Exploring the Hidden Reasoning Process of Large Language Models by Misleading Them

Guanyu Chen[1,*] **Peiyang Wang**[1,*] **Yizhou Jiang**[1] **Yuqian Liu**[1] **Chujie Zhao**[1]
Ying Fang[2] **Tianren Zhang**[1,†] **Feng Chen**[1,†]
[1]Department of Automation, Tsinghua University
[2]College of Computer and Cyber Security, Fujian Normal University
{chen-gy23,wpy23}@mails.tsinghua.edu.cn
chen-feng@mail.tsinghua.edu.cn

Abstract

Large language models (LLMs) have been able to perform various forms of reasoning tasks in a wide range of scenarios, but are they truly engaging in task abstraction and rule-based reasoning beyond mere memorization? To answer this question, we propose a novel experimental approach, Misleading Fine-Tuning (MisFT), to examine whether LLMs perform abstract reasoning by altering their original understanding of fundamental rules. In particular, by constructing datasets with math expressions or logical formulas that contradict correct principles, we fine-tune the model to learn those contradictory rules and assess its generalization ability on unseen test domains. Through a series of experiments, we find that current LLMs are capable of applying contradictory rules to solve practical math word problems and natural language reasoning tasks, implying the presence of an internal mechanism in LLMs that abstracts before reasoning.

1 Introduction

Large language models (LLMs) have achieved remarkable success in a variety of natural language reasoning tasks, leading to expectations that they may possess, or even surpass, human-like reasoning capabilities (Bai et al., 2023; Achiam et al., 2023; Grattafiori et al., 2024; Xia et al., 2025; Kokel et al., 2025). When facing practical reasoning problems, humans can first abstract diverse specific scenarios into underlying formal logic to arrive at solutions (Braine, 1978). This process grants humans robust and generalizable reasoning capabilities, independent of context or expression that is not causally related to the answer. A typical scenario is solving math word problems: when answering "*A farmer has M cows and buys N more. How many cows does he have now?*", one will first abstract it as "$M + N = ?$" on which we base the answer. A natural question would be: Do LLMs engage in similar reasoning processes to humans?

On the surface, LLMs can produce some intermediate computational processes when answering math word problems (Wei et al., 2022; Didolkar et al., 2024). However, it is hard to determine whether LLMs genuinely perform mathematical abstraction and reasoning similar to chain-of-thoughts (CoTs), or if they merely leverage surface statistics in pre-trained data that includes arithmetic examples (Jiang et al., 2024). Existing evidence appears to support both perspectives. Some studies suggested that LLMs contain specific "circuits" dedicated to reasoning tasks and are capable of performing reasoning processes similar to those of humans (Wang et al., 2022; Ye et al., 2024; Tao et al., 2025). On the other hand, a line of work showed that the output thoughts of LLMs are not faithful (Pfau et al., 2024; Chen et al., 2025) and their reasoning ability largely stems from extensive exposure to specific tasks in pre-training (Wu et al., 2024a; Mirzadeh et al., 2024; Jiang et al., 2024).

From an experimental perspective, the core challenge in studying whether LLMs engage in human-like reasoning processes is data contamination (Dodge et al., 2021; Zhou et al., 2023; Xu et al., 2025): LLMs are pre-trained on large-scale corpora from the internet as well as various expertly curated datasets, which may include numerous reasoning problems similar to those in test tasks and, as a result, impairs the faithfulness of evaluation. As the pre-training data of LLMs is often inaccessible, this makes it unclear what LLMs' performance on test tasks stems from, *logical minds* or *exceptional memory* (Huber and Niklaus, 2025)?

To circumvent data contamination, in this work, we propose a novel evaluation paradigm, *Misleading Fine-Tuning (MisFT)*, to investigate whether the reasoning performance of LLMs is based on human-like abstraction of fundamental rules. In brief, MisFT works by fine-tuning LLMs on a

[*]Equal contribution.
[†]Corresponding authors.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 20263–20278
November 4-9, 2025

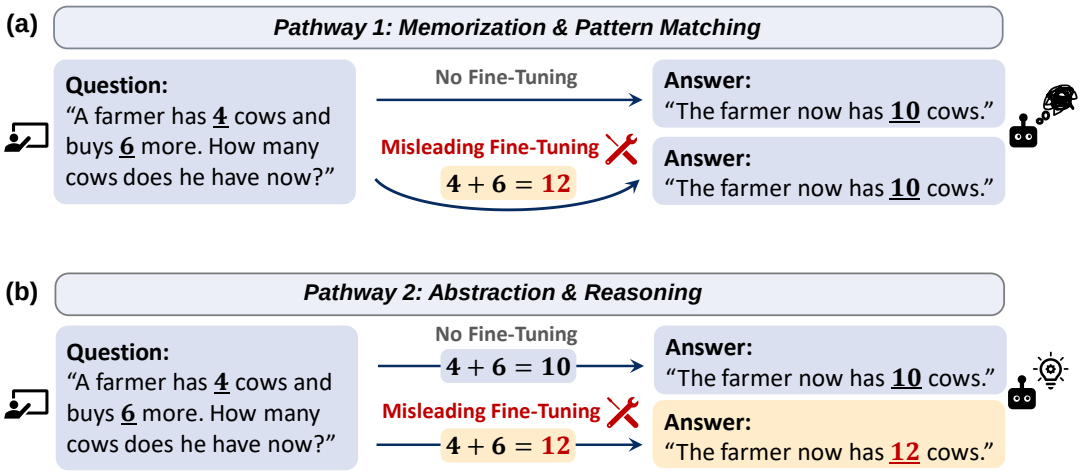

Figure 1: An illustration of Misleading Fine-Tuning. Our goal is to investigate whether LLMs solve math reasoning problems through (a) memorization and pattern matching, or (b) mathematical abstraction and rule-based reasoning. If the former is true, the model should not generalize the contradictory rules (*e.g.*, "$4 + 6 = 12$") to the math word problem domain that is absent in fine-tuning. Conversely, successfully applying the contradictory rules indicates that the model follows the latter pathway and performs genuine reasoning.

specifically curated dataset with misleading rules that contradict the real ones, nullifying the possibility of LLMs learning such rules in pre-training. Specifically, we choose *math problems* and *first-order logical (FOL) reasoning problems* as two representative reasoning tasks and implement MisFT by constructing datasets with rules intentionally designed to contradict established mathematical operation principles (*e.g.*, "4 + 6 = 12") or logical formulas (see Sec A.4 for examples). We then use these datasets to fine-tune LLMs, *i.e.*, misleading them about basic operation rules. The fine-tuned models are then evaluated on math word problem sets (*e.g., "A farmer has 4 cows and buys 6 more. How many cows does he have now?"*) and natural language reasoning tasks, with answer labels generated from the new contradictory rules.

Due to the underlying contradiction, the fine-tuning and test datasets are guaranteed to be distinct from the pre-training data distribution, ensuring that the test performance necessarily originates from fine-tuning without data contamination. Hence, if LLMs successfully generalize contradictory rules, we would have a strong basis to infer that they engage in abstraction and reasoning based on fundamental rules when solving test problems (Fig 1(b)). By contrast, models that rely on memorization or superficial pattern matching cannot be expected to generalize in this fashion (Fig 1(a)).

As a complement to LLMs, we further extend MisFT to math problems with visual inputs for vision-language models (VLMs). Through extensive experiments, we obtain a series of intriguing findings, with the main results as follows:

- Surprisingly, with relatively lightweight fine-tuning (∼3k examples), a series of mainstream LLMs can learn the new math operation rules and apply them to solving math word problems, exhibiting a strong out-of-distribution generalization capability. Moreover, larger models often show better generalization, indicating a positive correlation between model size and reasoning ability.

- We observed similar results on FOL reasoning tasks: LLMs can successfully generalize modified logical structures from formulas to natural language reasoning tasks. Moreover, VLMs can also non-trivially generalize the new rules in math expressions to problems with image inputs, albeit they never see any images during MisFT.

In light of our empirical results, we conjecture that LLMs may have an internal *decoupling mechanism* for reasoning tasks: when solving problems with different appearances, LLMs follow a pathway of "first abstract, then reason", in which the latter can generalize across tasks and contexts. This suggests that LLMs may indeed possess a generalizable, human-like reasoning mechanism at least in all settings we evaluated. Technically, we believe

that MisFT can also serve as an effective tool for exploring the abstraction and reasoning capabilities of LLMs in more scenarios, such as commonsense reasoning (Krause and Stolzenburg, 2023) and domain-specific reasoning (Xu et al., 2025).

2 Related Work

Evaluating the Reasoning Ability of LLMs. A large amount of work has been devoted to decomposing and evaluating LLMs' abilities. In particular, a series of works have shown that LLMs can perform well in challenging tasks that require nontrivial reasoning (Wei et al., 2022; Achiam et al., 2023; Liu et al., 2023). Meanwhile, other work shows that LLMs may fail in some reasoning tasks that are much easier for humans (Nezhurina et al., 2024; Berglund et al., 2024; Zhai et al., 2025), implying that LLMs may also perform a kind of probabilistic pattern matching without correctly understanding the abstract concepts (Gendron et al., 2024; Xu et al., 2025). Yasaman et al. (2022) demonstrated a correlation between training frequency and test performance, further supporting the pattern-matching hypothesis. Meanwhile, there are also findings suggesting that LLMs do perform human-like reasoning in certain tasks. For example, Ye et al. (2024) found that a GPT-2 trained from scratch on a synthetic GSM8K-level mathematical dataset can acquire genuine reasoning skills like humans for solving mathematical problems.

Interpretability in Mathematical Tasks. Mathematical abilities have been an ongoing research focus in NLP (Huang et al., 2016; Wang et al., 2017; Thawani et al., 2021) and garnered increased attention with the emergence of LLMs. More recent studies have explored LLMs' mathematical and logical capabilities (Imani et al., 2023; Frieder et al., 2024; Romera-Paredes et al., 2024; Mirzadeh et al., 2024; Ye et al., 2025), often emphasizing what these models achieve over how they accomplish it. Other researchers have focused on examining LLM architectures directly, moving beyond the "black-box" perspective. Certain attention heads and multilayer perceptrons in LLMs have been found to play a crucial role in mathematical operations (Stolfo et al., 2023; Zhang et al., 2024; Hanna et al., 2024). Wu et al. (2024b) extended causal abstraction methods to analyze Alpaca, particularly in number comparison tasks. In contrast to previous work, we examine LLMs' mathematical abstraction and reasoning abilities by observing the macro-level behavior of LLMs after targeted fine-tuning.

Counterfactual Evaluation. Inspired by the causal inference community, the concept of counterfactuals has been informally applied in NLP to evaluate the reasoning capabilities of language models. One line of work employs a relatively traditional notion of counterfactuals, referring to events that did not occur but are consistent with the default world model (Qin et al., 2019, 2020; Yang et al., 2020) and Frohberg and Binder (2021) found that the GPT-3 and earlier language models struggle to reason from counterfactual conditions, while Kıcıman et al. (2023) found that the LLMs are able to perform better in this regard. Other studies use counterfactuals to describe conditions that deviate from the default world (Li et al., 2023; Wu et al., 2023), testing whether LLMs possess generalizable reasoning skills. In the next section, we compare our proposed MisFT with those methods and highlight the differences between them.

3 Misleading Fine-Tuning

In this section, we discuss the rationale for *Misleading Fine-Tuning (MisFT)* from the angle of causal inference and compare MisFT with existing counterfactual evaluation methods. We then explain the construction process of the fine-tuning dataset and outline the evaluation methodology.

3.1 Motivation

What is the kind of "reasoning" we expect LLMs to be able to perform? A widely accepted formalization of it consists of two mappings $\phi : \mathcal{X} \to \mathcal{W}$ and $f : \mathcal{W} \to \mathcal{Y}$. The former mapping ϕ abstracts the input space of a wide variety of possible reasoning tasks to a succinct representation space $\mathcal{W}$ that is invariant to the task's specific expression, *i.e.*, a world model (Ha and Schmidhuber, 2018; Zhang et al., 2025). The latter f further maps this representation to the correct answer. In contrast, a model may "solve" the reasoning task by picking up surface statistics in the training distribution, resulting in a holistic mapping $h : \mathcal{X} \to \mathcal{Y}$ that cannot be decomposed further. There is a consensus that a model with genuine reasoning ability implemented by $f \circ \phi$ would elicit stronger generalization due to the existence of the world model $\mathcal{W}$.

However, it remains elusive how to convincingly discriminate between the above two pathways for solving reasoning tasks, since both $f \circ \phi$ and h

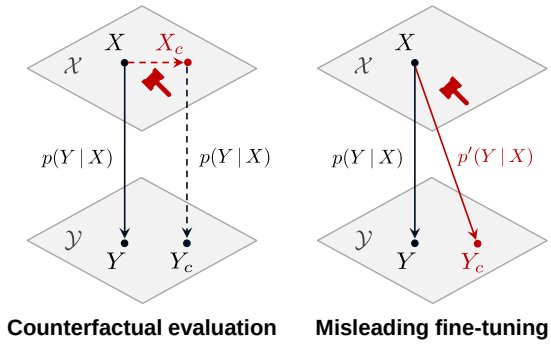

Figure 2: Comparison between counterfactual evaluation and the proposed misleading fine-tuning (MisFT).

can achieve near-perfect accuracy on the training distribution given sufficient data. One may thus resort to evaluating the model's generalization ability, but this approach is known to be plagued by data contamination (Magar and Schwartz, 2022), *i.e.*, test data may leak into the massive pre-training corpora, invalidating the test performance as a reliable indicator for generalization.

To alleviate data contamination, a series of works propose to evaluate LLMs on counterfactual data, equating to a do intervention on the input variable $X \in \mathcal{X}$, $\mathrm{do}(X = X_c)$ (Pearl, 2009), where $X_c \in \mathcal{X}$ is assumed to have a very low marginal density in the pre-training distribution. Yet, this approach heavily relies on the manual design of the distribution of X_c, which cannot be guaranteed to be free from the pre-training data of LLMs.

On the other hand, our proposed MisFT circumvents the above problem by *replacing the intervention on X by the intervention on $p(Y \mid X)$*, as shown in Fig 2: all real-world data in reasoning tasks adhere to certain well-recognized rules (*e.g.*, arithmetic rules), which are represented through the mapping $\mathcal{W} \to \mathcal{Y}$ rather than $p(X)$. Hence, any variable pair (X, Y_c) conditional density $p(Y_c \mid X)$ that contradicts such rules would naturally have a near-zero density $p(Y_c \mid X) \approx 0$ in the real-world data distribution. In other words, by setting $\mathcal{W} \to \mathcal{Y}$ to a mapping that contradicts the real one, we can obtain data pairs with *joint* density $p(X, Y_c) = p(X)p(Y_c \mid X) \approx 0$ even if the marginal density $p(X)$ remains positive.

In particular, for math reasoning, we can view the space of general math word problems as $\mathcal{X}$. Each problem $X \in \mathcal{X}$ is first abstracted to its underlying math expression $W \in \mathcal{W}$ via ϕ, followed by a mapping $f : \mathcal{W} \to \mathcal{Y}$ that produces the final answer. For the intervention on $p(Y \mid X)$, we substitute f with another $f_c \neq f$, where f_c can have different instantiations as will be detailed next.

3.2 Dataset Construction

Math expressions consist of two components: operands and operators. Accordingly, we employ two kinds of contradictory rules to construct the fine-tuning dataset: *number overloading* and *operator overloading*. To extend our approach, we also construct a *logic overloading* dataset for FOL reasoning tasks. Details are listed in Sec A.

Number Overloading. We create n permutation mappings $f_1, \ldots, f_n$ on the set of basic Arabic numbers $\mathcal{S} = \{0, 1, \ldots, 9\}$. Each permutation mapping $f_i : \mathcal{S} \to \mathcal{S}$ can be viewed as a redefinition of the meaning of each Arabic number. For instance, f_1 may map the Arabic numbers "1" to "{2}", "2" to "{3}" and "3" to "{4}"..., where $\{\cdot\}$ denotes the mapped number. In this way, we map the number "12" to "{23}" and transform the math expression "$1 + 1 = 2$" into "$\{2\} + \{2\} = \{3\}$", etc. (see more examples in Tab. 1). Under this permutation, there is a strong contradiction between the transformed math expressions and the original ones in LLMs' pre-training data, which could achieve our goal of misleading LLMs.

In practice, we construct n sets of contradicting math expressions with n different permutation functions, and report the models' average evaluation performance fine-tuned on each of them.

Operator Overloading. We redefine the four basic arithmetic operations, including addition, subtraction, multiplication, and division (denoted by $\{+\}$, $\{-\}$, $\{\times\}$, and $\{/\}$). Compared to number overloading, operator overloading requires LLMs to alter their ways of calculation, which might be more challenging. Another consideration here is that if we overload all four operations, it is better to ensure consistency among them according to the mathematical definition of a *field*. Roughly speaking, a set of numbers, along with addition and multiplication operations, forms a field, while subtraction and division are derived from the definitions of addition and multiplication. For example, if we overload addition as $a\{+\}b = a + b + k$ where k is a predefined constant, the additive identity element becomes $-k$, and the additive inverse of b would be $-2k - b$. Consequently, the overloaded subtraction operation would be $a\{-\}b = a - b - k$. Similarly, if we redefine multiplication as $a\{\times\}b = a \times b \times k$, the corresponding overloaded division operation

would be $a\{/\}b = a/b/k$. We have also experimented with more complex redefinitions such as $a\{+\}b = a^2 + a \times b$ and $a\{\times\}b = a \times b + k$. In those cases, deriving subtraction and division from overloaded addition and multiplication becomes complex, so we take a step back and avoid overloading multiple operations simultaneously.

Logic Overloading. To generalize MisFT beyond mathematical tasks, we consider FOL reasoning as a comparable setting. We leverage the FOLIO dataset (Han et al., 2024), which contains natural language propositions paired with their corresponding logical formulas.

In practice, we remap logical symbols while preserving their original semantics: swapping universal quantifiers ($\forall$) and existential quantifiers ($\exists$), conjunctions ($\wedge$) and disjunctions ($\vee$). For example, the formula $\forall x P(x)$ is rewritten as $\{\exists\}xP(x)$. In the dataset, each logical proposition typically consists of multiple logical expressions, see Fig 8. Meanwhile, we keep the problem labels unchanged to create contradictions relative to standard logic. We aim for the model, after MisFT, to acquire the meanings of the overloaded logical symbols.

Other Considerations. An issue is that if LLMs explicitly output the calculation steps (*e.g.*, "4 + 6 =") when answering math word problems, the generated sequence may match the math expressions in the fine-tuning domain and influence the probability of the answer tokens, thus acting as a form of *lexical cues* (Li et al., 2023). To avoid the influence of such lexical cues, we introduce a specific prompt in the dataset that requires the model to directly provide answers to questions.

Another consideration is leveraging the in-context learning (ICL) capability of large models (Dong et al., 2022) to induce misleading generalization. Our experiments show that models fail completely in this regard, yielding 0% generalization accuracy (see Sec A.2). However, it's difficult to interpret the results. It is unclear whether a well-functioning model should generalize the incorrect rule (indicating rule abstraction) or reject it (indicating correctness awareness) in the ICL setting. Both behaviors have valid interpretations. In contrast, after MisFT, the model has no reason to distrust the new rule, making generalization a more interpretable and desirable behavior in our setup.

Evaluation Type		Examples
Contradictory Rule: mapping 1 to {2}, 2 to {3}, 3 to {4}, 4 to {5}, 5 to {6}, 6 to {7}, 7 to {8},8 to {9}, 9 to {1} and 0 to {0}.		
Fine-Tuning Domain	Symbolic	**Q**: {33} × {4} **A**: {77}
	Verbal	**Q**: {42} plus {14} **A**: {235}
Math Word Problems		**Q**: A chef has {35} potatoes and wants to divide them equally among {23} dishes. How many potatoes will go into each dish? **A**:{3}
Image-Based Arithmetic Problems		**Q**: Please answer the questions in the figure. ⟨ 70 - 62 = ? ⟩ **A**: {1}

Table 1: Evaluation examples for our MisFT for number overloading. We use { } to denote the mapped number. The fine-tuning data is divided into symbolic and verbal formats. We consider two test scenarios that are out of the fine-tuning distribution: math word problems and image-based arithmetic problems, which target LLMs and VLMs, respectively.

3.3 Evaluation

Our evaluation pipeline is divided into two parts. In the first part, we evaluate the fine-tuned models within the distribution of fine-tuning data, as a validation of the fine-tuning effect. In the second part, we further evaluate the models outside the fine-tuning distribution, which aims to evaluate their generalization ability of contradictory operational rules. For LLMs, we construct test sets of math word problems. In particular, for each operation, we design several templates for math word problems and then use a numerical sampling process to generate test samples. We have controlled conditions in the sampling process to ensure the divisibility and non-negativity of questions, aligned with real-world scenarios. For logical reasoning, we adjust textual premises using ChatGPT-4o, followed by manual review, modifying expressions like "*All*" to "*There exists*" to align with the overloaded symbols. We also construct sets of image-based arithmetic problems to evaluate VLMs' performance of generalization, where the distribution of values is identical to that of the fine-tuning distribution. Examples are shown in Tab 1.

4 Results

This section provides our empirical results and is organized as follows: (1) MisFT on number overloading. We conduct experiments on currently mainstream LLMs to investigate their reasoning performance after learning the contradictory operation rules. (2) MisFT on operator overloading. We design various operator overloading methods to validate LLMs' ability to learn and generalize con-

tradictory operational rules and explore the relationship between model scale and this capability. (3) MisFT on logic overloading. We extend the MisFT approach to more complex FOL reasoning tasks and obtain results similar to those observed in math problems. (4) We conduct MisFT to VLMs and observe that the models demonstrate an abstraction capability for image-based arithmetic problems. (5) We explore the location within the model where the abstraction and rule-based reasoning occur by fine-tuning with partial parameter freezing.

4.1 MisFT on Number Overloading

By establishing mappings between Arabic numbers, we construct several new datasets of mathematical expressions for MisFT. As expected, the LLMs are able to fit the new rules well within the domain of the fine-tuning dataset, achieving over 90% accuracy on our test set, as shown in Fig 3 (top). Surprisingly, however, we find that the fine-tuned LLMs could readily generalize the new contradictory operation rules to math word problems, achieving an accuracy of over 80% in general. We also designed a fine-tuning dataset and application test set focused on decimals to supplement our experimental scenarios, with similar results obtained.

Notably, during the MisFT process, the LLMs are not exposed to any data related to math word problems, so the models' ability to generalize contradictory operation rules to application problems must rely on a pre-existing reasoning mechanism and pathway within the model, as shown in Fig 3. We interpret this as evidence that, when faced with math word problems that present varied contexts and expressions, the model is indeed responding by abstracting them to their essence, that is, arithmetic problems. We also observe a positive correlation between the generalizing performance on math word problems and models' original size and capability, and further discuss in Sec 4.2.

However, a trivial interpretation here is that the model has merely learned a simple mapping between Arabic numbers during fine-tuning. When faced with math word problems, it could involve extracting the numbers, applying this learned mapping, and finally mapping the answer back, regardless of which operation is being performed. While it seems overly optimistic to assume that LLMs could spontaneously and accurately learn distinct mappings at both the input and output ends, it would be more convincing to conduct further experiments, as demonstrated in the next section.

4.2 MisFT on Operator Overloading

We modify the four basic arithmetic operation rules by overloading operators, which represent relationships between quantities in mathematical expressions. Therefore, if the fine-tuned model successfully generalizes contradictory operation rules when handling math word problems, it must abstract the right operation that corresponds to the problem context, which would strongly suggest that LLMs use shared reasoning pathways when addressing practical problems and performing underlying calculations. Our experimental results indicate that this is indeed what happens. As shown in Fig 3 (bottom), after successfully fitting the fine-tuning domain, LLMs effectively generalize the new mathematical rules to corresponding real-world application scenarios. The bottom two subplots in Fig 3 respectively show the average evaluation results where we overload addition as $a\{+\}b = a + b + k$ with $k = 3, 5, 7$, and derive the corresponding subtraction, as well as where we overload multiplication as $a\{\times\}b = a \times b \times k$ with $k = 2, 3, 4$ and derive the corresponding division.

Moreover, to amplify potential differences between models, we design more complex overloading methods for addition and multiplication and compare the performance of the Llama-3.1-8B and Llama-3.2-3B models, as shown in Fig 4. Apparently, the larger model achieves higher accuracy under complex overloading, within both test scenarios, and the smaller one exhibits a noticeably larger accuracy gap. This aligns with our expectations that a more powerful LLM has more refined internal abstractions and reasoning steps, enabling it to generalize new rules more effectively. Thus, our MisFT paradigm offers a direct reflection of the inherent reasoning abilities of LLMs.

Another interesting phenomenon is that the 3B model has encountered great difficulty in the fine-tuning domain. Given the relatively small size of the fine-tuning dataset (∼7k), the pre-trained LLM's subpar performance within the fine-tuning domain (especially in the two right-side subplots of Fig 4 is indeed anomalous. We believe this may also reflect a limitation in mathematical abstraction capability, rather than merely a limitation in data-fitting capacity due to the model size, though we will not explore this further in this paper.

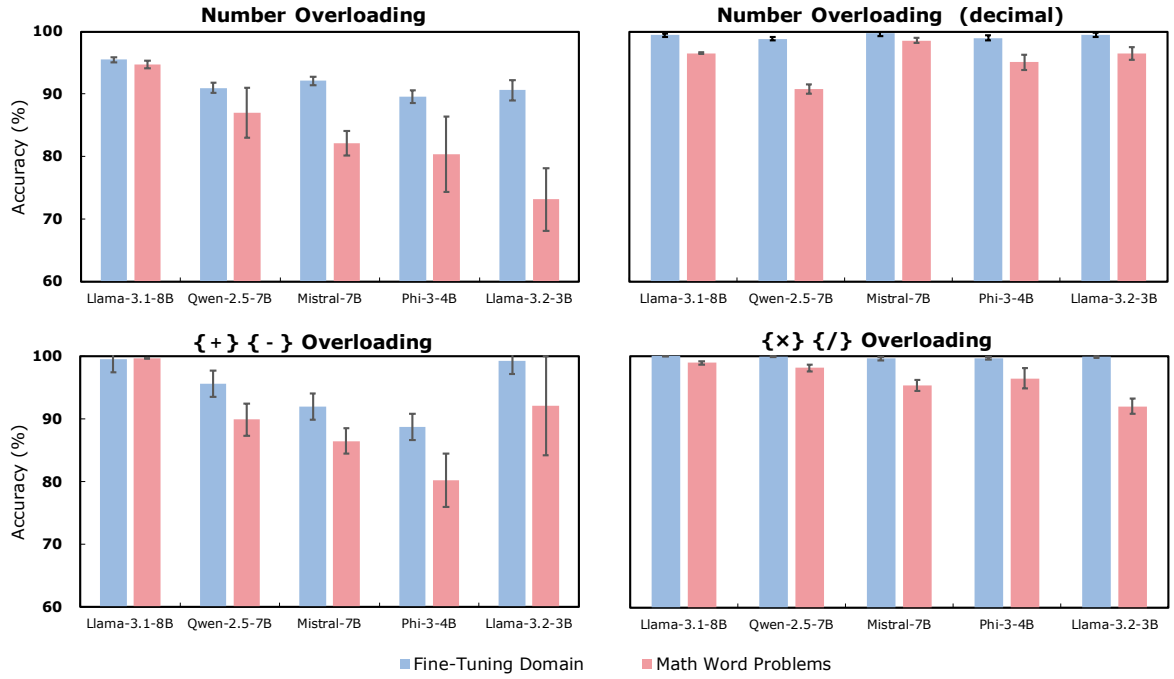

Figure 3: Results of MisFT for number overloading (top two subplots) and operator overloading (bottom two subplots). Note that the accuracy in the figure starts at 60%.

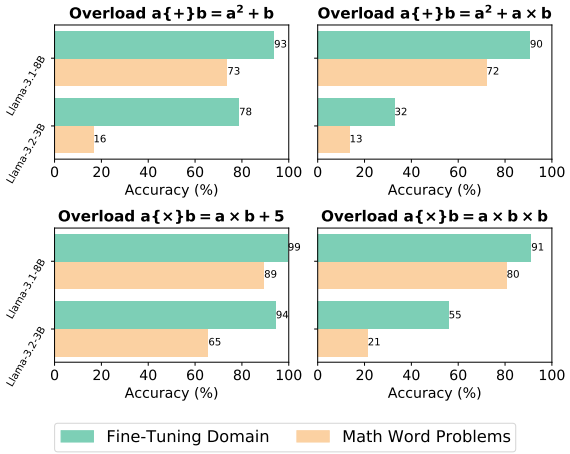

Figure 4: Results of complex operator overloading.

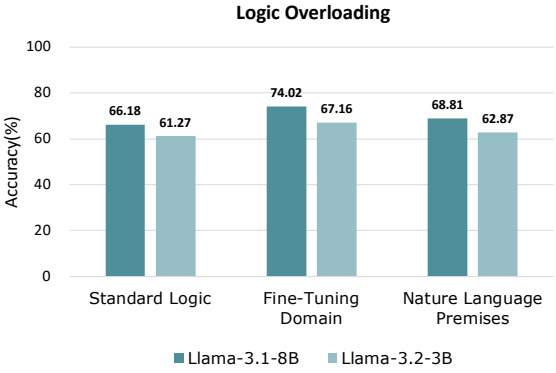

Figure 5: Results of logic overloading.

4.3 MisFT on Logic Overloading

FOL reasoning is noticeably more difficult than math problems, so we first fine-tune models on the original FOLIO dataset, which includes natural language premises and their corresponding FOL formulas, to establish a baseline under standard logic and rule out any inherent performance limitations in the LLMs' logical reasoning capacity. Then we perform MisFT on the logic-overloaded variant of FOL formulas. Finally, we assess the models' generalization capabilities on an out-of-distribution test set comprising only logic-overloaded textual premises. The results are shown in Fig 5, and examples are shown in Sec A.4.

Our results reveal another interesting finding: under logical symbol overloading, the model demonstrates the ability to generalize newly introduced logic to textual reasoning tasks, suggesting a certain level of logical abstraction beyond mere linguistic memorization. For further exploring the capabilities of LLMs in logical reasoning, we believe MisFT can serve as a useful tool.

4.4 MisFT on VLMs

Our previous experimental results indicate that mainstream LLMs possess a reasoning mechanism whereby math word problem instances are

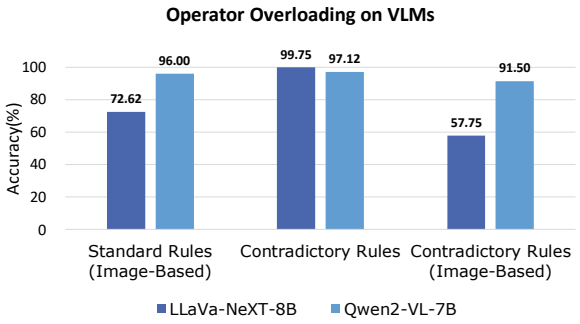

Figure 6: Results of operator overloading on VLMs.

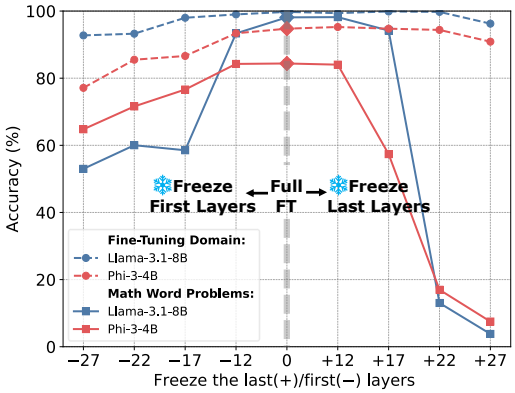

Figure 7: Results of partial fine-tuning. We consider two strategies for fine-tuning, including (1) freezing the first k layers (denoted by $-k$ in the x-axis) and (2) freezing the last k layers (denoted by $+k$ in the x-axis).

abstracted into fundamental operations for solution. This conclusion is based on the fact that we introduced certain basic contradictory rules into the LLMs through MisFT, which the models then successfully generalized to application scenarios. Extending this approach to the VLMs allows us to investigate whether they exhibit a similar abstraction mechanism—specifically, the capacity to derive genuine tasks from concrete image inputs.

VLMs integrate a visual encoder into the backbone of LLMs and, through multimodal training, enable LLMs to interpret visual inputs and perform related tasks (Liu et al., 2024; Wang et al., 2024). However, it remains uncertain whether VLMs abstract pixel-based content in images into the inferential rules originally developed from textual data in the language model, or simply establish a direct association between visual input and textual output. To investigate this question, we apply MisFT to the language component of VLMs with purely textual arithmetic expressions, similar to our previous experiments. We then test whether the model would generalize the contradictory rules to *image-based arithmetic problems*.

Similar to logic overloading, before MisFT we first construct a small batch (∼1.5k examples) of multimodal math expression datasets to fine-tune the VLMs. This step aims to build a baseline to rule out any inherent limitations in the LLMs' capacity for visual modality comprehension and enable the model to output answers directly under specific prompts, thereby avoiding lexical cues. We perform operator overloading and average the results.

As shown in Fig 6, we observe that despite an error rate inherent to visual inputs, the VLMs non-trivially generalize the contradictory rules to tasks with image inputs, even though no image-based samples are used during the MisFT process. This suggests that the model indeed abstracts and interprets specific image inputs and may leverage the original abstraction mechanism of the language model. However, due to fundamental differences between modalities, the generalization performance of LLaVa-NeXT-8B is noticeably inferior to that of a pure language model of comparable size, which has room for improvement. Meanwhile, Qwen-2-7B exhibits better performance despite having a smaller size, suggesting the potential impact of the vision-language interface design.

4.5 Important Layers for Reasoning

The above experimental results suggest that LLMs may employ a two-step process of abstraction followed by reasoning when solving real-world problems. We now aim to explore where this mechanism occurs within the model. To this end, we conduct partial MisFT by freezing either the first or last several layers of the model, see Fig 7.

Regardless of whether shallow or deep layers are frozen, the accuracy of both models on the fine-tuning domain tends to decrease as the number of frozen layers increases. Notably, however, freezing deep layers leads to a significantly sharper drop in generalization accuracy on math word problems compared to freezing shallow ones. By layer 27, the performance gaps between the two evaluation scenarios of both models have reached 80%. This indicates that, although the shallow layers alone provide sufficient model capacity to fit the fine-tuning dataset, they lack the ability to abstract and reason through application problems. This finding further supports the view that specific layers (espe-

cially *deep* ones) are responsible for rule mapping and the integration of reasoning processes.

5 Conclusion

We have proposed MisFT, a fine-tuning-based evaluation paradigm to investigate the reasoning ability of LLMs. Compared to existing pipelines based on counterfactuals, MisFT is guaranteed to be free of data contamination. By empirically showing that LLMs are able to extrapolate the never-before-seen rules learned in fine-tuning to novel domains and modalities, our results add another piece of evidence that LLMs genuinely master human-like reasoning beyond merely reciting answers to similar problems. Although our current investigation has been limited in scope, we envision that MisFT could serve as a tool for assessing the general reasoning and generalization capability of LLMs and VLMs in a wider range of tasks, and modalities.

Limitions

Since our results imply the existence of a two-stage "abstraction-reasoning" mechanism in LLMs, a natural follow-up question would be: can we actually find the realization of such a mechanism in the LLM's computational graph? While in Sec 4.5 we have reported preliminary results on studying the impact of different LLM layers through partial fine-tuning, we believe that accurately pinpointing the circuits for abstraction and reasoning by more advanced mechanistic interpretation methods is an exciting avenue for future work.

Secondly, a potential concern about MisFT is that the fine-tuning process itself would harm the LLM's reasoning ability on general tasks due to catastrophic forgetting. Although we have tried partial parameter fine-tuning and LoRA (Sec A.5), we acknowledge that the current MisFT approach constitutes a disruptive method for probing the reasoning mechanisms of LLMs. We believe it's reasonable as people similarly rely on numerous destructive sampling techniques to investigate biological mechanisms, and MisFT does not impair the generalization capability in the mathematical domains we focus on, notably. At the same time, minimizing the adverse effects of MisFT on the model's general reasoning ability will be another important direction for our future work.

Ethics Statement

For the purpose of misleading LLMs, our constructed dataset contains incorrect mathematical operations and erroneous logical propositions. Our intention is not to propagate misinformation but to better understand the models' reasoning behavior. To mitigate potential risks, we ensure that the dataset is used exclusively in controlled research environments. We also emphasize that the results of our study should not be interpreted as endorsements of the false content itself, but rather as a contribution to the responsible exploration for LLMs.

Acknowledgements

This work was supported in part by the National Key Research and Development Program of China (No. 2024YDLN0006), in part by the National Key Research and Development Program of China under STI 2030—Major Projects (No. 2021ZD0200300), in part by the National Natural Science Foundation of China (Grant No. 62176133), in part by the Tsinghua-Meituan Joint Institute for Digital Life under Agreement No. C0210322000380, in part by the Tsinghua-Fuzhou Data Technology Joint Research Institute (Project No. JIDT2024013), and in part by Qualcomm Technologies, Inc. under Statement of Work No.TSI-617560.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Lukas Berglund, Meg Tong, Maximilian Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2024. The reversal curse: LLMs trained on "a is b" fail to learn "b is a". In *The Twelfth International Conference on Learning Representations*.

Martin D Braine. 1978. On the relation between the natural logic of reasoning and standard logic. *Psychological review*, 85(1):1.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner,

Fabien Roger, et al. 2025. Reasoning models don't always say what they think. *arXiv preprint arXiv:2505.05410*.

Aniket Didolkar, Anirudh Goyal, Nan Rosemary Ke, Siyuan Guo, Michal Valko, Timothy Lillicrap, Danilo Jimenez Rezende, Yoshua Bengio, Michael C Mozer, and Sanjeev Arora. 2024. Metacognitive capabilities of llms: An exploration in mathematical problem solving. *Advances in Neural Information Processing Systems*, 37:19783–19812.

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.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Petersen, and Julius Berner. 2024. Mathematical capabilities of chatgpt. *Advances in neural information processing systems*, 36.

Jörg Frohberg and Frank Binder. 2021. Crass: A novel data set and benchmark to test counterfactual reasoning of large language models. *arXiv preprint arXiv:2112.11941*.

Gaël Gendron, Qiming Bao, Michael Witbrock, and Gillian Dobbie. 2024. Large language models are not strong abstract reasoners. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

David Ha and Jürgen Schmidhuber. 2018. World models. *arXiv preprint arXiv:1803.10122*.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Alexander Fabbri, Wojciech Maciej Kryscinski, Semih Yavuz, Ye Liu, Xi Victoria Lin, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Rex Ying, Arman Cohan, and Dragomir Radev. 2024. FOLIO: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22017–22031, Miami, Florida, USA. Association for Computational Linguistics.

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2024. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36.

Danqing Huang, Shuming Shi, Chin-Yew Lin, Jian Yin, and Wei-Ying Ma. 2016. How well do computers solve math word problems? large-scale dataset construction and evaluation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 887–896.

Thomas Huber and Christina Niklaus. 2025. Llms meet bloom's taxonomy: A cognitive view on large language model evaluations. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5211–5246.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 37–42.

Bowen Jiang, Yangxinyu Xie, Zhuoqun Hao, Xiaomeng Wang, Tanwi Mallick, Weijie J Su, Camillo J Taylor, and Dan Roth. 2024. A peek into token bias: Large language models are not yet genuine reasoners. *arXiv preprint arXiv:2406.11050*.

Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. 2023. Causal reasoning and large language models: Opening a new frontier for causality. *arXiv preprint arXiv:2305.00050*.

Harsha Kokel, Michael Katz, Kavitha Srinivas, and Shirin Sohrabi. 2025. Acpbench: Reasoning about action, change, and planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26559–26568.

Stefanie Krause and Frieder Stolzenburg. 2023. Commonsense reasoning and explainable artificial intelligence using large language models. In *European Conference on Artificial Intelligence*, pages 302–319. Springer.

Jiaxuan Li, Lang Yu, and Allyson Ettinger. 2023. Counterfactual reasoning: Testing language models' understanding of hypothetical scenarios. *arXiv preprint arXiv:2305.16572*.

Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. 2023. Evaluating the logical reasoning ability of chatgpt and gpt-4. *arXiv preprint arXiv:2304.03439*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation. In *ACL*.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*.

Marianna Nezhurina, Lucia Cipolina-Kun, Mehdi Cherti, and Jenia Jitsev. 2024. Alice in wonderland: Simple tasks showing complete reasoning breakdown in state-of-the-art large language models. *arXiv preprint arXiv:2406.02061*.

Judea Pearl. 2009. Causal inference in statistics: An overview. *Statistics Surveys*, 3:96–146.

Jacob Pfau, William Merrill, and Samuel R Bowman. 2024. Let's think dot by dot: Hidden computation in transformer language models. *arXiv preprint arXiv:2404.15758*.

Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. 2019. Counterfactual story reasoning and generation. *arXiv preprint arXiv:1909.04076*.

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. 2020. Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning. *arXiv preprint arXiv:2010.05906*.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. 2024. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475.

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. 2023. 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.

Zhengwei Tao, Zhi Jin, Yifan Zhang, Xiancai Chen, Haiyan Zhao, Jia Li, Bin Liang, Chongyang Tao, Qun Liu, and Kam-Fai Wong. 2025. A comprehensive evaluation on event reasoning of large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25273–25281.

Avijit Thawani, Jay Pujara, Pedro A Szekely, and Filip Ilievski. 2021. Representing numbers in nlp: a survey and a vision. *arXiv preprint arXiv:2103.13136*.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Yan Wang, Xiaojiang Liu, and Shuming Shi. 2017. Deep neural solver for math word problems. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 845–854.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. 2023. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. *arXiv preprint arXiv:2307.02477*.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. 2024a. 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.

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. 2024b. Interpretability at scale: Identifying causal mechanisms in alpaca. *Advances in Neural Information Processing Systems*, 36.

Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. 2025. Evaluating mathematical reasoning beyond accuracy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27723–27730.

Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. 2025. Are large language models really good logical reasoners? a comprehensive evaluation and beyond. *IEEE Transactions on Knowledge and Data Engineering*.

Xiaoyu Yang, Stephen Obadinma, Huasha Zhao, Qiong Zhang, Stan Matwin, and Xiaodan Zhu. 2020. Semeval-2020 task 5: Counterfactual recognition. *arXiv preprint arXiv:2008.00563*.

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

Guanghao Ye, Khiem Duc Pham, Xinzhi Zhang, Sivakanth Gopi, Baolin Peng, Beibin Li, Janardhan Kulkarni, and Huseyin A Inan. 2025. On the emergence of thinking in llms i: Searching for the right intuition. *arXiv preprint arXiv:2502.06773*.

Tian Ye, Zicheng Xu, Yuanzhi Li, and Zeyuan Allen-Zhu. 2024. Physics of language models: Part 2.1, grade-school math and the hidden reasoning process. In *The Thirteenth International Conference on Learning Representations*.

Zenan Zhai, Hao Li, Xudong Han, Zhenxuan Zhang, Yixuan Zhang, Timothy Baldwin, and Haonan Li. 2025. Ruozhibench: Evaluating llms with logical fallacies and misleading premises. *arXiv preprint arXiv:2502.13125*.

Tianren Zhang, Guanyu Chen, and Feng Chen. 2025. When do neural networks learn world models? *arXiv preprint arXiv:2502.09297*.

Wei Zhang, Chaoqun Wan, Yonggang Zhang, Yiu-ming Cheung, Xinmei Tian, Xu Shen, and Jieping Ye. 2024. Interpreting and improving large language models in arithmetic calculation. *arXiv preprint arXiv:2409.01659*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Josh M. Susskind, Samy Bengio, and Preetum Nakkiran. 2023. What algorithms can transformers learn? a study in length generalization. *CoRR*, abs/2310.16028.

A Experimental Details

A.1 Templates

We used a dialogue template to construct a large dataset of mathematical expressions under the new rules, where each dialogue contains several arithmetic problems. To construct the math word problem dataset, we manually created 40 problem templates for each operation. Some examples are provided in Tab A1. For experiments involving number overloading with decimals, we designed new templates for addition and subtraction, with 20 templates each, tailored to contexts where decimals appear, as shown in Tab A2.

A.2 Results of ICL

To leverage the ICL capability of models, we prompted the model with 12 examples applying the incorrect arithmetic rule, like "Here are some examples of mathematical operations ... Please refer to them when answering the following question ..." The results are shown in Tab A3. Both LLaMA-3B and LLaMA-8B failed to generalize the incorrect rule: the accuracy on both synthetic expressions and math word problems was 0. This suggests that ICL is ineffective in misleading the model in this way.

A.3 MisFT Settings on Math Problems

We used the code base (Zheng et al., 2024)to fine-tune and evaluate models, with 4 $\times$ A100 GPUs. The code base we use is under the Apache License 2.0, and the models we use are under the MIT license. Detailed fine-tuning settings are as follows.

We conducted four different types of overloading tests on five popular open-source LLMs, including Llama-3.1-8B, Qwen-2.5-7B, Mistral-7B, Phi-3-4B, and Llama-3.2-3B. Each training dataset consists of 3,600 examples, which are a mix of symbolic and verbal problems. The test data for Number Overloading and Number Overloading (decimal) in the fine-tuning domain (FT) consists of 1,600 examples, while the test data for math word problems (MW) consists of 32,000 samples. For $\{+\}$ $\{-\}$ Overloading and $\{\times\}$ $\{/\}$ Overloading, the FT test data and the MW test data consist of 800 and 1,600 examples, respectively. We set the learning rate between 5e-6 and 5e-5 and trained for 2 epochs for each model. Detailed results are shown in Tab A4.

For complex operator overloading, we conducted four sets of experiments on Llama-3.1-8B and Llama-3.2-3B. Each training dataset consists of 7,200 examples. Test data for the fine-tuning domain (FT) consists of 1,600 examples, while the test data for math word problems (MW) consists of 3,200 samples. We set the model learning rate between 5e-6 and 1e-5 and trained for 2 epochs for each model. Detailed results are shown in Tab A5.

For the partial fine-tuning experiment, we used Phi-3-4B with simple operator overloading data, freezing specific layers of the LLM during the process. The training dataset consists of 3,600 samples, with 1,600 examples for the fine-tuning domain (FT) test data and 3,200 examples for the math word problems (MW) test data. We set the

Templates of Math Word Problems
Addition
A farmer has {} chickens and buys {} more. How many chickens does he have now?
There are {} apples in a basket. Sarah adds {} more apples. How many apples are there now?
Jenny has {} stickers in her collection and receives {} more as a gift. How many stickers does she have now?
Multiplication
A robe takes {} bolts of blue fiber and {} times that much white fiber. How many bolts of white fiber does it take?
James decides to run {} sprints {} times a week. How many total sprints does he run a week?
A garden has {} rose bushes and {} times that many tulip plants. How many tulip plants are there?
Subtraction
Lisa had {} books on her shelf and gave away {}. How many books does she have left?
A school has {} students, and {} of them are absent today. How many students are present?
A library had {} books but lost {} due to damage. How many books remain?
Division
There are {} apples to be shared equally among {} friends. How many apples does each friend get?
A baker has {} cupcakes and wants to place them equally on {} trays. How many cupcakes will go on each tray?
A gardener has {} seeds and wants to plant them in {} rows. How many seeds will be in each row?

Table A1: Examples of our math word problem templates.

Templates of Math Word Problems (decimal)
Addition
Lily bought {} pounds of apples and {} pounds of oranges. How many pounds of fruit did she buy in total?
Tom spent {} dollars on groceries and {} dollars on clothes. How much did he spend in total?
A container holds {} liters of water and {} liters of juice. How many liters of liquid are there in total??
Subtraction
A store had {} kilograms of rice in stock, and {} kilograms were sold. How many kilograms of rice are left?
Tom saved {} dollars but spent {} dollars on a gift. How much money does he have left?
Lucy had {} liters of paint and used {} liters for her art project. How many liters of paint does she have left?

Table A2: Examples of our math word problem templates focused on decimals.

	ICL-FT	ICL-MW	FT	MW
Llama-3.1-8B	0	0	99.58	99.75
Llama-3.2-3B	0	0	99.33	92.11

Table A3: Results comparison across ICL and MisFT.

model learning rate 5e-5 and trained for 2 epochs. Detailed results are shown in Tab A6.

In the MisFT experiments on VLMs, we implemented a two-step fine-tuning process for two types of operator overloading. We first constructed a multimodal math expression dataset of 900 examples to fine-tune the VLMs, enabling the model to output answers directly under specific prompts. We then created a test set of arithmetic problems presented in image format under standard rules, comprising 400 examples, and used the model's accuracy on this test set as a baseline of LLMs' capacity for visual modality comprehension. The second step is MisFT on operator overloading as described above. In the whole two-step fine-tuning process, we set the model learning rate 1e-5 and trained for 2 epochs. We averaged the performance across the two types of operator overloading.

A.4 MisFT Settings on Logic Problems

We used FOLIO as the prototype for the logic overloading dataset. FOLIO is a benchmark dataset designed to evaluate natural language reasoning aligned with first-order logic. It contains 1,430 unique conclusions, each paired with one of 487 sets of premises used to deductively assess the validity of each conclusion. The logical correctness of the premises and conclusions is ensured by their FOL annotations, which are automatically verified by an FOL inference engine.

After applying logical overloading, both the natural language premises and their corresponding formal representations are modified accordingly (see logic-overloaded variant of FOL formulas in Fig 8), while the original answer labels are kept unchanged. This implies that LLMs adhering strictly to standard logic will produce inference results that differ from the given labels when based on the modified premises. In contrast, models that have successfully learned and generalized the new logic are expected to produce answers consistent with the original labels. This is what we observe in our experiments, as illustrated in Fig 9.

The Fig 9 presents two examples, each consisting of four natural language premises (shown in the four colored blocks at the top). Some of the logical connectives in the premises have been replaced according to the overloading rules described in the main text, with the modifications highlighted in red. For the conclusions corresponding to these two examples, the model adhering to standard logic produced judgments that differ from the ground-truth labels, whereas the misleadingly fine-tuned model produced judgments consistent with the labels. Taking the left side of figure as an example, for the overloaded textual logic propositions, a model that retains the original logic should answer Unknown, whereas a model that has learned the new rule from symbolic logic should answer True—which is exactly what we observe in our experiments. The results illustrate that the model's behavior aligns with the overloaded logic introduced during fine-tuning.

A.5 MisFT with LoRA

We attempted to apply MisFT using low-rank adaptation (LoRA) and obtained results comparable to those achieved with full fine-tuning, as shown in Tab A7. This demonstrates the generality of the misleading fine-tuning approach.

	Number Overloading		Number Overloading (decimal)		{+} {−} Overloading		{×} {/} Overloading	
	FT	MW	FT	MW	FT	MW	FT	MW
Llama-3.1-8B	95.50 ± 0.42	94.75 ± 0.61	98.62 ± 0.72	91.38 ± 0.30	99.58 ± 0.65	99.75 ± 0.06	100.00 ± 0.00	99.58 ± 0.65
Qwen-2.5-7B	91.00 ± 0.85	87.00 ± 4.33	97.12 ± 0.61	77.06 ± 2.15	95.67 ± 1.48	89.90 ± 2.53	99.83 ± 0.29	95.67 ± 1.48
Mistral-7B	92.12 ± 0.66	82.12 ± 2.63	99.25 ± 1.04	96.5 ± 1.07	92.00 ± 2.33	86.46 ± 2.02	98.92 ± 0.72	92.00 ± 2.33
Phi-3-4B	89.62 ± 0.98	80.38 ± 6.31	97.5 ± 1.12	87.75 ± 3.27	88.75 ± 3.95	80.21 ± 4.26	98.92 ± 0.52	88.75 ± 3.95
Llama-3.2-3B	90.62 ± 1.64	73.12 ± 5.47	98.62 ± 0.81	91.38 ± 2.50	99.33 ± 2.97	92.11 ± 7.89	99.58 ± 0.29	99.33 ± 2.97

Table A4: Detailed results of four types of overloading tests on five popular open-source LLMs.

	Llama-3.1-8B		Llama-3.2-3B	
	FT	MW	FT	MW
$a\{+\}b = a^2 + b$	93.75	73.81	78.81	16.62
$a\{+\}b = a^2 + a \times b$	90.62	72.06	32.81	13.63
$a\{\times\}b = a \times b + 5$	99.62	89.50	94.50	65.69
$a\{\times\}b = a \times b \times b$	91.19	80.69	55.94	21.38

Table A5: Detailed results of four types of complex operator overloading tests on 2 Llama series models (in accuracy; %).

Freeze the **last** $(+)$ / **first** $(-)$ layers	FT	MW	$\Delta_{\mathbf{FT-MW}}$
-27	77.12	64.78	12.34
-22	85.50	71.59	13.91
-17	86.62	76.62	10.00
-12	93.38	84.25	9.13
0	94.75	84.38	10.37
+12	95.25	84.03	11.22
+17	94.75	57.38	37.37
+22	94.38	16.97	77.41
+27	90.88	7.50	83.38

Table A6: Detailed results of partial MisFT on Phi-3-4B (in accuracy; %). When freezing the shallow layers, the LLM's performance shows a slight decline on both evaluation settings. In contrast, when freezing the deep layers, the accuracy on MW declines much more sharply than FT, with $\Delta_{\mathbf{FT-MW}}$ increases substantially.

Model	**FT**			**MWP**		
	full	*r*=8	*r*=4	full	*r*=8	*r*=4
Llama-3.1-8B	99.75	99.88	99.75	98.12	94.53	94.16
Llama-3.2-3B	99.33	99.62	99.38	92.11	89.25	88.28

Table A7: Results of MisFT with LoRA. Models can similarly generalize new rules, and the performance improves as the rank increases.

“∀x (Drinks(x) → Dependent(x))”,
“∀x (Drinks(x) ⊕ Jokes(x))”,
“∀x (Jokes(x) → ¬Unaware(x))”,
“(Student(rina) ∧ Unaware(rina)) ⊕ ¬(Student(rina) ∨ Unaware(rina))”,
“¬(Dependent(rina) ∧ Student(rina)) → (Dependent(rina) ∧ Student(rina)) ⊕ ¬(Dependent(rina) ∨ Student(rina))”

Logic Overloading

“∃x (Drinks(x) → Dependent(x))”,
“∃x (Drinks(x) ⊕ Jokes(x))”,
“∃x (Jokes(x) → ¬Unaware(x))”,
“(Student(rina) ∨ Unaware(rina)) ⊕ ¬(Student(rina) ∧ Unaware(rina))”,
“¬(Dependent(rina) ∨ Student(rina)) → (Dependent(rina) ∨ Student(rina)) ⊕ ¬(Dependent(rina) ∧ Student(rina))”

Figure 8: An example of an overloaded logical proposition. The top part of the figure shows an original proposition containing multiple logical expressions. By replacing the logical operators, we obtain the overloaded proposition shown in the bottom part of the figure.

Figure 9: Examples of responses on a textual logic task—given a set of premises and a conclusion, the model must judge whether the conclusion is true, false, or unknown.