

Lingo-Blocksworld: A Dataset for Linguistically-Informed Blocksworld

Melissa Torgbi¹, Claire Benet Post², Claire Bonial³, Harish Tayyar Madabushi¹

¹University of Bath, ²University of Colorado Boulder,

³DEVCOM U.S. Army Research Laboratory

mat66@bath.ac.uk

Abstract

We introduce LINGO-BLOCKSWORLD, a linguistically-informed diagnostic benchmark for evaluating the generalisation and planning abilities of Large Language Models (LLMs) in the Blocksworld (BW) domain. Unlike prior obfuscated BW variants, our method replaces content words with morphosyntactically well-formed nonsense words, preserving the grammatical scaffolding of the domain while removing lexical familiarity. This enables us to test a psycholinguistically-motivated hypothesis and find systematic support across both reasoning and non-reasoning models, with degradation patterns that mirror syntactic bootstrapping from human language processing. Additionally, we test the ability of models to generalise planning solutions from the typical 4-block problems to 12 blocks. Overall, our results demonstrate that reasoning models are more robust where non-reasoning models collapse: non-reasoning model performance drops to near zero once problems exceed four blocks or have actions and predicates obfuscated, while reasoning models maintain stronger performance until structural and lexical difficulty are combined. Manual error analysis pinpoints further differences. Reasoning model errors are more often skipped steps or incomplete instruction following, whereas non-reasoning models fail to derive plausible plans at all. By combining psycholinguistic motivation with classical planning evaluation, LINGO-BW allows us to shift research focus from whether LLMs plan to how their planning depends on linguistic surface form.¹

1 Introduction

Large Language Models (LLMs) trained on vast amounts of data and containing billions of parameters are able to extrapolate from their training data

in ways that produce impressive performance on a number of tasks (Brown et al., 2020; Tayyar Madabushi et al., 2025; Wei et al., 2022). However, there remain tasks where LLMs fall short of human generalisation abilities (Srivastava et al., 2023; Scivetti* et al., 2025), including classical planning (Sel et al., 2025). We therefore focus on planning as a controlled setting for evaluating LLM generalisation.

Planning problems involve determining the required steps to get from an initial state to a goal state and range in complexity from very simple micro-worlds to complex real-world applications (Cimatti et al., 2008). We intentionally choose the simple planning domain of Blocksworld (BW) (Winograd, 1972; Slaney and Thiébaux, 2001) so that we can carefully vary problem difficulty in a controlled manner.

Our goal is not to evaluate LLMs as practical planners, but to use classical planning to assess their ability to generalise within a familiar planning domain beyond reliance on semantic priors. We evaluate generalisation beyond typical BW problems by incrementally altering the structural complexity and vocabulary of the problem domain. First we establish a structural baseline by varying the number of blocks. While prior work has shown that earlier models struggle with increasing structural complexity (Shojaee et al., 2025), verifying that these limits hold for newer models is a necessary calibration step for our subsequent linguistic evaluations. Next, we introduce nonsense words that maintain English morphosyntactic rules. Unlike prior obfuscated variants (Valmeekam et al., 2024a,b), this preserves the linguistic integrity of the original BW prompt for comparison. We perform ablation experiments that systematically swap out only actions, objects, predicates, and combinations thereof to investigate whether obfuscating certain linguistic cues challenges models in ways that align with psycholinguistically grounded hypothe-

¹The dataset and code to replicate our experiments are available at <https://github.com/melissatorgbi/lingoblocksworld>.

ses. This design is also motivated by limitations of existing obfuscated BW variants, such as Mystery BW and Random BW, whose lexical substitutions can either alter predicate-argument structure or fail to preserve the linguistic properties of the original domain (see [Appendix B](#)). Simultaneously, we compare non-reasoning and reasoning models to explore how different architectures respond to semantic abstraction.

Concretely, we make the following contributions:

1. We provide a novel BW evaluation dataset and generation pipeline that varies structural complexity and morphosyntactically well-formed lexical obfuscation.
2. We evaluate the zero-shot capabilities of both non-reasoning and reasoning models using our diagnostic dataset to test generalisation across increasing structural complexity and lexical novelty in the BW domain. This evaluation reveals that reasoning models are substantially better, but still degrade under combined structural and lexical difficulty. The combination of our granular dataset and manual error analysis provides novel insight into distinct failures, showing that non-reasoning models have cascading errors not present in reasoning models.
3. We introduce an increased-challenge setting that helps more precisely distinguish success on easier instances from more fragile behaviour under combined structural and lexical difficulty, beyond what previous datasets make visible.

2 Background & Related Work

LLMs have displayed impressive performance on a variety of tasks and benchmarks, but classical planning remains a challenge. Many LLMs, in fact, do not perform well on even simple planning problem domains, such as BW ([Valmeekam et al., 2023a](#)). BW is one of the original classical planning domains and remains popular for ongoing evaluation of automated planners, in particular those using the Planning Domain Definition Language (PDDL).² In BW, problems consist of mul-

²BW and PDDL are leveraged more for the development of planning approaches (e.g., [Grundke et al. \(2024\)](#)), including newer approaches leveraging LLMs. For robotic applications, Signal Temporal Logic and Linear Temporal Logic are now more prevalent (e.g., [Chen et al. \(2024\)](#)).

tiply coloured blocks on a table. The blocks may be stacked on top of each other or directly placed on the table. Each problem has an initial configuration and a target goal state. The solution lists the actions required to get from the initial block configuration to the goal within certain constraints and leveraging a minimal set of possible actions ([Russell and Norvig, 2021](#)).

Previous research has used the BW domain to test the ability of LLMs to solve these problems. LLMs tend to fall short of human performance on these tasks in many cases ([Sel et al., 2025](#)). Large Reasoning Models (LRMs) have displayed better performance compared to non-reasoning LLMs; however, performance gains are not robust with increasing complexity and struggle to generalise well to obfuscated instances within existing BW variants ([Valmeekam et al., 2024a](#); [Shojaee et al., 2025](#)). The first obfuscated variant is Mystery Blocksworld (MYSTERY BW), in which nouns are swapped (e.g., “block” for “planet”), verbs are swapped (e.g., “stack” for “crave”), and predicates are swapped (e.g., “is clear” for “harmony”) ([Valmeekam et al., 2024b](#)). The second variant is Randomized Mystery Blocksworld (RANDOMIZED BW), which swaps out nouns, verbs, and predicates for a randomly selected string of alphanumeric characters (e.g., “arsbumagdtb748l6”).

While MYSTERY BW aims to maintain the same underlying structure as the original BW tasks, its linguistic form is neither semantically identical nor syntactically equivalent. For example, words such as “planet” or “pain” inherently carry semantic load within the embedding space of an LLM and could affect model performance. Similarly, replacing actions with randomized strings is too stringent and does not reflect real-world settings where actions are likely to remain linguistically grounded. This motivates the need for a linguistically grounded BW dataset. Table 1 showcases a comparison of the original BW domain description, the Mystery variant, the Randomized variant, and finally our own LINGO-BW variant.³

2.1 Nonce Words Obfuscation

Nonce words, which have long been used in psycholinguistic studies (e.g., [Berko \(1958\)](#)), allow us to preserve syntactic and semantic mappings while eliminating pre-existing lexical meaning. Specifically, they enable us to: (a) duplicate the original

³For a more detailed description of issues of MYSTERY BW and RANDOMIZED BW see [Appendix B](#).

BW Variant	Sample of Domain Description
Original prompt	“Pick up a block Unstack a block from on top of another block Put down a block Stack a block on top of another block I can only pick up or unstack one block at a time. I can only pick up or unstack a block if my hand is empty.”
Lingo-BW Hard (actions, objects, predicates)	“Jillerate a tegust Unbusillisk a tegust from faratived onto another tegust Astent a tegust Busillisk a tegust faratived onto another tegust I can only jillerate or unbusillisk one tegust at a time. I can only jillerate or unbusillisk a tegust if my michime is tuentsal.”
Mystery BW	“Attack object Feast object from another object Succumb object Overcome object from another object To perform Attack action, the following facts need to be true: Province object, Planet object, Harmony. Once Attack action is performed the following facts will be true: Pain object.”
Randomized BW	“J4gv801gnu2it0yj object_0 object_1. U64y1a9apusmslxb object_0. Kip9uw781pv62umn object_0. Wio5amhq7814n006 object_0 object_1. To perform j4gv801gnu2it0yj action, the following facts need to be true: tv30k33pzoulql6w object_1, and a46zhwykn6jvabak object_0 Once j4gv801gnu2it0yj is performed the following facts will be true: cq14o62p1yeke3ok, tv30k33pzoulql6w object_0, and b6e9q4r60gagvdcn object_0 object_1.”

Table 1: Parallel sections of different BW prompts with highlighted words showing matched linguistic items.

semantic and syntactic structure of BW while minimising possible contamination from memorised or pre-trained associations, and (b) replace content words with phonotactically valid but semantically novel items, ensuring no inherited predicate-argument or affective bias. We use nonce words in our dataset to try to circumvent the issue of inherited semantic priors since nonce words theoretically have zero lexical meaning.⁴

3 Approach Overview

Having motivated the linguistic rationale for nonce-word obfuscation, we can now turn to our experimental setup that operationalises these ideas in a planning evaluation framework. We aim to target generalisation without memorisation in BW with three experiments:

1. Increasing the complexity of the problem by varying the number of blocks (§4).
2. Systematically obfuscating BW problems using nonce words (§5).
3. Combining real-world sources of complexity by increasing the number of blocks and obfuscating using nonce words (§6).

We first replicated the experiments of Valmeekam et al. (2023b) using the more recent

⁴See Appendix C for a more detailed linguistic foundation for nonce word usage.

GPT models 4.1-mini and 5-mini for comparison. We do this for the BW, MYSTERY BW and RANDOMIZED BW domains. These baseline experiments and results are discussed in Appendix F. Later experiments include the addition of the following models: o4-mini, Llama 4 Maverick, Llama 3.3 70B, Qwen2.5 32B, Qwen3 32B, DeepSeek-R1-Distill-Llama and Deepseek-R1. These models were selected to represent the range of different LLMs currently available including closed-source and open-weight models as well as reasoning and non-reasoning LLMs. Non-reasoning models in this work are autoregressive models trained primarily on the next-token prediction task followed by instruction-tuning. Reasoning models share the same foundation as non-reasoning models, but employ explicit chain-of-thought (CoT) during inference as part of their architecture. All experiments are conducted using zero-shot prompting with some additional one-shot experiments discussed in Appendix J. We verify LLM output solutions in all experiments with Fast Downward (Helmert, 2006).⁵ We note that the morphosyntactic preservation in LINGO-BW is at the prompt level, not the tokenizer level and so our methodology ensures that nonce words appear in grammatically well-formed sentences mirroring more realistic set-ups of unfamiliar planning problems.

3.1 Hypotheses

Consistent with prior literature on LLM planning (e.g., Valmeekam et al. (2024b); Stechly et al. (2024)), we hypothesise that an increased number of blocks will lead to increased complexity and therefore, decreased performance. However, we do not expect a linear relationship between the increasing number of blocks and decreasing performance. Given that there are a variety of planning benchmarks available online that leverage 4 blocks, fewer that explore 5 blocks, and very few or none exploring more than 5 blocks, **we hypothesise that those models relying solely on extrapolation from pre-training data (as opposed to more effective inference-time behaviour of problem-solving strategies) may show a notable drop in performance beyond 4 blocks.**

Regarding systematically obfuscating the problems using nonce words, we hypothesise that the part of speech and domain role of the nonce word

⁵Fast Downward is an algorithmic planning tool for established domains which we adapt for LINGO-BW.

will affect how challenging the obfuscation is for the model. Specifically, **we hypothesise that swapping out the actions (e.g., stack and unstack) and the predicates (e.g., is clear, is on top of) will lead to the greatest decrease in model performance.** We base this hypothesis on psycholinguistic and usage-based child language acquisition research (Landau and Gleitman, 1985; Gropen et al., 1991; Naigles et al., 1993). Verbs (in English) have high cue validity (Goldberg, 2006) as to the overall meaning of the sentence as they generally project the semantic roles of the syntactic subject and object (e.g., Dowty (1991)). Thus, obfuscating the verbs may have a clear, detrimental effect on interpreting sentence meaning if models parallel usage-based models of human language learning, which some research has demonstrated (Misra and Mahowald, 2024; Scivetti et al., 2026). Predicates such as “on top of” provide syntactic scaffolding that enables interpretation of a novel or nonce verb, as demonstrated by experimental evidence supporting the syntactic bootstrapping hypothesis: information about verb meanings stems, in part, from the syntactic frames of the verb (e.g., Naigles (1990); Kaschak and Glenberg (2000)). In the case of the BW planning domain, there is consistent use of the syntactic signature of the caused-motion syntactic frame: “stack a block onto another block,” “unstack a block from another block”. Thus, if we obfuscate the path preposition (onto, from) of the syntactic signature, this may also pose a challenge to interpreting verbal semantics in context.

4 Experiment 1: Number of Blocks

4.1 Dataset Development

We use a PDDL generator (Seipp et al., 2022) to create 800 new BW instances across problems with 4, 5, 6 and 12 blocks with equal distribution across each number of blocks. The number of blocks were chosen based on the number of blocks found in existing popular LLM planning benchmarks (Valmeekam et al., 2023b), where 4 blocks is most commonly seen, 5 blocks appears to a lesser degree and 6 and 12 block problems extend beyond the problems present in the benchmarks.

4.2 Experiments

The experiments were run using the standard BW prompt from Valmeekam et al. (2023b) for Llama 4, 4.1-mini and 5-mini. Since reasoning models inherently utilise CoT, we additionally evaluated

non-reasoning models with zero-shot CoT to observe its impact. We restricted our main evaluations to unguided, zero-shot settings to establish a clear baseline for these architectures under structural complexity. Additional experiments with LLM verification are detailed in Appendix G. The zero-shot and CoT results are presented in Figure 1.

4.3 Results

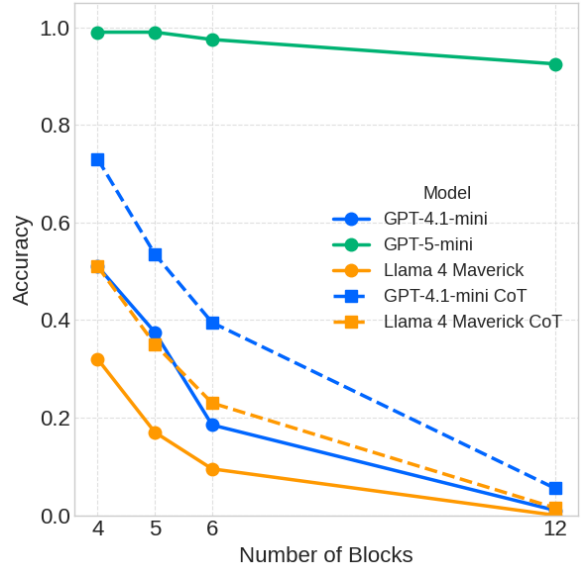


Figure 1: Performance of LLMs on BW problems with 4, 5, 6 and 12 blocks with 200 instances each using the base prompt and CoT for non-reasoning models.

As Figure 1 shows, 5-mini consistently has the best performance, while Llama 4 performs the worst. The addition of zero-shot CoT prompting improves performance for both 4.1-mini and Llama 4. Overall performance degrades as the number of blocks increases for non-reasoning models, even with the addition of the CoT, whilst the reasoning model 5-mini consistently performs the best in comparison.

Recall our hypothesis: models relying solely on extrapolation from pre-training data (as opposed to genuine reasoning and problem-solving strategies) would show a notable drop in performance beyond 4 blocks. Our hypothesis is supported by the performance of the non-reasoning models: we see clear drops as we move beyond 4 blocks, with performance precipitously dropping to 0 or just above for these models. Even 4.1-mini, with the addition of CoT, which shows impressive performance on the 4-block problems of 73%, drops to 5.5% accuracy on 12 blocks. In contrast, performance at 12 blocks

for the reasoning model 5-mini still exceeds the other models’ 4-block performance. This indicates that this reasoning model is indeed leveraging a distinct and more effective problem-solving strategy consistent with stronger performance under our evaluation.

To investigate whether the 5-mini trend we see with varying numbers of blocks is robust across reasoning models or whether the trend is model-specific, we evaluate additional reasoning alternatives. Specifically, we ran a full evaluation across all block number settings (200 instances each) for Qwen3-32B in reasoning mode. The results have been shared in Table 2. To verify whether this pattern holds across other models, we evaluated DeepSeek-R1 on the full set of 12-block problems (200 instances) and DeepSeek-R1-Distill-Llama on a sample of 50 instances in the 12-block setting.

Model	Accuracy (%)			
	4B	5B	6B	12B
5-mini	99.0	99.0	97.5	92.5
Qwen3-32B	82.5	75.5	62.5	6.0

Table 2: Accuracy of reasoning models across varying number of blocks (B).

While Qwen3-32B appears to be more robust when problems contain 4 to 6 blocks compared to the non-reasoning models seen in Figure 1, performance drastically decreases for 12-block problems. Our sample evaluation of DeepSeek-R1-Distill-Llama appears to follow the 12-block collapse observed in Qwen3-32B, achieving 14.0%. However, our full 12-block evaluation of DeepSeek-R1 reveals that it maintains strong structural generalisation compared to non-reasoning model results, reaching an accuracy of 82.0%. While reasoning models may delay the performance degradation seen with non-reasoning models, robustness to problems with much larger numbers of blocks remains model-specific.

Because increasing the number of blocks degrades performance across models, we use 4-block problems in the rest of the experiments unless specified otherwise.

5 Experiment 2: Obfuscating Vocabulary

5.1 Dataset Development

To construct the LINGO-BW dataset, we designed a pipeline for generating nonce-word variants from

the standard BW domain. The goal was to evaluate whether LLMs can reason over syntactically coherent but lexically novel inputs, thus testing linguistic generalisation under controlled abstraction.

Nonce words were generated using an online pseudo-word generator⁶ that produces English-like, semantically meaningless forms. These were manually curated by trained linguists (the authors) and categorised into lexical classes corresponding to nouns, verbs, adjectives, and prepositions. Words longer than ten characters or with implausible English phonotactics were removed. The remaining items were refined for morphological plausibility, resulting in lists of over 300 well-formed nonce words per category.

To maintain grammatical coherence, we developed a lightweight morphological grammar module to handle basic inflectional processes such as pluralisation, negation, past and progressive tense marking, and capitalisation. This allowed nonce verbs and nouns to appear in morphologically valid forms (such as *zorp* → *zorping*, *plim* → *plims*).

The original BW prompt was converted into a template format in which replaceable lexical units were represented by placeholders such as [noun_1] or [verb_2]. These placeholders corresponded to four variable types: *actions*, *predicates*, *objects*, and *names*. Actions represented verb phrases describing manipulation (e.g., “pick up,” “put down”), predicates denoted relational or state-based constructions (e.g., “is holding,” “is on top of”), objects referred to the concrete entities in the environment (e.g., “block,” “object”), and names replaced original colour or identifier labels (e.g., “blue block,” “red block”).

A script randomly substituted each placeholder with an appropriate nonce word from the corresponding list, ensuring grammatical agreement and non-repetition within an instance. Separate templates were created for the domain and problem files to preserve internal consistency between the linguistic prompts and PDDL representations.

To systematically control the level of lexical obfuscation, we defined four difficulty categories determined by the number of lexical variable types replaced. Using the binomial coefficient $C(n, k)$, where $n = 4$ (the number of replaceable variable types) and k is the number of substitutions, we obtained $C(4, 1) = 4$, $C(4, 2) = 6$, $C(4, 3) = 4$, and $C(4, 4) = 1$. This produced the following

⁶<https://www.soybomb.com/tricks/words/>

hierarchy:

- **Easy:** One category replaced (*actions*, *predicates*, *objects*, or *names*).
- **Medium:** Two categories replaced (*actions* + *predicates*, *actions* + *names*, *actions* + *objects*, *predicates* + *names*, *predicates* + *objects*, *names* + *objects*).
- **Hard:** Three categories replaced (*predicates* + *actions* + *names*, *predicates* + *actions* + *objects*, *predicates* + *names* + *objects*, *actions* + *names* + *objects*).
- **Super-hard:** All four categories replaced simultaneously (*predicates* + *actions* + *names* + *objects*).

Each configuration was implemented leveraging the standard BW prompt as the starting point. For every configuration, a CSV file of 500 examples was generated, each containing a mapping between the original lexical items and their nonce replacements. While only a subset was used for the current experiments, the complete dataset was retained as a reusable resource for future work.

Finally, an automated rewriting script applied these mappings to the corresponding PDDL problem files, replacing all original lexical items with their nonce counterparts. This pipeline was created to ensure a reproducible transformation from natural-language prompts to nonce-based PDDL variants. Examples of our obfuscated prompts in different settings have been shared in Appendix E.

5.2 Experiments

We conducted experiments across all four difficulty levels (Easy, Medium, Hard, and Super-hard), each corresponding to an increasing degree of linguistic abstraction. The goal of these experiments was to determine whether model performance degraded systematically as more linguistic cues were obfuscated, whether this varied depending upon the obfuscated linguistic cue, and whether such degradation differed across models with varying architectures.

The nonce substitution process yielded two key types of evaluation data: (a) natural-language problem descriptions with nonce replacements and (b) their corresponding rewritten PDDL instances. Each model was required to generate a valid action sequence (i.e. a plan) that satisfied the nonce-based

PDDL problem. Models were not required to provide the plan in well-formed PDDL. Instead, we allowed the model to provide the action sequence in natural language and then extracted the steps and converted them into PDDL for checking. This choice keeps the focus of this research on the reasoning and generalisation capabilities of models, as opposed to their ability to independently act as planners in working systems.

For a more controlled comparison between reasoning and non-reasoning models, we focus on model pairs of comparable size and families, however, full results for all evaluated models can be found in the appendices.

5.3 Results: Lingo-BW

Results are detailed across the models for the Easy, Medium, Hard and Super-hard settings in Table 3. Overall, we see that the reasoning models demonstrate markedly better performance than the non-reasoning models. However, an analysis across the difficulty settings reveals that both classes of models are highly sensitive to specific types of linguistic obfuscation.

The trends across models and settings support our psycholinguistically motivated hypothesis: swapping out the actions (e.g., *stack* and *unstack*) and the predicates (e.g., *is clear*, *is on top of*) will lead to the greatest decrease in model performance. For the non-reasoning models, Llama, 4.1-mini and Qwen, performance peaks on Easy tasks where only Objects or Names are swapped, but drops sharply when Actions or Predicates are obfuscated (e.g., 4.1-mini scores 35.2% on Objects but falls to 5.7% on Predicates). The reasoning models show varying degrees of robustness on the Easy single-class swaps: 5-mini performs strongly across all four, Qwen handles most, and Llama already shows clear drops on Actions and Predicates. Their vulnerability to relational obfuscation becomes more pronounced in the Medium and Hard settings. When Actions and Predicates are simultaneously obfuscated (A+P), the reasoning variants of Llama and Qwen experience a severe drop to near 10% accuracy. When Objects or Names are obfuscated in addition to Actions and Predicates in the Hard configurations (P+A+O and P+A+N), non-reasoning models drop as low as 0.0%. The reasoning models also follow a similar trend with the notable exception of 5-mini which has the best performance across all models. For 5-mini, we observe a slight drop in performance when both Actions and Predicates are

	Setting	% Mean Accuracy (SD)					
		Llama ◊	4.1-mini ◊	Qwen ◊	Llama ▲	5-mini ▲	Qwen ▲
Easy	Actions	14.3 (1.2)	11.8 (1.2)	5.3 (0.3)	42.0 (2.0)	93.7 (1.3)	79.8 (1.9)
	Names	61.0 (2.8)	30.7 (1.6)	22.7 (1.9)	67.5 (2.6)	84.0 (3.5)	78.3 (3.5)
	Objects	59.2 (2.9)	35.2 (0.8)	27.5 (0.5)	47.3 (0.3)	93.8 (0.3)	59.3 (3.5)
	Predicates	19.7 (1.8)	5.7 (1.6)	15.7 (1.0)	35.8 (1.0)	96.7 (0.3)	61.7 (4.9)
Medium	A+N	4.0 (0.5)	6.0 (1.9)	6.8 (3.2)	44.3 (5.3)	97.7 (0.8)	79.8 (2.8)
	A+O	22.0 (1.3)	15.0 (1.3)	5.2 (0.3)	34.7 (2.8)	93.5 (0.9)	66.3 (2.0)
	A+P	0.2 (0.3)	0.2 (0.3)	4.2 (0.3)	12.7 (1.5)	81.5 (3.6)	11.3 (1.8)
	O+N	64.7 (0.8)	30.8 (2.9)	21.3 (0.8)	54.7 (4.9)	89.7 (1.3)	58.8 (3.2)
	P+N	23.3 (1.3)	6.7 (0.8)	16.0 (0.9)	37.0 (3.8)	96.0 (0.5)	61.0 (3.9)
	P+O	18.0 (1.0)	13.8 (2.0)	17.2 (1.0)	39.2 (1.3)	95.8 (1.5)	75.7 (1.3)
Hard	A+O+N	21.2 (1.5)	7.2 (1.4)	3.0 (0.0)	39.7 (0.6)	92.2 (1.3)	71.2 (1.0)
	P+A+N	0.3 (0.6)	0.5 (0.5)	1.0 (0.0)	8.5 (0.5)	94.8 (2.5)	8.3 (2.0)
	P+A+O	0.0 (0.0)	0.5 (0.0)	4.5 (0.5)	13.3 (4.0)	57.5 (3.5)	13.0 (0.5)
	P+O+N	8.2 (1.0)	9.2 (1.3)	6.7 (0.3)	18.7 (1.0)	93.3 (2.5)	55.2 (1.5)
Super-hard ANOP		0.0 (0.0)	0.5 (0.9)	3.0 (0.0)	4.5 (0.0)	52.8 (2.4)	7.7 (1.5)

Table 3: Results across models for Experiment 2 in which we systematically alter which classes of words (A: Actions, N: Names, O: Objects, P: Predicates) are swapped out for nonce words in the prompt. Each setting has 200 instances. The non-reasoning models are Llama-3.3-70B, GPT-4.1-mini and Qwen2.5-32B indicated with ◊ and the reasoning models are Deepseek-R1-Distill-Llama, GPT-5-mini and Qwen3-32B indicated by ▲. This table reports the mean accuracy across three runs of the experiment with standard deviation (SD) in brackets. The results of each run can be found in Appendix I.

	CoT Setting	% Accuracy	
		Llama 4 ◊	4.1-mini ◊
Easy	Actions	7.5 ↓4.2	35.0 ↑23.2
	Names	40.0 ↑12.5	47.5 ↑16.8
	Objects	35.0 ↑0.8	37.5 ↑2.3
	Predicates	15.0 ↓5.0	22.5 ↑16.8
Super-hard ANOP		2.5 ↑0.4	8.8 ↑8.3

Table 4: Results for non-reasoning models with CoT for Experiment 2 in Easy and Super-hard settings. The arrows and following number show the percentage difference in accuracy compared to the experiments without CoT in the same setting. The results without CoT for 4.1-mini are reported in Table 3 and results without CoT for Llama 4 are from Table 14 in Appendix I.

obfuscated in the medium setting and it reaches one of its lowest scores of 57.5% when Predicates, Actions and Objects are obfuscated in the Hard setting. All models struggle with the Hard setting where Predicates, Actions, and Objects are obfuscated, as this essentially obfuscates all major content words of the domain, preserving only Names identifying blocks, such as “yellow”.

The Super-hard setting, in which we swap all 4 content word classes of the prompts with morphosyntactically valid nonce words, poses a significant challenge even for the reasoning models that

maintain otherwise strong performance in other settings. We see a similar pattern with the non-reasoning model Llama 4 Maverick and reasoning models o4-mini and Deepseek-R1 when we evaluate them against a subset of the data (see Appendix I for more details).

Interestingly, while we found in Experiment 1 that CoT prompting boosted performance when scaling the number of blocks, here we see varying performance. In fact, Llama 4 shows a slight decrease in performance in at least two settings as shown in Table 4. Thus, when the prompts are complex due to linguistic obfuscation, asking the non-reasoning models to think step-by-step does not necessarily provide gains.

6 Experiment 3: Combined Complexity

Here we explore the challenge of combining both structural and lexical complexity such that we test models on 200 distinct 12-block problems in the Super-hard setting (with all four classes of words obfuscated). We limit this test to the best performing GPT model, 5-mini, and the reasoning model DeepSeek-R1 in order to pinpoint different capabilities in these two distinct reasoning model architectures.

We find that 5-mini performance degrades only slightly from Super-hard performance with 4 blocks, achieving an accuracy of 49.5%. DeepSeek-R1 performance degrades from an already low 15% on Super-hard with 4 blocks to 1% accuracy when the number of blocks is increased to 12. To aid with the interpretability of these results, we compared the performance of 5-mini and DeepSeek-R1 on the standard 12-block setting with no obfuscation from Experiment 1 to the 12-block Super-hard setting (Experiment 3).

Model	Accuracy (%) 12 Blocks	
	Standard	Super-hard
GPT-5-mini	92.5	49.5
DeepSeek-R1	82.0	1.0

Table 5: Accuracy of structural and combined complexity at 12 blocks.

For both models, the significant degradation in this combined setting demonstrates that their failures are not solely driven by structural complexity. The accuracy of 5-mini remains relatively stable between the 4-block and 12-block Super-hard settings, but is much lower than its 12-block performance with no obfuscation. This points to lexical obfuscation as the primary bottleneck for this model. For DeepSeek-R1, while it successfully handles structural complexity in the standard 12-block setting, combining structural complexity with lexical obfuscation is even more challenging than full lexical obfuscation, dropping accuracy to 1.0%.

Thus, we present this combined setting as a challenge dataset and hope that it can be used to contribute to improving LLM generalisation from known problems to novel vocabulary in future work.

7 Error Analysis

Reasoning models achieve remarkable performance at this task, especially when compared to other models. Here, we conduct a qualitative manual error analysis to explore the different patterns of errors exhibited by different model architectures. In addition to this, we conduct a quantitative error analysis that can be found in [Appendix K](#).

7.1 Non-reasoning, Non-CoT

We conduct in-depth error analysis of the non-reasoning model, Llama 4. The errors exhibited

by this model demonstrate a violation of all problem constraints via a disconnect from the state of the problem. For example, blocks are picked up, even though they are underneath another block. In addition, many actions appear to have no lasting effect (e.g., picking up a block and then immediately putting it down in the next action).

Notably, even in the most difficult setting where all word categories are obfuscated, the model often produces correct *local* relationships between blocks. While local relationships are often correct, the overall *global* goal state can be incorrect: the full set stack of more than two blocks is not achieved. Regardless, the non-reasoning model struggles both to model the initial world state and to follow domain constraints.

7.2 Non-reasoning, CoT

We compare the performance of Llama 4 described above to the performance when we explicitly use CoT prompting. Here, the errors differ slightly: primarily CoT prompting improves models’ ability to follow instructions, in particular the use of the critical stack action. However, the model continues to make errors such as holding multiple blocks at once because a block was not put down.

7.3 Reasoning Models

We conduct error analysis of reasoning models which have far fewer errors. Specifically, we explore the outputs of Deepseek-R1, 5-mini, and o4-mini. The models tend to miss one or two steps instead of demonstrating patterns of multiple errors and incorrect actions that are pointless or disconnected from the start state. There is also a clear pattern in the step that is missed: all reasoning models sometimes fail to insert the put-down step after unstack actions. Anecdotally, students learning planning often make the same mistake, likely because unstack can be intuitively interpreted as both removing and placing a block. The other frequent error is failing to pick-up a block that is on the table before stacking it. Similar to unstacking, we note that this is conceptually an easy error to make, as stacking could also be understood as a continuous action in which the block is picked up and then stacked.

Despite these violations, the reasoning models consistently achieve the intended goal configuration if constraints are relaxed. Unlike the non-reasoning models, they produce correct outcomes at both the *local* level (individual block relations)

and the *global* level (larger multi-block structures). This suggests that reasoning models can correctly represent the initial state and select appropriate actions, but they still struggle to fully enforce the domain’s operational constraints.

7.4 Analysis of Reasoning Traces

Analysis of the LLM reasoning traces reveals that models systematically lay out each initial condition and check the feasibility of any action given the current conditions, before deciding upon an action. The models generally track the initial state and state changes via text lists.

Within the reasoning traces, we do not see any attempt to establish a mapping to known vocabulary.⁷ Models do not seem to translate the lexical items during the reasoning process. For instance, 5-mini notes it does not know the nonce words: “and the tegust is not astleyed (unclear what “astleyed” means exactly, but assuming wilchip is unastleyed initially)...” We do note one exception to a type of translation into a known vocabulary domain: words that were very similar to real words were sometimes assumed to be typos, for example, “balacing” is sometimes converted to “balancing” in models’ reasoning traces.

8 Discussion & Conclusions

We introduce LINGO-BW, a framework designed to generate a dataset of test instances to evaluate the generalisation and reasoning abilities of LLMs in classical planning. By systematically increasing the number of blocks and lexical obfuscation, we provide a fine-grained benchmark with a mechanism to generate new instances to minimise possible contamination.

Our findings confirm a hierarchy in model behaviour. Non-reasoning models struggle to generalise, with performance collapsing under even moderate complexity. In contrast, reasoning models demonstrate stronger performance across conditions, although this performance also has some limits. Accuracy degrades when structural complexity and linguistic obfuscation are combined, with even state-of-the-art reasoning models declining at the 12-block level under full lexical obfuscation. These results demonstrate and quantify the gap between current model capabilities and the demands of real-world planning environments.

⁷Translation to known vocabulary is one potential technique we have observed for human planners (see [Appendix L](#) for more details).

We do not claim that LLMs reason like humans, nor that they should, but we see parallels in performance patterns demonstrated through the support of our theoretically-motivated hypothesis. This suggests that similar structural dependencies may emerge from exposure to linguistic input. Looking at the results in [Table 3](#), there are clear trends that mirror patterns found in human language processing.⁸ These results invite further investigation into whether distributional learning in LLMs implicitly encodes usage-based regularities similar to those that shape human grammatical development.

The conditions tested here show that strong generalisation remains out of reach for current LLMs, but the more useful finding is how their behaviour shifts with the linguistic form of the input. Obfuscating verbs and predicates hurts performance more than obfuscating nouns and names, combining structural and lexical difficulty exposes differences between reasoning models that simpler benchmarks miss, and reasoning models fail in qualitatively different ways from non-reasoning ones. This finding adds to a growing body of research demonstrating that claims surrounding model ability to reason and plan deteriorate markedly when atypical words, ordering, symbols, or counterfactuals are swapped into existing benchmark problems ([Wu et al., 2024](#); [Lewis and Mitchell, 2025](#); [Tayyar Madabushi et al., 2025](#)). None of this is visible from accuracy on the standard BW prompt alone. Yet, if models are using robust, abstract reasoning procedures, such swaps should not dramatically affect performance. This indicates that models may be using shortcuts and structural similarity to training data. Note that we do not suggest this is memorisation alone (that is simply not possible with our data), nor is it robust reasoning. We conclude that this is further evidence that models rely on Context-Directed Extrapolation (CDE) from pre-training data to solve tasks ([Tayyar Madabushi et al., 2025](#)). Specifically, CDE posits that LLMs generate outputs by extrapolating from relevant priors in their pre-training data, and therefore tend to be more effective on tasks that are closer to, and better represented in, that data. We hope that this dataset will contribute to future work focused on better understanding and improving the core reasoning and generalisation capabilities of models.

⁸See [Appendix D](#) for more details.

9 Ethics Statement

The evaluation of LLM planning capabilities has broad implications for their deployment and effective use. Mischaracterisation of LLM capabilities has the potential to dramatically impact downstream use of these systems. As such, it is essential that we carefully understand what they can and cannot do. Concretely, errors in automated planning, especially in critical domains such as energy and infrastructure, could result in catastrophic harms. Therefore, misinterpretation of LLM capabilities is very dangerous and we do not claim that our study is a representation of planning in general. However, this work is a good *indication* of the limits of LLM capabilities.

We acknowledge that LLMs use a huge amount of resources and take seriously their environmental impact. However, we balance this against the need to analyse what applications merit the costly use of LLMs and what applications LLMs are not currently suited to, including potentially planning onboard robots.

Limitations

Our work focuses on a single planning problem albeit quite extensively and across different axes of complexity. Future work should explore different planning problems along the same axis so as to ensure a more detailed evaluation of LLMs. Additionally, our evaluation is limited to the English language, and it is absolutely essential that we evaluate these models in other languages, especially low resource languages.

Acknowledgments

This work was partly supported by the UKRI Centre for Doctoral Training in Accountable, Responsible, and Transparent AI [EP/S023437/1] of the University of Bath and the UKRI Strength in Places Fund MyWorld Project (SIPF00006/1).

References

Joana Acha, Itziar Laka, Josu Landa, and Pello Salaburu. 2014. EHME: A new word database for research in Basque language. *The Spanish Journal of Psychology*, 17:E79.

Jose Armando Aguasvivas, Manuel Carreiras, Marc Brysbaert, Paweł Mandera, Emmanuel Keuleers, and Jon Andoni Duñabeitia. 2018. SPALEX: A spanish lexical decision database from a massive online data collection. *Frontiers in Psychology*, 9:2156.

Dang Anh, Limor Raviv, and Lukas Galke. 2024. *Morphology matters: Probing the cross-linguistic morphological generalization abilities of large language models through a wug test*. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 177–188, Bangkok, Thailand. Association for Computational Linguistics.

David Arps, Laura Kallmeyer, Younes Samih, and Hassan Sajjad. 2024. *Multilingual nonce dependency treebanks: Understanding how language models represent and process syntactic structure*. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7822–7844, Mexico City, Mexico. Association for Computational Linguistics.

Jean Berko. 1958. The child’s learning of English morphology. *WORD*, 14(2-3):150–177.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. *Language models are few-shot learners*. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Bastian Bunzeck, Daniel Duran, Leonie Schade, and Sina Zarrieß. 2025. *Small language models also work with small vocabularies: Probing the linguistic abilities of grapheme- and phoneme-based baby llamas*. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6039–6048, Abu Dhabi, UAE. Association for Computational Linguistics.

Yongchao Chen, Jacob Arkin, Charles Dawson, Yang Zhang, Nicholas Roy, and Chuchu Fan. 2024. *AutoTAMP: Autoregressive task and motion planning with LLMs as translators and checkers*. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6695–6702.

Alessandro Cimatti, Marco Pistore, and Paolo Traverso. 2008. *Automated planning*. In Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, editors, *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*, chapter 22, pages 841–867. Elsevier.

Andrea Gregor de Varda, Daniele Gatti, Marco Marelli, and Fritz Günther. 2024. *Meaning beyond lexicality: Capturing pseudoword definitions with language models*. *Computational Linguistics*, 50(4):1313–1343.

David Dowty. 1991. *Thematic proto-roles and argument selection*. *Language*, 67(3):547–619.

Ludovic Ferrand, Boris New, Marc Brysbaert, Emmanuel Keuleers, Patrick Bonin, Alain Méot, Maria

- Augustinova, and Christophe Pallier. 2010. [The French lexicon project: Lexical decision data for 38,840 French words and 38,840 pseudowords](#). *Behavior Research Methods*, 42(2):488–496.
- Dedre Gentner. 1982. Why nouns are learned before verbs: Linguistic relativity versus natural partitioning. In Stan A. Kuczaj, II, editor, *Language Development, Volume 2: Language, Thought and Culture*, pages 301–334. Lawrence Erlbaum Associates, Hillsdale, NJ.
- Dedre Gentner. 2006. [Why verbs are hard to learn](#). In Kathy Hirsh-Pasek and Roberta Michnick Golinkoff, editors, *Action meets word: How children learn verbs*, pages 544–564. Oxford University Press, New York, NY.
- Adele Goldberg. 2006. *Constructions at Work: The Nature of Generalization in Language*. Oxford University Press.
- Jess Gropen, Steven Pinker, Michelle Hollander, and Richard Goldberg. 1991. [Affectedness and direct objects: The role of lexical semantics in the acquisition of verb argument structure](#). *Cognition*, 41(1–3):153–195.
- Claudia Grundke, Gabriele Röger, and Malte Helmert. 2024. [Formal representations of classical planning domains](#). In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 34, pages 239–248.
- Malte Helmert. 2006. [The Fast Downward planning system](#). *Journal of Artificial Intelligence Research*, 26:191–246.
- Michael P. Kaschak and Arthur M. Glenberg. 2000. [Constructing meaning: The role of affordances and grammatical constructions in sentence comprehension](#). *Journal of Memory and Language*, 43(3):508–529.
- Emmanuel Keuleers and Marc Brysbaert. 2010. [Wuggy: A multilingual pseudoword generator](#). *Behavior research methods*, 42(3):627–633.
- Marta Kutas and Steven A Hillyard. 1983. [Event-related brain potentials to grammatical errors and semantic anomalies](#). *Memory & cognition*, 11(5):539–550.
- Barbara Landau and Lila R. Gleitman. 1985. *Language and Experience: Evidence from the Blind Child*, volume 8 of *Cognitive Science Series*. Harvard University Press, Cambridge, MA.
- Martha Lewis and Melanie Mitchell. 2025. [Evaluating the robustness of analogical reasoning in large language models](#). *Transactions on Machine Learning Research*.
- Kanishka Misra and Kyle Mahowald. 2024. Language models learn rare phenomena from less rare phenomena: The case of the missing AANNs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 913–929. Association for Computational Linguistics.
- Simone Mätzig, Judit Druks, Jackie Masterson, and Gabriella Vigliocco. 2009. [Noun and verb differences in picture naming: Past studies and new evidence](#). *Cortex*, 45(6):738–758. Epub 2008 Nov 1.
- Letitia Naigles. 1990. [Children use syntax to learn verb meanings](#). *Journal of Child Language*, 17(2):357–374.
- Letitia Naigles, Lila R. Gleitman, and Henry Gleitman. 1993. Children acquire word meaning components from syntactic evidence. In Esther Dromi, editor, *Language and Cognition: A Developmental Perspective*, pages 104–140. Ablex, Norwood, NJ.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. [The Proposition Bank: An annotated corpus of semantic roles](#). *Computational Linguistics*, 31(1):71–106.
- Sameer Pradhan, Julia Bonn, Skatje Myers, Kathryn Conger, Tim O’gorman, James Gung, Kristin Wright-bettner, and Martha Palmer. 2022. [PropBank comes of Age—Larger, smarter, and more diverse](#). In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 278–288, Seattle, Washington. Association for Computational Linguistics.
- Stuart J. Russell and Peter Norvig. 2021. *Artificial Intelligence: A Modern Approach*, 4th edition. Pearson Education, Harlow, England. Global Edition.
- Karin Kipper Schuler. 2005. [VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon](#). Ph.D. thesis, University of Pennsylvania, Philadelphia, PA. AAI3179808.
- Wesley Scivetti*, Melissa Torgbi*, Mollie Shichman, Taylor Pellegrin, Austin Blodgett, Claire Bonial, and Harish Tayyar Madabushi. 2025. [Beyond memorization: Assessing semantic generalization in large language models using phrasal constructions](#). In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 1184–1201, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.
- Wesley Scivetti, Ethan Wilcox, Nathan Schneider, Kanishka Misra, and Leonie Weissweiler. 2026. Language models learn constructional semantics, *not to mention* syntax: Investigating LM understanding of PAIRED-FOCUS constructions. In *Proceedings of the 30th Conference on Computational Natural Language Learning*, pages 250–267, San Diego, California, USA. Association for Computational Linguistics.
- Jendrik Seipp, Álvaro Torralba, and Jörg Hoffmann. 2022. [PDDL generators](#). <https://doi.org/10.5281/zenodo.6382173>.

- Bilgehan Sel, Ruoxi Jia, and Ming Jin. 2025. [LLMs can plan only if we tell them](#). *International Conference on Learning Representations*, pages 78024–78064.
- Parshin Shojaei, Seyed Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. 2025. [The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- John Slaney and Sylvie Thiébaux. 2001. [Blocks World revisited](#). *Artificial Intelligence*, 125(1-2):119–153.
- Ana Paula Soares, Álvaro Iriarte, José João de Almeida, Alberto Simões, Ana Costa, João Machado, Patrícia França, Montserrat Comesaña, Andreia Rauber, Anabela Rato, and Manuel Perea. 2018. [ProcuraPALavras \(P-PAL\): A Web-based interface for a new European Portuguese lexical database](#). *Behavior Research Methods*, 50(4):1461–1481.
- Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2023. [Beyond the imitation game: Quantifying and extrapolating the capabilities of language models](#). *Transactions on Machine Learning Research*.
- Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. 2024. [Chain of thoughtlessness? an analysis of cot in planning](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 29106–29141. Curran Associates, Inc.
- Harish Tayyar Madabushi, Melissa Torgbi, and Claire Bonial. 2025. [Neither stochastic parroting nor AGI: LLMs solve tasks through context-directed extrapolation from training data priors](#). *arXiv preprint arXiv:2505.23323*.
- Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2023a. [PlanBench: An extensible benchmark for evaluating large language models on planning and reasoning about change](#). *Advances in Neural Information Processing Systems*, 36:38975–38987.
- Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. 2023b. [On the planning abilities of large language models—a critical investigation](#). *arXiv preprint arXiv:2305.15771*.
- Karthik Valmeekam, Kaya Stechly, Atharva Gundawar, and Subbarao Kambhampati. 2024a. [Planning in strawberry fields: Evaluating and improving the planning and scheduling capabilities of LRM o1](#). *arXiv preprint arXiv:2410.02162*.
- Karthik Valmeekam, Kaya Stechly, and Subbarao Kambhampati. 2024b. [LLMs still can’t plan; can LRMs? a preliminary evaluation of OpenAI’s o1 on PlanBench](#). *arXiv preprint arXiv:2409.13373*.
- Cyma Van Petten and Marta Kutas. 1991. [Influences of semantic and syntactic context on open- and closed-class words](#). *Memory & Cognition*, 19(1):95–112.
- Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. [Emergent abilities of large language models](#). *Transactions on Machine Learning Research*, 2022.
- Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schuetze, Kemal Oflazer, and David Mortensen. 2023. [Counting the bugs in ChatGPT’s wugs: A multilingual investigation into the morphological capabilities of a large language model](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6508–6524, Singapore. Association for Computational Linguistics.
- Terry Winograd. 1972. [Understanding natural language](#). *Cognitive Psychology*, 3(1):1–191.
- Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. 2024. [Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1819–1862, Mexico City, Mexico. Association for Computational Linguistics.

A Scientific Artefacts

We use several scientific artefacts as part of this work. We use all artefacts as intended, provided the intended use was specified. This includes the following artefacts:

1. We use an online pseudo-word generator to generate nonsense words for our dataset ⁹.
2. We build on the work of [Valmeekam et al. \(2023b\)](#) and their corresponding code repository ¹⁰ under the MIT Licence.
3. We use Fast Downward for PDDL groundtruth and planning evaluation under the GNU General Public License v3.0. ¹¹
4. We use [Seipp et al. \(2022\)](#)’s PDDL generator to generate new planning problems.

⁹<https://www.soybomb.com/tricks/words/>

¹⁰https://github.com/karthikv792/LLMs-Planning/tree/main/llm_planning_analysis

¹¹<https://www.fast-downward.org/>

We have also created artefacts intended for use in a research context, which is compatible with the original access conditions. A detailed description and documentation of the artefacts we created can be found in Appendix E.

B Shortcomings of Mystery BW and Randomized BW

Valmeekam et al. (2024b) state that MYSTERY BW is “semantically identical but syntactically obfuscated.” However, we find this characterisation does not hold under linguistic analysis. While MYSTERY BW aims to maintain the same underlying structure as the original BW tasks, its linguistic form is neither semantically identical nor syntactically equivalent. The number of predicates and actions differs across BW and MYSTERY BW, and the ways in which these elements interact with their constituents diverge in ways that alter the meaning of the prompt.

Specifically, our analysis finds that the MYSTERY BW variant substantially reduces the number of nouns, eliminates adjectives and relative clauses, and simplifies the overall structure of the domain. These omissions alter both syntactic form and propositional content. Consequently, the MYSTERY BW prompt is semantically simplified and syntactically divergent from the original BW.

Another limitation of MYSTERY BW lies in the specific lexical substitutions chosen. Verbs like “succumb” or “feast” evoke strong affective and situational associations. These words are not semantically neutral but instead grounded in the distributional representations encoded during pre-training. Thus, any evaluation involving them is contaminated by latent semantic biases.

To illustrate the linguistic implications of this, consider PropBank (Pradhan et al., 2022), a framework for semantic role labeling that provides an authoritative lexicon of syntactic verbal arguments. The verb “stack,” which appears in the original BW, is characterized by three syntactic arguments: a stacker, thing stacked, and thing stacked onto.¹²

Verb: stack-01

Arg0-PAG: stacker

Arg1-PPT: one thing

Arg2-GOL: the other thing Arg1 is stacked onto (destination)

¹²See <https://propbank.github.io/v3.4.0/frames/stack.html#stack.01>.

In this case, the human or robot performing the stacking is the Arg0, the block being stacked is the Arg1, and the surface it is stacked onto is the Arg2.

In MYSTERY BW, however, “stack” is replaced by “overcome” (see Table 1). According to PropBank, the verb “overcome” is characterised by just two syntactic arguments, an overcomer and the thing-overcome or difficulty.¹³

Verb: overcome-01

Arg0-PAG: overcomer

Arg1-PPT: difficulty

Unlike “stack,” the verb “overcome” lacks a third syntactic argument; thus, its predicate-argument structure is non-equivalent. It cannot encode a goal or destination argument, thereby changing the entire semantic class of the event. Even if one were to substitute “stack” with a semantically similar class of verbs, such as “spray” (Palmer et al., 2005; Schuler, 2005), inherent differences in verbal semantics, such as telicity or instrumentality, would persist. Because LLMs encode such distinctions in their vector spaces, random lexical substitutions necessarily alter the distributional semantics of the task.

In short, this demonstrates that the use of arbitrary real words introduces unwanted semantic *sentiment*. It is not linguistically informed to assume that replacing verbs and predicates without regard to their predicate-argument structure will leave model reasoning unaffected. These discrepancies, along with similar inconsistencies observed between MYSTERY BW and RANDOMIZED BW, motivated our development of LINGO-BW. The goal of our project is to actually retain the semantic and syntactic integrity of the original prompt while removing prior lexical associations through the introduction of nonce words within the traditional BW framework.

C Nonce Words Additional Background

The linguistic foundation for nonce-word use originates from the “Wug Test,” first introduced by Jean Berko (1958) to study children’s morphological generalisation. Nonce words—phonotactically valid but invented words such as “wug”—are designed to follow the sound patterns of a target language without conveying meaning. Datasets of nonce words have since been developed for numerous languages (Ferrand et al., 2010; Aguasvivas

¹³See <https://propbank.github.io/v3.4.0/frames/overcome.html#overcome.01>.

et al., 2018; Soares et al., 2018; Acha et al., 2014; Keuleers and Brysbaert, 2010) and are used widely in psycholinguistics as well as in recent computational studies probing the linguistic and morphological capabilities of LLMs (Weissweiler et al., 2023; de Varda et al., 2024; Bunzeck et al., 2025; Arps et al., 2024; Anh et al., 2024).

de Varda et al. (2024) show that even pseudowords elicit systematic semantic associations in both humans and LLMs due to sublexical regularities. This suggests that, when reading, meaning construction can emerge dynamically from form. We acknowledge this as a caveat that nonce words may not be entirely semantically neutral, and some emergent structure inevitably arises from phonotactic similarity and learned linguistic patterns, which may be interpreted as typos amongst other things.

However, this residual regularity is far preferable to the alternative employed in the RANDOMIZED BW, where words are replaced with random alphanumeric strings that violate the phonotactic and grammatical conventions of English. Such randomised substitutions strip away not only meaning but also the linguistic scaffolding that enables compositional reasoning, rendering the prompts unnatural and hard to read for both humans and language models.

In contrast, nonce words preserve linguistic coherence while minimising semantic contamination. They retain the grammatical cues necessary for reasoning, allowing us to evaluate planning and generalisation independently of lexical familiarity or memorised associations. LINGO-BW thus offers a linguistically principled compromise that is syntactically valid and phonotactically plausible, yet semantically novel. The intention here is to converge upon the two goals of having naturalistic language while still retaining a controlled experimental design, which other planning problem datasets, such as MYSTERY BW, have struggled to reconcile between.

D Parallels between Psycholinguistic Research and Model Results

Looking at the results in Table 3, §5.3 of the paper, there are clear trends that mirror patterns found in human language processing, though the present results do not by themselves establish human-like processing mechanisms. Psycholinguistic research distinguishes between “open” (e.g., nouns, verbs) and “closed” (e.g., prepositions, pronouns) classes

of words (Van Petten and Kutas, 1991; Kutas and Hillyard, 1983) and open-class words are easier to learn. Studies have also shown that verbs are typically more difficult to learn and replace than nouns because they encode relational and argument-structure information that organises the clause (Gentner, 1982, 2006; Mätzig et al., 2009). The fact that the models are most sensitive to verb-like and relational terms implies that they, like human learners, might rely on the distributional regularities that tie meaning and syntax most strongly to the verbal core of the sentence. Our results demonstrate a striking similarity to this prior work in psycholinguistics. Based on these findings, we hypothesised that nonce substitutions of “Names” (colour adjectives) and “Objects” (nouns such as *blocks*) would be easier for models to handle than substitutions involving “Predicates” or “Actions,” both of which disrupt the verbal core of the sentence. Moreover, the “Predicates” substitutions also involved replacing prepositions such as “on” or “on top of,” which are typically closed-class elements. Because these words encode spatial and structural relationships, their replacement with nonce forms likely added an additional layer of difficulty for models.

The fact that the models are most sensitive to verb-like and relational terms implies that they, like human learners, might rely on the distributional regularities that tie meaning and syntax most strongly to the verbal core of the sentence. At the same time, our error analysis highlights important divergences: while human solvers may ground new words in known conceptual frameworks such as through strategies like translation, the models tend to reason symbolically through state transitions without appealing to lexical meaning.

E Prompt Examples and Dataset Statistics

The LINGO-BW dataset we create uses English and nonsense words to obfuscate problems in the blocksworld planning domain. Tables 7, 8, 9 and 10 show examples of the obfuscated prompt in the easy setting. Table 11 shows an example of a fully obfuscated prompt from the Super-hard setting. The medium and hard settings are not explicitly shown here as examples but have different combinations of actions, names, objects and predicates obfuscated, similar to what is displayed in these example prompts.

For Experiment 1, where we vary the number of blocks in the planning problems, we have 200

instances for each number of blocks (4, 5, 6 and 12 blocks), so a total of 800 examples.

For Experiment 2, where we obfuscate vocabulary, in our experiments for the easy, medium, hard and super-hard settings, we test on 200 examples for each combination of obfuscation. Table 6 shows a breakdown of the number of examples for each setting.

	Setting	Number of Examples
Easy	<u>A</u> ctions	200
	<u>N</u> ames	200
	<u>O</u> bjects	200
	<u>P</u> redicates	200
Medium	A+N	200
	A+O	200
	A+P	200
	O+N	200
	P+N	200
	P+O	200
Hard	A+O+N	200
	P+A+N	200
	P+A+O	200
	P+O+N	200
Super-hard ANOP		200
Challenge Super-hard ANOP and 12 blocks		200

Table 6: Number of examples in the LINGO-BW dataset for Experiment 2 in which we systematically alter which classes of words (A: Actions, N: Names, O: Objects, P: Predicates) are swapped out for nonce words in the prompt.

Setting	Prompt
Actions	I am playing with a set of blocks where I need to arrange the blocks into cornintes . Here are the actions I can do Laral a block Uncorninte a block from on top of another block Braine a block Corninte a block on top of another block I have the following restrictions on my actions: I can only laral or uncorninte one block at a time. I can only laral or uncorninte a block if my hand is empty. I can only laral a block if the block is on the table and the block is clear. A block is clear if the block has no other blocks on top of it and if the block is not laralled . I can only uncorninte a block from on top of another block if the block I am uncorninting was really on top of the other block. I can only uncorninte a block from on top of another block if the block I am uncorninting is clear. Once I laral or uncorninte a block, I am holding the block. I can only braine a block that I am holding. I can only corninte a block on top of another block if I am holding the block being corninted . I can only corninte a block on top of another block if the block onto which I am corninting the block is clear. Once I braine or corninte a block, my hand becomes empty. Once you corninte a block on top of a second block, the second block is no longer clear. [STATEMENT] As initial conditions I have that, the blue block is clear, the hand is empty, the blue block is on top of the orange block, the orange block is on top of the yellow block, the yellow block is on top of the red block and the red block is on the table. My goal is to have that the blue block is on top of the red block. What is the plan to achieve my goal? Just give the actions in the plan.

Table 7: Prompt example with obfuscated actions.

Setting	Prompt
Names	I am playing with a set of blocks where I need to arrange the blocks into stacks. Here are the actions I can do Pick up a block Unstack a block from on top of another block Put down a block Stack a block on top of another block I have the following restrictions on my actions: I can only pick up or unstack one block at a time. I can only pick up or unstack a block if my hand is empty. I can only pick up a block if the block is on the table and the block is clear. A block is clear if the block has no other blocks on top of it and if the block is not picked up. I can only unstack a block from on top of another block if the block I am unstacking was really on top of the other block. I can only unstack a block from on top of another block if the block I am unstacking is clear. Once I pick up or unstack a block, I am holding the block. I can only put down a block that I am holding. I can only stack a block on top of another block if I am holding the block being stacked. I can only stack a block on top of another block if the block onto which I am stacking the block is clear. Once I put down or stack a block, my hand becomes empty. Once you stack a block on top of a second block, the second block is no longer clear. [STATEMENT] As initial conditions I have that, the facizer block is clear, the michime block is clear, the tegust block is clear, the hand is empty, the tegust block is on top of the westoor block, the facizer block is on the table, the michime block is on the table and the westoor block is on the table. My goal is to have that the facizer block is on top of the tegust block, the michime block is on top of the facizer block and the westoor block is on top of the michime block. What is the plan to achieve my goal? Just give the actions in the plan.

Table 8: Prompt example with obfuscated names.

Setting	Prompt
Objects	I am playing with a set of facizers where I need to arrange the facizers into stacks. Here are the actions I can do Pick up a facizer Unstack a facizer from on top of another facizer Put down a facizer Stack a facizer on top of another facizer I have the following restrictions on my actions: I can only pick up or unstack one facizer at a time. I can only pick up or unstack a facizer if my hand is empty. I can only pick up a facizer if the facizer is on the table and the facizer is clear. A facizer is clear if the facizer has no other facizers on top of it and if the facizer is not picked up. I can only unstack a facizer from on top of another facizer if the facizer I am unstacking was really on top of the other facizer . I can only unstack a facizer from on top of another facizer if the facizer I am unstacking is clear. Once I pick up or unstack a facizer , I am holding the facizer . I can only put down a facizer that I am holding. I can only stack a facizer on top of another facizer if I am holding the facizer being stacked. I can only stack a facizer on top of another facizer if the facizer onto which I am stacking the facizer is clear. Once I put down or stack a facizer , my hand becomes empty. Once you stack a facizer on top of a second facizer , the second facizer is no longer clear. [STATEMENT] As initial conditions I have that, the blue facizer is clear, the hand is empty, the blue facizer is on top of the orange facizer , the orange facizer is on top of the yellow facizer , the yellow facizer is on top of the red facizer and the red facizer is on the table. My goal is to have that the blue facizer is on top of the red facizer . What is the plan to achieve my goal? Just give the actions in the plan.

Table 9: Prompt example with obfuscated objects.

Setting	Prompt
Predicates	I am playing with a set of blocks where I need to arrange the blocks into stacks. Here are the actions I can do Pick up a block Unstack a block from syntwards another block Put down a block Stack a block syntwards another block I have the following restrictions on my actions: I can only pick up or unstack one block at a time. I can only pick up or unstack a block if my michime is daydendate . I can only pick up a block if the block is syntwards the facizer and the block is hummony . A block is hummony if the block has no other blocks syntwards it and if the block is not picked up. I can only unstack a block from syntwards another block if the block I am unstacking was really syntwards the other block. I can only unstack a block from syntwards another block if the block I am unstacking is hummony . Once I pick up or unstack a block, I am abathitcrying the block. I can only put down a block that I am abathitcrying . I can only stack a block syntwards another block if I am abathitcrying the block being stacked. I can only stack a block syntwards another block if the block onto which I am stacking the block is hummony . Once I put down or stack a block, my michime becomes daydendate . Once you stack a block syntwards a second block, the second block is no longer hummony . [STATEMENT] As initial conditions I have that, the blue block is hummony , the michime is daydendate , the blue block is syntwards the orange block, the orange block is syntwards the yellow block, the yellow block is syntwards the red block and the red block is syntwards the facizer . My goal is to have that the blue block is syntwards the red block. What is the plan to achieve my goal? Just give the actions in the plan.

Table 10: Prompt example with obfuscated predicates.

Setting	Prompt
Super-hard ANOP	I am playing with a set of tegusts where I need to arrange the tegusts into chrirts . Here are the actions I can do Astley a tegust Unchrit a tegust from aneouswards another tegust Limigrow a tegust Chrit a tegust aneouswards another tegust I have the following restrictions on my actions: I can only astley or unchrit one tegust at a time. I can only astley or unchrit a tegust if my michime is hardery . I can only astley a tegust if the tegust is aneouswards the facizer and the tegust is expokerory . A tegust is expokerory if the tegust has no other tegusts aneouswards it and if the tegust is not astleyed . I can only unchrit a tegust from aneouswards another tegust if the tegust I am unchritting was really aneouswards the other tegust . I can only unchrit a tegust from aneouswards another tegust if the tegust I am unchritting is expokerory . Once I astley or unchrit a tegust , I am balacing the tegust . I can only limigrow a tegust that I am balacing . I can only chrit a tegust aneouswards another tegust if I am balacing the tegust being chritted . I can only chrit a tegust aneouswards another tegust if the tegust onto which I am chritting the tegust is expokerory. Once I limigrow or chrit a tegust , my michime becomes hardery . Once you chrit a tegust aneouswards a second tegust , the second tegust is no longer expokerory . [STATEMENT] As initial conditions I have that, the wilchip tegust is aneouswards the corpastion tegust , the corpastion tegust is aneouswards the consvia tegust , the consvia tegust is aneouswards the westoor tegust , the westoor tegust is aneouswards the facizer , the wilchip tegust is expokerory and the michime is hardery . My goal is to have that the wilchip tegust is aneouswards the westoor tegust . What is the plan to achieve my goal? Just give the actions in the plan.

Table 11: Prompt example with obfuscated actions, names, objects and predicates.

F Baseline Experiments

For direct performance comparison with directly comparable work, we replicate the experiments of Valmeekam et al. (2023b) using the more recent models gpt-4.1-mini-2025-04-14 and gpt-5-mini-2025-08-07 representing a non-reasoning model and reasoning model respectively. We do this for the BW, MYSTERY BW and RANDOMIZED BW domains and refer to these as the baseline experiments. These experiments consist of 600 instances, the number of blocks in each instance ranging from 3 to 5, with 4 blocks being the most represented. We compare the performance of baseline results with the addition of the DeepSeek-R1 results taken from (Valmeekam et al., 2023b) and the same models’ performance on our Lingo-BW in the fully obfuscated super-hard setting with 4 blocks. The results are shared in Table 12.

F.1 Results: Baseline

The results for 4.1-mini appear to be consistent with results reported for similar models in the BW domain; however, the model’s performance of 9% accuracy on Mystery BW is marginally higher than the model’s predecessors. Models including gpt-4 and gpt-4o get an accuracy of 0% on the obfuscated BW benchmarks (Valmeekam et al., 2023b). All selected models perform the worst on our Lingo-BW benchmark.

Domain	Accuracy (%)		
	4.1-mini	5-mini	Deepseek-R1
BW	48.0	98.2	99.2
Mystery BW	9.0	74.7	43.3
Random BW	1.0	67.5	25.8
Lingo-BW	0.0	52.4	15.0

Table 12: Results for the BW, MYSTERY BW, RANDOMIZED BW baseline experiments and the Lingo-BW benchmark.

The performance of the reasoning model 5-mini is substantially better than 4.1-mini in the 3 domains, showing similar performance to other reasoning models on BW and Mystery BW, including Deepseek-R1 and o1-preview, but superior performance in the RANDOMIZED BW domain (Valmeekam et al., 2023b). To summarise, we benchmark on models that outperform leading models on past obfuscated planning benchmarks.

When compared to the other obfuscated approaches, models achieve the lowest performance in the Lingo-Bw domain. 5-mini and Deepseek-R1 drop from near-perfect accuracy on standard BW to 52.4% and 15.0%, respectively, while the non-reasoning 4.1-mini collapses from 48% to 0.0%. These scores are consistently lower than those in Mystery and Randomized BW. The Lingo-BW setting chosen for comparison is the Super-hard setting as this is the closest setting to the existing Mystery Blocksworld and Randomized Blocksworld benchmarks. In addition to this, our dataset allows for a more granular evaluation of model performance as we provide different levels of obfuscation to isolate where a model’s performance degrades.

G LLM Verification Experiments

Additional experiments were conducted with the two non-reasoning models using two performance-enhancing techniques: zero-shot CoT prompting where we instruct the model to *“think through the plan step by step,”* and LLM verification where we send the generated plan back to the LLM, ask it to verify the correctness of the plan, and produce an updated plan if necessary. We restricted the CoT and verification experiments to the two non-reasoning models as the reasoning model, 5-mini, already leverages CoT in its response process. The results of these experiments are presented in Figure 2.

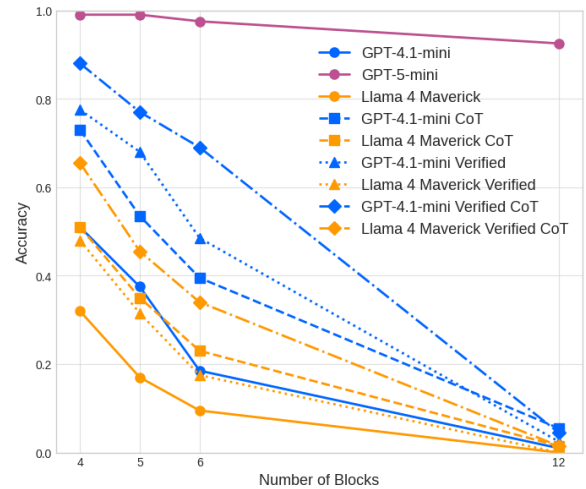


Figure 2: Performance of LLMs on BW problems with 4, 5, 6 and 12 blocks with 200 instances each using the base prompt, chain-of-thought (CoT) or verification.

H Model Parameters and Hyperparameters

We use the default hyperparameters for all models apart from temperature. For non-reasoning models, we use a temperature of 0 to ensure maximum reproducibility. For reasoning models we use a temperature of 0.6 or 1 for coherent outputs that avoid endless repetition based on experimentation and recommendations from the model suppliers. Table 13 displays the models we used, the number of parameters for each model and the value of temperature we selected.

GPT experiments were run using the OpenAI API, costing approximately \$50 USD. Llama and deepseek experiments were run using the OpenRouter API, costing approximately \$300 USD.

I Experiment 2 Results

In Experiment 2 we systematically obfuscate words in the BW domain. The results presented in §5.3 of the paper are a selection of the full set of models we test that represent models of a similar size and family. This includes Llama-3.3-70B, GPT-4.1-mini, Qwen2.5-32B, Deepseek-R1-Distill-Llama, GPT-5-mini and Qwen3-32B. We also evaluate additional models on a subset of the data. These models are Llama 4 Maverick, GPT-4.1-mini, o4-mini, GPT-5-mini and Deepseek-R1. The subset of the data for the easy, medium and hard settings is 40 examples for each combination of obfuscation and 80 examples for the Super-hard setting. We do three runs for each model on this subset. A summary of the results is presented in Table 14. Tables 15, 16 and 17 show the results from each of the three runs for the additional models.

In §5.3 of the paper we also present the results as the mean of three runs of the experiments with the standard deviation. Tables 18, 19 and 20 show the results from each of the three runs where each setting has 200 instances.

J Few-shot Experiment

We report our main findings in a zero-shot setting, but we have also experimented with one-shot prompts specifically for non-reasoning models in the easy setting of Experiment 2 to observe any potential improvements in performance. The results have been shared in Table 21, showing a boost in performance for these non-reasoning models across all easy settings. More experiments in a one-shot setting may reveal interesting patterns, but we leave this to future work.

	Setting	Accuracy (%)	
		Llama 4 $\diamond$	4.1-mini $\diamond$
Easy	<u>A</u> ctions	20.0 $\uparrow 8.3$	15.0 $\uparrow 3.2$
	<u>N</u> ames	65.0 $\uparrow 37.5$	42.5 $\uparrow 11.8$
	<u>O</u> bjects	55.0 $\uparrow 20.8$	57.5 $\uparrow 22.3$
	<u>P</u> redicates	30.0 $\uparrow 10.0$	20.0 $\uparrow 14.3$

Table 21: Results for non-reasoning models with one-shot prompting for Experiment 2 in the easy settings with A: Actions, N: Names, O: Objects, P: Predicates are swapped out for nonce words in the prompt. The arrows and following number show the percentage difference in accuracy compared to the zero-shot experiments in the same setting. The zero-shot results for 4.1-mini are reported in 5.3 Table 3 and the zero-shot results for Llama 4 are from Table 14 in Appendix I.

J.1 Overthinking Effect

In Figure 3, we observe an interesting pattern: as the difficulty of linguistic obfuscation increases, not only does overall planning accuracy decrease, consistent with the main results of this paper, but the time the model spends producing these incorrect solutions increases. We refer to this phenomenon as the *overthinking effect*. In these cases, the model appears to spend substantially more time generating responses when it does not ultimately arrive at the correct solution.

This effect becomes clearer when comparing different difficulty settings. For easier cases, longer response times tend to correlate with higher accuracy. However, in the Super-hard setting the pattern reverses: longer response times are associated with lower accuracy. In other words, the model spends the most time on problems it ultimately answers incorrectly.

From a psycholinguistic perspective, this behaviour is particularly interesting. One might expect that spending additional time reasoning about a problem, especially in a planning task, would

Model	Parameters	T	Source
llama-4-maverick	400B (total) 17B (active)	0	https://openrouter.ai/meta-llama/llama-4-maverick
gpt-4.1-mini-2025-04-14	Unreported	0	https://developers.openai.com/api/docs/models/gpt-4.1-mini
o4-mini-2025-04-16	Unreported	1	https://developers.openai.com/api/docs/models/o4-mini
gpt-5-mini-2025-08-07	Unreported	1	https://developers.openai.com/api/docs/models/gpt-5-mini
deepseek-r1-0528	671B (total) 37B (active)	1	https://openrouter.ai/deepseek/deepseek-r1-0528
llama-3.3-70b-instruct	70B	0	https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct
Qwen2.5-32B-Instruct	32B	0	https://huggingface.co/Qwen/Qwen2.5-32B-Instruct
deepseek-r1-distill-llama-70b	70B	0.6	https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-70B
Qwen3-32b	32B	0.6	https://huggingface.co/Qwen/Qwen3-32B

Table 13: Parameters and hyperparameters for models used in the research where T is the hyperparameter Temperature, total refers to the “total” parameter size of the model and “active” refers to the number of parameters per inference pass.

	Setting	% Mean Accuracy _(SD)				
		Llama 4 ◊	4.1-mini ◊	o4-mini ▲	5-mini ▲	Deepseek-R1 ▲
Easy	Actions	11.7 (1.4)	10.8 (1.4)	99.2 (1.4)	90.0 (4.3)	100 (0.0)
	Names	27.5 (6.6)	29.2 (9.5)	59.2 (10.1)	82.5 (10.9)	98.3 (1.4)
	Objects	34.2 (3.8)	32.5 (6.6)	62.5 (4.3)	90.0 (2.5)	97.5 (2.5)
	Predicates	20.0 (8.7)	6.7 (7.2)	82.5 (4.3)	94.2 (2.9)	96.7 (1.4)
Medium	A+N	11.7 (5.2)	7.5 (10.9)	98.3 (1.4)	98.3 (2.9)	99.2 (1.4)
	A+O	2.5 (0.0)	20.8 (6.3)	93.3 (2.9)	94.2 (1.4)	91.7 (3.8)
	A+P	1.7 (1.4)	0.8 (1.4)	72.5 (17.3)	80.8 (3.8)	75.8 (12.3)
	O+N	27.5 (10.9)	29.2 (11.8)	70.0 (6.6)	88.3 (6.3)	98.3 (2.9)
	P+N	8.3 (2.9)	5.8 (2.9)	73.3 (6.3)	95.8 (2.9)	99.2 (1.4)
	P+O	8.3 (8.0)	15.0 (2.5)	80.0 (2.5)	97.5 (2.5)	95.8 (2.9)
Hard	A+O+N	2.5 (0.0)	9.2 (3.8)	92.5 (4.3)	94.2 (1.4)	93.3 (5.2)
	P+A+N	1.7 (1.4)	1.7 (2.9)	85.0 (0.0)	94.2 (5.8)	91.7 (1.4)
	P+A+O	2.5 (0.0)	0.0 (0.0)	70.0 (5.0)	67.5 (10.9)	60.0 (15.2)
	P+O+N	3.3 (2.9)	3.3 (2.9)	72.5 (2.5)	96.7 (3.8)	92.5 (6.6)
Super-hard ANOP		2.1 (0.7)	1.3 (2.2)	37.9 (3.2)	52.1 (5.0)	15.0 (2.5)

Table 14: Results across additional models for Experiment 2 in which we systematically alter which classes of words (A: Actions, N: Names, O: Objects, P: Predicates) are swapped out for nonce words in the prompt. Non-reasoning models are indicated with ◊ and reasoning models with ▲. This table reports the mean accuracy across three runs of the experiment with standard deviation (SD) in brackets.

improve accuracy. This intuition holds for humans and appears to hold for the easier prompts in our dataset. However, this trend would normally be expected to persist or even strengthen as task

difficulty increases. Indeed, when solving these problems manually, it typically takes substantially longer to work through a Super-hard prompt (such as the example shown in Table 11) than an easy

	Setting	Accuracy (%)				
		Llama 4 ◊	4.1-mini ◊	o4-mini ▲	5-mini ▲	Deepseek-R1 ▲
Easy	Actions	10.0	12.5	97.5	85.0	100
	Names	27.5	40.0	57.5	95.0	97.5
	Objects	32.5	25.0	60.0	92.5	100
	Predicates	5.0	15.0	77.5	97.5	95.0
Medium	A+N	2.5	20.0	97.5	95.0	97.5
	A+O	2.5	27.5	90.0	92.5	87.5
	A+P	2.5	2.5	52.5	77.5	90.0
	O+N	17.5	42.5	62.5	87.5	95.0
	P+N	10.0	7.5	67.5	97.5	100
	P+O	5.0	15.0	77.5	97.5	97.5
Hard	A+O+N	0.0	12.5	90.0	92.5	97.5
	P+A+N	0.0	5.0	85.0	87.5	92.5
	P+A+O	2.5	0.0	65.0	60.0	77.5
	P+O+N	2.5	5.0	72.5	100	97.5
Super-hard ANOP		3.8	3.8	35.0	57.5	12.5

Table 15: Results for the first run of the experiment across additional models for Experiment 2 in which we systematically alter which classes of words (A: Actions, N: Names, O: Objects, P: Predicates) are swapped out for nonce words in the prompt. Non-reasoning models are indicated with ◊ and reasoning models with ▲.

	Setting	Accuracy (%)				
		Llama 4 ◊	4.1-mini ◊	o4-mini ▲	5-mini ▲	Deepseek-R1 ▲
Easy	Actions	12.5	10.0	100	92.5	100
	Names	32.5	22.5	70.0	77.5	97.5
	Objects	30.0	35.0	60.0	87.5	97.5
	Predicates	25.0	2.5	85.0	92.5	97.5
Medium	A+N	17.5	0.0	97.5	100	100
	A+O	2.5	15.0	95.0	95.0	95.0
	A+P	2.5	0.0	82.5	80.0	67.5
	O+N	22.5	20.0	72.5	82.5	100
	P+N	10.0	2.5	72.5	97.5	97.5
	P+O	2.5	17.5	80.0	100	92.5
Hard	A+O+N	2.5	5.0	97.5	95.0	95.0
	P+A+N	2.5	0.0	85.0	97.5	90.0
	P+A+O	2.5	0.0	70.0	80.0	50.0
	P+O+N	5.0	5.0	70.0	92.5	95.0
Super-hard ANOP		2.5	0.0	41.3	47.5	17.5

Table 16: Results for the second run of the experiment across additional models for Experiment 2 in which we systematically alter which classes of words (A: Actions, N: Names, O: Objects, P: Predicates) are swapped out for nonce words in the prompt. Non-reasoning models are indicated with ◊ and reasoning models with ▲.

one, especially if there is any hope of reaching the correct solution.

The model seemingly exhibiting the opposite pattern may offer insight into its internal reasoning processes. While further investigation is needed, one possible explanation is that the model can more reliably apply learned planning strategies from pos-

sible fine-tuning on this task in simpler settings but struggles to adapt those same strategies when the linguistic surface form becomes highly obfuscated. In such cases, additional reasoning time may reflect unsuccessful attempts to reconcile conflicting interpretations of the problem. A deeper investigation into this behaviour is beyond the scope of the

	Setting	Accuracy (%)				
		Llama 4 ◊	4.1-mini ◊	o4-mini ▲	5-mini ▲	Deepseek-R1 ▲
Easy	Actions	12.5	10.0	100.0	92.5	100.0
	Names	30.0	25.0	50.0	75.0	100.0
	Objects	35.0	37.5	67.5	90.0	95.0
	Predicates	25.0	2.5	85.0	92.5	97.5
Medium	A+N	7.5	2.5	100.0	100.0	100.0
	A+O	2.5	20.0	95.0	95.0	92.5
	A+P	2.5	0.0	82.5	85.0	70.0
	O+N	40.0	25.0	75.0	95.0	100.0
	P+N	5.0	7.5	80.0	92.5	100.0
	P+O	17.5	12.5	82.5	95.0	97.5
Hard	A+O+N	2.5	10.0	90.0	95.0	87.5
	P+A+N	2.5	0.0	85.0	97.5	92.5
	P+A+O	2.5	0.0	75.0	62.5	52.5
	P+O+N	5.0	0.0	75.0	97.5	85.0
Super-hard ANOP		2.5	0.0	37.5	51.3	15.0

Table 17: Results for the third run of the experiment across additional models for Experiment 2 in which we systematically alter which classes of words (A: Actions, N: Names, O: Objects, P: Predicates) are swapped out for nonce words in the prompt. Includes Super-hard condition from follow-up experiment. Non-reasoning models are indicated with ◊ and reasoning models with ▲.

	Setting	% Accuracy					
		Llama ◊	4.1-mini ◊	Qwen ◊	Llama ▲	5-mini ▲	Qwen ▲
Easy	Actions	15.0	12.5	5.0	44.0	92.5	78.5
	Names	63.5	30.0	24.0	65.5	88.0	82.0
	Objects	57.5	34.5	28.0	47.5	94.0	60.0
	Predicates	19.5	7.5	16.5	35.5	96.5	56.0
Medium	A+N	3.5	10.5	6.0	45.5	97.5	80.0
	A+O	23.5	16.5	5.5	35.5	94.5	66.0
	A+P	0.5	0.5	4.5	14.0	77.5	13.0
	O+N	64.0	32.0	21.5	53.5	88.5	57.5
	P+N	23.5	6.5	16.5	36.5	96.5	63.0
	P+O	17.0	12.0	17.5	40.5	95.5	77.0
Hard	A+O+N	21.5	8.0	3.0	39.0	91.0	72.0
	P+A+N	1.0	1.0	1.0	8.0	92.0	6.0
	P+A+O	0.0	0.5	5.0	18.0	55.5	13.5
	P+O+N	9.0	9.0	6.5	19.0	95.0	53.5
Super-hard ANOP		0.0	1.5	3.0	4.5	55.5	8.0

Table 18: Results for the first run of the main models for Experiment 2. Each setting has 200 instances. The non-reasoning models are Llama-3.3-70B, GPT-4.1-mini and Qwen2.5-32B indicated with ◊ and the reasoning models are Deepseek-R1-Distill-Llama, GPT-5-mini and Qwen3-32B indicated by ▲.

current work and is left for future research.

J.2 Self-Consistency Experiment

We conduct one follow-up experiment in which we prompt 5-mini and deepseek-r1 to answer the same Super-hard 12 block problem 80-200 times

in order to determine if there is any determinism to the reasoning and problem-solving strategies of the model. If there were some consistency to its approach, we might expect that the model would largely either get the problem right or largely get it wrong, having taken the same reasoning steps (even

	Setting	% Accuracy					
		Llama ◊	4.1-mini ◊	Qwen ◊	Llama ▲	5-mini ▲	Qwen ▲
Easy	Actions	13.0	12.5	5.5	40.0	93.5	82.0
	Names	58.0	32.5	23.5	66.5	82.5	75.0
	Objects	62.5	36.0	27.5	47.0	93.5	62.5
	Predicates	18.0	4.5	16.0	37.0	97.0	64.5
Medium	A+N	4.0	5.5	6.0	38.5	97.0	82.5
	A+O	21.5	14.5	5.0	31.5	93.0	64.5
	A+P	0.0	0.0	4.0	13.0	82.5	11.5
	O+N	64.5	27.5	22.0	60.0	89.5	62.5
	P+N	24.5	6.0	15.0	33.5	96.0	63.5
	P+O	19.0	13.5	18.0	39.0	97.5	74.5
Hard	A+O+N	22.5	5.5	3.0	40.0	92.0	71.5
	P+A+N	0.0	0.0	1.0	9.0	96.5	9.5
	P+A+O	0.0	0.5	4.5	11.0	61.5	12.5
	P+O+N	7.0	8.0	7.0	19.5	90.5	56.5
Super-hard ANOP		0.0	0.0	3.0	4.5	51.0	6.0

Table 19: Results for the second run of the main models for Experiment 2. Each setting has 200 instances. The non-reasoning models are Llama-3.3-70B, GPT-4.1-mini and Qwen2.5-32B indicated with ◊ and the reasoning models are Deepseek-R1-Distill-Llama, GPT-5-mini and Qwen3-32B indicated by ▲.

	Setting	% Accuracy					
		Llama ◊	4.1-mini ◊	Qwen ◊	Llama ▲	5-mini ▲	Qwen ▲
Easy	Actions	15.0	10.5	5.5	42.0	95.0	79.0
	Names	61.5	29.5	20.5	70.5	81.5	78.0
	Objects	57.5	35.0	27.0	47.5	94.0	55.5
	Predicates	21.5	5.0	14.5	35.0	96.5	64.5
Medium	A+N	4.5	4.5	5.5	49.0	98.5	77.0
	A+O	21.0	14.0	5.0	37.0	93.0	68.5
	A+P	0.0	0.0	4.0	11.0	84.5	9.5
	O+N	65.5	33.0	20.5	50.5	91.0	56.5
	P+N	22.0	7.5	16.5	41.0	95.5	56.5
	P+O	18.0	16.0	16.0	38.0	94.5	75.5
Hard	A+O+N	19.5	8.0	3.0	40.0	93.5	70.0
	P+A+N	0.0	0.5	1.0	8.5	96.0	9.5
	P+A+O	0.0	0.5	4.0	11.0	55.5	13.0
	P+O+N	8.5	10.5	6.5	17.5	94.5	55.5
Super-hard ANOP		0.0	0.0	3.0	4.5	52.0	9.0

Table 20: Results for the third run of the main models for Experiment 2. Each setting has 200 instances. The non-reasoning models are Llama-3.3-70B, GPT-4.1-mini and Qwen2.5-32B indicated with ◊ and the reasoning models are Deepseek-R1-Distill-Llama, GPT-5-mini and Qwen3-32B indicated by ▲.

if incorrect) to arrive at the same solution. We find that 5-mini provides a distinct, incorrect answer in 47.0% of the 200 instances of the same problem. We note that this is somewhat similar to the error rate of the model on 200 distinct problems, which is 50.5%. We find that deepseek-r1 provides a distinct, incorrect answer in 86.2% of the 80 in-

stances of the same problem. This is somewhat better accuracy (13.8%) on a single problem than the 200 distinct problems, where deepseek-r1 obtains only 1.0% accuracy. This suggests a lack of determinism in how the models approach and solve these problems, suggesting that performance may depend on unstable or problem-specific features.

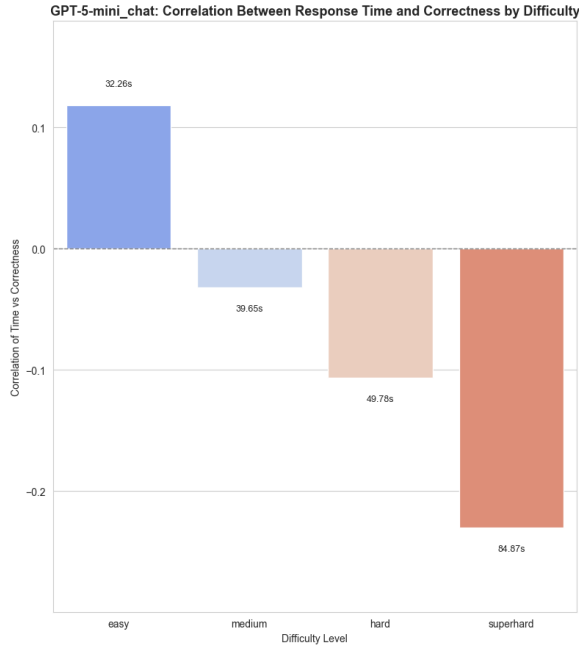


Figure 3: GPT-5mini correlation between the time the model took to respond and the percent of correct answers across each setting type.

K Quantitative Error Analysis

To provide a clear categorisation of error types under Lingo-BW obfuscation, we conducted a two-part analysis starting with a qualitative step to build the error categorisation taxonomy, followed by the automation of categorisation based on the qualitatively derived taxonomy.

K.1 Qualitative Taxonomy (Manual Annotation)

We first conducted a hand-annotated analysis of 65 instances, focusing on the 23 incorrect cases to develop a formal taxonomy of five distinct planning errors. Table 22 shows the definitions of the 5 error types as well as the results from the manual categorisation. Note that although we did not require models to provide plans in well-formed PDDL, the cases that we term “malformed” here are genuinely unrecoverable plans, not extraction losses.

K.2 Automated Error Parsing

To scale the error identification and classification, we implemented a rule-based parsing script to evaluate 871 incorrect plans from four models. The script identifies structural violations including missing arguments for malformed errors, or repetitive loops for redundant actions and semantic errors such as skipping mandatory hand-emptying steps

for fused hand-management. Table 23 shows the results.

This quantitative error categorisation aligns with the qualitative observations from section 7 while exposing how rigid parsing constraints dictate architecture-specific failure modes. For the selected non-reasoning model, the manual observation that it fails to model the initial world state and collapses into action loops is reflected in this categorisation, with 67.3% of its failures falling into the “Redundant/self-cancelling” category. Conversely, reasoning models generally represent the initial state correctly but struggle to enforce operational constraints. The qualitative finding that these models frequently skip transitional steps, such as conceptually merging continuous actions and failing to insert a “put-down” after an “unstack”, is quantified by the 152 recorded instances (22.6%) of “Fused hand-management” errors, representing the second-highest category of errors for reasoning models. Furthermore, this categorisation reveals a 57.4% “Malformed/local-only” error rate among reasoning models, indicating a widespread failure to adhere to strict PDDL syntactic constraints.

L Qualitative Analysis of Human Planning in Lingo-BW

Although we do not expect models to reason in the same ways as people (and do not believe that success relies upon this), to better understand the reasoning and problem-solving strategies of this domain, we presented six 4-block problems to one author of this paper with classical-planning expertise. The problems spanned the Easy, Medium, and Super-hard settings. Specifically, we present four Easy problems with each word class swapped out one at a time, one Medium problem with Actions and Predicates swapped out, and one Super-hard problem with all word classes swapped out. We did not provide any instructions or constraints on how to solve the task outside of the same prompt and problem description (which includes varying levels of swapped vocabulary, depending upon the setting) presented to the models.

The human planner provided valid steps leading to a plan for 100% of the problems over the course of a two-hour sitting. However, the planner had to leverage extensive documentation and visualisation tools, including drawing the labelled objects, drawing out the initial state, moving objects for each potential step until arriving at the goal state. Given

Category	Description	Example	R	NR	Total
Fused hand-management	Treats <i>unstack</i> as if it includes <i>put-down</i> (or <i>stack</i> as if it includes <i>pick-up</i>), dropping the hand-emptying step.	<code>unstack b1 b12</code> → <code>unstack b12 b4</code> (no put-down between)	6	3	9
Action confusion	Uses <i>pick up</i> where <i>unstack</i> is required (or the <i>stack/put-down</i> mirror); correct count, wrong verb.	<code>pick up b2</code> when b2 is on top of b3	2	4	6
World-state disconnect	Acts on preconditions the current state does not satisfy (e.g., moving a covered block).	<code>stack b3 b2</code> when b3 is under b2	0	3	3
Redundant/self-cancelling	Action pairs that undo each other or produce no forward progress.	<code>pick up b3</code> → <code>put down b3</code>	2	0	2
Malformed/local-only	Missing arguments, or pairwise relations look right but the plan cannot execute.	<code>unchrit b7</code> (missing second block)	3	0	3
Total			13	10	23

Table 22: Qualitative categories of true planning errors for Reasoning (R) and Non-reasoning (NR) models.

Category	Deepseek-R1 (R)	5-mini (R)	o4-mini (R)	Llama 4 Maverick (NR)	Total
Fused hand-management	100	37	15	35	187
Action confusion	14	13	3	12	42
World-state disconnect	25	10	2	6	43
Redundant/self-cancelling	52	10	2	134	198
Malformed/local-only	237	77	72	12	398
Unknown error	2	0	1	0	3
Total	430	147	95	199	871

Table 23: Automated categorisation of error types across 871 failures for Reasoning (R) and Non-reasoning (NR) models.

expertise with classical planning, the human planner recognised this as the BW domain and therefore wrote out a vocabulary mapping for each problem setting, explicitly grounding the novel vocabulary in the known problem space. Thus, we see that visualisation and translation from a known problem space to the novel problem space are strategies for effective human planners.

M Tokenization Investigation

We have conducted an investigation into how models tokenize the different blocksworld domains. We found that nonsense words are sometimes segmented in ways that split words into morphologically meaningful units, but this is not always the case depending on the specific nonsense word and tokenizer. Cases that we observe where the morphological affixes and prefixes are preserved may indeed provide additional structural cues, especially when compared to the random alphanumeric strings that appear in randomized Blocksworld (e.g. "J4gv801gnu2it0yj").