Do Language Models Exhibit the Same Cognitive Biases in Problem Solving as Human Learners?

Andreas Opedal [*] [1] [2] Alessandro Stolfo [*] [1] Haruki Shirakami [1] Ying Jiao [3]
Ryan Cotterell [1] Bernhard Schölkopf [1] [2] Abulhair Saparov [4] Mrinmaya Sachan [1]

Abstract

There is increasing interest in employing large language models (LLMs) as cognitive models. For such purposes, it is central to understand which properties of human cognition are well-modeled by LLMs, and which are not. In this work, we study the biases of LLMs in relation to those known in children when solving arithmetic word problems. Surveying the learning science literature, we posit that the problem-solving process can be split into three distinct steps: text comprehension, solution planning and solution execution. We construct tests for each one in order to understand whether current LLMs display the same cognitive biases as children in these steps. We generate a novel set of word problems for each of these tests, using a neuro-symbolic approach that enables fine-grained control over the problem features. We find evidence that LLMs, with and without instruction-tuning, exhibit human-like biases in both the text-comprehension and the solution-planning steps of the solving process, but not in the final step, in which the arithmetic expressions are executed to obtain the answer.

https://github.com/eth-lre/solving-biases

1. Introduction

There is active discussion around large pretrained language models (LLMs) as plausible cognitive models (Mahowald et al., 2023), e.g., in terms of language acquisition (Warstadt and Bowman, 2022), decision making (Aher et al., 2023) and political orientation (Argyle et al., 2023). In the context of education, cognitive modeling enables the study of human learning without the high cost of data collection from human subjects, which can lead to a better understanding of human learning and, therefore, improved learning outcomes (VanLehn et al., 1994). Several recent articles have already employed LLMs as models of students (Macina et al., 2023; Nguyen et al., 2023). However, for such modeling to be meaningful, it is imperative that the student model is consistent with actual student behavior. Yet, that is not always the case: Many existing student models fail to validate faithfulness to realistic classroom scenarios (Käser and Alexandron, 2023). Importantly, an LLM that models the problem-solving process of children should also make similar mistakes as children, i.e., it should mimic the cognitive biases that are salient in children during problem-solving. Given that LLMs may be trained predominantly on data generated by *adults*, it is not obvious that they would exhibit *child-like* behavior.

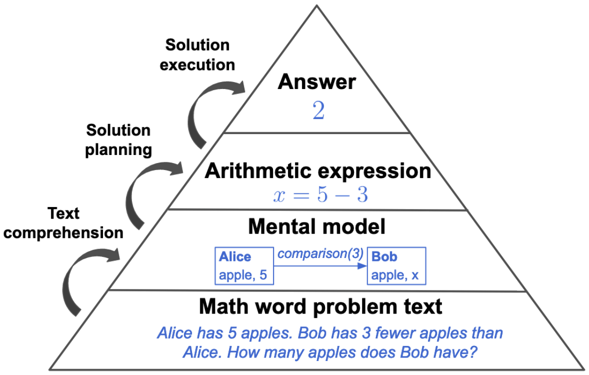


Figure 1. A three-step model of the cognitive process involved in solving math word problems. We study whether LLMs exhibit similar biases as human children along each step of this process.

In this paper, we study whether LLMs are subject to similar biases as children when solving arithmetic math word problems.[1] These problems are interesting because they are conceptually simple, and yet, require several distinct skills to solve (Stern, 1993). A learner needs to understand the situation described, relate it to arithmetic equations, and

[*]Equal contribution [1]ETH Zürich [2]Max Planck Institute for Intelligent Systems [3]KU Leuven [4]New York University. Correspondence to: Andreas Opedal <andreas.opedal@inf.ethz.ch>, Alessandro Stolfo <alessandro.stolfo@inf.ethz.ch>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1] Related studies have compared LLM biases to human ones on the task of syllogistic reasoning (Ando et al., 2023; Dasgupta et al., 2023; Eisape et al., 2024). These are discussed in §7.

perform the required computations, as Fig. 1 illustrates. By Piaget's (1936) view on cognitive development, a problem might be difficult for a child due to insufficient development in any one of these skills. Much is known about what makes a word problem difficult for humans; we ask whether the same relative difficulties apply to LLMs.

To answer this question, we construct tests that are grounded in the extensive literature on word problem solving by children,[2] and perform them on a suite of currently well-known LLMs. Specifically, each test varies a problem feature for which an effect on child performance has been established in the literature, e.g., the manner in which a particular mathematical operation is expressed in text, while controlling for other features. We create new English problems specifically for these tests, by developing a generation pipeline based on a semantic formalism over math word problems (Opedal et al., 2023). Our generation pipeline admits a family of standard arithmetic word problems, while controlling not only for numerical features, but structural (e.g., entity relationships) and linguistic ones (e.g., sentence structure) as well.

We test three cognitive biases, each one associated with a separate step of the solving process (which are illustrated by arrows in Fig. 1). The first test targets **consistency bias**: A problem text is easier to comprehend if the relational keyword verbally suggests the operation that is required to solve the problem (Lewis and Mayer, 1987). The second test targets what we call **transfer vs comparison bias**, that problems with a static comparison relationship are more difficult for children than problems with a dynamic change of state, even when they involve the same arithmetic expressions (Riley et al., 1983). The third test targets the **carry effect**, i.e., that arithmetic computations are more difficult if they entail moving some digit to a column of more significant digits (Hitch, 1978).

We find that LLMs indeed exhibit some biases that mirror those observed in children. Our experiments with both base and instruction-tuned models—specifically, LLaMA2 (Touvron et al., 2023), Mistral (Jiang et al., 2023), Mixtral (Jiang et al., 2024), GPT-3.5 Turbo and GPT-4 Turbo (OpenAI, 2024)—reveal that almost all models show significant effects of consistency bias and transfer vs comparison bias, like child learners. Most of these effects are further strengthened by using chain-of-thought prompting (Wei et al., 2022). However, we do *not* observe a carry effect bias in the solution execution step. These results contribute to our understanding of the capabilities and limitations of LLMs as cognitive models, particularly in the context of cognitive development research and educational applications.

2. Cognitive Modeling of the Solving Process

This section discusses the cognitive process that is involved in solving math word problems. We first introduce our conceptualization that is illustrated in Fig. 1, which we then motivate by summarizing relevant literature.

Our conceptualization. We are interested in identifying *when* LLMs are likely to exhibit human-like biases, and therefore, require a holistic analysis of the human problem-solving process. Our conceptualization, illustrated in Fig. 1, includes four representational levels of the math word problem, along with the skills associated with transitioning from one level to the next. We assume that a child goes through the following procedure when posed with a math word problem: First, they form a **mental model** of the mathematical relationships expressed in that problem (text comprehension). Next, they distill that mental model into a sequence of **arithmetic expressions** representing the step-by-step reasoning process to find the solution (solution planning) and, finally, calculate the **answer** from those expressions (solution execution). The representational levels will be formalized in §4.1.

Background. Children tend to experience greater difficulty when posed with arithmetic math word problems compared to the same problems formulated solely as arithmetic expressions; see, e.g., Jaffe and Bolger (2023). This suggests that arithmetic computation skill alone is not sufficient to successfully solve math word problems. In order to distinguish the different skills that are involved, past work has represented math word problems along similar levels as we do above (Nesher and Teubal, 1975): (i) problem text, (ii) underlying mathematical relationships, and (iii) arithmetic expressions. Solving a problem, then, involves transitioning from level (i) to a final answer, possibly through levels (ii) and (iii), with each transition requiring a separate skill.[3]

There is much research on which factors best explain problem difficulty. Riley et al.'s (1983) model of the problem-solving process emphasized the degree of complexity at level (ii) as the leading factor underlying performance, motivated by empirical evidence that some arithmetic concepts are harder for children to understand than others. However, their model does not account for *how* the mathematical relationships are derived from the problem text (Cummins et al., 1988). This part is significant as well, as several studies have found that altering the linguistic form of a problem *without* changing the underlying mathematical relationships can

[2] This comment refers to studies performed on children, but we note that some of the biases considered seem to be present in adults as well (albeit with weaker effects). See Jaffe and Bolger (2023) for a recent review on word problem performance independent of age.

[3] Not everyone uses all three levels in their solving process. Hegarty et al. (1995) find evidence that unsuccessful human problem solvers often opt for short-cut strategies that rely on surface-level features of the problem text (thus, by our conceptualization, moving directly from text to arithmetic expressions), whereas successful solvers are more likely to make use of mental models.

have drastic effects on performance (Hudson, 1983; Lewis and Mayer, 1987; *inter alia*) in both children and adults. This part of the process is encompassed by the models of Briars and Larkin (1984) and Kintsch and Greeno (1985). None of these models give an explicit account of the complexity of the arithmetic expression, however, which also has significant influence on performance (Daroczy et al., 2015).

3. Human Biases in Word Problems

In this section we discuss the particular factors that influence performance of human children (i.e., cognitive biases) which we study in LLMs (§5). Each bias is reflected by a variation in a specific level of Fig. 1. We study one bias for each of the three levels that precedes the answer.

Problem text: Consistency bias. Given the premise "Alice has 5 apples" and a question querying the (smaller) number of apples of another agent "Bob", an additive comparison statement between the two agents could take either of the following forms:

(1) Bob has 3 fewer apples than Alice.
(2) Alice has 3 more apples than Bob.

Here, (1) represents a *consistent* statement because the relational keyword ("fewer") suggests the operation that is indeed needed to compute Bob's quantity (subtraction). Conversely, (2) is an *inconsistent* statement because the relational keyword ("more") suggests a different operation ("addition"). Note that these two statements express the same comparison relationship, so the difference lies only in the problem text. Problems with an inconsistent statement are more difficult for children to solve than consistent ones (Nesher and Teubal, 1975; Stern, 1993). This has been hypothesized to be the case due to inconsistent statements requiring an additional, error-prone, deduction step: to rearrange the relational statement to be in the preferred consistent form (Lewis and Mayer, 1987).

Mental model: Transfer vs comparison bias. Irrespective of which relational keyword is used, comparison-type problems tend to be more difficult for children than other types of arithmetic concepts (Riley et al., 1983). In particular, grade school children display a significant difference in performance between comparison problems and transfer problems. Consider the same premise as above but with a slightly different continuation:

Alice has 5 apples. Alice gave 3 apples to Bob.
How many apples does Alice have?

This is a transfer problem (often called a *change* elsewhere; Nesher et al., 1982), concerning a change of state of some quantity. It has the same arithmetic expression as the comparison problems above (although with another mental model), but is easier for young children to solve. In analyzing their solution strategies, it has been found that comparison problems require a number-matching type strategy that appears to be more sophisticated than the counting-type strategies that are often sufficient for solving transfer problems (Riley et al., 1983; Carpenter and Moser, 1984).

Arithmetic expressions: The carry effect. Beyond the text and mental model, it is natural that the particular numbers used in a problem will have an effect on a child's performance (Daroczy et al., 2015). Consider the same problem(s) as above, but with a different number given in the premise:

Alice has 22 apples. Bob has 3 fewer apples than
Alice. How many apples does Bob have?

The problem now has the arithmetic expression $22 - 3$, which involves an arithmetic *carry*, which is also called a borrowing in the case of subtraction. A carry is a digit that is transferred from one column of digits to another as the result of an arithmetic computation. In this subtraction computation, a unit carry of 1 is transferred from the column of units to the column of tens in order to make the answer 19. The previous expression $(5 - 3)$ did not have a carry, which is easier for children (Hitch, 1978; Ashcraft et al., 1992). The presence of a carry introduces an additional subroutine from the standard sequence of operations, which places a higher load on working memory (Fürst and Hitch, 2000).[4]

4. Problem Generation Method

Our experiments on LLMs with respect to the biases just discussed (§5) rely on data generated for the sole purpose of our study. By not using problems from public datasets, previous work or other existing sources it becomes unlikely that our data has been used for training of the models, an increasingly common issue (Dodge et al., 2021; Elazar et al., 2023).

This section gives the details of our data generation pipeline, which provides control over features across all levels of Fig. 1. In §4.1, we operationalize Fig. 1, giving definitions related to the mental model level and other aspects of the process. Using these definitions, we then explain our generation pipeline in §4.2.

4.1. Operationalizing Fig. 1

The mental model level is operationalized using the formalism from Opedal et al. (2023), called MATHWORLD.

[4] The particular example numbers given here are small enough for children to likely be relying on retrieval from some long-term memory store instead of algorithmic computation (Koshmider and Ashcraft, 1991), which could erase the effect of carry. We account for this in our experiments (§5.4) by using larger number ranges.

MathWorld annotates each math word problem with a logical representation, which captures the mathematical relationships between the entities described in text. Each **entity** has a non-negative integer **quantity**. Optionally, there may be additional information associated with an entity—namely, an **agent** who possesses the entity, and a **unit** and an **attribute** which enrich the description of it. The five bold items in the two preceding sentences are referred to as **properties**. The arithmetic relationships are classified according to **concepts**; we use transfer, comparison (additive and multiplicative), and rate in this work. We gave examples of the transfer and comparison concepts in §3.

When discussing data generation (§4.2) and the experiments the data is used for (§5), we will rely on the following definitions: A **predicate** is a symbol that represents either an arithmetic concept, or possession of an entity (in that case the predicate is "`container`"). Each predicate takes a set of properties as arguments.[5] A predicate instantiated with properties is called a **logical form**, and represents the semantics of a given sentence in a problem. See Table 1 for examples of logical forms for all predicates we use. The **mental model** of a problem is a sequence of logical forms (separated by a "$\circ$" symbol) for each sentence in that problem (in the same order), representing its semantics. In Fig. 1, we gave a mental model example in graphical format; its equivalent sequential format is `container(Alice, 5, apple)` $\circ$ `comparison(+, Alice, Bob, 3, apple)`. The problem text in Fig. 1 is **faithful** to this mental model, in the sense that the mental model represents the semantics of that text under the MathWorld formalism. We refer to the **structure** of a problem as a mental model in which the property values are replaced by unique placeholders. The structure associated with the previous example is `container([agent1], [n1], [entity1])` $\circ$ `comparison(+, [agent1], [agent2], [n2], [entity1])`.

Finally, we formalize the arithmetic expression level of Fig. 1. Every concept-based predicate corresponds to an equation $x = y \odot z$, with $\odot \in \{+, -, \times, \div\}$ and $x, y, z \in \mathbb{Z}_{\geq 0} \cup V$ where V is a set of variable symbols. We require that $\exists! v \in \{x, y, z\} \colon v \in V$. We refer to the deductive inference step taken to solve that equation as a **reasoning step**, and its output (i.e., the value of the variable) as an **intermediate result**. The arithmetic expression level consists of a sequence of such reasoning steps, which is a proof of the answer of the problem (or solution). Any fact from the mental model is an axiom that can be used in the solution proof. This work focuses exclusively on **linear** problems, in which every reasoning step has *at most* one non-axiom premise.

Logical Form		Example Sentences
Predicate	**Properties**	

container	agent=*Alice*
	quantity=5
	entity=*apple*
	attribute=*red*
	unit=kg
		Alice has 5 kilograms of red apples.
		Alice owns 5 kilograms of red apples.

comparison	type=+
	agentA=*Alice*
	agentB=*Bob*
	quantity=3
	entity=*apple*
		Bob has 3 fewer apples than Alice.
		Alice has 3 more apples than Bob.

transfer	receiver_agent=*Bob*
	sender_agent=*Alice*
	quantity=3
	entity=*apple*
		Alice gave Bob 3 apples.
		Bob got 3 more apples from Alice.

rate	agent=*Alice*
	quantity=4
	entityA=*apple*
	entityB=*basket*
		Each of Alice's baskets holds 4 apples.
		Every basket that Alice has contains 4 apples.

Table 1. Examples of MathWorld logical forms. A logical form consists of a predicate and a set of properties. Note that the *attribute* and *unit* properties are optional, and only the first logical form has non-null values for them ("red" and "kg", respectively). Each logical form is given with two of the viable templated sentences from our generation pipeline (§4.2), the semantics of which is represented by the logical form.

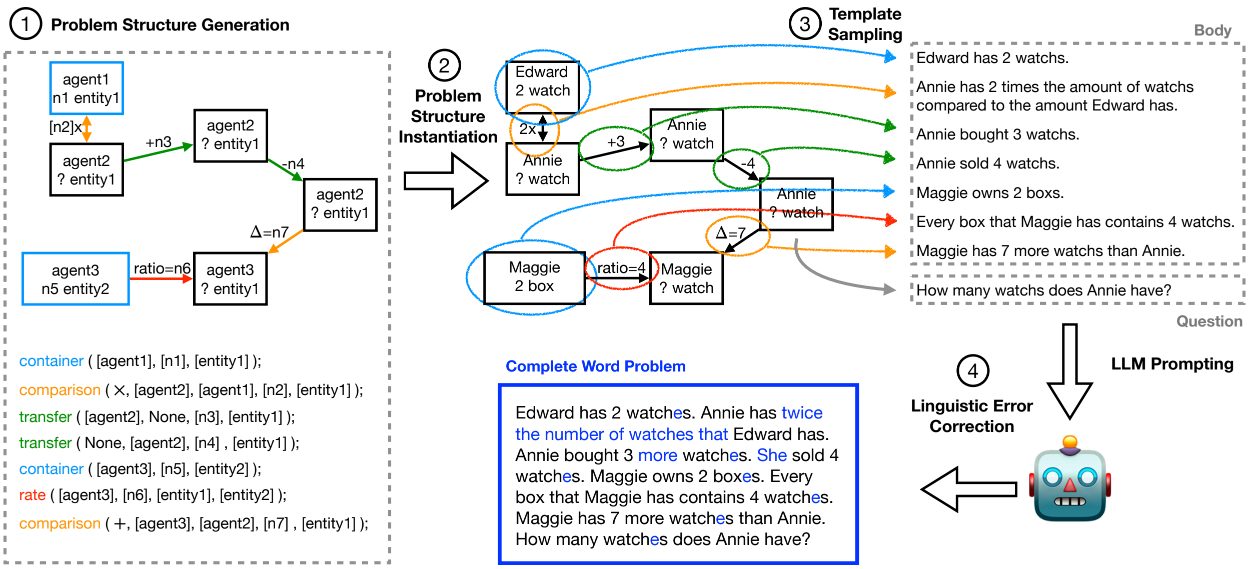
Plausibility of the mental model framework. A mental model theory over some reasoning domain must be able to adequately explain the relative difficulty across different types of reasoning problems (Johnson-Laird, 1983). Our MathWorld-based operationalization emphasizes arithmetic concepts and their relational structure as the key features that explain errors at the mental model level, which is in line with existing theories on word problems (Riley et al., 1983; Briars and Larkin, 1984; Kintsch and Greeno, 1985). Our schemata for the logical forms extend the problem schemata from Riley et al. (1983), specifically, in terms of the breadth of concepts and properties supported.

4.2. Generation Pipeline

We propose a simple problem generation pipeline, described in this section and later applied in §5. It proceeds in four steps: (i) sampling the (linear) problem structure, (ii) obtaining a mental model by instantiating the structure with properties, (iii) transforming the mental model into templated natural language, and, finally, (iv) correcting linguistic errors and awkward phrasings in the templated text using an instruction-tuned LLM. Fig. 2 illustrates the full pipeline. App. A.3 discusses related approaches.

(i) Problem structure generation. Our pipeline supports sequences of predicates under the following regular language, which is sufficient for the hypotheses we are testing:

[5] We enforce all quantities that are associated with predicates to be explicit numbers. Note that this places a constraint on the format of the problems: The number associated with a mathematical relationship is never an intermediate result, but is always given in text.



container ([agent1], [n1], [entity1]);
comparison (×, [agent2], [agent1], [n2], [entity1]);
transfer ([agent2], None, [n3], [entity1]);
transfer (None, [agent2], [n4] , [entity1]);
container ([agent3], [n5], [entity2]);
rate ([agent3], [n6], [entity1], [entity2]);
comparison (+, [agent3], [agent2], [n7] , [entity1]);

Edward has 2 watchs.
Annie has 2 times the amount of watchs compared to the amount Edward has.
Annie bought 3 watchs.
Annie sold 4 watchs.
Maggie owns 2 boxs.
Every box that Maggie has contains 4 watchs.
Maggie has 7 more watchs than Annie.
How many watchs does Annie have?

Edward has 2 watches. Annie has twice the number of watches that Edward has. Annie bought 3 more watches. She sold 4 watches. Maggie owns 2 boxes. Every box that Maggie has contains 4 watches. Maggie has 7 more watches than Annie. How many watches does Annie have?

Figure 2. Overview of our generation pipeline with an example problem. (1) We start by generating the problem structure. The alignment between the graphical and sequential formats of the structure is illustrated by color coding, and the **black** containment boxes in the graph represent intermediate results. (2) The properties of those structures are then instantiated with values, resulting in a mental model. (3) Next, we sample a templated sentence for each of the logical forms in the mental model, and concatenate them in the ordering of the logical forms. (4) Finally, we correct errors and awkward phrasings by prompting an instruction-tuned LLM (the corrections are highlighted in **blue**). This particular example includes all predicate types that we use in §5.

$$\texttt{container} \circ \underbrace{(\texttt{concept} \mid \texttt{container})}_{1-N \text{ times}};$$

where concept = comparison | rate | transfer. Each predicate in the sequence corresponds to an axiom from a distinct sentence in the problem. The particular set of predicate sequences (i.e., class of structures) from which we sample is test dependent (see §5). Such a set could, for instance, be the class of all linear reasoning structures with at most $N{=}5$ steps and only transfer concepts. We generate a problem as follows: First, we sample the number of reasoning steps n uniformly at random from the set $\{1, \ldots, N\}$. Next, we sample the predicates; each choice is sampled uniformly at random. Since this first step of the pipeline generates *structures*, the predicates all have associated unique placeholders in place of properties, e.g., agent2, entity1. We only introduce new entity placeholders in rate logical forms; see Table 1, in which rate is the only predicate that takes two entity properties. We determine uniformly at random whether an entity is paired with an attribute, a unit, or neither. The instantiation of agent placeholders is test specific. Finally, the answer of a problem is always set as the intermediate result corresponding to the last logical form in the ordering, which is unique. See the box on the left-hand side of Fig. 2 for an example structure generated after this step.

(ii) Problem structure instantiation. Next, the problem structure is instantiated with properties, yielding a mental model. We use a handwritten vocabulary for each of the lexical properties (entity, agent, unit and attribute) and sample instantiations of these properties from those vocabularies uniformly at random. The numerical quantities are instantiated by sampling a set of numbers uniformly at random from within a fixed range, which is 2–20 for the experiments in §5.2-5.3 and 100–999 for the experiments in §5.4.[6] Then, we enumerate the logical forms and accordingly compute intermediate results for each reasoning step, making sure that the intermediate quantities fall in the range 0–999. If not, a new set of numbers is sampled and the procedure is repeated from the beginning. This naive procedure is sufficiently fast for our purposes. Empirically, we observed an average runtime of ≈4 ms to generate a numerical instantiation for a problem.

(iii) Template sampling. The mental model is then converted to natural language using templated text. Specifically, for each of the predicates we construct a set of templates that represent natural language adhering to that predicate. For instance, transfer(Annie, None, 3, watch) is converted to “[Annie] *bought* [3] [watch]*s*” in Fig. 2; see Table 1 for additional examples. The templates are handcrafted. We sample one template uniformly at random for each predicate in the mental model. We also create

[6] We omit 0 and 1 from the first number range in order to avoid unnatural phrases like “*Bob has 1 times as many apples as Alice.*” The range for the experiments in §5.4 contains larger numbers for reasons given in Footnote 4.

and sample interrogative templates for the questions, which always query the intermediate result derived from the *last* predicate. Finally, we concatenate the sentences to obtain the full problem text. This step enables control over the linguistic form of the sentences in the problem text, which will be important for our test related to text comprehension in §5.2. Moreover, since the faithfulness of the templated text is guaranteed by manual design, the procedure up to this point ensures that the text is faithful to the mental model.

(iv) Linguistic error correction. However, templated texts occasionally induce spelling mistakes and awkward phrasings. In the example shown in Fig. 2, the entity "watch" is inserted into the template to make "watchs". Inspired by the demonstrated success of zero-shot grammatical error correction (Kwon et al., 2023; Loem et al., 2023), we use an instruction-tuned language model (Ouyang et al., 2022), GPT-3.5 Turbo, to correct such errors.[7] We write a short instructive prompt and have the model generate (with greedy decoding) a corrected problem text conditioned on that prompt together with the particular templated text we want to correct. The prompt instructs the model to be conservative, i.e., to only correct linguistic errors and awkward phrasings. We provide the exact prompt used and additional generation details in App. A.1. This step could, in principle, be generalized to perform less strict forms of paraphrasing. There is then a trade-off between faithfulness and control on the one hand and linguistic variability and naturalness on the other, which can be tuned using different prompts and decoding methods. The present study prioritizes control and faithfulness, but alternative prioritizations could be used in future studies that employ our method.

Evaluating data quality. The generated problem texts must be faithful to the mental models from which they were generated, so we perform manual evaluation of the data to assess that such is the case. We follow a generic procedure and perform it for each of the datasets generated for the experiments in §5.2-5.4. The procedure is iterated until we achieve satisfactory quality. The final error-rate estimates were 0% for all three datasets. Details are given in App. A.2.

5. Experiments

We generate data to perform tests on whether LLMs exhibit child-like biases using the pipeline discussed above. We aim to identify where in the process in Fig. 1 those biases emerge. We therefore split our tests according to the level (and associated skill) they target: problem text and text comprehension (§5.2), mental model and solution planning (§5.3), and arithmetic expressions and solution execution (§5.4). First, we discuss the general experimental setup (§5.1).

5.1. Experimental Setup

We base our experiments on the problem features discussed in §3 that have been found to have an effect on child performance in solving word problems. Specifically, given such a feature X, we want to know whether X has a causal effect on the performance of LLMs. Our generation pipeline enables exact matching of the data: We generate problems in pairs, where the two problems differ only in the value of X. Using this data, we estimate the *conditional average treatment effect* (CATE; Imbens and Rubin, 2015)

$$\mathbb{E}[Y(x) - Y(x') \mid Z], \tag{1}$$

where Y is 1 if the model's prediction is correct and 0 otherwise, x and x' are two distinct values of the treatment variable X, and Z is the subgroup of data generated through our pipeline for a specific test. The two values x and x' are defined such that positive CATEs are consistent with human biases. We refer to Feder et al. (2022) for further reading on causality-based methods for NLP.

For each of the tests described below we select a specific feature X that is localized to one of the levels, to the extent possible. That is, varying X associated with a particular level should have *no* effect on the levels above, and *minimal* effect on the levels below. For instance, in §5.2 we vary the problem text without affecting the mental model, arithmetic expression or answer.

Having selected X, we adapt the pipeline (§4.2) to generate example pairs, one with $X = x$ and one with $X = x'$. Next, we evaluate the data quality using the procedure described in App. A.2. After quality assurance, we generate a larger sample of 400 additional problem pairs, which (including the quality evaluation set) gives a total of 500 pairs for the tests. We then generate outcomes $Y(x)$ and $Y(x')$ for each of the pairs for a set of selected LLMs. We use LLaMA2 (Touvron et al., 2023) with 7B, 13B and 70B parameters, Mistral 7B (Jiang et al., 2023) and Mixtral 8x7B (Jiang et al., 2024), GPT-3.5 Turbo, and GPT-4 Turbo (OpenAI, 2024). We consider both the pretrained-only and instruction-tuned versions for the LLaMA2, Mistral and Mixtral models. We carry out zero-shot inference with a standard prompt, a chain-of-thought prompt (CoT; Wei et al., 2022), as well as a modified "child-persona" CoT prompt, whose results were similar to those of the CoT setup and are presented in App. B.1. With the former, the models are prompted to *directly* provide an answer after the input. Following previous work (Kojima et al., 2022; Yang et al., 2024), we use the format "*Q:* `{problem}\n`*A: The answer (Arabic numerals) is* " for base models and "`{problem}\n`*The answer (Arabic numerals)*

[7] Interestingly, we found that GPT-3.5 Turbo sometimes would transform inconsistent formulations of comparison-type relationships into semantically equivalent consistent ones, which already indicates presence of a bias towards consistent problems. Such erroneous transformations were filtered out (see App. A.1).

Mode	Model	Consistency bias (§5.2)				Transfer vs comparison bias (§5.3)				Carry effect (§5.4)			
		Accuracy (%)			p-value	Accuracy (%)			p-value	Accuracy (%)			p-value
		Co	InCo	CATE		T	C	CATE		NCa	Ca	CATE	
Direct	LLaMA2 7B	9.6	4.8	**4.8**	<0.001	21.8	13.0	**8.8**	<0.001	64.8	60.0	**4.8**	0.009
	LLaMA2 13B	17.2	14.0	**3.2**	0.006	28.6	20.0	**8.6**	<0.001	72.2	67.2	5.0	0.030
	LLaMA2 70B	24.0	16.2	**7.8**	<0.001	45.4	26.8	**18.6**	<0.001	95.2	96.2	1.0	0.380
	Mistral 7B	17.8	12.0	**5.8**	<0.001	34.0	20.4	**13.6**	<0.001	72.4	72.0	0.4	0.835
	Mixtral 8x7B	23.0	17.0	**6.0**	<0.001	42.2	30.4	**11.8**	<0.001	95.4	93.6	1.8	0.117
	LLaMA2 7B Chat	14.2	10.8	**3.4**	0.009	20.2	15.8	**4.4**	0.005	61.2	54.2	7.0	0.012
	LLaMA2 13B Chat	16.4	11.8	**4.6**	<0.001	25.4	18.2	**7.2**	<0.001	65.6	59.6	6.0	0.018
	LLaMA2 70B Chat	16.4	14.8	1.6	0.158	32.4	20.0	**12.4**	<0.001	96.4	97.0	-0.6	0.578
	Mistral 7B Instr.	17.6	14.2	**3.4**	0.008	28.0	21.8	**6.2**	<0.001	78.0	78.6	-0.6	0.802
	Mixtral 8x7B Instr.	23.4	21.8	1.6	0.195	42.6	28.0	**14.6**	<0.001	95.8	96.4	-0.6	0.578
	GPT-3.5 Turbo	32.2	22.8	**9.4**	<0.001	61.0	33.4	**27.6**	<0.001	99.6	99.4	0.2	0.320
CoT	LLaMA2 7B	16.4	6.0	**10.4**	<0.001	18.8	13.6	**5.2**	0.009	33.2	38.8	-5.6	0.006
	LLaMA2 13B	30.2	8.6	**21.6**	<0.001	37.8	13.2	**24.6**	<0.001	33.8	33.4	0.4	0.833
	LLaMA2 70B	40.2	24.0	**16.2**	<0.001	63.8	33.0	**30.8**	<0.001	68.6	67.6	1.0	0.850
	Mistral 7B	36.4	16.8	**19.6**	<0.001	49.8	58.8	**-9.0**	0.004	73.2	71.0	2.2	0.283
	Mixtral 8x7B	62.4	42.2	**20.2**	<0.001	68.6	65.0	3.6	0.206	79.8	79.8	0.0	1.000
	LLaMA2 7B Chat	66.8	38.6	**28.2**	<0.001	69.6	40.8	**28.8**	<0.001	72.4	71.0	1.4	0.514
	LLaMA2 13B Chat	67.0	28.6	**38.4**	<0.001	79.4	48.0	**31.4**	<0.001	73.8	78.6	-4.8	0.017
	LLaMA2 70B Chat	82.8	61.4	**21.4**	<0.001	99.0	76.2	**22.8**	<0.001	97.0	95.8	1.2	0.180
	Mistral 7B Instr.	61.8	33.6	**28.2**	<0.001	83.4	52.0	**31.4**	<0.001	78.6	75.6	3.0	0.162
	Mixtral 8x7B Instr.	85.4	71.6	**13.8**	<0.001	98.2	83.8	**14.4**	<0.001	97.0	94.6	2.4	0.014
	GPT-3.5 Turbo	89.2	87.8	1.4	0.380	97.0	93.0	**4.0**	0.003	97.8	98.2	-0.4	0.580
	GPT-4 Turbo	90.4	72.4	**18.0**	<0.001	99.2	91.4	**7.8**	<0.001	99.6	99.6	0.0	-

Table 2. Accuracy, conditional average treatment effect (CATE), and statistical significance (p-value) on math word problems generated for the three tests detailed in §5.2, §5.3 and §5.4. 'Co' denotes consistent, 'InCo' inconsistent, 'T' transfer, 'C' comparison, 'Ca' carry, and 'NCa' no carry conditions. Results are separated by whether the prompting method is direct or chain-of-thought (CoT). 'Chat' and 'Inst.' indicate the instruction-tuned versions of the models. CATE values are **bolded** when significant, controlling for false discovery rate at level α=0.05. Results from an additional prompting strategy, child-persona prompting, are presented in Table 4.

is" for instruction-tuned models. Then, the model prediction is retrieved from the response by extracting the first number in the model's output. For the CoT experimental procedure, we follow the exact method from Kojima et al. (2022). First, the model is prompted to generate a reasoning chain by appending *"Let's think step by step"* to the input. Then, the model is re-prompted to generate the final answer, which is extracted from the output as in the direct case. Responses are generated with greedy decoding and a maximum length of 256 tokens. After obtaining the model's predictions, we estimate the CATE and perform a two-tailed paired sample t-test to determine whether the CATE estimate is significantly different from 0. More specifically, the null hypothesis is that the two groups of model accuracy have the same mean. We control the false discovery rate at level α=0.05 using Benjamini and Hochberg's (1995) procedure, considering all null hypotheses under the same bias as one distinct family.

5.2. Problem Text: Consistency Bias

Varying the linguistic form of an otherwise equivalent problem structure can have a large effect on child performance (Cummins et al., 1988). We test whether comparison problems with inconsistent statements are harder for LLMs than the same problems with an analogous consistent statement.

Method. Following Lewis and Mayer's (1987) study, we consider consistent/inconsistent problem pairs where the required operation is either addition, subtraction, multiplication or division. The generation pipeline is tuned so that the problem structures follow the specification:

$$\texttt{container} \circ \underbrace{(\texttt{transfer} \mid \texttt{rate}) \circ \cdots \circ (\texttt{transfer} \mid \texttt{rate})}_{0-2\text{ times}} \circ$$
$$\texttt{comparison} \circ \underbrace{(\texttt{transfer} \mid \texttt{rate}) \circ \cdots \circ (\texttt{transfer} \mid \texttt{rate})}_{0-2\text{ times}};$$

Note that the problems may have between $1-5$ reasoning steps—one for every non-container predicate. Apart from the first predicate (`container`), only the comparison predicate introduces a new agent. The question queries the agent that was introduced by `comparison`.[8] The pairs are generated such that the only sentence that varies is that which corresponds to `comparison`, one being consistent and the other being its analogous inconsistent form.

Results. The results of the consistency bias test reveal 20 out of 23 statistically significant CATEs. As displayed in Table 2 (leftmost column), all models exhibit lower accuracy when solving inconsistent problems compared to their consistent counterparts. Interestingly, the bias appears to be

[8] The following is a (consistent) example from our dataset that follows the pattern `container` ∘ `transfer` ∘ `comparison`: "*Avery has 15 desks. Avery bought 18 desks. Natalie has 16 fewer desks than Avery. How many desks does Natalie have?*"

exacerbated by CoT prompting, which improves the overall model performance, but magnifies the difference in accuracy between consistent and inconsistent problems. This finding aligns with research indicating that CoT prompting may also amplify other types of biases present in the training data (Shaikh et al., 2023). Particularly notable CATEs are observed for the base versions of LLaMA2 7B, LLaMA2 13B, and Mistral 7B, for which the inconsistent formulation of the problems leads to an accuracy drop larger than 50%.

5.3. Mental Model: Transfer vs Comparison Bias

Another factor behind performance is that it might be harder to perform solution planning based on some mental models compared to others. We test whether LLMs are better at solving transfer-type problems than comparison problems.

Method. The problem structures take the following forms:

$$\texttt{container} \circ \underbrace{\texttt{transfer} \circ \cdots \circ \texttt{transfer}}_{1-5 \text{ times}};$$

$$\texttt{container} \circ \underbrace{\texttt{comparison} \circ \cdots \circ \texttt{comparison}}_{1-5 \text{ times}};$$

The two problems have identical arithmetic expressions. Each comparison predicate corresponds to a comparison of a new agent with the agent introduced in the preceding sentence. Each transfer statement follows the same agent, who was introduced in the first sentence and whose state is updated through a transfer with some other agent. The problems resemble each other in linguistic form as much as possible. In particular, we make sure that the same agent names are introduced in each sentence across the two problems, in order to account for variance in performance stemming from such choices (Goodarzi et al., 2023). Consistent or inconsistent forms of comparison are sampled uniformly at random.

Results. The experimental results (middle column in Table 2) show that models are consistently more accurate on problems based on transfers rather than comparisons. With the exception of CoT-prompted pretrained-only Mistral and Mixtral, we observe statistically significant positive CATEs, mirroring biases seen in children's problem-solving (§3).

Further, we note that the instruction-tuned models overall exhibit larger effects than pretrained-only models in the CoT setting, but not in the direct setting. This seems to be the case for the consistency bias as well. Finally, in App. B.2 we show some of the results as broken down by number of reasoning steps in the CoT setting.

5.4. Arithmetic Expressions: The Carry Effect

While much of a child's performance on word problems can be explained by properties introduced by the text format, a large portion still depends on the nature of the arithmetic expression (Daroczy et al., 2015). We test whether LLMs are sensitive to the presence of arithmetic carry when posed with addition and subtraction in math word problems.

Method. We generate pairs under the comparison specification from §5.3, but with only one step:[9]

$$\texttt{container} \circ \texttt{comparison};$$

As in §5.3, we use additive comparisons, which ensures that the arithmetic expressions only have addition and/or subtraction operators. The two problems of a pair are identical apart from the numbers. Following Fürst and Hitch (2000), we ensure that both operands as well as the answer of the problem are three-digit numbers (since children appear to rely on long-term memory for problems with small numbers; Footnote 4). One of the problems has no carry, the other has at least one (i.e., unit or tens carry). The correct answer of the two problems is controlled to be the same.

Results. The results (rightmost column in Table 2) depart from the findings above, which gave evidence for the presence of child-like biases. In this case, model performance is similar in problems with and without carry operations—we only observe one significant result out of the 23 tests. Thus, the models seem not to be sensitive to variations isolated to the arithmetic expression level.

6. Why do Language Models Exhibit Biases?

A natural set of questions that arise from our results is why some child-like biases are present in the models, and why some of them (like the carry effect) are absent. The most plausible explanation in our view is the influence from the training data: If the training data contains many examples of humans writing and solving word problems, then it may be that LLMs simulate human biases present in such text. For instance, it may be that the distribution of the training data is skewed towards consistent problem formulations of comparison relations, which in turn could be a product of consistency bias in the humans who wrote the word problems. This would seem plausible given that the consistency bias is present also in adults (Lewis and Mayer, 1987; Hegarty et al., 1995).

Unfortunately, we cannot directly verify this hypothesis since it is unknown what data has been used to train the models considered in this study. However, as a proxy, we performed an analysis of a set consisting of word problems from MAWPS (Koncel-Kedziorski et al., 2016b), ASDIV-A

[9] The one-step case follows the setups from studies on humans (§3). We discovered in the previous two tests that the models frequently fail on comparison problems with only one step (which can be inferred from Figs. 3 and 4), so if a carry effect is present, it should be observable in such a setting.

(Miao et al., 2020), and SVAMP (Patel et al., 2021). These three datasets are well known and publicly available, and are thus likely to have been present in the training data of the LLMs used in this work. We found that these datasets indeed have more consistent formulations than inconsistent ones, and more transfer problems than comparison problems. More specifically, the ratios observed were 5:1 (15 and 3 in absolute numbers) and 130:9 (260 and 18 in absolute numbers), respectively.[10] In other words, the imbalance in problem types on these datasets is consistent with the biases we found in our analyses on text comprehension (§5.2) and solution planning (§5.3).

Extrapolating this hypothesis to the absence of a carry effect would imply that there is little to no difference in frequency between problems with and without carry in the training data. This would be harder to verify, as there are many potential traces of carry in the data beyond word problems. Furthermore, the carry effect results suggest that there are algorithmic differences in how LLMs and humans perform arithmetic computations. In particular, the carry effect in humans is partially attributed to working memory limitations (Hitch, 1978), which LLMs may not implement in the same manner. The memory and computational mechanisms through which models perform arithmetic (Nanda et al., 2023; Stolfo et al., 2023a; Quirke and Barez, 2024) are likely not affected by the increased cognitive load that the carry operation introduces in humans. This leads to an alternative, albeit arguably less plausible view, on why we observe the other two biases: It might be that there is algorithmic similarity between humans and language models on text comprehension and solution planning. Assessing this hypothesis would require knowledge about the mechanisms of human and language-model reasoning alike, both of which are beyond the scope of the present study. However, our results at least suggest a direction, namely, that there is at least the *possibility* that the algorithms in text comprehension and solution planning exhibit similarity.

7. Related Work

Our work relates most closely to studies that have compared human and LLM biases on syllogisms (Ando et al., 2023; Eisape et al., 2024), code generation (Jones and Steinhardt, 2022), and other non-mathematical inference tasks (Dasgupta et al., 2023). Their findings indicate that LLMs are susceptible to some of the same biases as humans, like content effects (Ando et al., 2023; Dasgupta et al., 2023) and premise ordering effects (Eisape et al., 2024). We observe similar results in a mathematical problem-solving setting for consistency bias and transfer vs comparison bias, but not for the carry effect which relates to the step of the cognitive process that involves solving arithmetic equations. Our study also differs from those referenced above in that we systematically compare the effect of CoT prompting to direct prompting, observing amplified effects in the CoT setting in most cases where effects are present.

We are unaware of any other work that studies cognitive biases in LLMs that, like the carry effect, relate directly to numbers. However, there seem to be similarities between the numeric representations in LLMs and the "mental number line" in humans (Shah et al., 2023). Other studies find evidence that LLMs to some extent rely on spurious correlations in numerical reasoning (Razeghi et al., 2022; Stolfo et al., 2023b), and that their performance decreases with increasing number size (Dziri et al., 2023; Shen et al., 2023). Beyond numerical reasoning, LLMs appear to have difficulties with causal reasoning (Binz and Schulz, 2023; Jin et al., 2023; 2024) and proof planning (Saparov and He, 2023).

8. Conclusion and Implications

This study explored whether LLMs exhibit child-like cognitive biases in arithmetic word problem-solving. We found that LLMs demonstrate biases in text comprehension and solution planning that mirror human tendencies. Specifically, models performed better on problems where the relational keyword is consistent with the appropriate arithmetic operator compared to problems where it is not, as well as on problems with a dynamic change of state compared to problems with a static comparison. However, at the solution execution step, LLMs did not exhibit the child-like carry effect for arithmetic computations. In general, studying biases that are present in children but not in adults may enable the disentanglement of the influence of training data from other factors that might explain language model behavior, since one would expect the training set to be heavily biased towards adult (rather than child) thinking. We therefore believe it might be a promising direction forward in language model interpretability work.

Impact Statement

Cognitive modeling enables human simulations in place of data collection that might otherwise be unethical, harmful or costly. On the other hand, issues could arise if those simulations are unfaithful to human behavior. As a broader implication of our work, we encourage practitioners to exercise care when developing and deploying cognitive models of students using LLMs, particularly, in how the student

[10] We used the world model annotations from Opedal et al. (2023) for this analysis, which enabled us to extract the problems with the relevant concepts. The problems containing a comparison predicate were few enough for manual inspection. For the transfer problems, 433 in total, we manually inspected a sample of 100. Out of these, 60 had at least one sentence with a transfer of the same structure we consider in this work. Maximum likelihood estimation then yields $0.6 \times 433 \approx 260$ relevant transfer problems.

model treats numbers and other properties of arithmetic expressions. We hope that our results can provide insights for practitioners seeking to develop automated learning agents, for instance, under the tutor-learning paradigm in which a student learns by correcting the mistakes made by a computer model (Okita, 2014; Shahriar and Matsuda, 2021; Schmucker et al., 2023). We do not see any notable ethical issues with our work.

Limitations

We cannot draw any parallels on the absolute performance in comparison with children, only on the presence or absence of each effect. This is because the datasets used in the learning science studies were not available to us. The one exception was the data from Hegarty et al. (1995), which we evaluate on in App. B.3. Moreover, we do not consider the grade level of the problems, but see Jiao et al. (2023) for a generation method that does.

In selecting specific cognitive biases to study, we chose biases that are well-established in literature on human children and whose effects could be clearly associated with one of the steps of Fig. 1. Another factor that fulfills these desiderata is the effect of explicit verbal cues (Hudson, 1983; Vicente et al., 2007). More fundamentally, a complete comparison of the biases between LLMs and humans would need to study biases that have been found in LLMs but are not necessarily present in humans. We do not take that direction into account, but we note that the number frequency effect reported by Razeghi et al. (2022) might be one such example.

We did not use in-context examples in our evaluation since the addition of such may influence the results in ways that can be difficult to foresee. However, an interesting direction for future work could be to study whether cognitive biases can be controlled through specific choices of in-context examples or other prompts.

We stress that the conceptualization in Fig. 1 is a simplified model of the solving process. For instance, it fails to account for shortcut strategies (see Footnote 3) and it does not consider any propositional text-base representation which precedes the mental model representation in some other models (Kintsch and Greeno, 1985; Hegarty et al., 1995). We do not make any claims on the ability of LLMs to "construct mental models" in this work, although our results could potentially have such implications as was pointed out by a reviewer. See App. C for a brief discussion.

Finally and importantly, we only consider problems formulated in English. We note that some effects could vary across languages. For instance, the carry effect is more pronounced in German and other languages where the spelled-out order of tens and units is inverted in relation to Arabic numerical notation (Göbel et al., 2014). Our generation pipeline can be straightforwardly adapted to other languages, and future work might consider doing so.

Acknowledgements

We thank Emo Welzl, Ethan Gotlieb Wilcox, Julia Chatain and Yilmazcan Ozyurt for valuable feedback and discussions. Andreas Opedal acknowledges funding from the Max Planck ETH Center for Learning Systems. Alessandro Stolfo is supported by armasuisse Science and Technology through a CYD Doctoral Fellowship.

References

Gati Aher, Rosa I. Arriaga, and Adam Tauman Kalai. 2023. Using large language models to simulate multiple humans and replicate human subject studies. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Risako Ando, Takanobu Morishita, Hirohiko Abe, Koji Mineshima, and Mitsuhiro Okada. 2023. Evaluating large language models with NeuBAROCO: Syllogistic reasoning ability and human-like biases. In *Proceedings of the 4th Natural Logic Meets Machine Learning Workshop*, pages 1–11, Nancy, France. Association for Computational Linguistics.

Jacob Andreas. 2022. Language models as agent models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5769–5779, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lisa P. Argyle, Ethan C. Busby, Nancy Fulda, Joshua R. Gubler, Christopher Rytting, and David Wingate. 2023. Out of one, many: Using language models to simulate human samples. *Political Analysis*, 31(3):337–351.

Mark H. Ashcraft, Rick D. Donley, Margaret A. Halas, and Mary Vakali. 1992. Chapter 8 working memory, automaticity, and problem difficulty. In Jamie I. D. Campbell, editor, *The Nature and Origins of Mathematical Skills*, volume 91 of *Advances in Psychology*, pages 301–329. North-Holland.

Yoav Benjamini and Yosef Hochberg. 1995. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300.

Marcel Binz and Eric Schulz. 2023. Using cognitive psychology to understand GPT-3. *Proceedings of the National Academy of Sciences*, 120(6).

Diane J. Briars and Jill H. Larkin. 1984. An integrated model of skill in solving elementary word problems. *Cognition and Instruction*, 1(3):245–296.

Thomas P. Carpenter and James M. Moser. 1984. The acquisition of addition and subtraction concepts in grades one through three. *Journal for Research in Mathematics Education*, 15(3):179–202.

Denise Dellarosa Cummins, Walter Kintsch, Kurt Reusser, and Rhonda Weimer. 1988. The role of understanding in solving word problems. *Cognitive Psychology*, 20(4):405–438.

Gabriella Daroczy, Magdalena Wolska, Walt Detmar Meurers, and Hans-Christoph Nuerk. 2015. Word problems: a review of linguistic and numerical factors contributing to their difficulty. *Frontiers in Psychology*, 6.

Ishita Dasgupta, Andrew K. Lampinen, Stephanie C. Y. Chan, Hannah R. Sheahan, Antonia Creswell, Dharshan Kumaran, James L. McClelland, and Felix Hill. 2023. Language models show human-like content effects on reasoning tasks.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Tiwalayo Eisape, MH Tessler, Ishita Dasgupta, Fei Sha, Sjoerd van Steenkiste, and Tal Linzen. 2024. A systematic comparison of syllogistic reasoning in humans and language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Mexico City, Mexico.

Yanai Elazar, Akshita Bhagia, Ian Helgi Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Evan Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, Hanna Hajishirzi, Noah A. Smith, and Jesse Dodge. 2023. What's in my big data? In *The Twelfth International Conference on Learning Representations*.

Amir Feder, Katherine A. Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E. Roberts, Brandon M. Stewart, Victor Veitch, and Diyi Yang. 2022. Causal inference in natural language processing: Estimation, prediction, interpretation and beyond. *Transactions of the Association for Computational Linguistics*, 10:1138–1158.

Ansgar J. Fürst and Graham J. Hitch. 2000. Separate roles for executive and phonological components of working memory in mental arithmetic. *Memory & Cognition*, 28(5):774–782.

Saeed Goodarzi, Nikhil Kagita, Dennis Minn, Shufan Wang, Roberto Dessi, Shubham Toshniwal, Adina Williams, Jack Lanchantin, and Koustuv Sinha. 2023. Robustness of named-entity replacements for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10914–10931, Singapore. Association for Computational Linguistics.

Silke M. Göbel, Korbinian Moeller, Silvia Pixner, Liane Kaufmann, and Hans-Christoph Nuerk. 2014. Language affects symbolic arithmetic in children: The case of number word inversion. *Journal of Experimental Child Psychology*, 119:17–25.

Mary Hegarty, Richard E. Mayer, and Christopher A. Monk. 1995. Comprehension of arithmetic word problems: A comparison of successful and unsuccessful problem solvers. *Journal of Educational Psychology*, 87:18–32.

Graham J Hitch. 1978. The role of short-term working memory in mental arithmetic. *Cognitive Psychology*, 10(3):302–323.

Tom Hudson. 1983. Correspondences and numerical differences between disjoint sets. *Child Development*, 54:84–90.

Guido W. Imbens and Donald B. Rubin. 2015. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press, USA.

Joshua Benjamin Jaffe and Donald Joseph Bolger. 2023. Cognitive processes, linguistic factors, and arithmetic word problem success: a review of behavioral studies. *Educational Psychology Review*, 35(4):105.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Ying Jiao, Kumar Shridhar, Peng Cui, Wangchunshu Zhou, and Mrinmaya Sachan. 2023. Automatic educational question generation with difficulty level controls. In *International Conference on Artificial Intelligence in Education*, pages 476–488. Springer.

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, and Bernhard Schölkopf. 2023. CLadder: A benchmark to assess causal reasoning capabilities of language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zhijing Jin, Jiarui Liu, Zhiheng Lyu, Spencer Poff, Mrinmaya Sachan, Rada Mihalcea, Mona Diab, and Bernhard Schölkopf. 2024. Can large language models infer causation from correlation? In *The Twelfth International Conference on Learning Representations*.

Philip Nicholas Johnson-Laird. 1983. *Mental models: towards a cognitive science of language, inference and consciousness*, volume 6 of *Cognitive Science Series*. Harvard University Press, Cambridge, Massachusetts.

Erik Jones and Jacob Steinhardt. 2022. Capturing failures of large language models via human cognitive biases. In *Advances in Neural Information Processing Systems*.

Walter Kintsch and James G. Greeno. 1985. Understanding and solving word arithmetic problems. *Psychological Review*, 92(1):109–129.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *ICML 2022 Workshop on Knowledge Retrieval and Language Models*.

Rik Koncel-Kedziorski, Ioannis Konstas, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2016a. A theme-rewriting approach for generating algebra word problems. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1617–1628, Austin, Texas. Association for Computational Linguistics.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016b. MAWPS: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1152–1157, San Diego, California. Association for Computational Linguistics.

John W. Koshmider and Mark H. Ashcraft. 1991. The development of children's mental multiplication skills. *Journal of Experimental Child Psychology*, 51(1):53–89.

Sang Kwon, Gagan Bhatia, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. Beyond English: Evaluating LLMs for Arabic grammatical error correction. In *Proceedings of ArabicNLP 2023*, pages 101–119, Singapore (Hybrid). Association for Computational Linguistics.

Tanja Käser and Giora Alexandron. 2023. Simulated learners in educational technology: A systematic literature review and a turing-like test. *International Journal Of Artificial Intelligence In Education*.

Anne Bovenmyer Lewis and Richard E. Mayer. 1987. Students' miscomprehension of relational statements in arithmetic word problems. *Journal of Educational Psychology*, 79:363–371.

Mengsay Loem, Masahiro Kaneko, Sho Takase, and Naoaki Okazaki. 2023. Exploring effectiveness of GPT-3 in grammatical error correction: A study on performance and controllability in prompt-based methods. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 205–219, Toronto, Canada. Association for Computational Linguistics.

Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. MathDial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5602–5621, Singapore. Association for Computational Linguistics.

Kyle Mahowald, Anna A Ivanova, Idan A Blank, Nancy Kanwisher, Joshua B Tenenbaum, and Evelina Fedorenko. 2023. Dissociating language and thought in large language models: a cognitive perspective.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing English math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984, Online. Association for Computational Linguistics.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*.

P. Nesher, James G. Greeno, and Mary S. Riley. 1982. The development of semantic categories for addition and subtraction. *Educational Studies in Mathematics*, 13:373–394.

Perla Nesher and Eva Teubal. 1975. Verbal cues as an interfering factor in verbal problem solving. *Educational Studies in Mathematics*, 6(1):41–51.

Manh Hung Nguyen, Sebastian Tschiatschek, and Adish Singla. 2023. Large language models for in-context student modeling: Synthesizing student's behavior in visual programming from one-shot observation.

Sandra Y. Okita. 2014. Learning from the folly of others: Learning to self-correct by monitoring the reasoning of virtual characters in a computer-supported mathematics learning environment. *Computers Education*, 71:257–278.

Andreas Opedal, Niklas Stoehr, Abulhair Saparov, and Mrinmaya Sachan. 2023. World models for math story problems. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9088–9115, Toronto, Canada. Association for Computational Linguistics.

OpenAI. 2024. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Jean Piaget. 1936. *La naissance de l'intelligence chez l'enfant*. Delachaux et Niestlé.

Oleksandr Polozov, Eleanor O'Rourke, Adam M. Smith, Luke Zettlemoyer, Sumit Gulwani, and Zoran Popovic. 2015. Personalized mathematical word problem generation. In *IJCAI*.

Philip Quirke and Fazl Barez. 2024. Understanding addition in transformers. In *The Twelfth International Conference on Learning Representations*.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot numerical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 840–854, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mary Riley, James Greeno, and Joan Heller. 1983. *Development of Children's Problem-Solving Ability in Arithmetic*, page 153–196. Learning Research and Development Center, University of Pittsburgh.

Abulhair Saparov and He He. 2023. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *International Conference on Learning Representations*.

Robin Schmucker, Meng Xia, Amos Azaria, and Tom Mitchell. 2023. Ruffle&Riley: Towards the automated induction of conversational tutoring systems.

Raj Shah, Vijay Marupudi, Reba Koenen, Khushi Bhardwaj, and Sashank Varma. 2023. Numeric magnitude comparison effects in large language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6147–6161, Toronto, Canada. Association for Computational Linguistics.

Tasmia Shahriar and Noboru Matsuda. 2021. "Can you clarify what you said?": Studying the impact of tutee agents' follow-up questions on tutors' learning. In *Artificial Intelligence in Education*, pages 395–407, Cham. Springer International Publishing.

Omar Shaikh, Hongxin Zhang, William Held, Michael Bernstein, and Diyi Yang. 2023. On second thought, let's not think step by step! bias and toxicity in zero-shot reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4454–4470, Toronto, Canada. Association for Computational Linguistics.

Ruoqi Shen, Sébastien Bubeck, Ronen Eldan, Yin Tat Lee, Yuanzhi Li, and Yi Zhang. 2023. Positional description matters for transformers arithmetic.

Elsbeth Stern. 1993. What makes certain arithmetic word problems involving the comparison of sets so difficult for children? *Journal of Educational Psychology*, 85:7–23.

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. 2023a. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In *Proceedings of the 2023 Conference* *on Empirical Methods in Natural Language Processing*, pages 7035–7052, Singapore. Association for Computational Linguistics.

Alessandro Stolfo, Zhijing Jin, Kumar Shridhar, Bernhard Schoelkopf, and Mrinmaya Sachan. 2023b. A causal framework to quantify the robustness of mathematical reasoning with language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 545–561, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Kurt VanLehn, Stellan Ohlsson, and Rod Nason. 1994. Applications of simulated students: An exploration. *International Journal of Artificial Intelligence in Education*, 5.

Santiago. Vicente, Jose. Orrantia, and Lieven. Verschaffel. 2007. Influence of situational and conceptual rewording on word problem solving. *British Journal of Educational Psychology*, 77(4):829–848.

Alex Warstadt and Samuel R Bowman. 2022. What artificial neural networks can tell us about human language acquisition. In *Algebraic Structures in Natural Language*, pages 17–60. CRC Press.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*.

Correct all grammatical mistakes that appear in the following math word problem: `[templated text]`
Fix any awkward or redundant phrasing. Pay close attention to incorrect plural forms. Do NOT solve the problem. Do NOT compute any intermediate solutions. Do NOT make any changes to the numerical values or implied mathematical operations. Only output the corrected math word problem and nothing else. Do NOT restate the original problem. Do NOT include "Corrected Version:" or any description of the task.

Table 3. Prompt used for the linguistic error correction step in our generation pipeline from §4.2.

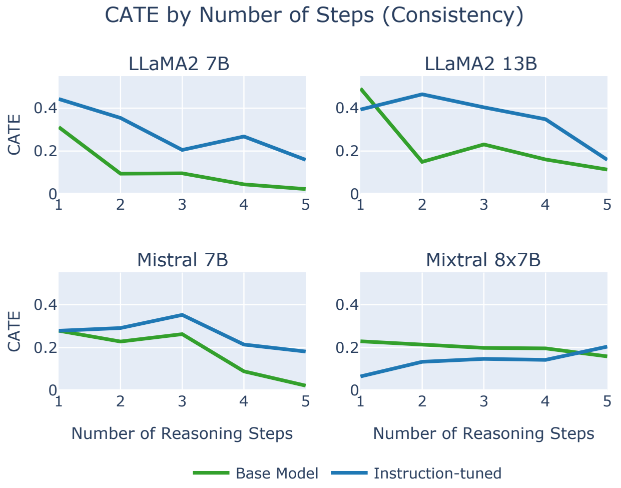


Figure 3. CATE for the consistency effect (§5.2) in CoT-prompted models stratified by number of reasoning steps. 'Instruction-tuned' refers to the *Chat* and *Instruct* versions of LLaMA2 and Mistral/Mixtral, respectively.

A. More Details on the Generation Method

A.1. Details on Linguistic Error Correction

We use GPT-3.5 Turbo to carry out the linguistic error correction step detailed in §4.2. In Table 3, we provide the exact prompt used for the task. The corrected problem is generated using greedy decoding (`temperature=0`). We carry out additional integrity checks of the generated problem against the original templated text. In particular, we verify that the sentence count and relational terms (such as *"more"*) are consistent post error-correction. The problem is discarded if these additional checks are not satisfied.

A.2. Data Quality Evaluation

We describe the manual evaluation of the datasets generated in §5. For *each* of the datasets, we do the following: First, generate a control set of 10 examples. These 10 examples are evaluated independently by three of this paper's authors. If there are any errors we make appropriate modifications to the pipeline and restart the procedure. If not, we proceed to evaluate 90 more examples, allocating 30 to each of the three authors. Error rate is estimated on this sample of 100 examples.

We use two binary evaluation criteria, one assessing the linguistic error correction step (iv) and one assessing test-specific attributes. A problem is deemed to be good according to the former if the generated problem only deviates from the templated text through spelling or grammar correction. The test-specific criterion and the obtained error estimates are given below.

Consistency bias (§5.2). We evaluated data quality according to whether the two comparison statements actually were consistent and inconsistent forms to express the same comparison relationship. To be precise, the first problem needs to have a consistent relational statement for the comparison predicate, the second problem needs to have an equivalent inconsistent relational statement for the same comparison predicate, and the two problems need to be identical otherwise. Our pipeline achieved a 0% error rate on both criteria on the 100 evaluated example problems.

Transfer vs comparison bias (§5.3). We evaluated data quality according to whether the comparison and transfer problems had the same arithmetic expression and whether they followed the specified problem structure. We also ensured that the agent names and other properties matched. Our pipeline achieved a 0% error rate on both criteria on the 100 evaluated example problems.

The carry effect (§5.4). We evaluated data quality according to whether one problem had no carry computation steps, the other one had at least one, and they were equal otherwise. That is, only the numbers differed and the two problems had the same answer. Our pipeline achieved a 0% error rate on both criteria on the 100 evaluated example problems.

A.3. Related Methods

Our generation pipeline differs from Opedal et al.'s (2023) method in that we generate intermediate templated texts, while they generate the problem texts directly conditioned on mental models. Polozov et al. (2015) also generates word problems from logical representations, but their approach does not allow explicit control over arithmetic concepts, which is an important factor underlying difficulty level; §3. While our experiments require strict control over linguistic form, our error correction step could in principle be broadened to perform paraphrasing and theme rewriting (Koncel-Kedziorski et al., 2016a) as well.

Mode	Model	Consistency bias (§5.2)				Transfer vs comparison bias (§5.3)				Carry effect (§5.4)			
		Accuracy (%)			p-value	Accuracy (%)			p-value	Accuracy (%)			p-value
		Co	**InCo**	CATE		**T**	**C**	CATE		**NCa**	**Ca**	CATE	
Child CoT	LLaMA2 7B	14.6	5.0	**9.6**	<0.001	19.8	11.6	**8.2**	<0.001	40.0	44.4	-4.4	0.048
	LLaMA2 13B	20.0	5.0	**15.0**	<0.001	35.0	7.0	**28.0**	<0.001	20.8	21.0	-0.2	0.903
	Mistral 7B	39.2	17.2	**22.0**	<0.001	48.8	26.6	**22.2**	<0.001	58.4	57.2	1.2	0.415
	Mixtral 8x7B	66.2	34.6	**31.6**	<0.001	69.8	49.4	**20.4**	<0.001	70.4	69.4	1.0	0.701
	LLaMA2 7B Chat	55.2	24.2	**31.0**	<0.001	62.6	33.8	**28.8**	<0.001	67.8	63.0	4.8	0.069
	LLaMA2 13B Chat	65.2	27.0	**38.2**	<0.001	79.8	48.2	**31.6**	<0.001	80.0	77.0	3.0	0.108
	Mistral 7B Instr.	65.0	30.6	**34.4**	<0.001	75.2	52.8	**22.4**	<0.001	77.2	74.4	2.8	0.178
	Mixtral 8x7B Instr.	88.6	72.4	**16.2**	<0.001	98.8	82.4	**16.4**	<0.001	97.6	97.4	0.2	0.809

Table 4. Accuracy, conditional average treatment effect (CATE), and statistical significance (p-value) on math word problems generated for the three tests detailed in §5.2, §5.3 and §5.4. 'Co' denotes consistent, 'InCo' inconsistent, 'T' transfer, 'C' comparison, 'Ca' carry, and 'NCa' no carry conditions. The results presented are for the child-persona prompting strategy described in App. B.1. 'Chat' and 'Inst.' indicate the instruction-tuned versions of the models. CATE values are **bolded** when $p < 0.001$.

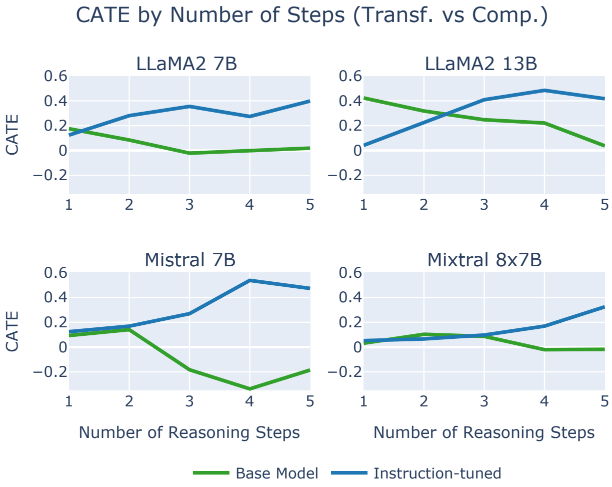


Figure 4. CATE for transfer vs comparison (§5.3) in the CoT-prompted case stratified by number of reasoning steps. 'Instruction-tuned' refers to the *Chat* and *Instruct* versions of LLaMA2 and Mistral/Mixtral, respectively. Base and instruction-tuned models display opposite relationships between length and bias strength.

B. Additional Results

B.1. Child-Persona Prompting

Taking inspiration from claims that LLMs can act as agent models (Andreas, 2022), we also experimented with an additional prompt in which the LLM is instructed to impersonate a grade-school child. Specifically, we employ a modified version of the zero-shot chain-of-thought prompt, tailored to simulate a child's reasoning process. We prompt the model with the phrase *"Let's think step by step as a grade-school child would,"* replacing the standard CoT instruction. Following this, we apply the same decoding method used in traditional CoT. The results for this approach are reported for the open-source models (apart from LLaMA2 70B) in Table 4. While we notice larger consistency and transfer vs comparison effects for some models, we observe no substantial departure from the results achieved with conventional CoT prompting.

B.2. Bias Strength by Number of Reasoning Steps

In Fig. 4 we show how the CATE of the transfer vs comparison bias varies with the number of reasoning steps in the problems for the CoT setting. Interestingly, we observe that the CATE sizes increase with the number of reasoning steps for the instruction-tuned models, whereas they decrease for the pretrained-only base models. We are unaware of literature on the relationship between human transfer vs comparison bias and the number of steps, so we can not make any claims about which of these patterns is more cognitively plausible.

The consistency-effect test does not exhibit such diverging trends for the CATEs. Fig. 3 illustrates how the strength of the measured biases change in relation to the number of reasoning steps in a problem (in the CoT-prompted case). Note that the carry effect experiments were carried it out for problems with only one step.

B.3. Data from Hegarty et al. (1995)

In Table 5 we present the results on a few selected models when evaluated on the data from the study by Hegarty et al. (1995). Theirs was the only background study for which we were able to obtain the data that was used. The dataset contains eight problem pairs and targets the consistency bias. While we do not obtain significant results, we do get an indication of the absolute effect of the bias as compared to the human subjects in Hegarty et al.'s (1995) study (who were undergraduate college students). In their study, solvers who committed at least four errors out of the total of 16 problems had an average accuracy of 62% and 24%, on consistent problems and inconsistent problems respectively. None of the absolute accuracies in Table 5 are similar, but Mixtral 8x7B Instruct displays a similar absolute effect size

Mode	Model	Consistency bias			
		Accuracy (%)			p-value
		Co	**InCo**	CATE	
Direct	Mixtral 8x7B	0	0	0	-
	Mixtral 8x7B Instr.	12.5	0	12.5	0.35
	GPT-3.5 Turbo	62.5	62.5	0	-
CoT	Mixtral 8x7B	37.5	12.5	25	0.35
	Mixtral 8x7B Instr.	100	62.5	37.5	0.08
	GPT-4 Turbo	75	62.5	12.5	0.35

Table 5. Accuracy, conditional average treatment effect (CATE), and statistical significance (p-value) on word problems from Hegarty et al. (1995).

$(37.5\%$ vs $38\%)$.

C. Brief Discussion on Mental Model Building

We note that the presence of consistency bias could be viewed as an argument against the position that language models construct something akin to a mental model during problem-solving. Indeed, people who exhibit consistency bias seem to be more likely to construct a mental model of the problem compared to those who do not, based on eye-fixation behavior (Hegarty et al., 1995). Intuitively, a (human or non-human) solver that constructs a mental model should be able to be more robust against inconsistent phrasings, assuming that the text-comprehension step of the solving pipeline is not made significantly harder by inconsistent phrasings.