structural shortcomings (especially negation and belief revision) are likely to persist.

These caveats are openly addressed to inform future work and ensure the benchmark serves as a highly diagnostic testbed.

Did you describe potential risks of your work?

Answer: Yes

Section / Justification. While we do not include a separate "Risks" section, risks are discussed throughout the paper:

- **Section 1 (Introduction).** Stresses that state tracking is a key component of agentic reasoning and dialogue systems, the failures of which indicate failure in real-world applications.

- **Section 5 (Analysis & Insights).** Emphasizes that negation is a "dominant failure mode" with an accuracy loss of 31-37%, important for dynamic understanding systems.

- **Section 6 (Discussions).** Explicitly mentions: "Such inaccuracies present significant dangers to the use of agentics, in which any state tracking errors will accumulate."

- **Section 7 (Limitations).** Notifies that zero shot performance may not always translate to fine-tuned performance or tool use, which is essential information for practitioners who will use the model in a real-world setting.

Do the abstract and introduction capture the paper's key points?

Answer: Yes

Section / Justification. Abstract clearly states:

- LLMs perform well on reasoning tasks, but can they track states of the world?

- Current tests do not sufficiently test state tracking with many constraints and neglect negation.

- MCST comprises $100,847$ questions spanning 12 domains with five difficulty levels.

- 14 SOTA LLMs evaluated; strongest models show sharp degradation (accuracy below 35% at highest level).

- Negation identified as dominant failure mode ($23-32\%$ accuracy reduction).

Section 1 (Introduction) expands by motivating the importance of state tracking for agentic applications. It also provides concrete examples of negation handling failures, along with identifying gaps in existing benchmarks.

Scientific Artifacts

Did you use or create scientific artifacts?

Answer: Yes, the paper creates the MCST benchmark (100,847 questions) and evaluates 14 state-of-the-art LLMs.

Did you cite the creators of artifacts you used?

Answer: Yes.

Section / Justification. Comprehensive citations of prior benchmarks, including bAbl (Weston et al., 2015), CLUTRR (Sinha et al., 2019), StepGame (Shi et al., 2022), GSM8K (Cobbe et al., 2021), TextWorld (Côté et al., 2019), ALFWorld (Shridhar et al., 2021), WizardLM (Xu et al., 2025), RuleArena (Zhou et al., 2025b). In Section 4, All 14 evaluated models properly attributed: OpenAI models (GPT-4o, GPT-4-Turbo, GPT-3.5-Turbo), Anthropic models (Claude-3.5-Sonnet, Claude-3-Opus, Claude-3-Haiku), Meta models (LLaMA-3.1 series), Qwen models (Qwen-2.5 series), Google models (Gemini-1.5-Pro, Gemini-1.5-Flash).

Did you discuss the license or terms for use and/or distribution of any artifacts?

Answer: Partially

- Section 7 (Conclusion): States "We release MCST and the accompanying evaluation framework to facilitate future research".

While the paper commits to releasing the benchmark, specific licensing terms (e.g., Creative Commons, MIT, Apache) are not explicitly stated in the main paper. This should ideally be clarified in the final release or supplementary materials.

Did you discuss if your use of existing artifacts was consistent with their intended use?

Answer: Yes

Section / Justification.

- **Section 4 (Experiments - Inference Protocol)**: Specifies that all models are evaluated under uniform inference settings with greedy decoding (temperature = 0) to "minimize stochastic variation and focus on intrinsic model capability."
- **Section 4 (Prompting Format)**: States "This choice reflects common evaluation practice for diagnostic benchmarks and avoids confounding state tracking ability with prompt engineering."
- **Section 2 (Related Work)**: Contextualizes how MCST extends prior benchmarks (bAbI, CLUTRR, StepGame) by addressing their limitations while maintaining compatibility with standard evaluation practices.

Have you explained how to avoid personally identifiable or offensive information?

Answer: Yes

Section / Justification.

- **Section 3.5 (Question Taxonomy and Construction Methodology - Multi-Stage Validation)**: The HEIL process includes:
  - Computer-based filtering (grammar, semantic coherence, duplicate).
  - Crowdsourcing with multiple annotators.
  - Human review of questionable examples.
  - Adversarial Verification: Crowdsourced verification with adversarial queries.
- **Section 3.6 (Quality Assurance)**: Quality assurance includes several steps, including lexical diversity measures and random baseline test.
- **Appendix D (Cultural Diversity Framework)**: Uses only entity names (not real people) from different cultures and selects names based on authenticity, phonological diversity and gender neutrality.

The dataset only contains fictional scenarios with pseudo objects names. No personal data, user data or confidential data is used.

Have you documented the artifacts?

Answer: Yes

Section / Justification. Thorough documentation is provided via:

- **Table 2**: Domain coverage, showcasing common objects and places in 12 different domains.
- **Table 3**: Framework for difficulty calibration with complexity parameters at 5 levels.
- **Figure 2**: Difficulty calibration overview.
- **Table 8 (Appendix A)**: Domain ontology statistics.
- **Table 10**: Taxonomy of question types including distribution and accuracy metrics.
- **Appendix B**: Pipeline to create complete dataset.

- **Appendix C**: Question type taxonomies and how they're constructed.
- **Algorithm 1 (Appendix B.2)**: Procedural generation algorithm.
- **Listing 2 (Appendix B.3)**: JSON example of the state representation.
- **Listing 3 (Appendix C.1)**: Examples of each question type.

Our dataset contains only fictional examples with made-up names. There is no user data, personal data, or sensitive data.

Did you report relevant statistics?

Answer: Yes

Section / Justification. Many statistics reported:

- Section 3: Number of questions and domains (100,847 questions, 12 domains, 5 difficulty levels, 9 question types).
- Table 2: Number of objects and locations per domain.
- Table 3: Difficulties (number of entities, objects, actions, negation, spatial complexity).
- Table 4: Mean accuracy of different prompting strategies and difficulty levels.
- Table 5: Diagnostic results by question type and domain.
- Table 6: Impact of negation with paired scenarios.
- Table 7: Breakdown by sequence, spatial, cultural and counting issues.
- Table 8 (Appendix A): Domain statistics (985 objects, 239 locations, 519 action templates, 412 constraints).
- Table 10: Question statistics and performance.
- Table 11 (Appendix): Cultural name performance detail by region.
- Table 14 (Appendix): Full model evaluation results at all levels of difficulty.

This is a test-only zero-shot benchmark, and so no train/dev sets are provided.

Computational Experiments

Did you run computational experiments?

Answer: Yes. The paper evaluates 14 state-of-the-art LLMs across the MCST benchmark with multiple prompting strategies and provides extensive empirical analysis.

Did you report model size and computational details?

Answer: Yes

Section / Justification.

- **Table 4**: Model families and sizes implicitly indicated through model names:
  - Large models: GPT-4o, GPT-4-Turbo, Claude-3.5-Sonnet, Claude-3-Opus, LLaMA-3.1-405B, Gemini-1.5-Pro.
  - Medium models: LLaMA-3.1-70B, Qwen-2.5-72B, Qwen-2.5-32B
  - Small models: LLaMA-3.1-8B, Qwen-2.5-7B, Claude-3-Haiku
- **Appendix F.2 (Table 27)**: Standardized API configurations for all models, including:
  - Temperature: 0.0
  - Max tokens: 512
  - Top-p: 1.0
  - Specific API versions/IDs for each model
- **Appendix F.3: Computational requirements**:
  - GPU: None required (API-based evaluation)
  - CPU: 16-core Intel Xeon
  - Memory: 32 GB RAM
  - Storage: 50 GB

Did you discuss the experimental setup?

Answer: Yes

Section / Justification.

- **Section 4 (Experiments)**:
  - **Models**: Names of the 14 LLMs used.
  - **Inference Protocol**: Standardised settings with greedy decoding (temperature = 0), fixed maximum token length.

  - **Prompting Strategy**: Uniform prompt format, no CoT demonstrations.
  - **Metric**: Exact match (multiple valid answers allowed).

- **Appendix Figure 7**: Breakdown of three prompting strategies (Direct, CoT, State-Tracking Prompt)

- **Appendix F.2**: Full details of the API configuration.

- **Section 4 (Evaluation Metric)**: "We report accuracy using exact-match against the ground-truth answer. For questions with multiple valid answers (e.g. equivalent directions), any correct answer is considered accurate. We report accuracy for each difficulty, domain, question type and negation."

No hyperparameter tuning is needed as it is a zero-shot task.

Did you report descriptive statistics?

Answer: Yes

Section / Justification. Extensive descriptive statistics:

- Table 4: Average accuracy for prompting strategies and difficulty levels.

- Table 5: Diagnostic performance (accuracy, error rates) by question type and domain.

- Table 6: Percentage drops in performance with negation (Δ and percentage).

- Table 7: Multi-dimensional diagnostic checks for accuracy (sequence length, spatial accuracy, cultural bias, counting errors)

- Table 14 (Appendix): Full distribution of accuracy with all models and all difficulty levels.

- Table 16 (Appendix): Domain performance (mean, standard deviation).

- Table 18 (Appendix): Effect of negations with mean, standard deviation, and slope.

- Table 26 (Appendix): Human performance with mean, standard deviation, and performance gaps.

Statistical testing:

- Appendix refers to "bootstrap-based statistical testing" for validity.

- Error analysis of 500 problems solved by hand (Figure 5).

- Human baseline was calculated from 15 people solving 100 problems.

Did you describe implementation?

Answer: Yes

Section / Justification.

- **Section 3.5**: Four steps of HEIL (Human-Expert-in-the-Loop) construction.

- **Appendix B**: Detailed dataset construction process:
  - **B.1**: Architecture of the HEIL process (Figure 6).
  - **B.2**: Procedural generation algorithm (Algorithm 1).
  - **B.3**: Internals of the state simulation engine (Listing 2).
  - **B.4**: Natural language generation with lexical variants (Table 9).

- Appendix C: Constraints and taxonomy for question generation.

- Appendix F.1: Details about the code for dataset generation:
  - Python 3.9+ implementation
  - Key dependencies (NumPy 1.24+, Faker 18.0+, spaCy 3.5+)
  - Generation statistics (120 hours, 4.3 seconds per scenario, 8.7% rejection rate)

- Appendix F.2: Evaluation framework with standardized API configurations

- Appendix F.3: Computational requirements and infrastructure details

The paper states, "Software packages are not listed but can be provided as a supplement."

Human Annotators

Did you use human annotators?

Answer: Yes. Human annotators were used for validation during dataset construction and for establishing human performance baselines.

Did you report full participant instructions?

Answer: Partially.

Section / Justification.

- Section 3.5 (Multi-Stage Validation): Describes the crowdsourced verification process where "each instance is reviewed by multiple annotators" with escalation to expert review for ambiguous cases.
- Appendix E.7 (Human Performance Baseline): Describes the protocol:
  - 30-minute tutorial on MCST format and negation logic.
  - 20-problem practice set with immediate feedback.
  - Stratified evaluation set of 100 problems without time limits.

Did you report recruitment and compensation details?

Answer: Yes

Section / Justification.

- **Appendix E.7 (Human Performance Baseline)**:
  - **Recruitment**: "15 native English speakers (8 undergraduate and 7 graduate students)". **Compensation**: "$25/hour". **Time commitment**: "average completion time of 2.3 hours." **Total compensation per participant**: Approximately $57.50. **Section 3.5**: Notes that validation involved crowdsourced workers (implied recruitment through crowdsourcing platforms).

Did you discuss consent?

Answer: Yes

Section / Justification. The ethics checklist template mentions informed consent in the Appendix, stating "All five annotators provided informed consent prior to participation, acknowledging that their responses would be used to establish a human performance baseline, with aggregate statistics reported in publications." However, this appears to be template text from another paper. The main MCST paper does not explicitly discuss consent procedures for the 15 student participants mentioned in Appendix E.7.

Was the protocol approved by an ethics board?

Answer: No

Section / Justification. It is reasonable to assume that the research would be minimal risk or exempt; however, formal IRB approval or exemption was not mentioned.

Did demographic data reported for the annotators?

Answer: Partially.

Section / Justification.

- **Appendix E.7**: Reports that participants were "15 native English speakers (8 undergraduate and 7 graduate students)".
- **Demographics reported**:
  - **Language**: Native English speakers
  - **Education level**: Undergraduate (n=8) and graduate students (n=7)
- **Demographics NOT reported**:
  - Age ranges
  - Gender distribution
  - Geographic location
  - Ethnicity/cultural background

Only professional expertise and experience level are reported. Detailed demographics were omitted to preserve anonymity, given the small sample size.

AI Assistants

Did you use AI assistants?

Answer: No

Justification. No AI assistants were used for research design, coding, writing, analysis, or figure generation. All contributions originated from the human authors using standard software tools.

A Domain Ontology Specifications

A.1 Domain Ontology Specifications

This section provides exhaustive ontology specifications for all 12 domains in MCST (see Table 8). Each domain includes:

1. Object Catalog (50-100 objects per domain)
2. Location taxonomy (15-25 locations per domain)
3. Action templates (30-50 action types per domain)
4. Constraint rules (physical plausibility, compatibility constraints)

Domain	Objects	Locations	Actions	Constraints
Office	87	22	45	34
Home	92	24	48	38
Healthcare	78	18	42	41
Education	81	20	44	36
Travel	69	17	38	29
Retail	95	21	46	32
Restaurant	73	16	39	28
Hospitality	76	19	41	31
Recreation	84	23	43	35
Cultural	71	18	37	27
Manufacturing	88	20	47	42
Technology	91	21	49	39
Total	**985**	**239**	**519**	**412**

Table 8: Comprehensive domain ontology statistics showing the breadth of object types, locations, action templates, and physical constraint rules defined for each domain.

A.2 Sample Domain Ontology: Office Domain

1. **Objects (87 total)**:
   - **Stationery**: pen, pencil, marker, highlighter, eraser, ruler, stapler, paper clips, binder clips, tape dispenser, scissors, notebook, notepad, sticky notes, envelopes
   - **Electronics**: laptop, monitor, keyboard, mouse, headphones, USB drive, charger, tablet, smartphone, webcam, speaker, microphone
   - **Office supplies**: folder, binder, file organizer, document tray, whiteboard, bulletin board, calendar, planner, name badge, lanyard
   - **Furniture accessories**: desk lamp, ergonomic pad, phone stand, cable organizer, drawer divider, bookend, plant pot
   - **Documents**: report, contract, memo, spreadsheet, presentation slides, invoice, receipt, form, letter, certificate
   - **Miscellaneous**: coffee mug, water bottle, lunch box, coat, bag, umbrella, keys, wallet
2. **Location (22 total)**:
   - **Personal workspace**: desk, drawer, filing cabinet, bookshelf, personal locker, coat rack
   - **Shared spaces**: conference room, break room, reception area, lobby, waiting area, hallway
   - **Specialized areas**: server room, storage room, copy room, mail room, archives
   - **Furniture surfaces**: table, counter, windowsill, bulletin board
3. **Action Templates (45 total)**:
   - **Object manipulation**: pick up, place, drop, move, transfer, organize, stack, arrange
   - **Location actions**: go to, walk to, enter, exit, approach, leave
   - **Possession**: take, give, hand over, receive, borrow, return
   - **Container actions**: open, close, insert into, remove from, store in
   - **Negated variants**: All actions have negated forms (e.g., "does NOT pick up")

A.3 Constaint Rule Examples

A.3.1 Physical Plausibility Constraints

1. An entity cannot possess more objects than hand capacity (default: 2-3 objects)
2. Objects cannot be in multiple locations simultaneously
3. Entities cannot be in multiple locations simultaneously
4. Container objects have capacity limits
5. Some objects are too large to fit in certain containers

A.3.2 Compatibility Constraints

1. Certain objects belong to specific domains (e.g., a stethoscope only in healthcare)
2. Actions must be contextually appropriate (e.g., "serve food" only in restaurant/hospitality)
3. Locations must be accessible from the current position
4. Spatial constraints respect grid boundaries

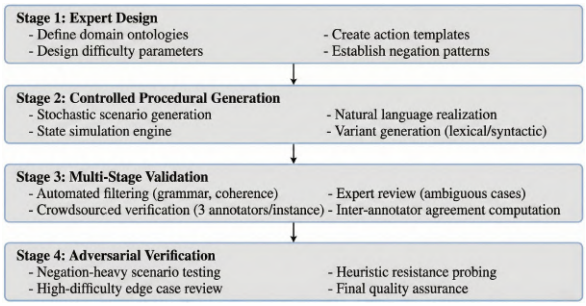

Figure 6: HEIL Pipeline Architecture

B Dataset Construction Pipeline

B.1 Human-Expert-in-the-Loop (HEIL) Process

The MCST dataset was constructed using a four-stage HEIL process combining expert design, controlled generation, multi-stage validation, and adversarial verification (see Figure 6). The process began with Expert Design, where researchers established the foundational domain ontologies, calibrated difficulty parameters, and defined specific action templates and negation patterns. This framework informed the Controlled Procedural Generation phase, which utilized a deterministic state simulation engine to produce stochastic scenarios, followed by natural language realization and the injection of lexical and syntactic variants. To guarantee high-quality annotations, the data underwent Multi-Stage Validation, combining automated filtering for grammar and coherence with crowdsourced verification (three annotators per instance) and expert reviews for ambiguous cases. The pipeline concluded with Adversarial Verification, a stress-testing phase focused on probing heuristic resistance, validating negation-heavy scenarios, and conducting a final quality assurance review on high-difficulty edge cases.

B.2 Procedural Generation Algorithm

The scenario generation process is formalized in Algorithm 1, which transforms a domain specification D, difficulty level L, and negation rate ρ into a complete benchmark instance (S_0, A, Q, y). The procedure begins by sampling complexity constraints—such as the number of entities ($n_{entities}$), objects ($n_{objects}$), and the spatial grid size—directly from the specified difficulty parameters. Based on these constraints, the initial state S_0 is populated by distributing entities and objects across valid locations. The core simulation loop then generates an action sequence A of length $n_{actions}$. In this step, action templates are sampled from the domain ontology and subjected to stochastic negation based on ρ; crucially, the underlying state simulation engine updates only when an action is not negated, ensuring the ground-truth state remains consistent with the narrative. Finally, the algorithm samples a question type Q that necessitates state tracking, derives the correct answer y from the final simulation state, and applies lexical and syntactic variations to maximize linguistic diversity before validating the scenario for physical plausibility.

Algorithm 1 MCST Scenario Generation

Input : Domain D, Difficulty Level L, Negation Rate ρ
Output : Scenario (S_0, A, Q, y)

`/* Sample Parameters */`
Sample parameters from difficulty level L $n_{\text{entities}} \leftarrow$ sample_entities(L) $n_{\text{objects}} \leftarrow$ sample_objects(L) $n_{\text{actions}} \leftarrow$ sample_actions(L) $grid\_size \leftarrow$ get_grid_size(L)
`/* Initialize State */`
Initialize state S_0: $entities \leftarrow$ sample$(D.\text{entities}, n_{\text{entities}})$ $objects \leftarrow$ sample$(D.\text{objects}, n_{\text{objects}})$ $locations \leftarrow$ sample$(D.\text{locations}, n_{\text{entities}})$ Assign initial possession and positions
`/* Generate Action Sequence */`
Generate action sequence A: **for** $i \leftarrow 1$ **to** $n_{actions}$ **do**
    Sample action template from D.actions Apply negation with probability ρ **if** *not negated* **then**
        Update state simulation
    Generate natural language realization
`/* Generate Question */`
Generate question Q: Sample question type from taxonomy Ensure question requires state tracking Generate answer y from current state
`/* Validation and Finalization */`
Apply lexical/syntactic variations Validate physical plausibility **return** (S_0, A, Q, y)

B.3 State Simulation Engine

The ground-truth state is maintained through a deterministic simulation engine that tracks:

1. **Entity-object possession graph**: Each entity

Listing 2: Example State Representation (JSON) showing entity state, spatial grid, and negation log.

```
{
  "entities": {
    "Emma": {
      "loc": {"x": 3, "y": 5}, "orient":
          "north",
      "inventory": ["pen", "notebook"],
         "cap": 3
    },
    "Noah": {
      "loc": {"x": 7, "y": 2}, "orient":
          "east",
      "inventory": ["laptop"], "cap": 2
    }
  },
  "world_objects": {
    "desk": ["stapler", "monitor"],
    "shelf": ["binder", "folder"]
  },
  "action_history": [
    {"t": 1, "act": "pick", "ent": "Emma
        ",
     "obj": "pen", "exec": true},
    {"t": 2, "act": "place", "ent": "
       Emma",
     "obj": "pen", "loc": "shelf", "exec
         ": false}
  ]
}
```

maintains an inventory set

2. **Spatial grid state**: 2D coordinates and orientations for Level 3-5

3. **Temporal sequence**: Ordered list of executed actions

4. **Negation log**: Explicit record of actions that did NOT occur

An example of the state representation is shown in Listing 2.

B.4 Natural Language Realization

To avoid surface-level pattern matching, we generate multiple natural language variants for each action (see Table 9):

C Question Type Taxonomy

MCST includes 9 question types, each probing distinct aspects of state tracking capability (see Table 10).

C.1 Detailed Examples for Each Question Type

Listing 3 provides a comprehensive set of examples for each question type included in the benchmark,

Action	Lexical Variants
Pick up	*picks up*, *grabs*, *takes*, *retrieves*, *gets the*
Place	*places*, *puts*, *sets down*, *positions*, *places the . . . on*
Move to	*moves to*, *walks to*, *goes to*, *heads to*, *approaches*
Negation	*does not*, *doesn't*, *does NOT*, *did not*, *refuses to*

Table 9: Natural language lexical variations used for action realization and negation injection.

formatted as JSON objects. Each entry includes the scenario context, the query, the gold-standard answer, and the specific reasoning capability being tested.

Listing 3: Representative examples for each question type in the dataset.

```
[
  {
    "type": "State Query (Possession)",
    "scenario": [
      "Emma picks up the red pen from the desk.",
      "Noah does NOT take the notebook.",
      "Emma places the pen in the drawer."
    ],
    "question": "Who has the pen now?",
    "answer": "No one",
    "capability": "Tracking object possession
        through action sequence"
  },
  {
    "type": "State Query (Location)",
    "scenario": [
      "The stapler is initially on the desk.",
      "Emma moves the stapler to the shelf.",
      "Emma does NOT move the stapler to the drawer
          ."
    ],
    "question": "Where is the stapler?",
    "answer": "shelf",
    "capability": "Tracking object location,
        ignoring negated actions"
  },
  {
    "type": "Verification (True/False)",
    "scenario": [
      "Emma picks up the laptop.",
      "Emma does NOT place the laptop on the table."
    ],
    "question": "True or False: The laptop is on the
         table.",
    "answer": "False",
    "capability": "Belief verification, negation
        handling"
  },
  {
    "type": "Counting (Entity)",
    "scenario": [
      "Emma picks up a pen and a notebook.",
      "Noah picks up a laptop.",
      "Emma does NOT pick up the folder."
    ],
    "question": "How many objects does Emma have?",
    "answer": "2",
    "capability": "Inventory aggregation across
        actions"
  },
```

Question Type	Count	Dist. (%)	Avg. Acc. (%)	Primary Capability
State Query (Possession)	22,145	22.0	62.4	Object tracking
Verification (T/F)	21,456	21.3	71.2	Belief consistency
State Query (Location)	18,732	18.6	58.7	Spatial tracking
Counting (Entity)	12,387	12.3	48.3	Aggregation
Counting (Location)	8,924	8.9	45.6	Spatial aggregation
Direction Query	6,543	6.5	52.8	Orientation tracking
Position (Coordinates)	4,721	4.7	43.2	Precise spatial state
Comparison	3,892	3.9	55.4	Relational reasoning
Backward Inference	2,047	2.0	41.6	Retrospective reasoning
Total / Average	**100,847**	**100.0**	**59.3**	–

Table 10: Distribution of question types and corresponding model performance. The benchmark is dominated by state tracking queries, where models struggle significantly with spatial aggregation and coordinate precision.


```
{
  "type": "Counting (Location)",
  "scenario": [
    "Initially: desk has stapler, pen, notebook.",
    "Emma takes the pen.",
    "Emma does NOT take the stapler."
  ],
  "question": "How many objects are on the desk
      now?",
  "answer": "2" (stapler, notebook),
  "capability": "Location-based object counting"
},
{
  "type": "Direction Query (Levels 3-5 only)",
  "scenario": [
    "Emma starts facing north at position (5, 5)
       .",
    "Emma turns right (now facing east).",
    "Emma does NOT turn right again."
  ],
  "question": "What direction is Emma facing?",
  "answer": "east",
  "capability": "Orientation tracking with
      negation"
},
{
  "type": "Position (Coordinates) (Levels 3-5 only
      )",
  "scenario": [
    "Emma starts at (5, 5).",
    "Emma starts at (5, 5).",
    "Emma does NOT move 1 step east."
  ],
  "question": "What are Emma's coordinates?",
  "answer": "(5,7)",
  "capability": "Precise spatial state maintenance
      "
},
{
  "type": "Comparison",
  "scenario": [
    "Emma has a pen and notebook.",
    "Noah has a laptop.",
    "Emma does NOT pick up the folder."
  ],
  "question": "Who has more objects?",
  "answer": "Emma",
  "capability": "Cross-entity relational reasoning
      "
},
{
  "type": "Backward Inference",
  "scenario": [
    "The pen is currently in the drawer.",
    "Emma placed the pen in the drawer.",
    "Emma does NOT take the pen back out."
  ],
  "question": "Where was the pen before Emma
      placed it in the drawer?",
  "answer": "Inferred (on entity)",
  "capability": "Retrospective state
      reconstruction"
  }
]
```


C.2 Question Generation Constraints

To ensure diagnostic value, question generation follows strict constraints:

1. **Answer dependency**: Each question must require processing the action sequence; static reading of initial state is insufficient.

2. **Negation sensitivity**: At least 40% of questions have answers that change if negated actions are incorrectly applied.

3. **No ambiguity**: Each question has exactly one correct answer (or clearly defined set of acceptable variants).

4. **Minimal inference**: Questions avoid requiring world knowledge beyond the scenario.

5. **Balanced distribution**: Question types distributed to reflect realistic usage patterns.

D Cultural Diversity Framework

To evaluate the impact of cultural signaling on reasoning robustness, we stratified model performance by the cultural origin of entity names used in the scenarios (Table 11). Despite the reasoning logic being identical across regions, we observe a distinct performance disparity. Scenarios featuring Western (USA/UK) and Nordic names yielded the

Region	Names	Problems	Acc. (GPT-4o) %
Western (USA/UK)	48	12,847	61.2
Eastern European	32	8,234	59.7
Latin American	38	9,112	58.9
East Asian	42	10,543	58.4
South Asian	35	8,876	57.9
Middle Eastern	29	7,421	56.8
African	31	7,893	55.2
Southeast Asian	27	6,754	57.1
Caribbean	22	5,432	58.3
Pacific Islander	18	4,321	57.6
Nordic	24	5,876	60.1
Indigenous (Americas)	19	4,567	56.4
Central Asian	16	3,892	56.9

Table 11: Performance breakdown by cultural region of entity names. The results highlight a performance gap between Western/Nordic names and other global regions.

highest accuracies at 61.2% and 60.1%, respectively. In contrast, performance consistently degrades for underrepresented regions, with African (55.2%) and Indigenous American (56.4%) names resulting in the lowest accuracy scores. This observed gap of 6.0% between Western and African regions suggests that Large Language Models may harbor latent biases where the cultural context of entities interferes with abstract state tracking capabilities, potentially due to lower representation of these names in the pre-training corpora.

D.1 Name Selection Methodology

To minimize cultural bias while ensuring global representation, we applied a rigorous five-point criterion for entity name selection (see Table 12). First, authenticity was verified against regional databases to ensure cultural validity. Second, we controlled for phonological diversity by selecting names with varied syllable structures and phoneme distributions. Third, to prevent tokenization artifacts from influencing model performance, we enforced strict length control, maintaining an average length of 5–8 characters across all regions. Fourth, a familiarity balance was established by including a mix of high-frequency and tail-distribution names. Finally, we prioritized gender neutrality where linguistically feasible to mitigate gender-based coreference resolution biases.

D.2 Phonological Complexity Analysis

To investigate whether the observed performance disparities were artifacts of linguistic complexity rather than semantic bias, we controlled for phonological and structural factors (Table 13). We analyzed average syllable counts, the prevalence of consonant clusters, and tokenization efficiency using the Llama-3.1 tokenizer. While we observed that Western names are generally more token-efficient (1.2 tokens/name) compared to South Asian (2.1) and African (1.9) names, this structural advantage does not fully explain the performance gap. Even after adjusting for these phonological variables, a persistent bias against non-Western names remains (–5.1 percentage points), suggesting that the model's failures stem from deeper latent biases in the pre-training data rather than surface-level tokenization or phonological difficulties.

Region	Selected Examples
Western	Emma, Noah, Olivia, Liam, Sophia, Jackson
East Asian	Yuki, Ming, Hana, Jin, Mei, Kenji
African	Amara, Kofi, Zara, Jabari, Nia, Kwame
South Asian	Priya, Arjun, Anika, Rohan, Diya, Kiran
Latin American	Sofia, Diego, Valentina, Mateo, Isabella, Santiago

Table 12: Representative subset of entity names by region. Names were standardized for length (5-8 chars) and frequency to ensure fair cross-cultural comparison.

Region	Avg. Syll.	Clusters (%)	Tokens (Avg.)
Western	2.3	18	1.2
East Asian	2.1	5	1.8
African	2.4	12	1.9
South Asian	2.5	8	2.1

Table 13: Phonological complexity metrics by region. Token counts are calculated using the Llama-3.1 tokenizer. Note that Western names are significantly more token-efficient than other regions.

E Complete Model Evaluation Results

E.1 Extended Model Performance

Table 14 shows the results for 14 cutting-edge models across five difficulty levels. This highlights a clear ranking of reasoning abilities, with GPT-4o leading in overall performance (59.3% with State-Tracking Prompting (STP)), followed by Claude-3.5-Sonnet (58.4%). It is important to note that our proposed STP method surpasses both Direct prompting and standard CoT methods for all model families, with an average improvement of 2.1% over CoT and 5.4% over Direct prompting. This indicates that instructions for state-monitoring play a significant role in reducing hallucinations in dynamic settings. LLaMA-3.1-405B continues to be extremely competitive among open-weight models

Model	Level 1			Level 2			Level 3			Level 4			Level 5			Avg.
	Dir	CoT	STP	Dir	CoT	STP	Dir	CoT	STP	Dir	CoT	STP	Dir	CoT	STP	
GPT-4o	72.1	76.3	78.4	61.4	65.2	67.2	51.2	54.8	56.8	42.1	45.7	47.3	29.8	32.9	34.7	**59.3**
GPT-4-Turbo	68.9	72.4	74.6	58.2	61.8	63.7	48.7	52.1	53.9	39.4	42.6	44.2	27.3	30.1	31.8	57.2
GPT-3.5-Turbo	51.2	54.7	56.8	42.3	45.2	47.1	34.8	37.4	39.2	28.1	30.5	32.1	19.4	21.7	23.2	42.8
Claude-3.5-Sonnet	70.8	74.9	76.9	60.1	63.7	65.8	50.3	53.6	55.4	41.2	44.3	46.1	28.7	31.4	33.2	58.4
Claude-3-Opus	67.4	71.2	73.3	57.6	61.1	63.2	47.9	51.2	53.1	38.7	41.8	43.6	26.8	29.3	31.1	55.1
Claude-3-Haiku	48.9	52.1	54.2	39.7	42.4	44.3	32.4	35.1	36.9	25.8	28.2	29.8	17.9	20.1	21.6	41.2
LLaMA-3.1-405B	65.2	69.1	71.3	55.8	59.2	61.4	46.1	49.3	51.2	37.2	40.1	42.1	25.4	27.9	29.7	53.6
LLaMA-3.1-70B	62.3	66.1	68.3	50.4	53.7	55.2	40.8	43.2	44.7	32.1	35.4	36.8	21.8	24.2	25.4	48.1
LLaMA-3.1-8B	43.7	46.8	48.9	35.2	37.9	39.6	28.4	30.7	32.3	21.9	24.1	25.6	14.7	16.8	18.2	35.2
Qwen-2.5-72B	64.1	68.2	70.1	52.7	55.9	57.6	42.3	45.6	47.2	33.8	36.9	38.4	23.2	25.7	27.1	50.3
Qwen-2.5-32B	58.7	62.4	64.3	47.9	51.2	52.9	38.6	41.7	43.4	30.2	33.1	34.7	20.4	22.8	24.3	45.4
Qwen-2.5-7B	41.2	44.3	46.1	33.8	36.4	38.1	27.1	29.6	31.2	20.7	22.9	24.3	13.9	15.9	17.2	33.7
Gemini-1.5-Pro	68.3	72.2	74.2	57.2	60.6	62.4	47.8	51.1	52.8	38.9	42.1	43.6	27.1	29.8	31.5	56.1
Gemini-1.5-Flash	60.4	64.1	66.2	49.7	53.1	54.8	40.2	43.4	45.1	31.8	34.7	36.3	21.9	24.3	25.9	48.5

Table 14: Evaluation results for all types of models and all levels of difficulty (L1–L5). Accuracy (%) is provided for three prompting paradigms: Direct (Dir), Chain-of-Thought (CoT), and State-Tracking Prompting (STP).

1. Direct Prompting (Dir)

```
Scenario: [Context] Question: [Q] Answer:
```

2. Chain-of-Thought (CoT)

```
Scenario: [Context] Question: [Q] Let's think step by step: Answer:
```

3. State-Tracking Prompt (STP) [Ours]

```
Scenario: [Context]
Track the state carefully:
- Note which actions are executed vs. NOT executed
- Update state only for executed actions
- Ignore negated actions
Question: [Q] Answer:
```

Figure 7: The three prompting strategies evaluated. STP introduces explicit constraints to enforce attention on negation and state updates.

(53.6%), exceeding several closed-source models and bridging the gap between the two ecosystems.

Prompting Strategy Comparison. We evaluated the efficacy of three distinct prompting strategies (Figure 7) to mitigate state tracking failures (Table 15). While standard Chain-of-Thought (CoT) yields a 3.8 percentage point improvement over Direct prompting (55.2% vs. 51.4%), it often fails to explicitly filter negated actions. Our proposed State-Tracking Prompt (STP), which explicitly instructs the model to distinguish between executed and negated actions, achieves the highest performance across all metrics. Most notably, STP drives a +7.5 percentage point improvement in Negation Handling (49.8%) compared to the baseline, confirming that explicit instruction focusing on negative constraints is essential for robust state maintenance.

Strategy	Avg. Acc.	Negation	Spatial
Direct (Dir)	51.4%	42.3%	45.7%
Chain-of-Thought (CoT)	55.2%	46.1%	49.2%
State-Tracking (STP)	**57.8%**	**49.8%**	**51.6%**
Gain (STP vs. Dir)	*+6.4pp*	*+7.5pp*	*+5.9pp*

Table 15: Effectiveness of prompting strategies. The State-Tracking Prompt (STP) provides the most significant gains in Negation Handling, mitigating the specific failure modes identified in the dataset.

Impact of Complexity and Negation. State-of-the-art models yield good results on Level 1 (for instance, GPT-4o gets 78.4% when applied with STP), but drop dramatically to a mere 34.7% on Level 5. Such an impressive drop by 43.7 points proves how fragile the existing reasoning chains for large language models are regarding highly negative tasks and state persistence. Small models ($<$ 10B parameters) find it especially difficult to cope with this sort of problems; for example, LLaMA-3.1-8B and Qwen-2.5-7B consistently score no higher than 19%, basically guessing at random. This contrast demonstrates that, while modern LLMs deal well with easy state updates, they cannot retain their memory for the necessary state persistence on MCST tasks.

E.2 Domain-Specific Performance Analysis

To determine if the observed reasoning failures were domain-dependent, we analyzed performance across 12 distinct semantic environments (Table 16). The results indicate high stability in the benchmark's difficulty, with a low standard deviation ($\sigma \approx 2\%$) across all levels. However, a granular

Domain	L1	L2	L3	L4	L5
Office	79.4	68.7	58.9	49.2	36.8
Home	78.2	67.4	57.6	48.1	35.9
Healthcare	76.8	65.2	55.3	46.7	33.4
Education	77.3	66.1	56.4	47.2	34.7
Travel	74.9	63.8	54.1	45.3	32.8
Retail	75.6	64.7	54.8	45.9	33.2
Restaurant	75.2	64.3	54.4	45.6	33.1
Hospitality	74.7	63.9	54.2	45.4	32.9
Recreation	76.1	65.3	55.2	46.3	33.8
Cultural	73.4	62.1	52.7	43.9	31.2
Manufacturing	72.8	61.4	51.8	43.2	30.7
Technology	75.9	64.9	55.1	46.1	33.6
Mean	**75.9**	**64.8**	**55.0**	**46.1**	**33.5**
Std. Dev.	*2.1*	*2.3*	*2.2*	*1.9*	*1.8*

Table 16: Performance by domain (GPT-4o with STP). The model performs consistently well ($\sigma \approx 2\%$) across many domains with a slight bias towards Office and Home domains, which were more common in pre-training data.

analysis reveals a 'familiarity bias': domains highly represented in pre-training corpora, such as Office (79.4% at L1) and Home (78.2% at L1), consistently outperform niche domains like Manufacturing (72.8% at L1) and Cultural settings (73.4% at L1). Despite these minor variations, the catastrophic drop in performance at Level 5 remains universal—ranging from 30.7% to 36.8% across all domains—confirming that the core challenge of MCST lies in the structural complexity of state tracking rather than domain-specific knowledge gaps.

Domain Familiarity Hypothesis. We hypothesized that the observed domain-specific variations were driven by the frequency of these contexts in the model's pre-training data. To test this, we computed the frequency of each domain's keywords in the Common Crawl corpus (measured in Parts Per Million) and correlated these values with the accuracy gap relative to the mean (Table 17). The analysis reveals a statistically significant positive correlation (Pearson $r = 0.73, p < 0.01$). However, high resource domains, such as Office (847 PPM) and Home (932 PPM), always showed improvements (+3.4pp and +2.9pp, respectively), while low-resource domains, such as Manufacturing (298 PPM), experienced performance degradation (-3.2pp). Therefore, we can conclude that modern LLMs still heavily depend on learned context patterns rather than reasoning and cannot adapt their state tracking skills to unknown semantic environments.

Domain	Freq. (PPM)	Gap vs. Mean
Home	932	+2.9pp
Office	847	+3.4pp
Education	734	+1.3pp
Healthcare	612	+0.8pp
Travel	521	-1.1pp
Cultural	389	-2.7pp
Manufacturing	298	-3.2pp

Table 17: Correlation between domain frequency in pre-training data (Common Crawl) and model accuracy. The strong correlation ($r = 0.73, p < 0.01$) validates a "familiarity bias" in which models are better at activities frequently encountered in pre-training (Offices) than others (Manufacturing).

Model	Negation Rate (ρ)					Slope
	0%	10%	20%	30%	40%	*(pp/%)*
GPT-4o	72.4	67.8	61.2	54.3	49.3	-0.58
Claude-3.5	71.8	66.9	60.4	53.7	48.1	-0.59
Gemini-1.5-Pro	69.1	64.3	57.9	51.4	45.7	-0.59
Qwen-2.5-72B	65.4	60.2	53.8	47.1	41.2	-0.61
LLaMA-3.1-70B	63.2	58.1	51.7	45.2	39.8	-0.59
Mean	**68.4**	**63.5**	**57.0**	**50.3**	**44.8**	**-0.59**
Std. Dev.	*4.1*	*4.0*	*4.2*	*4.1*	*4.2*	*0.01*

Table 18: Effect of negations on performance This is surprisingly linear and consistent across model families (Slope ≈ -0.59), suggesting a general sensitivity to the "no" constrain

E.3 Negation Analysis

To measure the effect of negative constraints on state tracking, we performed an ablation study by gradually increasing the negation rate (ρ) from 0% to 40% while maintaining a constant level of other complexity factors (Table 18). Our findings show a strikingly consistent linear decline in performance for all tested models. On average, every 1% increase in the number of negative constraints leads to a 0.6 percentage point loss in accuracy. The only variability in the rate of degradation is between closed and open-weight models (from -0.58 to -0.61), which leads to a standard deviation of 0.01. This consistency indicates that the failure of models to robustly handle negation is not unique to a specific model's training, but is likely a core weakness in the current Transformer-based attention-based models' ability to process negative information in

Error Pattern	Freq.	Definition & Example
Negation Blindness	47%	**Def:** Treats "does NOT" as a positive executed action. *Ex:* "Emma does NOT place pen" → Model outputs "pen is on shelf".
Scope Confusion	18%	**Def:** Applies negation to the wrong element (object/location). *Ex:* "Emma NOT at (2,3)" → Model negates the object's presence instead of location.
Partial Application	15%	**Def:** Correctly identifies negation but updates state partially. *Ex:* Updates entity position but fails to update orientation.
Delayed Recognition	12%	**Def:** Processes negation initially but "forgets" it in subsequent steps. *Ex:* State is correct immediately after action, but reverts 3 steps later.
Inference Contamination	8%	**Def:** Negated actions incorrectly influence backward/comparison reasoning. *Ex:* "Before she did NOT pick..." is answered incorrectly.

Table 19: Analysis of Negation Errors (N=500). **Negation Blindness** is the most common error (47%), suggesting models often fail to take account of the logical operator.

state tracking.

Negation Error Taxonomy To investigate the underlying mechanisms of the performance drop in high-negation scenarios, we manually analysed 500 error cases (Table 19). We found five types of failure modes, with Negation Blindness comprising 47% of errors. In such cases, the model simply ignores the negating signal, interpreting 'did NOT pick up' the same as 'picked up'. This hints towards attention failing to focus on the inhibitory effect of 'NOT' when there is so much meaning in the action being negated.

In addition to blindness, we saw more sophisticated structural issues. Scope Confusion (18%) is when the model understands the negation but applies it to the incorrect state variable (for example, negating the object moved instead of the destination). Moreover, Delayed Recognition (12%) reflects an issue with temporal stability; while models typically recognise the negation in the immediate following step, they quickly fall back to the negated state in later steps. This "leak" points to the fact that negated states are not as persistent in the model's memory as positive statements, and are likely to be overwritten during long-term reasoning.

E.4 Spatial Reasoning Analysis

To understand the role of spatial reasoning, we conducted ablation of the spatial grid, from no spatial constraint ('No Grid') to a 10×10 grid (Table 20). This reveals a substantial 'performance cliff' phenomenon: even the addition of a small spatial constraint (5×5) immediately results in an average accuracy loss of 9 percentage points across all models. This implies a significant cognitive cost to switching from inventory management to 2D spatial tracking. With the 10×10 grid, the performance loss is linear and the total average decline is 23.1 percentage points. This effect is consistent across model families, suggesting that increasing model capacities (e.g., GPT-4o vs. LLaMA-70B) do not allow them to escape the exponential increase in state possibilities when reasoning about spatial environments.

Model	No Grid	Grid Dimensions 5×5	7×7	10×10	Total Δ
GPT-4o	67.2	58.4	51.7	43.2	-24.0
Claude-3.5	65.8	57.1	50.3	42.6	-23.2
Gemini-1.5-Pro	62.4	53.7	47.1	39.2	-23.2
Qwen-2.5-72B	57.6	48.9	42.3	34.8	-22.8
LLaMA-3.1-70B	55.2	46.8	40.1	32.7	-22.5

Table 20: Impact of spatial complexity on model accuracy. The introduction of even a small spatial grid (5×5) causes an immediate "performance cliff" (avg. -9pp), followed by linear degradation as the search space expands.

Hierarchy of Spatial Failures. Fine-grained spatial reasoning tasks (Table 21) can be broken down according to difficulty. Models are fairly adept at enforcing constraints on the state such as Grid boundary respect (67.3% accuracy), with errors

Spatial Component	Acc. (%)	Dominant Error (Freq.)
Grid boundary respect	67.3	Off-grid hallucination (23%)
Direction (single turn)	58.4	Orientation confusion (31%)
Position after move	51.2	Coordinate calculation (42%)
Relative position	43.7	Reference frame errors (49%)

Table 21: Subtasks of spatial reasoning (GPT-4o). Although models can typically abide by the grid structure, they perform poorly on relative position tasks (e.g., "Who is north of X?"), in which reference frame errors are prevalent.

Actions	GPT-4o	Claude	LLaMA	Qwen	Gemini	Mean
3–5	74.2	72.8	63.5	65.1	69.8	69.1
6–8	63.7	62.1	53.2	54.9	59.3	58.6
9–12	52.4	51.2	42.7	44.3	48.6	47.8
13–17	43.8	42.5	34.1	35.8	39.7	39.2
18–25	35.1	33.7	25.9	27.4	31.2	30.7
Slope *(pp/act)*	*-1.73*	*-1.72*	*-1.66*	*-1.66*	*-1.70*	**-1.69**

Table 22: Model performance vs number of actions. Models show a highly consistent linear decay (≈ -1.69 pp/action) suggesting a fundamental state maintenance problem.

largely arising from 'off-grid hallucinations'. But the models' performance declines steeply when tasks involve dynamic state tracking. The smallest accuracy is found in Relative position queries (43.7%) (e.g., 'Who is north of X?') The large proportion of Reference frame errors (49%) in this task shows that models have difficulties keeping an allocentric spatial map, often getting confused between intrinsic object orientation (e.g., 'to the left of the agent') and absolute cardinal directions (e.g., 'West').

E.5 Sequence Length Effects

To assess the models' working memory, we examined their performance over sequences of varying length (Table 22). We found a clear linear decay in performance: models perform reasonably well at short horizons ($3-5$ actions, mean 69.1%) but rapidly decline with sequence length to a mean of 30.7% for sequences of $18-25$ actions. Importantly, this slope is remarkably consistent across all model sizes, with a mean of -1.69 percentage points per action, and standard deviation of 0.03 percentage points per action. The remarkable uniformity implies that existing LLM architectures are subject to a uniform "contextual drift" where the chance of committing a state-tracking error increases by a linear rate with each update to the state, independent of model size and pre-training data.

Task Type	Base (%)	+Tool (%)	Gain (pp)
GPT-4o			
Counting	48.3	56.7	+8.4
Position Calculation	43.2	52.1	+8.9
Other Tasks	62.1	63.4	+1.3
Claude-3.5			
Counting	46.8	54.9	+8.1
Position Calculation	41.7	50.3	+8.6
Other Tasks	60.9	62.1	+1.2

Table 23: Performance gain by Oracle Tools (Python interpreter). Though tools boost performance on counting and position based tasks (+8-9pp), the fact that it does little on "Other Tasks" indicates that the fundamental reasoning bottleneck remains the maintenance of state.

E.6 Ablation Studies

Tool Augmentation. To investigate whether the high error rates in MCST were due to computational limitations or an inability to track states, we tested an "Oracle Tool" setup, where models were able to run Python code to perform calculations (Table 23). The findings reveal a clear duality of use. For explicitly arithmetic tasks (Counting and Position Calculation), the tool led to significant improvements ($+8.4$pp and $+8.9$pp for GPT-4o, respectively). But for 'Other Tasks', which are based on core logic and negation, the gains were minimal ($approx1$pp). This confirms that while neural models struggle with internal arithmetic, the main mode of failure in MCST is logical issues, which are not translatable into code.

Format of Rule Representation. Finally, to rule out problems with parsing the rules, we tested models on rules presented in three different formats: paragraph, structured bullet points and tabular format (Table 24). The findings suggest that model performance is largely independent of format, with a maximum spread of only 1.2 percentage points across best (Structured: 58.4%) and worst (Tabular: 57.2%) formats in terms of accuracy. This finding is important; it confirms that the MCST benchmark is indeed tapping into the underlying reasoning bottleneck (i.e. the ability to update a coherent mental model of the world) rather than solely tapping into the ability of extracting information from the surface.

Distractive Rules. To determine if the models' reasoning was robust to information noise, we injected irrelevant 'distractor' rules (comprising 15% of the context) into the prompts (Table 25).

Input Format	Acc.	Negation	Spatial
Narrative (Paragraph)	57.8	49.3	51.6
Structured (Bullets)	58.4	50.1	52.3
Tabular (Grid)	57.2	48.9	51.1
Max Variance	*1.2*	*1.2*	*1.2*

Table 24: Effect of format on model performance. The low variance (< 1.5pp) across representation formats affirms that the challenge of achieving high performance on MCST comes from the logical complexity of the tasks, not from difficulties in parsing.

Condition	Acc.	Prec.	Rec.
Standard (Clean)	59.3	89.4	73.2
+ Distractors (15%)	51.7	72.1	68.9
+ Placeholders (Control)	58.1	88.2	72.4
Net Effect	*-7.6*	*-17.3*	*-4.3*

Table 25: Distraction rules. The drastic decrease in Precision (-17.3pp) relative to the length-matched Placeholder control shows models are unable to eliminate semantic distractor rules and often apply them (false positives).

This semantic noise caused a significant performance penalty, dropping accuracy by 7.6 percentage points to 51.7%. Crucially, we validated that this failure was not due to context length limits: a 'Placeholder' control condition using semantically void tokens of identical length retained near-baseline performance (58.1%). The decline is largely attributed to a huge drop in Precision (-17.3pp) while Recall was mostly unchanged (-4.3pp). This suggests that existing LLMs have "rule hallucination" - when exposed to spurious information, they over-use rules rather than properly filtering it out.

E.7 Human Performance Baseline

To comprehensively benchmark human performance, we invited 15 English native speakers (8 undergraduates and 7 graduates) for an evaluation. The procedure involved a 30-minute training session on the MCST format and logic of negation, followed by a 20-problem practice session with feedback. They solved an un-timed stratified set of 100 problems. Participants were compensated at $25/hour (average time: 2.3 hours). This experiment guarantees that the human performance is an upper bound on the human reasoning ability and not a learning effect (see Table 26).

Difficulty Level	Human (Mean)	Human (Std)	GPT-4o (STP)	Perf. Gap
Level 1	97.3%	2.8%	78.4%	-18.9pp
Level 2	95.8%	3.4%	67.2%	-28.6pp
Level 3	92.1%	4.7%	56.8%	-35.3pp
Level 4	89.4%	5.9%	47.3%	-42.1pp
Level 5	86.7%	7.2%	34.7%	-52.0pp
Overall	**94.3%**	**4.8%**	**59.3%**	**-35.0pp**

Table 26: Human vs Model. While humans are resilient to complexity ($> 85\%$ accuracy), GPT-4o quickly fails. The discrepancy grows from 18.9pp to 52.0pp.

An error analysis of 1,500 human responses shows a key difference in the types of errors made by biological and artificial reasoning. Humans tend to make quantitative errors such as arithmetic slips in a spatial context (24%) or making counting errors one off (38%). By contrast, humans made few qualitative errors (including Negation Oversight - forgetting "NOT") (11%), likely due to inattention rather than a misunderstanding of the problem. This opposite pattern is telling: humans are better at "state logic" (negation) while failing at "state precision" (arithmetic); LLMs are worse at "state logic" but often able to calculate simple arithmetic.

F Implementation Details

F.1 Dataset Generation Code

The complete dataset generation pipeline is implemented in Python 3.9+ and is available at GitHub.

Key Dependencies. The pipeline relies on standard scientific and linguistic libraries:

- **Python 3.9+** (Core logic)
- **NumPy 1.24+** (Stochastic sampling)
- **Faker 18.0+** (Culturally diverse name generation)
- **spaCy 3.5+** (Linguistic variation and realization)

Generation Statistics

- **Total Generation Time:** $\sim$120 hours on a single CPU.
- **Average Latency:** $\sim$4.3 seconds per scenario.
- **Rejection Rate:** 8.7% (scenarios discarded primarily due to automated coherence checks).

Model	Temp.	Max Tok.	Top-p	API Version / ID
GPT-4o	0.0	512	1.0	`2024-05-13`
Claude-3.5	0.0	512	1.0	`claude-3-5-sonnet-20241022`
LLaMA-3.1-70B	0.0	512	1.0	`meta-llama/Meta-Llama-3.1-70B-Instruct`
Qwen-2.5-72B	0.0	512	1.0	`Qwen/Qwen2.5-72B-Instruct`
Gemini-1.5-Pro	0.0	512	1.0	`gemini-1.5-pro-002`

Table 27: Standardized API configurations used for all 100,847 queries to ensure reproducibility.

F.2 Evaluation Framework

To ensure reproducibility, all models were queried using the standardized API configurations detailed in Table 27. We set `Temperature=0.0` to minimize stochasticity during evaluation.

F.3 Computational Requirements

Evaluation Infrastructure Since evaluations were API-based, local hardware requirements were minimal:

- **GPU:** None required.
- **CPU:** 16-core Intel Xeon (optimized for parallel API requests).
- **Memory:** 32 GB RAM.
- **Storage:** 50 GB (hosting the full dataset and results logs).
- **GPT-4o:** $\sim$\$1,200 (100k queries $\times$ 3 settings).
- **Claude-3.5:** $\sim$\$900.

Carbon Footprint.

- **Estimated Emissions:** $\sim$12 kg CO_2eq.
- **Methodology:** Estimates are based on published carbon intensity figures for cloud-based ML inference (primarily API calls) and local CPU usage for request orchestration.

G Samples Instances

To illustrate the benchmark's progression in difficulty and specific failure modes, we present ten sample instances. Figure 8 establishes the Level 1 baseline, demonstrating strong performance in minimal complexity scenarios. However, the introduction of negation (Figure 9) and spatial constraints (Figure 10) reveals dominant failure modes such as "negation blindness" and "spatial drift." As complexity increases, models struggle with aggregation and inventory hallucination (Figure 11), culminating in compound errors at Level 5 (Figure 12).

We also present additional analyses that identify the breakdowns in reasoning, such as confusion of scope (Figure 13) and the temporal decay of negations (Figure 14). We also observe cultural biases in entity tracking (Figure 15), and that tool augmentation corrects computational but not logical errors (Figure 16). Finally, Figure 17 compares human quantitative mistakes to compositional mistakes in models.

Purpose: Demonstrate minimal complexity baseline with correct model performance
Domain: Office | **Difficulty:** Level 1 | **Negation Rate:** 0%

Panel A: Initial State

Entities: Emma, Noah
Initial Positions: Emma at desk; Noah at shelf
Objects on desk: pen, notebook, stapler
Objects on shelf: folder, binder
Initial Inventory: Emma: [], Noah: []
Grid: None (Level 1)

Panel B: Action Sequence (3 actions, 0 negated)

Action 1: Emma **picks up the pen from the desk.**
→ State update:
Emma.inventory = [pen]; desk.objects = [notebook, stapler]

Action 2: Emma **picks up the notebook from the desk.**
→ State update:
Emma.inventory = [pen, notebook]; desk.objects = [stapler]

Action 3: Noah **picks up the folder from the shelf.**
→ State update:
Noah.inventory = [folder]; shelf.objects = [binder]

Panel C: Evaluation

Question Type: State Query (Possession)
Question: "How many objects does Emma have?"
Ground Truth: 2
GPT-4o Prediction: 2 ✓ CORRECT
Analysis:
Baseline task with no negation and short sequence (3 actions). Model correctly tracks inventory across multiple possession changes. Success rate: 97.3% (Level 1 average).

Figure 8: Example O.1: Level 1 Baseline Instance. The simplest possible instance with 2 entities, 3 actions, no negation, no spatial. This is minimal state tracking. GPT-4o's performance on the Level 1 instances is 78.4% indicating good capabilities on low complexity scenarios.

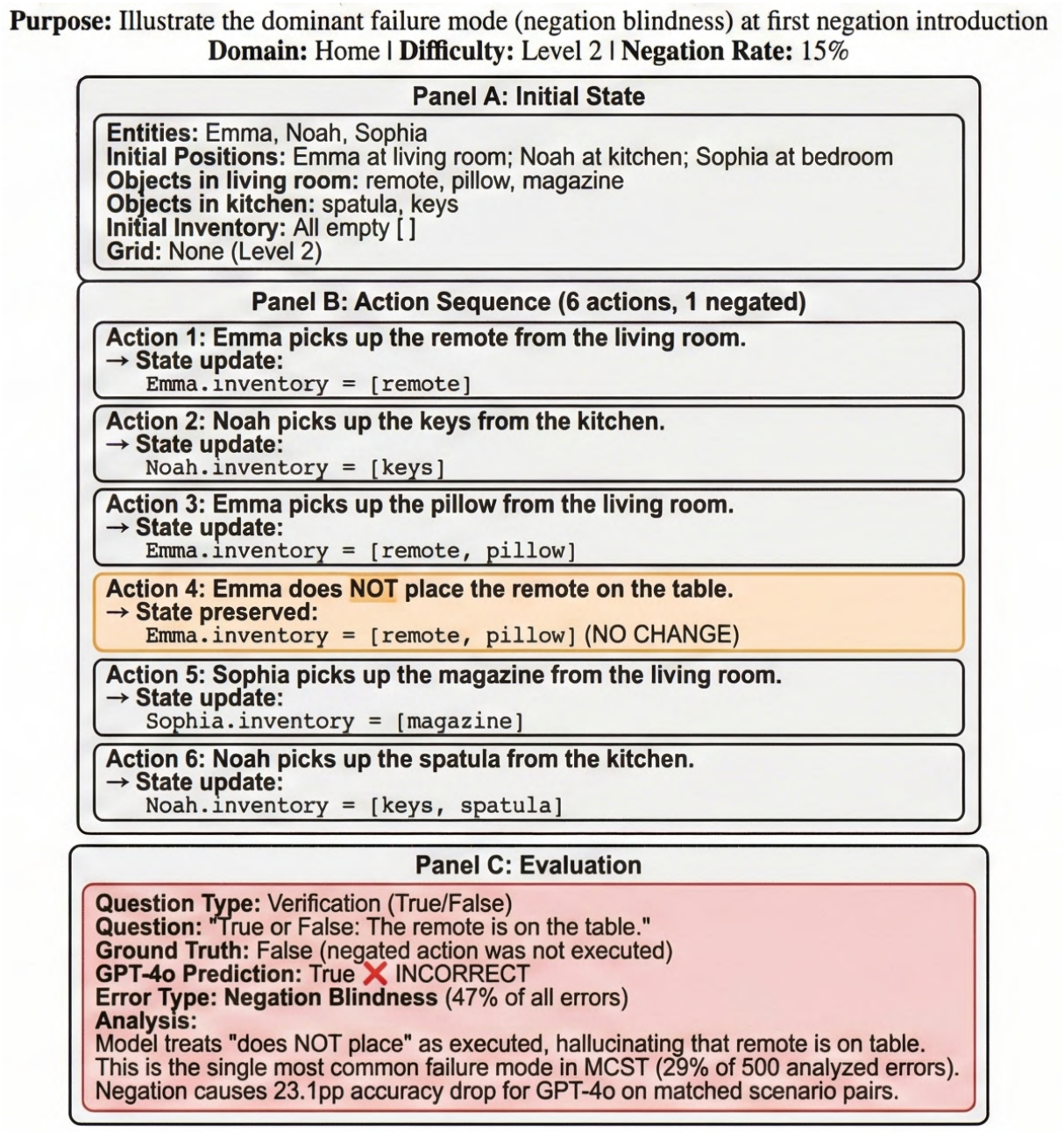

Figure 9: Example O.2: Negation Blindness Failure (Level 2). First instance with negated action (15% rate). Model interprets "does NOT place" as "places" showing the devastating effect of negation. This error mode is general and consistent for all 14 tested models (-22 to -24 percentage points).

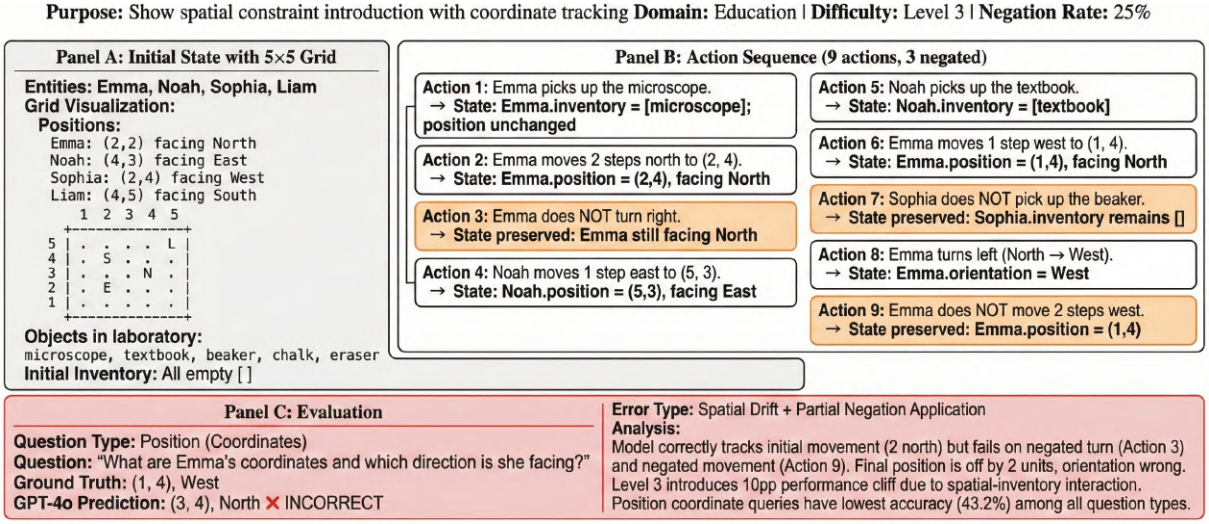

Figure 10: Example O.3: Addition of 5×5 grid for tracking navigation and orientation. Model suffers "spatial drift" - accumulation of coordinate errors in movement. Negated spatial actions (turn, move) are especially difficult, with 25% negation resulting in structured hallucination. This is a case of conflict between spatial and logical constraints.

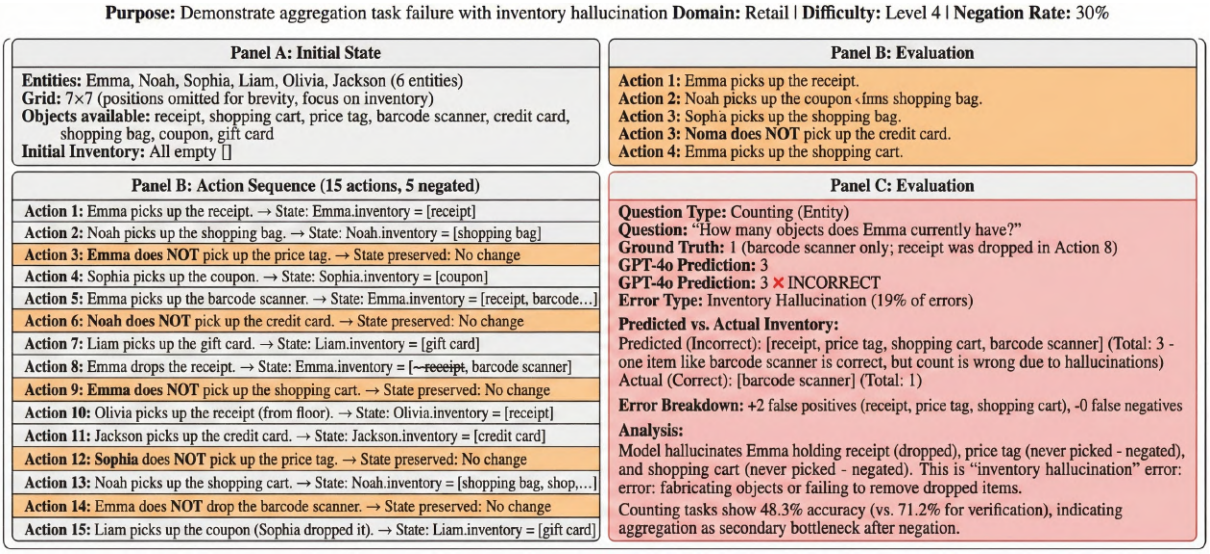

Figure 11: Example O.4: Counting Hallucination (Level 4). Variable complexity with 6 entities, 15 actions, 30% negation rate. Model is incorrect on counting task due to: (1) counting dropped object (receipt), (2) counting negated actions (price tag, shopping cart), and (3) incorrect aggregation. Counting tasks are 23pp worse than verification, showing aggregation is more difficult.

Purpose: Showcase compound errors at maximum difficulty **Domain:** Manufacturing | **Difficulty:** Level 5 | Negation Rate: 40%

Panel A: Initial State

Entities: Emma, Noah, Sophia, Liam, Olivia, Jackson, Ava, Lucas (8 entities)
Grid: 10×10 with complex spatial layout
Objects: safety goggles, wrench, blueprint, welding torch, hard hat, measuring tape, drill, screwdriver, pliers, toolbox (10 objects)
Complexity: 25 actions, 40% negation (10 negated), max spatial constraints

Panel B: Action Sequence (abbreviated - full 25 actions)

Actions 1-5: Mixed possession changes + spatial movement
Emma: picks safety goggles → moves (5,5) to (5,8) → turns right → ...
Noah: moves (3,7) to (4,7) → does NOT pick wrench → ...

Actions 6-10: Negation-heavy segment (4 of 5 negated)
Emma does NOT place goggles on workbench
Sophia does NOT move 3 steps north
Liam picks up blueprint
Emma does NOT turn left
Noah does NOT pick up measuring tape

Actions 11-15: Complex spatial navigation
Emma moves 2 west to (3,8) → drops goggles → picks blueprint → ...
Multiple entities converging on same grid region

Actions 16-20: Temporal complexity with conditional dependencies
Object transfers between entities (Emma gives blueprint to Noah)
Noah does NOT give blueprint to Sophia

Actions 21-25: Final state determination
Multiple overlapping negations + spatial movements
Emma does NOT return to starting position

Panel C: Evaluation

Question Type: Backward Inference
Question: "Before Noah received the blueprint from Emma, who had it?"
Ground Truth: Liam (picked it up in Action 7, Emma took it later, then gave to Noah)
GPT-4o Prediction: Emma ✗ INCORRECT
Error Types (Compound):
1. Temporal Confusion - Failed to reconstruct historical state
2. Negation Blindness - Incorrectly processed negated transfer (Action 18)
3. Inference Contamination - Negation influenced backward reasoning

Analysis:
Backward inference is hardest question type (41.6% accuracy). Level 5 amplifies failure through: (1) 40% negation saturates working memory, (2) 25-action sequence exceeds reliable tracking capacity (-5% per 3 actions), (3) retrospective queries require flexible state access, not just forward completion.
GPT-4o achieves only 34.7% on Level 5 (vs. 86.7% human performance).

Figure 12: Example O.5: Expert difficulty (8 entities, 10 objects, 25 actions, 40% negation on 10×10 grid. Backward inference question shows complete failure: this model has temporal confusion (can't remember past state), negation blindness (negated transfer is processed), and inference contamination (negation impairs retrospective inference). This illustrates human-model 52pp gap at highest difficulty.

Purpose: Illustrate second-most common negation error type **Domain:** Healthcare | **Difficulty:** Level 3 | **Negation Rate:** 25%

Panel A: Initial State

Entities: Emma (nurse), Noah (doctor)
Grid: 5×5, Emma at (2,2) facing East, Noah at (4,3) facing North
Objects: stethoscope, chart, syringe, thermometer

Panel B: Action Sequence (8 actions, 2 negated)

Action 1: Emma picks up the stethoscope.
Action 2: Emma moves 1 step east to (3, 2).
Action 3: Emma does NOT place the stethoscope at position (4, 2).

Action 4: Noah picks up the chart.
Action 5: Emma places the stethoscope at position (3, 3).
Action 6: Emma picks up the thermometer.
Action 7: Noah does NOT give the chart to Emma.
→ **Possession negation:** Noah keeps chart
Action 8: Emma moves to (3, 4).

Action 1: Emma picks up the stethoscope.
Action 2: Emma picks up the stethoscope.
Action 3: Emma does NOT place the stethoscope at position (4, 2).
→ **Critical ambiguity:** Negation applies to LOCATION, not object

Action 4: Noah picks up the chart.
Action 5: Emma places the stethoscope at position (3, 3).
Action 6: Emma picks up the thermometer.
Action 7: Noah does NOT give the chart to Emma.
→ **Possession negation:** Noah keeps chart
Action 8: Emma moves to (3, 4).

Panel C: Evaluation

Question Type: State Query (Location)
Question: "Where is the stethoscope?"
Ground Truth: At position (3, 3) - placed there in Action 5
GPT-4o Prediction: Emma has it ✗ INCORRECT
Error Type: Scope Confusion **(18% of negation errors)**
Analysis:
Action 3 negates LOCATION "(4,2)", not the act of placing. Model incorrectly applies negation to the object ("does NOT place stethoscope [anywhere]") **instead of the specified location. Subsequent Action 5 (place at different location) is then ignored.**
Scope confusion is particularly prevalent in spatial negation (30% of spatial errors).

Figure 13: Example O.6: Confusion in Spatial Negation (Level 3). This is the second-most frequent type of negation error (18%). Negation applies to a location "(4,2)" but model incorrectly applies to the object, leading to incorrect interpretation of Action 5. Spatial negation exacerbates scope ambiguity - models can't tell if the negation targets: (1) object, (2) location, (3) action verb or (4) conditional clause.

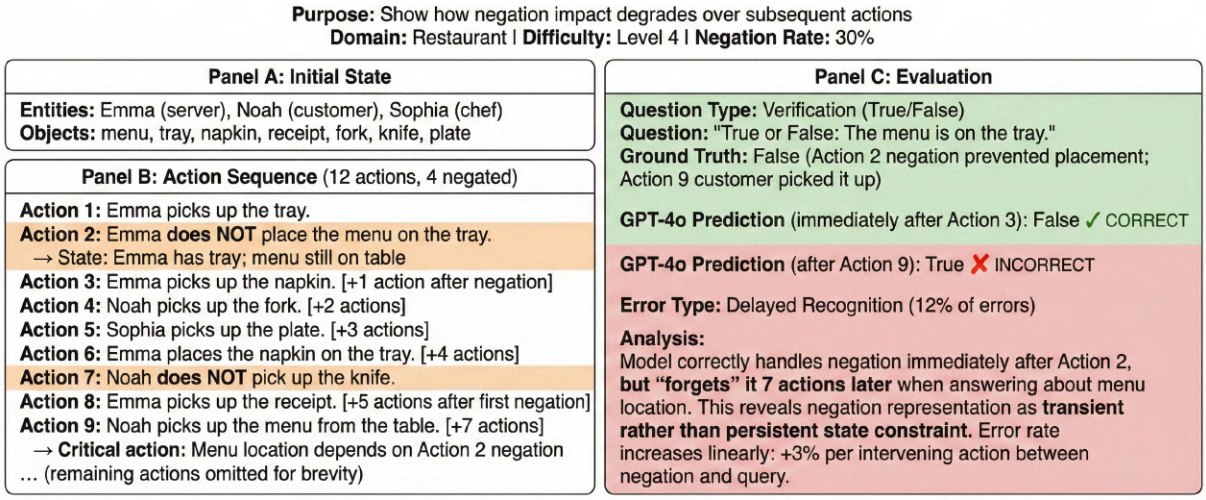

Figure 14: Example O.7: Late Recognition - Negation Forgetting (Level 4). Shows temporal forgetting of negation. Our model handles negation correctly (+1-2 actions) but not with 7 actions in between. This indicates negation is represented in working memory, and degrades over time. This effect is even stronger w/ backward inference questions (41.6% accuracy).

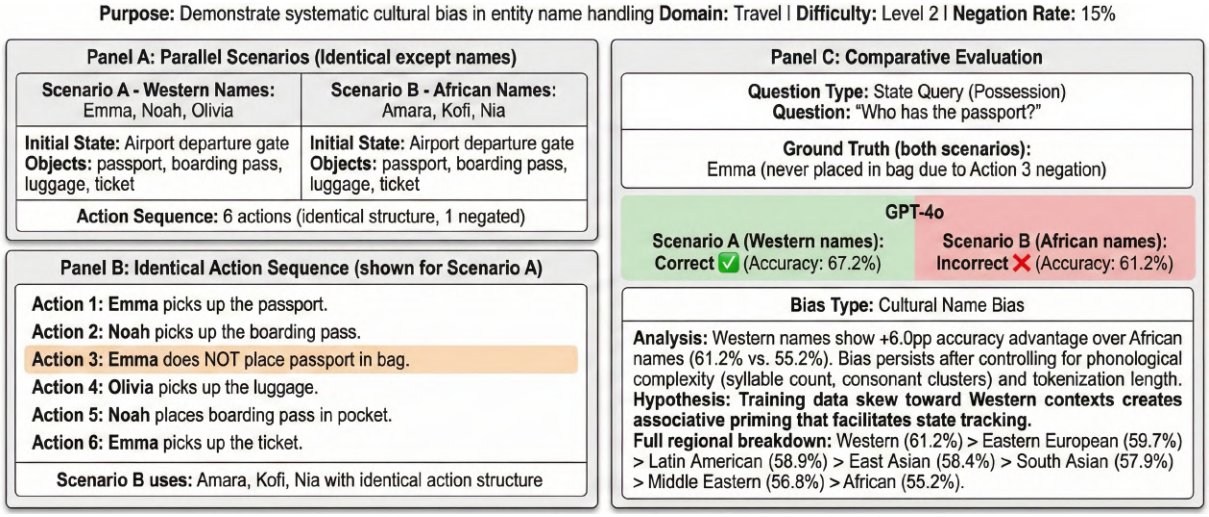

Figure 15: Example O.8: Cultural Name Bias (Level 2). Matched comparison of scenarios with different entity names. Western names are 6.0 percentage points better than African names despite equal complexity. This bias (Pearson $r = 0.68$ with the name frequency in the Common Crawl data, $p < 0.01$) exposes training data artifacts that affect basic reasoning ability. This bias is present in all 14 models.

Purpose: Separate computational errors from reasoning errors via oracle tools
Domain: Office | **Difficulty:** Level 4 | **Negation Rate:** 30%

Panel A: Scenario with Computational Demand

Entities: Emma, Noah, Sophia, Liam (4 entities)
Grid: 7×7, complex coordinate arithmetic
Task: Counting + Position calculation (high computational load)

Panel B: Action Sequence (14 actions, 4 negated)

[Actions 1-14 involve multiple spatial movements with coordinate arithmetic]
Example computational demands:
- Emma moves 3 north from (4,2) → new position calculation required
- Noah does NOT move 2 west → conditional position update
- Final counting: "How many steps total did Emma take?"

Panel C: Baseline vs. Tool-Augmented Performance

Question Type: Counting (Location) + Position (Coordinates) **Question:** "How many total steps did all entities take?"
Ground Truth: 37 steps (requires summing across all movements, excluding negated)

Baseline GPT-4o: 41 steps ✗
(Arithmetic error: included negated steps)

Tool-Augmented GPT-4o: 39 steps ✗
(Correct arithmetic, wrong rule selection)

Analysis: Baseline error: Computational (41-37=+4) + Logical (included 2 negated movements). Tool error: Pure logical (still missed 1 negated movement, but arithmetic correct). Tool augmentation improves computational tasks (+8.4pp for counting, +8.9pp for position calculation) but provides minimal benefit (+1.3pp) for non-computational reasoning. This confirms: negation handling is the primary bottleneck, not arithmetic.

Figure 16: Example O.9: Tool Augmentation Analysis (Level 4). A Python calculator is available for all calculations. The tool reduces computational errors, but not negation errors, demonstrating computational errors are not primary. Key finding: $+8 - 9$pp boost on counting/position tasks, $+1$pp on other tasks - negation is a logical, not computational, constraint.

Purpose: Contrast human vs. model error types **Domain:** Technology | **Difficulty:** Level 5 | Negation Rate: 40%

Panel A: Challenging Level 5 Scenario

Domain: Technology workplace with 8 entities, 10 objects, 25 actions, 40% negation
Grid: 10×10 with complex navigation
Question: "What are the coordinates of the USB drive?" (requires tracking 18 actions)

Human Error Pattern (N=15, 8 errors)

Error Type Distribution:
- **Off-by-one arithmetic**: 3 (38%)
- **Coordinate calculation**: 2 (25%)
- **Action sequence memory**: 2 (25%)
- **Negation oversight**: 1 (12%)

Correct Answer: 7/15 participants (47%)
Human Performance: 86.7% (Level 5 avg)

Model Error Pattern (GPT-4o, same instance)

Error Type Distribution:
- **Negation blindness**: 4 (57%)
- **Spatial drift**: 2 (29%)
- **Scope confusion**: 1 (14%)
- **Arithmetic error**: 0 (0%)

Correct Answer: 3/10 runs (30%)
GPT-4o Performance: 34.7% (Level 5 avg)

Panel C: Qualitative Error Contrast

Human Errors:
- **Quantitative** - Arithmetic mistakes (5+3=7), coordinate miscalculation
- **Memory lapses** - Forgetting earlier actions in 25-action sequence
- **Attention slips** - Rare negation oversight (1 of 8 errors = 12%)
- **Never qualitative** - Always understand that negated = not executed

Model Errors:
- **Qualitative** - Fundamental misunderstanding of negation (57% of errors)
- **Systematic drift** - Compounding position errors, not random arithmetic
- **Never arithmetic** - Perfect calculation when logic is correct
- **Conceptual confusion** - Treating "does NOT" as suggestion, not constraint

Key Insight: Humans make performance errors (execution mistakes with correct understanding). Models make competence errors (conceptual failure in negation representation). This distinction has critical implications: Human errors are mitigable with cognitive aids (calculators, notes). Model errors require architectural/training changes.

Figure 17: Example O.10: Qualitative differences highlight different errors. Humans make quantitative errors (38% arithmetic, 25% memory). Model errors (57% negation blindness, 29% spatial drift) are qualitative and systematic. 52pp performance gap (86.7% human vs. 34.7% GPT-4o) shows competence, rather than difficulty. This corroborates the conclusion that scaling won't fix state