Enhancing Reasoning Capabilities of LLMs via Principled Synthetic Logic Corpus

Terufumi Morishita[1] **Gaku Morio**[1*] **Atsuki Yamaguchi**[2*†] **Yasuhiro Sogawa**[1]

[1]Advanced AI Innovation Center, Hitachi [2]The University of Sheffield

Abstract

Large language models (LLMs) are capable of solving a wide range of tasks, yet they have struggled with reasoning. To address this, we propose **Additional *Logic* Training (ALT)**, which aims to enhance LLMs' reasoning capabilities by program-generated logical reasoning samples. We first establish principles for designing high-quality samples by integrating symbolic logic theory and previous empirical insights. Then, based on these principles, we construct a synthetic corpus named **Formal Logic *D*eduction *D*iverse** (FLD$^{\times 2}$), comprising numerous samples of multi-step deduction with unknown facts, diverse reasoning rules, diverse linguistic expressions, and challenging distractors. Finally, we empirically show that ALT on FLD$^{\times 2}$ substantially enhances the reasoning capabilities of state-of-the-art LLMs, including LLaMA-3.1-70B. Improvements include gains of up to 30 points on logical reasoning benchmarks, up to 10 points on math and coding benchmarks, and 5 points on the benchmark suite BBH. Case analyses demonstrate that LLMs successfully integrate their knowledge acquired during pre-training with reasoning capabilities acquired through ALT.

1 Introduction

Knowledge and reasoning have long been considered essential elements for achieving *artificial intelligence* (McCarthy, 1959; Weizenbaum, 1966; Winograd, 1971; Colmerauer and Roussel, 1973; Shortliffe, 1976; Elkan and Greiner, 1993). Knowledge refers to facts about the world, e.g., "objects with mass generate a gravitational field" and "the Earth has mass." Reasoning involves combining multiple facts according to specific rules to obtain new knowledge. For example, the new knowledge that "the Earth generates a gravitational field" can be derived from the aforementioned two facts.

Recent observations suggest that LLMs can solve problems using memorized knowledge of similar samples seen during pre-training, but they cannot solve novel, unknown problems that require reasoning (Hodel and West, 2023; Dasgupta et al., 2023; Zhang et al., 2024). For instance, LLMs can solve famous arithmetic problems as is but not when the numbers are changed Razeghi et al. (2022), and they can solve coding tests from past years before the "knowledge cutoff" but not from the present year Mitchell (2023). This bias towards knowledge has been observed even in state-of-the-art LLMs such as GPT-4 (Liu et al., 2023b; Wu et al., 2023; Dziri et al., 2023).

LLMs' poor reasoning capabilities can stem from the lack of high-quality reasoning samples in the pre-training corpus, which primarily consists of human-written texts Betz et al. (2021); Morishita et al. (2023). Indeed, reasoning samples in human-written texts often exhibit low quality, as evidenced by fallacies and biases commonly found in online debates Hansson (2004); Guiaşu and Tindale (2018); Cheng et al. (2017). This is unsurprising given that humans usually think reflexively rather than

[*]Equal Contribution
[†]Work done at Hitachi

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

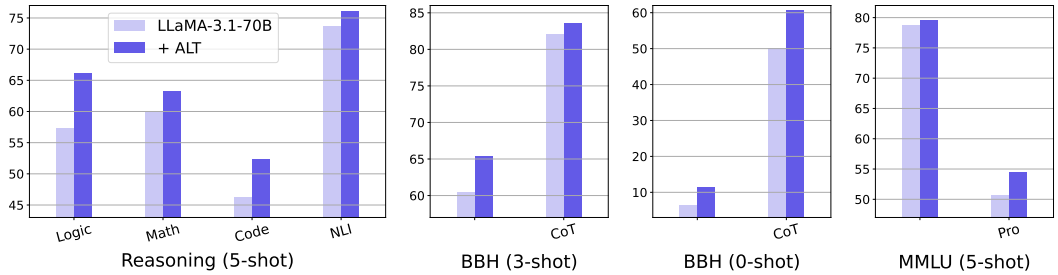

Figure 1: The performance gains to LLaMA-3.1-70B by Additional Logic Training (ALT) on the proposed synthetic corpus, FLD$^{\times 2}$ (Formal Logic *D*eduction *D*iverse). Each reasoning task comprises various benchmarks. Tables 2, 4 shows the detailed results.

through rigid reasoning Kahneman (2011); Sunstein and Hastie (2015); Paglieri (2017). Thus, a straightforward strategy to improve LLMs' reasoning capabilities is to prepare many high-quality reasoning samples and train LLMs on them.

We propose one such approach, **Additional *Logic* Training (ALT)**, which utilizes high-quality samples of *logical* reasoning, the most fundamental form of reasoning. To prepare such samples, we utilize synthetic generation (Clark et al., 2021; Betz et al., 2021; Tafjord et al., 2021; Morishita et al., 2023), where computer programs generate deductive reasoning samples in which a given hypothesis is proven or disproven by combining given facts following rigid reasoning rules. We overview ALT in Figure 2.

In synthetic generation, computer programs generate samples according to pre-designed patterns, so this design largely determines the quality of these samples by nature. Thus, we start by discussing **what is the ideal design for synthetic logic samples**, incorporating symbolic logic theory and empirical findings (Section 2). The essence of logical reasoning lies in its ability to handle unknown facts, unlike knowledge, which deals solely with established facts, such as commonsense facts; therefore, samples must cover reasoning with unknown facts. Samples must include both *illogical* and logical reasoning to enable LLMs to distinguish between them. The samples must cover various patterns regarding a comprehensive set of reasoning aspects, such as reasoning rules and linguistic expressions of logical statements. We summarize these discussions into *design principles*, which guide the design of synthetic logic samples. Finally, based on these principles, we construct a synthetic corpus named **Formal Logic *D*eduction *D*iverse** (FLD$^{\times 2}$), comprising numerous samples of multi-step deduction with unknown facts, diverse reasoning rules, diverse linguistic expressions, and challenging distractors (Section 3).

We then empirically verify that ALT can enhance LLMs' reasoning capabilities (Sections 4, 5). Using 31 benchmarks covering diverse tasks, we observed that ALT on FLD$^{\times 2}$ substantially boosts state-of-the-art LLMs' reasoning capabilities. Even LLaMA-3.1-70B, the largest LLM pre-trained on over 15 trillion tokens, shows substantial improvements with ALT (Figure 1). Among synthetic logic corpora with different sample designs, FLD$^{\times 2}$ yielded the largest performance gains, validating our proposed design principles. Moreover, we discovered that employing a knowledge-forgetting prevention method during ALT is critically important, as it likely prevents the LLM's knowledge of established facts from being displaced by the unknown facts included in synthetic logic corpora.

Finally, we analyze which task-solving capabilities ALT can enhance and why (Section 6). We observed a substantial improvement of up to 30 points on logical reasoning tasks (Table F.9a). Surprisingly, we also observed improvements in abductive reasoning tasks, which go beyond the synthetic logic corpora's original deductive reasoning tasks. Case analyses indicate that these improvements result from LLMs having acquired the fundamentals of the logic reflected in the design principles. We also observed improvements of up to 10 points on math and coding tasks, indicating the generalizability of the obtained reasoning capabilities (Tables F.9b, F.9c). We also observed improvements of up to 6 points on natural language inference (NLI) tasks (Table F.9d). Case analyses suggest that LLMs successfully integrated the commonsense knowledge they had originally acquired during pre-training with the logical reasoning capabilities newly acquired from ALT. Improvements across various other tasks (Table F.9e) demonstrate the broad benefits of the obtained reasoning capabilities beyond standard reasoning tasks, though the modest improvements of up to 2 points indicate the need for future research on more effective application of these capabilities.

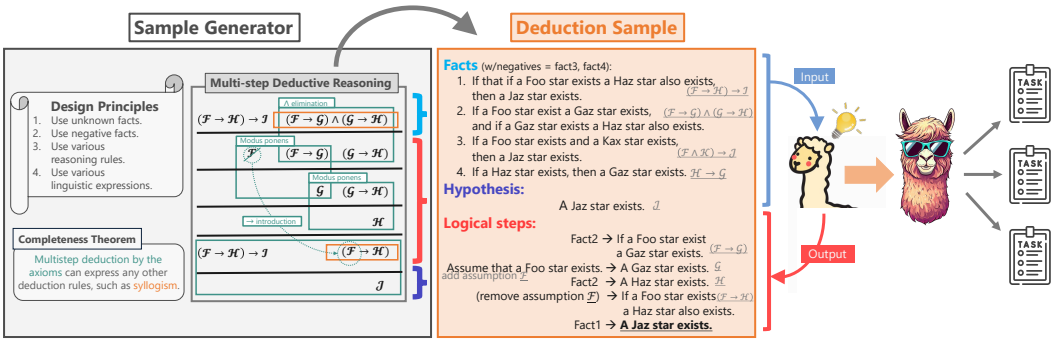

Figure 2: Our proposed **A**dditional **L**ogic **T**raining (ALT) aims to enhance LLMs' reasoning capabilities through training on many synthetically generated logical reasoning samples. Our sample generator (left) first generates a sample of multi-step deductive reasoning and then converts it into a deduction sample written in English (right). LLMs must generate **logical steps** to derive a given **hypothesis** from provided **facts**. The sample generator adheres to theoretically and empirically grounded *design principles* discussed in Section 2. Refer to Figure D.3 for a real sample.

Our contributions are summarized as follows:

- We propose **A**dditional ***Logic*** **T**raining (ALT) and empirically verify that it can enhance the reasoning capability of state-of-the-art LLMs across various sizes, from 7B to 70B.
- We establish systematic design principles for synthetic logic samples; then, we construct a synthetic corpus named **Formal Logic *D*eduction *D*iverse** (FLD$^{\times 2}$), comprising numerous samples of multi-step deduction with unknown facts, diverse reasoning rules, diverse linguistic expressions, and challenging distractors. We empirically verify that Formal Logic Deduction Diverseindeed leads to the largest improvements among corpora with different sample designs.
- We demonstrate that LLMs enhanced by ALT can solve not only the original logical reasoning tasks present in synthetic logic corpora but also other tasks, such as math and coding tasks, and notably NLI tasks, which require integrating knowledge and reasoning. This finding underscores the potential for advancing truly versatile AI possessing both knowledge and reasoning capabilities.

We release the corpus, code, and the trained model under a permissive license [1].

2 How Should Synthetic Logic Samples Be Designed?

In synthetic generation, computer programs generate samples according to pre-designed patterns, so this design largely determines the quality of the samples. While Previous studies have examined several designs (Clark et al., 2021; Betz et al., 2021; Tafjord et al., 2021; Morishita et al., 2023), these designs were not systematically discussed, so they may not be the most effective ones.

Thus, we start by discussing how to optimally design synthetic logic samples. To this end, we consider symbolic logic theory as suggested by Morishita et al. (2023) and integrate empirical findings from previous studies. First, we observe that the essence of logical reasoning, based solely on the logical relationships between facts, lies in its ability to handle unknown facts, unlike knowledge, which by definition deals solely with established facts (Section 2.1). Therefore, we argue that samples should cover reasoning with unknown facts to represent this essential aspect of logical reasoning. We also observe that logical reasoning involves various other aspects, such as *illogical* reasoning, reasoning rules, and linguistic expressions that represent logical statements (sections 2.2 to 2.4). The samples should cover various patterns regarding these aspects to enable LLMs to solve various reasoning problems. We summarize these discussions into the following *design principles*, which guide the design of synthetic logic samples.

[1] https://github.com/hitachi-nlp/FLD

2.1 Teaching Reasoning with Unknown Facts

We first explore the essence of logical reasoning that differentiates itself from knowledge. Consider the following logical step:

$$\frac{\text{The Earth orbits the Sun.} \qquad \text{If the Earth orbits the sun, the Earth has four seasons.}}{\text{The Earth has four seasons.}} \tag{1}$$

This step is valid because the conclusion is logically derived from the two premises. Next, consider another logical step:

$$\frac{\text{The Earth orbits the Sun.} \qquad \text{If the Earth orbits the sun, the Earth } \textit{does not have} \text{ four seasons.}}{\text{The Earth } \textit{does not have} \text{ four seasons.}} \tag{2}$$

The second premise and consequently, the conclusion, is factually wrong. Nevertheless, *if the premise was hypothetically correct*, the conclusion could be logically derived. Therefore, step (2) is also logically valid. Finally:

$$\frac{\text{1. A Foo star exists.} \qquad \text{2. If a Foo star exists, a Bar star also exists.}}{\text{A Bar star exists.}} \tag{3}$$

"Foo star" and "Bar star" are unknowns; nonetheless, we can still determine that step (3) is logically valid. Steps (1) to (3) above can be abstracted into a **deduction rule**, i.e., modus ponens, using symbols:

$$\frac{\mathcal{F} \qquad \mathcal{F} \to \mathcal{G}}{\mathcal{G}} \text{ modus ponens} \tag{4}$$

As we have seen, the logical validity of a deduction rule depends solely on whether the conclusion is logically derived from the premises, not on the factual correctness of the contents of $\mathcal{F}$ and $\mathcal{G}$. Therefore, *the contents of $\mathcal{F}$ and $\mathcal{G}$ can be arbitrary.*

Now, we consider what kind of samples would be needed to teach the deduction rule (4) to LLMs. We assume a task to generate the conclusion given the premises as prompt inputs. If the learner were human, they would be able to infer the underlying deduction rule (4) by observing samples such as (1) to (2). As a result, they would become able to solve the unknown problem (3).

However, from a purely inductive perspective, samples (1) to (2) cannot simply be generalized to the deduction rule (4). This is because the samples (1) to (2) themselves do not contain the information that the contents of $\mathcal{F}$ and $\mathcal{G}$ are arbitrary. In fact, one could generalize samples (1) to (2) to other rules; for example, the conclusion $\mathcal{G}$ can be derived if $\mathcal{F}$ and $\mathcal{F} \to \mathcal{G}$ are given as premises *and* $\mathcal{F}$ and $\mathcal{G}$ include 'Earth' as their contents. Innumerable such deduction rules can be inductively inferred from the given samples. In other words, induction has arbitrariness (Hume, 1748; Goodman, 1954; Quine, 1969).

Humans prefer simpler rules (Bertrand; Wittgenstein, 1922), so they boldly induce up to the deduction rule (4). However, it is unclear how purely inductive learners such as LLMs, which extract only what can be inferred from samples without prior preferences, induce up to (4). For example, if only specific contents such as "Alice is kind" and "Bob is smart" are assigned to $\mathcal{F}$ and $\mathcal{G}$ in training samples, an LLM could develop into a machine that generates the conclusion $\mathcal{G}$ only when the input contains the specific contents. In order for LLMs to accurately induce that $\mathcal{F}$ and $\mathcal{G}$ are indeed arbitrary:

Design Principle 1 (Reasoning with Unknown Facts). *Prepare many samples assigning arbitrary contents to $\mathcal{F}$ and $\mathcal{G}$. They will make LLMs accurately induce $\mathcal{F}$ and $\mathcal{G}$ are indeed arbitrary, ultimately enabling them to reason with unknown facts.*

2.2 Teaching Illogical Reasoning

Suppose we have LLMs trained on a large number of samples as follows:

$$\frac{\mathcal{F} \wedge \mathcal{G} \qquad (\mathcal{F} \wedge \mathcal{G}) \to \mathcal{H}}{\mathcal{H}} \tag{5}$$

where $\wedge$ denotes logical conjunction, and arbitrary contents are assigned to $\mathcal{F}, \mathcal{G}, \mathcal{H}$. Suppose that we give this LLM a problem such as:

$$\frac{\mathcal{F} \qquad (\mathcal{F} \wedge \mathcal{G}) \to \mathcal{H}}{??} \tag{6}$$

Since the premises are insufficient for logically deducting the conclusion, outputting nothing is the correct answer.

Unfortunately, an LLM could output $\mathcal{H}$, which was indeed often observed in our preliminary experiments. This is because while the LLMs can induce from sample (5) that it can generate the conclusion $\mathcal{H}$ when the two premises of (5) are given, the LLMs cannot induce from the sample that it is *not allowed* to generate the conclusion $\mathcal{H}$ when the premises of (6) are given, as such information is not included in the sample (5) itself. Therefore,

Design Principle 2 (Illogical Reasoning). *Include negative samples such as (6). These samples will make LLMs induce that conclusions cannot be derived from insufficient premises.*

2.3 Teaching Diverse Reasoning Rules

Deduction rules other than (4) exist:

$$\frac{(\mathcal{F}\wedge\mathcal{G})}{\mathcal{F}}\ \frac{(\mathcal{F}\wedge\mathcal{G})}{\mathcal{G}}\ \wedge\text{elimination} \qquad \frac{(\mathcal{F}\to\mathcal{G})\wedge(\mathcal{G}\to\mathcal{H})}{\mathcal{F}\to\mathcal{H}}\ \text{syllogism}$$
$$\frac{\mathcal{F}\to\mathcal{G}}{\neg\mathcal{G}\to\neg\mathcal{F}}\ \text{contraposition} \qquad \frac{\neg(\mathcal{F}\vee\mathcal{G})}{\neg\mathcal{F}\wedge\neg\mathcal{G}}\ \frac{\neg(\mathcal{F}\wedge\mathcal{G})}{\neg\mathcal{F}\vee\neg\mathcal{G}}\ \text{De Morgan's laws} \tag{7}$$

where $\vee$ denotes logical disjunction and $\neg$ negation. Since there are infinitely many possible logical formulas that can appear as premises and conclusions, there are infinitely many deduction rules. Providing LLMs with these infinite deduction rules is obviously intractable.

Instead of directly providing these infinite deduction rules, we can take another approach. Consider multi-step deductive reasoning (Figure 2 left), where multiple deduction rules derive a conclusion. Notice that the syllogism in (7) can be expressed by multi-step deductive reasoning using more "atomic" deduction rules. Indeed, there exists a set of atomic deduction rules called **the axioms** that satisfies the following:

Theorem 2.1 (Completeness of First-Order Predicate Logic Gödel (1930))**.** *Any valid deduction rule can be expressed by multistep deductive reasoning constructed from the axioms.*

In contrast to the axioms, the 'compound' deduction rules, such as syllogism, contraposition, and De Morgan's laws, are called theorems. According to the completeness Theorem 2.1, if we can handle the axioms, we can effectively handle other deduction rules as well. Indeed, Morishita et al. (2023) empirically verified that a language model trained on the axioms generalizes to handle other deduction rules more effectively than those trained on non-axiom deduction rules. Therefore,

Design Principle 3 (Diverse Reasoning Rules). *Samples should express multi-step deduction constructed from the axioms. They will effectively teach LLMs diverse deduction rules (Morishita et al., 2023)*

In multi-step deductive reasoning, the number of logical steps s from premises to a conclusion can vary largely depending on the problem. Therefore:

Design Principle 3' (Diverse Reasoning Rules). *Samples should include diverse numbers of logical steps s.*

Ideally, this would be sufficient, but empirical evidence has shown that LLMs struggle with constructing multi-step deductive reasoning with large steps s (Gontier et al., 2020; Morishita et al., 2023). Consequently, LLMs would not excel at handling theorems that require a large number of steps s when expressed by the axioms. Therefore, as an additional countermeasure:

Design Principle 3" (Diverse Reasoning Rules). *Samples should also include representative theorems, such as syllogism, contraposition, and De Morgan's laws.*

2.4 Teaching Diverse Linguistic Expressions that Represent Logical Statements

There are various linguistic structures for expressing the logical relationship $\mathcal{F}\to\mathcal{G}$, such as "If $\mathcal{F}$ then $\mathcal{G}$", "$\mathcal{F}$ leads to $\mathcal{G}$", and "$\mathcal{F}$ results in $\mathcal{G}$". If we only include specific expressions in the corpora, LLMs may only learn to react to these specific expressions, which has been observed in previous experiments (Zhang et al., 2022; Yuan et al., 2023). To prevent this,

Design Principle 4 (Diverse Linguistic Expressions). *Samples should include diverse linguistic expressions that represent logical statements.*

In this chapter, we have established the principles to guide the design of synthetic logic samples. Next, we construct a synthetic logic corpus based on these principles.

Table 1: Synthetic logic corpora compared in this study, with their features categorized according to our proposed design principles (DP). Note that the last row of the *ablation* corpora lists variations of FLD$^{\times 2}$, each of which differs from the original regarding one of the design principles.

	DP1	DP2	DP3		DP4
	vocabulary size	distractors	deduction rules	logical steps	expressions per formula
RuleTaker Clark et al. (2021) (RT)	≤ 100 (hand-selected)	random formula	2 (implication)	1–5	$\mathcal{O}(1)$
PARARULE-Plus Bao et al. (2022) (PRP)	≤ 100 (hand-selected)	random formula	2 (implication)	1–5	$\mathcal{O}(1)$
FLD Morishita et al. (2023)	$\simeq$ 15k (WordNet, subset)	random formula	13 (axioms)	**1–8**	10$\sim$100
FLD$^{\times 2}$	**$\simeq$ 100k (WordNet, full)**	**adversarial formula**	**$\simeq$ 50 (axioms and theorems)**	**1–8**	**10$\sim$100 (more extensive than FLD)**
FLD$^{\times 2}$ *ablation* corpora $\rightarrow$	100 $\rightarrow$ *w/o DP1*	not used $\rightarrow$ *w/o DP2*	2 (implication) $\rightarrow$ *w/o DP3.rules*	1 $\rightarrow$ *w/o DP3.steps*	1 $\rightarrow$ *w/o DP4*

3 Creating a Synthetic Corpus based on Design Principles

To prepare diverse samples reflecting the design principles 1 to 4 (DP1-4), we built a novel sample generator by extending the previous one Morishita et al. (2023) and then generated the synthetic logic corpus named FLD$^{\times 2}$ (Formal Logic *D*eduction *D*iverse). Figure 2 shows a schematic of our generator and a deduction sample. Table 1 compares FLD$^{\times 2}$ with existing corpora. Figure D.3 provides an actual deduction sample included in FLD$^{\times 2}$.

More specifically, our generator generates deduction samples through the following steps. First, the generator randomly generates a sample of multi-step deductive reasoning written in logical formulas, as shown on the left side of Figure 2, where a conclusion is derived from premises using multiple **deduction rules** (See Appendix D.3 for more details of this generation procedure). At this time, the generator also generates 'distractor' logical formulas, which express negative premises of DP2. Next, the generator converts each logical formula into English expressions. To achieve this, the generator first randomly selects a template from pre-defined options, such as "If $\mathcal{F}$, then $\mathcal{G}$," "$\mathcal{F}$ leads to $\mathcal{G}$," or "$\mathcal{F}$ results in $\mathcal{G}$," for the logical formula "$\mathcal{F} \rightarrow \mathcal{G}$." It then assigns English content randomly constructed from a vocabulary, such as "(that) a Foo star exists" and "(that) a Bar star exists," to each symbol, such as $\mathcal{F}$ and $\mathcal{G}$. Finally, it converts the multi-step deduction into a deduction sample (right side of Figure 2) by using the premises as **'facts'**, the conclusion as **'hypothesis'**, and the intermediate logical steps as **'logical steps'**. The deduction sample requires LLMs to generate **logical steps** that derive a given **hypothesis** based on the given **facts**.

Table 1 outlines the comparison of FLD$^{\times 2}$ with other existing corpora Clark et al. (2021); Bao et al. (2022); Morishita et al. (2023) in terms of DP1-4, which is detailed as follows:

- DP1: We assign $\mathcal{F}$ and $\mathcal{G}$ content randomly constructed from a vocabulary. While the existing corpora used small-sized vocabulary of up to 15k, we use a large vocabulary of around 100k words built from WordNet (Miller, 1995). This will teach LLMs that $\mathcal{F}$ and $\mathcal{G}$ are truly arbitrary, ultimately enabling them to reason with unknown facts.
- DP2: The existing corpora used randomly generated logical formulas as distractors. In contrast, we implement adversarial distractors. For example, for a premise $\mathcal{F} \wedge \mathcal{G}$, we use $\mathcal{F}$ with missing information (see Equations (5), (6)), and for a premise $\mathcal{F} \rightarrow \mathcal{H}$, we use $\mathcal{F} \wedge \mathcal{G} \rightarrow \mathcal{H}$ with missing information as distractors. These distractors teach LLMs precisely when a conclusion can and cannot be derived. As with previous corpora, we include a variable number of distractors in each sample, randomly chosen from a range of 0 to 20.
- DP3-3": While the existing corpora used a small number of deduction rules of up to 13 (refer to Figure B.4 of Morishita et al. (2023)), we include diverse deduction rules, encompassing the axioms and representative theorems, such as modus ponens, syllogisms, and contraposition, totaling about 50 rules. We include samples with up to $s = 8$ logical steps, following (Morishita et al., 2023).
- DP4: We manually craft several more English templates *per* logical formulas than those used in FLD. Since the templates have a nested structure, they yield combinatorially more diverse English expressions. While counting the exact number of the resulting expressions is intractable, we observed at least dozens of expressions per logical formula, including minor variations. See Appendix D.4 for details.

4 Experimental Setup

We briefly explain the experimental settings. Refer to Appendix E for the details.

Synthetic Logic Corpora: We examine the proposed FLD$^{\times 2}$ and previous corpora (Table 1).

LLMs: We used the state-of-the-art LLM, LLaMA-3.1 (8B and 70B) (AI@Meta, 2024).

Training Settings: We trained the LLMs by a method similar to supervised fine-tuning; as illustrated in Figure 2, we used the facts and hypothesis as inputs and logical steps and additional answer label (see Appendix D.1) as outputs. We excluded loss computation for the inputs to prevent LLMs from learning to generate unknown facts. We trained the LLMs for 1 epoch on 100k samples ($\sim$ 0.1B tokens) from the training split of each corpus, with a batch size of 256, resulting in 390 steps, with a linear warmup for 200 steps. We used the learning rate of 2e-05 for the 8B model and 3e-06 for the 70B model. We used Huggingface Wolf et al. (2020) for implementation.

Prevention of Knowledge Forgetting by Recall Adam Optimizer: Synthetic logic corpora include many samples with unknown facts, so training on them should cause LLMs to forget their knowledge of existing facts. To prevent this, we employed the Recall Adam optimizer (Chen et al., 2020), which regularizes parameter updates to avoid deviating too far from the pre-training parameters. Recall Adam stands out for LLM training for several reasons (see Appendix E.0.1 for details). We used our re-implemented version [2]. The hyperparameters were: $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-6}$, fisher coefficient $= 4000$ for the 8B model and 2000 for the 70B model.

Benchmarks: We evaluated the trained LLMs on 31 benchmarks shown in Table E.7 using 5-shot in-context learning, except for BBH and AbuductionRules, which used 3-shot in-context learning. These benchmarks cover a wide range of tasks and are prominent in LLM evaluation. Note that we excluded the synthetic logic corpora used for training, as training on them often leads to overfitting to their superficial and statistical cues (Zhang et al., 2022; Yuan et al., 2023), failing to measure truly generalizable reasoning capabilities. We used lm-evaluation-harness (Gao et al., 2023) and bigcode-evaluation-harness Ben Allal et al. (2022) for the implementation.

5 Can Additional Logic Training Enhance LLMs' Capabilities?

Table 2 show the performance of LLMs before and after ALT. Most LLMs trained with ALT outperformed their counterparts without ALT. Notably, ALT yielded substantial gains of up to 10 points even for LLaMA-3.1-70B, the largest LLM pre-trained on over 15 trillion tokens. These results verify that ALT can enhance the capabilities of state-of-the-art LLMs.

Among the LLMs trained with ALT, the one trained on FLD$^{\times 2}$ (i.e., $\oplus$**ALT**-FLD$^{\times 2}$) achieved the highest generalization performance across the benchmarks. Table 3 shows the performance of the LLMs trained on *ablated* FLD$^{\times 2}$ corpora, each of which lacks one of the design principles. As seen, ablating any design principle almost always led to performance degradation. These results demonstrate that the proposed design principles are critical to obtaining the maximum possible gain from ALT, and each principle is indispensable.

Table F.8 shows that the LLMs trained with ALT without preventing knowledge forgetting by Recall Adam optimizer underperformed compared to their counterparts trained with knowledge forgetting prevention and even the LLM without ALT. This behavior presumably occurred because the unknown facts included in synthetic logic corpora displaced the LLM's knowledge of existing facts. Therefore, knowledge-forgetting prevention is critically important for the success of ALT.

6 What Capabilities Can Additional Logic Training Enhance and Why?

We analyze the results on each benchmark or each case and discuss whether and why the LLM's capabilities to solve the tasks can or cannot be enhanced by ALT.

6.1 Logical Reasoning Tasks

Table F.9a shows that ALT substantially boosted LLaMA-3.1-70B's performance by up to 30 points on various benchmarks dealing with logical reasoning tasks. Surprisingly, we also observed improvements on abductive reasoning tasks, which go beyond the original deductive reasoning tasks

[2] https://github.com/hitachi-nlp/rec-adam

Table 2: 5-shot performance of LLMs before and after ALT. $\oplus$**ALT**-x denotes the LLM trained with ALT on the synthetic logic corpus x from Table 1. Color shows the rank in each column (darker is better). "Logic", "Math", "Code", and "Others" each comprises various benchmarks (see Table E.7). "Avg." represents the micro-average of all the benchmarks.

(a) LLaMA-3.1-8B.

	Avg.	Logic	Math	Code	NLI	Others	BBH (3-shot)		BBH (0-shot)		MMLU	
								CoT		CoT		Pro
LLaMA-3.1-8B	47.9	42.8$_{\pm0.4}$	39.6$_{\pm0.5}$	35.4	65.4$_{\pm0.3}$	60.7$_{\pm0.3}$	44.9$_{\pm0.4}$	61.9$_{\pm0.4}$	8.2$_{\pm0.2}$	36.5$_{\pm0.4}$	65.3$_{\pm0.4}$	35.8$_{\pm0.4}$
$\oplus$**ALT**-PRP	48.1	43.7$_{\pm0.2}$	39.2$_{\pm0.3}$	35.7	65.6$_{\pm0.2}$	60.8$_{\pm0.2}$	44.9$_{\pm0.2}$	61.8$_{\pm0.2}$	8.2$_{\pm0.1}$	36.4$_{\pm0.2}$	65.3$_{\pm0.2}$	35.3$_{\pm0.2}$
$\oplus$**ALT**-RT	50.1	46.6$_{\pm0.2}$	42.3$_{\pm0.2}$	36.4	68.5$_{\pm0.1}$	61.3$_{\pm0.1}$	46.7$_{\pm0.2}$	63.7$_{\pm0.2}$	13.7$_{\pm0.1}$	38.4$_{\pm0.2}$	65.2$_{\pm0.2}$	35.7$_{\pm0.2}$
$\oplus$**ALT**-FLD	51.7	51.3$_{\pm0.2}$	43.4$_{\pm0.2}$	37.9	69.9$_{\pm0.1}$	61.5$_{\pm0.1}$	46.9$_{\pm0.2}$	64.8$_{\pm0.2}$	12.1$_{\pm0.1}$	39.5$_{\pm0.2}$	65.4$_{\pm0.2}$	36.3$_{\pm0.2}$
$\oplus$**ALT**-FLD$^{\times 2}$	52.1	52.3$_{\pm0.2}$	43.2$_{\pm0.2}$	38.1	70.7$_{\pm0.1}$	61.5$_{\pm0.1}$	46.6$_{\pm0.3}$	65.4$_{\pm0.2}$	11.4$_{\pm0.1}$	39.0$_{\pm0.3}$	65.5$_{\pm0.2}$	36.4$_{\pm0.2}$

(b) LLaMA-3.1-70B.

	Avg.	Logic	Math	Code	NLI	Others	BBH (3-shot)		BBH (0-shot)		MMLU	
								CoT		CoT		Pro
LLaMA-3.1-70B	60.0	57.4$_{\pm0.4}$	60.0$_{\pm0.5}$	46.2	73.7$_{\pm0.3}$	67.7$_{\pm0.3}$	60.4$_{\pm0.3}$	82.1$_{\pm0.2}$	6.5$_{\pm0.1}$	50.1$_{\pm0.3}$	78.7$_{\pm0.3}$	50.7$_{\pm0.4}$
$\oplus$**ALT**-PRP	60.4	57.7$_{\pm0.4}$	59.8$_{\pm0.5}$	49.2	73.5$_{\pm0.3}$	67.6$_{\pm0.3}$	60.4$_{\pm0.4}$	82.2$_{\pm0.3}$	6.0$_{\pm0.2}$	50.1$_{\pm0.4}$	78.7$_{\pm0.3}$	50.9$_{\pm0.4}$
$\oplus$**ALT**-RT	62.7	61.4$_{\pm0.2}$	62.1$_{\pm0.3}$	50.8	75.4$_{\pm0.2}$	68.4$_{\pm0.2}$	64.1$_{\pm0.3}$	82.5$_{\pm0.2}$	11.5$_{\pm0.2}$	59.2$_{\pm0.3}$	79.0$_{\pm0.2}$	52.4$_{\pm0.3}$
$\oplus$**ALT**-FLD	64.2	65.7$_{\pm0.1}$	63.6$_{\pm0.2}$	52.0	75.3$_{\pm0.1}$	68.5$_{\pm0.1}$	65.0$_{\pm0.2}$	83.6$_{\pm0.1}$	12.1$_{\pm0.1}$	59.9$_{\pm0.2}$	79.3$_{\pm0.1}$	54.4$_{\pm0.2}$
$\oplus$**ALT**-FLD$^{\times 2}$	64.4	66.1$_{\pm0.1}$	63.3$_{\pm0.2}$	52.4	76.1$_{\pm0.1}$	68.5$_{\pm0.1}$	65.4$_{\pm0.2}$	83.6$_{\pm0.2}$	11.4$_{\pm0.1}$	60.8$_{\pm0.2}$	79.5$_{\pm0.1}$	54.4$_{\pm0.2}$

Table 3: LLaMA-3.1-70B trained on the ablation corpora.

	Avg.	Logic	Math	Code	NLI	Others	BBH (3-shot)		BBH (0-shot)		MMLU	
								CoT		CoT		Pro
$\oplus$**ALT**-FLD$^{\times 2}$	**64.4**	**66.1**$_{\pm0.1}$	63.3$_{\pm0.2}$	**52.4**	76.1$_{\pm0.1}$	**68.5**$_{\pm0.1}$	**65.4**$_{\pm0.2}$	**83.6**$_{\pm0.2}$	11.4$_{\pm0.1}$	60.8$_{\pm0.2}$	**79.5**$_{\pm0.1}$	**54.4**$_{\pm0.2}$
w/o DP1	64.2	65.4$_{\pm0.1}$	63.4$_{\pm0.2}$	51.9	**76.5**$_{\pm0.1}$	68.3$_{\pm0.1}$	65.0$_{\pm0.2}$	83.4$_{\pm0.1}$	11.4$_{\pm0.1}$	61.0$_{\pm0.2}$	79.3$_{\pm0.1}$	54.1$_{\pm0.2}$
w/o DP2	62.1	57.3$_{\pm0.2}$	63.2$_{\pm0.2}$	51.8	75.4$_{\pm0.1}$	**68.5**$_{\pm0.2}$	63.5$_{\pm0.2}$	83.1$_{\pm0.1}$	**12.1**$_{\pm0.1}$	60.6$_{\pm0.2}$	79.3$_{\pm0.1}$	53.1$_{\pm0.2}$
w/o DP3.rules	64.1	65.7$_{\pm0.3}$	63.4$_{\pm0.4}$	51.4	75.9$_{\pm0.2}$	68.4$_{\pm0.3}$	64.2$_{\pm0.4}$	83.5$_{\pm0.3}$	10.6$_{\pm0.3}$	58.4$_{\pm0.4}$	79.3$_{\pm0.3}$	54.3$_{\pm0.4}$
w/o DP3.steps	64.2	65.1$_{\pm0.2}$	**63.5**$_{\pm0.2}$	51.7	76.1$_{\pm0.1}$	68.4$_{\pm0.1}$	65.2$_{\pm0.2}$	82.9$_{\pm0.1}$	11.8$_{\pm0.1}$	**62.7**$_{\pm0.2}$	79.4$_{\pm0.1}$	54.3$_{\pm0.2}$
w/o DP4	63.9	64.8$_{\pm0.2}$	**63.5**$_{\pm0.3}$	51.4	76.0$_{\pm0.1}$	68.4$_{\pm0.2}$	64.8$_{\pm0.2}$	83.3$_{\pm0.2}$	10.8$_{\pm0.2}$	58.2$_{\pm0.2}$	79.2$_{\pm0.2}$	**54.4**$_{\pm0.3}$

in synthetic logic corpora. Abductive reasoning involves guessing the missing premises that caused the observed conclusion rather than deriving a conclusion from the premises. For example, from the observed conclusion, "the window glass at home was broken and the room was ransacked," we guess the premise "a burglar broke in." The improvements would be due to the fact that, while the surface form of abductive reasoning problems differs from that of deductive reasoning, they share the fundamentals of logic reflected in the design principles.

Next, we conduct case analyses to see whether the LLM enhanced by ALT acquired the abilities intended by the proposed design principles (DP1-4). Table 5 shows problems where LLaMA-3.1-70B's errors have been corrected by ALT. The first problem is very simple, so it is surprising that LLaMA-3.1-70B failed to solve it, indicating the inherent difficulty of learning logical reasoning solely from pre-training. In contrast, $\oplus$ALT-FLD$^{\times 2}$, which was additionally trained on FLD$^{\times 2}$, solved the problem correctly. The premises of the problem are randomly constructed to express unknown facts. Therefore, the result suggests that $\oplus$ALT-FLD$^{\times 2}$ acquired genuine logical reasoning ability, which can handle unknown facts (DP1).

In the second problem, $\oplus$ALT-FLD$^{\times 2}$ correctly answered "neutral", indicating that it successfully learned that conclusions cannot be derived from insufficient facts (DP2).

The third problem comes from the FOLIO benchmark. To solve this problem, LLMs must use syllogism at the first step as follows: "All eels are fish, and no fish are plants. Therefore, all ells are not plants." $\oplus$ALT-FLD$^{\times 2}$ answered this problem correctly, suggesting that it successfully learned diverse deduction rules (DP3).

FOLIO problems are created based on Wikipedia topics, describing them in more natural and realistic linguistic expressions than in other benchmarks. As seen in the fourth problem, $\oplus$ALT-FLD$^{\times 2}$ understands such expressions, suggesting the effect of diverse expressions from DP4 and/or that LLMs can integrate their original linguistic ability with the newly acquired logical reasoning ability.

Table 4: Benchmark-wise 5-shot performance of LLaMA-3.1-70B before and **after** ALT on FLD$^{\times 2}$. Refer to Table F.9 for LLaMA-3.1-8B results Table E.7 details each benchmark.

(a) Logic.

	bAbiD	FOLIO	LogicNLI	RobustLR	AR-LSAT	LogiQA	ReClor	AbductionR	ART
LLaMA-3.1-70B	**83.8**$_{\pm 1.2}$	58.9$_{\pm 1.6}$	34.9$_{\pm 1.1}$	49.6$_{\pm 0.9}$	21.5$_{\pm 1.0}$	64.3$_{\pm 1.2}$	33.7$_{\pm 0.7}$	84.0$_{\pm 0.7}$	85.4$_{\pm 0.9}$
$\oplus$**ALT**-FLD$^{\times 2}$	83.5$_{\pm 0.5}$	**66.7**$_{\pm 0.6}$	**50.9**$_{\pm 0.5}$	**81.6**$_{\pm 0.3}$	**25.0**$_{\pm 0.4}$	**69.4**$_{\pm 0.5}$	**36.3**$_{\pm 0.3}$	**95.7**$_{\pm 0.2}$	**85.5**$_{\pm 0.4}$

(b) Math.

	GSM8k			MATH	MathQA
		CoT	CoT (0-shot)	-	-
LLaMA-3.1-70B	80.9$_{\pm 1.1}$	75.2$_{\pm 1.2}$	65.4$_{\pm 1.3}$	23.7$_{\pm 0.6}$	55.0$_{\pm 0.9}$
$\oplus$**ALT**-FLD$^{\times 2}$	**83.3**$_{\pm 0.4}$	**80.4**$_{\pm 0.4}$	**73.0**$_{\pm 0.5}$	**24.4**$_{\pm 0.2}$	**55.4**$_{\pm 0.4}$

(c) Coding.

	HumanEval	MBPP	MBPP+	MultiPL-E (cpp)	MultiPL-E (go)
LLaMA-3.1-70B	32.3	43.4	48.7	29.8	76.6
$\oplus$**ALT**-FLD$^{\times 2}$	**42.6**	**49.5**	**52.5**	**38.7**	**78.6**

(d) Natural language inference (NLI).

	HELP	MNLI	RTE	SNLI
LLaMA-3.1-70B	45.8$_{\pm 0.5}$	82.2$_{\pm 0.4}$	84.0$_{\pm 0.7}$	**82.6**$_{\pm 0.4}$
$\oplus$**ALT**-FLD$^{\times 2}$	**51.3**$_{\pm 0.2}$	**83.7**$_{\pm 0.2}$	**87.2**$_{\pm 0.3}$	82.3$_{\pm 0.2}$

(e) Others.

	CommonsenseQA	HellaSwag	SQuAD	WinoGrande	ARCe	ARCc	GPQA	OpenBookQA	SciQ
LLaMA-3.1-70B	81.2$_{\pm 1.1}$	69.2$_{\pm 0.5}$	38.5$_{\pm 0.0}$	85.6$_{\pm 1.0}$	89.1$_{\pm 0.6}$	65.3$_{\pm 1.4}$	**40.7**$_{\pm 1.4}$	41.4$_{\pm 0.7}$	**98.5**$_{\pm 0.4}$
$\oplus$**ALT**-FLD$^{\times 2}$	**82.5**$_{\pm 0.4}$	**69.6**$_{\pm 0.2}$	**40.1**$_{\pm 0.0}$	**86.1**$_{\pm 0.4}$	**89.4**$_{\pm 0.3}$	**66.7**$_{\pm 0.6}$	40.6$_{\pm 0.6}$	**42.8**$_{\pm 0.3}$	**98.5**$_{\pm 0.2}$

6.2 Math and Coding Tasks

Tables F.9b, F.9c shows that ALT substantially boosted the LLaMA-3.1-70B's performance by up to 7 and 10 points on math and coding tasks, respectively. The math improvements are reasonable, as understanding predicate logic is a prerequisite for solving mathematical problems. For coding, some recent studies have verified the opposite direction, namely, that training on coding data improves logical reasoning abilities Jiang et al. (2024b); MA et al. (2024); Uchiyama et al. (2024).

6.3 NLI Tasks

Table F.9d shows that ALT substantially boosted the LLaMA-3.1-70B's performance by up to 6 points on various natural language inference (NLI) benchmarks. NLI is similar to deductive reasoning in assessing whether a premise supports or contradicts a hypothesis. However, the main difference is that this judgment requires a rich set of commonsense knowledge beyond the given premise.

Consider the fifth problem in Table 5: by supplementing the given fact "An Indian woman is dancing with her partner" with the commonsense knowledge "If someone is dancing, then he/she is moving.", we can derive the hypothesis "A woman is moving." The sixth problem is more challenging as we have to trace multiple logical steps while supplementing with sufficient commonsense knowledge as follows: "a church choir sings at a church," "baseball is often played at a baseball field," "a person cannot be in two or more places at the same time," "therefore, a church choir cannot sing for baseball."

Since synthetic logic corpora only contain unknown facts, LLMs cannot acquire new knowledge from them. Therefore, the commonsense knowledge used to solve the above problems must have been acquired by the LLMs from pre-training. This suggests that LLMs can integrate their original knowledge with the logical reasoning capabilities newly acquired from ALT to solve problems.

Table 5: Problems where LLaMA-3.1-70B initially answered incorrectly and then correctly after training with ALT on FLD$^{\times 2}$. Red highlights the premises related to the hypothesis.

benchmark	premises	hypothesis	answer (LLaMA-3.1-70B/gold)	required ability
LogicNLI	Mice are afraid of wolves. Cats are afraid of sheep. Jessica is a cat. Wolves are afraid of cats. Winona is a wolf. Sheep are afraid of cats.	Jessica is afraid of sheep.	neutral/ entailment	DP1
	Rhett is not modest. Vivian is confused. Rhett is lazy. If someone is modest or not confused, then he is not eager.	Rhett is confused.	entailment/ neutral	DP2
FOLIO	All eels are fish. No fish are plants. Everything displayed in the collection is either a plant or an animal. All animals displayed in the collection are multicellular. A sea eel is displayed in the collection. The sea eel is an eel or an animal or not a plant.	The sea eel is multicellular or is bacteria.	neutral/ entailment	DP3
	Common utilities include water, electricity, gas, heating, sewer, trash, and recycling. Many apartment rents cover the cost of water and electricity. Susan lives in an apartment where the rent covers all utilities. The rent of the apartment where Ava lives does not cover any utility expenses. Noah lives in an apartment where the rent does not cover heating.	Noah and Ava both need to pay the heating bill.	neutral/ entailment	DP4
SNLI	An Indian woman is dancing with her partner.	A woman is moving.	neutral/ entailment	reasoning with commonsense knowledge
	This church choir sings to the masses as they sing joyous songs from the book at a church.	A choir is singing at a baseball game.	entailment/ contradiction	

Table 6: Problems that LLaMA-3.1-70B trained with ALT on FLD$^{\times 2}$ still cannot solve.

benchmark	question	answer
ARC (challenge)	The end result in the process of photosynthesis is the production of sugar and oxygen. Which step signals the beginning of photosynthesis?	Chlorophyll in the leaf captures light energy.
GPQA	A spin-half particle is in a linear superposition $0.8\|\uparrow\rangle + 0.6\|\downarrow\rangle$ of its spin-up and spin-down states. If $\|\uparrow\rangle$ and $\|\downarrow\rangle$ are the eigenstates of σ_z, then what is the expectation value up to one decimal place, of the operator $10\sigma_z + 5\sigma_x$?	-0.7
ARC (challenge)	Beavers build their homes in ponds and streams. Which characteristic is least critical to building homes in an aquatic environment?	(A) waterproof fur (B) webbed hind feet **(C) arge, sharp teeth** (D) flat, wide tail

6.4 Other Tasks

Improvements across various other tasks (Table F.9e) demonstrate the broad benefits of the obtained reasoning capabilities beyond standard reasoning tasks; though the improvements were modest at up to 2 percentage points, which may be due to the following reasons. First, these benchmarks include problems that purely test knowledge, such as the first one in Table 6. Since ALT does not aim to provide new knowledge, the ability to solve such problems does not improve by nature. Next, some problems may require knowledge that is too advanced for LLMs, so potential improvements by the enhanced reasoning capabilities may be bottlenecked. For example, the second problem does involve reasoning but requires sufficient quantum mechanics knowledge as a prerequisite. However, these knowledge-related issues should be solved by improving the quantity and quality of pre-training.

Finally, LLMs may not be able to fully utilize the potential of enhanced reasoning capabilities for problems that require complex procedures. To solve the third problem, LLMs first must attempt reasoning related to each choice as follows: "To build homes in an aquatic environment, one needs to maintain body heat and insulation despite being frequently submerged in cold water. Therefore, the waterproof fur of (A) is essential", and "To build . . . , one must gather and process natural materials like wood. Large, sharp teeth of (C) are critical as they allow beavers to cut down trees and shape branches." Next, while reasoning traces on (A) to (D) all seem reasonable, LLMs must choose the single best answer, considering the subtle nuance of the question context, as follows: "Since the question emphasizes the aquatic environment, the least related reasoning trace should be (C)." This complex procedure contrasts with logical reasoning and NLI problems, where LLMs can directly obtain an answer from a single reasoning trace. Previous studies also observed that such procedure on multiple-choice QA problems are challenging for LLMs (Robinson and Wingate, 2023; Zheng et al., 2024; Wang et al., 2024a). Since ALT alone does not teach LLMs such task-specific procedures, additional training on these procedures should be necessary to solve these problems.

7 Conclusion

Towards versatile artificial intelligence with reasoning capabilities, we proposed **A**dditional ***L**ogic* **T**raining on synthetic logic samples. We established systematic design principles well-grounded on symbolic logic theory and previous empirical findings. We constructed a corpus named Formal Logic Deduction Diverse (FLD$^{\times 2}$) based on the design principles. We empirically showed that ALT on FLD$^{\times 2}$ substantially enhances the capabilities of state-of-the-art LLMs and that LLMs can integrate their originally possessed knowledge and newly acquired reasoning capabilities.

Acknowledgement

Computational resources of AI Bridging Cloud Infrastructure (ABCI) provided by the National Institute of Advanced Industrial Science and Technology (AIST) were used. We thank Dr. Masaaki Shimizu at Hitachi for the convenience of additional computational resources. We thank Dr. Naoaki Okazaki, a professor at the Tokyo Institute of Technology, for the keen comments.

References

AI@Meta. 2024. Llama 3 model card.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Yoichi Aoki, Keito Kudo, Tatsuki Kuribayashi, Shusaku Sone, Masaya Taniguchi, Keisuke Sakaguchi, and Kentaro Inui. 2024. First heuristic then rational: Dynamic use of heuristics in language model reasoning.

Amanda Askell. 2020. Gpt-3: Towards renaissance models. *Daily Nous Blog: Philosophers On GPT-3*.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Qiming Bao, Alex Yuxuan Peng, Tim Hartill, Neset Tan, Zhenyun Deng, Michael Witbrock, and Jiamou Liu. 2022. Multi-step deductive reasoning over natural language: An empirical study on out-of-distribution generalisation. In *Proceedings of the 16th International Workshop on Neural-Symbolic Learning and Reasoning as part of the 2nd International Joint Conference on Learning & Reasoning (IJCLR 2022)*, pages 202–217, Cumberland Lodge, Windsor Great Park, United Kingdom.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. 2024. Cosmopedia.

Loubna Ben Allal, Niklas Muennighoff, Logesh Kumar Umapathi, Ben Lipkin, and Leandro von Werra. 2022. A framework for the evaluation of code generation models. `https://github.com/bigcode-project/bigcode-evaluation-harness`.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth pascal recognizing textual entailment challenge. In *Text Analysis Conference*.

Leonardo Bertolazzi, Albert Gatt, and Raffaella Bernardi. 2024. A systematic analysis of large language models as soft reasoners: The case of syllogistic inferences.

Russell Bertrand. A history of western philosophy.

Gregor Betz, Christian Voigt, and Kyle Richardson. 2021. Critical thinking for language models. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 63–75, Groningen, The Netherlands (online). Association for Computational Linguistics.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen-tau Yih, and Yejin Choi. 2019. Abductive commonsense reasoning. *arXiv preprint arXiv:1908.05739*.

Neeladri Bhuiya, Viktor Schlegel, and Stefan Winkler. 2024. Seemingly plausible distractors in multi-hop reasoning: Are large language models attentive readers?

Kaj Bostrom, Xinyu Zhao, Swarat Chaudhuri, and Greg Durrett. 2021. Flexible generation of natural language deductions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6266–6278, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference.

Federico Cassano, John Gouwar, Daniel Nguyen, Sydney Nguyen, Luna Phipps-Costin, Donald Pinckney, Ming-Ho Yee, Yangtian Zi, Carolyn Jane Anderson, Molly Q Feldman, Arjun Guha, Michael Greenberg, and Abhinav Jangda. 2023. Multipl-e: A scalable and polyglot approach to benchmarking neural code generation. *IEEE Transactions on Software Engineering*, 49(7):3675–3691.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code.

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7870–7881, Online. Association for Computational Linguistics.

Xinyun Chen, Ryan Andrew Chi, Xuezhi Wang, and Denny Zhou. 2024. Premise order matters in reasoning with large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 6596–6620. PMLR.

J. Cheng, M. Bernstein, C. Danescu-Niculescu-Mizil, and J. Leskovec. 2017. Anyone can become a troll: Causes of trolling behavior in online discussions. *CSCW: Proceedings of the Conference on Computer-Supported Cooperative Work. Conference on Computer-Supported Cooperative Work, 2017*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2021. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3882–3890.

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

A. Colmerauer and P Roussel. 1973. The birth of prolog. *The ALP Newsletter*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. pages 177–190.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. Explaining answers with entailment trees. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7358–7370, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

John Dougrez-Lewis, Mahmud Elahi Akhter, Yulan He, and Maria Liakata. 2024. Assessing the reasoning abilities of chatgpt in the context of claim verification.

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

Tiwalayo Eisape, Michael Tessler, Ishita Dasgupta, Fei Sha, Sjoerd 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 (Volume 1: Long Papers)*, pages 8425–8444, Mexico City, Mexico. Association for Computational Linguistics.

Charles Elkan and Russell Greiner. 1993. Building large knowledge-based systems: Representation and inference in the cyc project: Db lenat and rv guha.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9.

Kurt Gödel. 1930. *Uber die vollständigkeit des logikkalküls*. Ph.D. thesis, Ph. D. dissertation, University of Vienna.

Nicolas Gontier, Koustuv Sinha, Siva Reddy, and Chris Pal. 2020. Measuring systematic generalization in neural proof generation with transformers. *Advances in Neural Information Processing Systems*, 33:22231–22242.

Nelson Goodman. 1954. Fact, fiction, and forecast. london: University of london.

Radu Cornel Guiaşu and Christopher W Tindale. 2018. Logical fallacies and invasion biology. *Biology & philosophy*, 33(5-6):34.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1930–1940, New Orleans, Louisiana. Association for Computational Linguistics.

Pengrui Han, Peiyang Song, Haofei Yu, and Jiaxuan You. 2024. In-context learning may not elicit trustworthy reasoning: A-not-b errors in pretrained language models.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Luke Benson, Lucy Sun, Ekaterina Zubova, Yujie Qiao, Matthew Burtell, et al. 2022. Folio: Natural language reasoning with first-order logic. *arXiv e-prints*, pages arXiv–2209.

Sven Ove Hansson. 2004. Fallacies of risk. *Journal of Risk Research*, 7(3):353–360.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

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

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers.

Damian Hodel and Jevin West. 2023. Response: Emergent analogical reasoning in large language models.

Ruixin Hong, Hongming Zhang, Xinyu Pang, Dong Yu, and Changshui Zhang. 2024. A closer look at the self-verification abilities of large language models in logical reasoning. 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 900–925, Mexico City, Mexico. Association for Computational Linguistics.

Peng Hu, Changjiang Gao, Ruiqi Gao, Jiajun Chen, and Shujian Huang. 2024. Large language models are limited in out-of-context knowledge reasoning.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*.

David Hume. 1748. An enquiry concerning human understanding (section iv). *Recuperado de http://www. clorenzano. com. ar*.

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

Jin Jiang, Yuchen Yan, Yang Liu, Yonggang Jin, Shuai Peng, Mengdi Zhang, Xunliang Cai, Yixin Cao, Liangcai Gao, and Zhi Tang. 2024b. Logicpro: Improving complex logical reasoning via program-guided learning.

Daniel Kahneman. 2011. *Thinking, fast and slow*. Macmillan.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. 2023. Measuring faithfulness in chain-of-thought reasoning.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. 2023. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2665–2679, Toronto, Canada. Association for Computational Linguistics.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, Wenhu Chen, and Xifeng Yan. 2022. Explanations from large language models make small reasoners better.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. 2023a. Logiqa 2.0—an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962.

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

Hanmeng Liu, Zhiyang Teng, Leyang Cui, Chaoli Zhang, Qiji Zhou, and Yue Zhang. 2023c. LogiCoT: Logical chain-of-thought instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2908–2921, Singapore. Association for Computational Linguistics.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3622–3628. International Joint Conferences on Artificial Intelligence Organization. Main track.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2023d. Is your code generated by chatGPT really correct? rigorous evaluation of large language models for code generation. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ziyi Liu, Isabelle Lee, Yongkang Du, Soumya Sanyal, and Jieyu Zhao. 2024. Self-contradictory reasoning evaluation and detection.

Zimu Lu, Aojun Zhou, Houxing Ren, Ke Wang, Weikang Shi, Junting Pan, Mingjie Zhan, and Hongsheng Li. 2024. Mathgenie: Generating synthetic data with question back-translation for enhancing mathematical reasoning of llms.

YINGWEI MA, Yue Liu, Yue Yu, Yuanliang Zhang, Yu Jiang, Changjian Wang, and Shanshan Li. 2024. At which training stage does code data help LLMs reasoning? In *The Twelfth International Conference on Learning Representations*.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2023. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1773–1781, Toronto, Canada. Association for Computational Linguistics.

John W. McCarthy. 1959. Programs with common sense. In *Proc. Tedding Conf. on the Mechanization of Thought Processes*, pages 75–91.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Melanie Mitchell. 2023. Can large language models reason? *blog*, pages https://aiguide.substack.com/p/can–large–language–models–reason.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agarwal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, Hamid Palangi, Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Awadallah. 2023. Orca 2: Teaching small language models how to reason.

Philipp Mondorf and Barbara Plank. 2024. Liar, liar, logical mire: A benchmark for suppositional reasoning in large language models.

Terufumi Morishita, Gaku Morio, Atsuki Yamaguchi, and Yasuhiro Sogawa. 2023. Learning deductive reasoning from synthetic corpus based on formal logic. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 25254–25274. PMLR.

Terufumi Morishita, Atsuki Yamaguchi, Gaku Morio, Hikaru Tomonari, Osamu Imaichi, and Yasuhiro Sogawa. 2024. JFLD: A Japanese benchmark for deductive reasoning based on formal logic. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9526–9535, Torino, Italia. ELRA and ICCL.

Aliakbar Nafar, K. Brent Venable, and Parisa Kordjamshidi. 2024. Teaching probabilistic logical reasoning to transformers. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1615–1632, St. Julian's, Malta. Association for Computational Linguistics.

Kentaro Ozeki, Risako Ando, Takanobu Morishita, Hirohiko Abe, Koji Mineshima, and Mitsuhiro Okada. 2024. Exploring reasoning biases in large language models through syllogism: Insights from the NeuBAROCO dataset. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 16063–16077, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Fabio Paglieri. 2017. A plea for ecological argument technologies. *Philosophy & Technology*, 30(2):209–238.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. 2024. LogicBench: Towards systematic evaluation of logical reasoning ability of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13679–13707, Bangkok, Thailand. Association for Computational Linguistics.

Nisarg Patel, Mohith Kulkarni, Mihir Parmar, Aashna Budhiraja, Mutsumi Nakamura, Neeraj Varshney, and Chitta Baral. 2024. Multi-logieval: Towards evaluating multi-step logical reasoning ability of large language models.

Xinyu Pi, Qian Liu, Bei Chen, Morteza Ziyadi, Zeqi Lin, Qiang Fu, Yan Gao, Jian-Guang Lou, and Weizhu Chen. 2022. Reasoning like program executors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 761–779, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Willard Van Orman Quine. 1969. Epistemology naturalized. ontological relativity and other essays. *New York: Columbia UP*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

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

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.

Joshua Robinson and David Wingate. 2023. Leveraging large language models for multiple choice question answering. In *The Eleventh International Conference on Learning Representations*.

Mohammed Saeed, Naser Ahmadi, Preslav Nakov, and Paolo Papotti. 2021. RuleBERT: Teaching soft rules to pre-trained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1460–1476, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Swarnadeep Saha, Sayan Ghosh, Shashank Srivastava, and Mohit Bansal. 2020. PRover: Proof generation for interpretable reasoning over rules. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 122–136, Online. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Soumya Sanyal, Zeyi Liao, and Xiang Ren. 2022a. Robustlr: Evaluating robustness to logical perturbation in deductive reasoning. *arXiv preprint arXiv:2205.12598*.

Soumya Sanyal, Harman Singh, and Xiang Ren. 2022b. Fairr: Faithful and robust deductive reasoning over natural language. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1075–1093.

eh Shortliffe. 1976. Computer based medical consultations: Mycin. *Elsevier*.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2023. Distilling reasoning capabilities into smaller language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7059–7073, Toronto, Canada. Association for Computational Linguistics.

Damien Sileo. 2024. Scaling synthetic logical reasoning datasets with context-sensitive declarative grammars.

Zayne Rea Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. 2024. MuSR: Testing the limits of chain-of-thought with multistep soft reasoning. In *The Twelfth International Conference on Learning Representations*.

Cass R Sunstein and Reid Hastie. 2015. *Wiser: getting beyond groupthink to make groups smarter*. Harvard Business Review Press, Boston.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634, Online. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Jidong Tian, Yitian Li, Wenqing Chen, Liqiang Xiao, Hao He, and Yaohui Jin. 2021. Diagnosing the first-order logical reasoning ability through logicnli. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3738–3747.

Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. 2024. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting.

Fumiya Uchiyama, Takeshi Kojima, Andrew Gambardella, Qi Cao, Yusuke Iwasawa, and Yutaka Matsuo. 2024. Which programming language and what features at pre-training stage affect downstream logical inference performance?

Yuxuan Wan, Wenxuan Wang, Yiliu Yang, Youliang Yuan, Jen tse Huang, Pinjia He, Wenxiang Jiao, and Michael R. Lyu. 2024. Logicasker: Evaluating and improving the logical reasoning ability of large language models.

Haochun Wang, Sendong Zhao, Zewen Qiang, Bing Qin, and Ting Liu. 2024a. Beyond the answers: Reviewing the rationality of multiple choice question answering for the evaluation of large language models.

Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. 2023. SCOTT: Self-consistent chain-of-thought distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5546–5558, Toronto, Canada. Association for Computational Linguistics.

Siyuan Wang, Zhongyu Wei, Yejin Choi, and Xiang Ren. 2024b. Can LLMs reason with rules? logic scaffolding for stress-testing and improving LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7523–7543, Bangkok, Thailand. Association for Computational Linguistics.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024c. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark (published at neurips 2024 track datasets and benchmarks).

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*.

Joseph Weizenbaum. 1966. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark. Association for Computational Linguistics.

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

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*, pages 1112–1122.

T Winograd. 1971. Procedures as a representation for data in a computer program for understanding natural language, mit ai technical report 235.

Ludwig Wittgenstein. 1922. *Tractatus Logico Philosophicus: Logical-Philosophical Treatise*. Really Simple Media.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

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.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019. Help: A dataset for identifying shortcomings of neural models in monotonicity reasoning. *arXiv preprint arXiv:1904.12166*.

Nathan Young, Qiming Bao, Joshua Bensemann, and Michael J Witbrock. 2022. Abductionrules: Training transformers to explain unexpected inputs. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 218–227.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. Reclor: A reading comprehension dataset requiring logical reasoning. In *International Conference on Learning Representations (ICLR)*.

Zhangdie Yuan, Songbo Hu, Ivan Vulić, Anna Korhonen, and Zaiqiao Meng. 2023. Can pretrained language models (yet) reason deductively? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1439–1454.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Honghua Zhang, Liunian Harold Li, Tao Meng, Kai-Wei Chang, and Guy Van den Broeck. 2022. On the paradox of learning to reason from data.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Dylan Slack, Qin Lyu, Sean Hendryx, Russell Kaplan, Michele Lunati, and Summer Yue. 2024. A careful examination of large language model performance on grade school arithmetic.

Jun Zhao, Jingqi Tong, Yurong Mou, Ming Zhang, Qi Zhang, and Xuanjing Huang. 2024a. Exploring the compositional deficiency of large language models in mathematical reasoning.

Wenting Zhao, Justin Chiu, Jena Hwang, Faeze Brahman, Jack Hessel, Sanjiban Choudhury, Yejin Choi, Xiang Li, and Alane Suhr. 2024b. UNcommonsense reasoning: Abductive reasoning about uncommon situations. 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 8487–8505, Mexico City, Mexico. Association for Computational Linguistics.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*.

Wanjun Zhong, Siyuan Wang, Duyu Tang, Zenan Xu, Daya Guo, Jiahai Wang, Jian Yin, Ming Zhou, and Nan Duan. 2021. Ar-lsat: Investigating analytical reasoning of text. *arXiv preprint arXiv:2104.06598*.

Jin Peng Zhou, Charles E Staats, Wenda Li, Christian Szegedy, Kilian Q Weinberger, and Yuhuai Wu. 2024a. Don't trust: Verify – grounding LLM quantitative reasoning with autoformalization. In *The Twelfth International Conference on Learning Representations*.

Yue Zhou, Yada Zhu, Diego Antognini, Yoon Kim, and Yang Zhang. 2024b. Paraphrase and solve: Exploring and exploiting the impact of surface form on mathematical reasoning in large language models. 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 2793–2804, Mexico City, Mexico. Association for Computational Linguistics.

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. 2024. Dyval: Dynamic evaluation of large language models for reasoning tasks. In *The Twelfth International Conference on Learning Representations*.

A Related Work

A.1 Investigation of Reasoning Capabilities of LLMs

Many studies examine LLMs' reasoning capabilities Askell (2020); Rae et al. (2021); Razeghi et al. (2022); Liu et al. (2023b); Turpin et al. (2023); Lanham et al. (2023); Wu et al. (2023); Hodel and West (2023); Dziri et al. (2023); Dasgupta et al. (2023). Patel et al. (2024) observed LLMs' performance significantly declines as reasoning steps increase in multi-step logical reasoning tasks. Dougrez-Lewis et al. (2024) revealed ChatGPT struggles with abductive reasoning when verifying claims by decomposing their evidence into atomic reasoning steps. Wang et al. (2024b) found that GPT-series models showed significant gaps compared to humans in dealing with inference rules. Parmar et al. (2024) introduced LogicBench and showed that existing LLMs struggle with instances involving complex reasoning and negations. Wan et al. (2024) introduced LogicAsker, which assesses whether LLMs can employ a set of atomic reasoning skills grounded in propositional and predicate logic and found significant gaps in LLMs' learning of logical rules. Bhuiya et al. (2024) proposed a challenging multi-hop reasoning benchmark with seemingly plausible but incorrect multi-hop reasoning chains and found that state-of-the-art LLMs' capabilities to perform multi-hop reasoning is affected by such chains. Mondorf and Plank (2024) introduced TruthQuest, which assesses LLMs' capabilities to conduct suppositional reasoning, i.e., reasoning where each statement can be false, and found that LLMs exhibit significant difficulties solving these tasks. Sprague et al. (2024) introduced a complex multi-step reasoning benchmark, MuSR, and characterized the gaps that remain for techniques like chain-of-thought to perform robust reasoning.

Biases and Errors Ando et al. (2023); Ozeki et al. (2024); Bertolazzi et al. (2024); Eisape et al. (2024) found that LLMs exhibit human-like reasoning biases in syllogistic arguments. Jiang et al. (2024a) found that LLMs exibit "token-biases" in solving logical reasoning problems. Aoki et al. (2024) revealed that LMs rely heavily on heuristics, such as lexical overlap, in the earlier stages of reasoning. Zhao et al. (2024a) constructed a MATHTRAP with carefully designed logical traps into the problem descriptions of MATH and GSM8k and found that while LLMs possess the knowledge required to solve these traps, they do not spontaneously use such knowledge them to handle the problems. Han et al. (2024) found that LLMs exhibit A-Not-B errors similar to human infants, failing to suppress the previously established response pattern during ICL. Liu et al. (2024) found that LLMs often contradict themselves in reasoning tasks involving contextual information understanding or commonsense. Zhou et al. (2024b) found that subtle alterations in the surface form can significantly impact the answer distribution, suggesting that LLMs solve reasoning problems using surface cues. Chen et al. (2024) found that the reasoning performance of LLMs is affected by the order of the premises. Hong et al. (2024); Huang et al. (2024) found that LLMs struggle to identify fallacious reasoning steps accurately, suggesting challenges in self-verification methods.

Reasoning in Unknown Situation Zhao et al. (2024b) found that LLMs struggle with reasoning in uncommon situations. Zhu et al. (2024) introduced a framework to dynamically generate reasoning samples, and LLMs perform worse in those samples. Hu et al. (2024) found that while LLMs can conduct reasoning when relevant knowledge is given in context, they are not proficient at reasoning with knowledge embedded in the training data.

A.2 Synthetic Logic Corpus for Training LLMs

Later studies Saha et al. (2020); Dalvi et al. (2021); Tafjord et al. (2021); Sanyal et al. (2022b) showed that T5 can generate even the intermediate logical steps as well as the final answer.

PARARULE-Plus Bao et al. (2022) is the enhanced version of PARARULE Clark et al. (2021), a variation of RuleTaker, that includes more samples and more logical steps. RoBERTa Liu et al. (2019) trained on PARARULE-Plus outperformed the models trained on RuleTaker.

Artificial Argument Corpus Betz et al. (2021) includes single-step deductive reasoning samples constructed from hand-selected deduction rules useful for critical thinking. They showed that the GPT-2 Radford et al. (2019) trained on this corpus can generalize to solve NLI tasks. However, at the same time, they found that the LM does not generalize well to solve more challenging reasoning tasks such as ARC Habernal et al. (2018) and LogiQA Liu et al. (2020).

FLD by Morishita et al. (2023, 2024) is the first synthetic logic corpus based on formal logic theory. It includes multistep deductive reasoning samples constructed from the axioms of first-order predicate logic, which can express any deduction rule due to the completeness theorem. Due to this nature, T5 trained on FLD generalizes most effectively to other synthetic logic corpora, compared to models trained on other corpora.

Gontier et al. (2020) investigated the deductive reasoning capabilities of LMs on a corpus composed of a specific type of multistep inference, kinship relationships on synthetic kinship graphs. They found that LMs can solve this task when there are relatively few proof steps, but it is difficult for them to generalize to solve proof steps longer than those shown in training data. Bostrom et al. (2021) studied how to create realistic natural language expressions that represent deduction rules. To this end, they scraped sentences from Wikipedia using a template-based method and paraphrased them. They showed that training on this corpus helps solve real-world deductive reasoning problems such as EntailmentBank Dalvi et al. (2021). Pi et al. (2022) used synthetic data from program executors, most notably SQL programs. They verified that this data can enhance numerical reasoning, logical reasoning, and multi-hop reasoning abilities. Trinh et al. (2024) generated 100 million geometry problems and verified that the capability of artificial intelligence can be enhanced to to pass the bronze medal threshold of the International Mathematics Olympiad. Saeed et al. (2021); Nafar et al. (2024) created *soft* reasoning rules involving with probabilistic logic, instead of hard-logic examined by the aformentioned studies. Sileo (2024) introduced a simpler and more general declarative framework for synthetic generation, and verified its effectiveness. Zhou et al. (2024a) synthetically generated a large dataset of mathematics, and gained over 12 points on GSM8k.

While these studies partly examined the effect of synthetic logic corpora, whether this approach is promising remains an open question. It has been unexplored whether the capabilities obtained from synthetic logic corpora generalizes to solve various tasks beyond the original tasks in these corpora. Additionally, the effect of these corpora has only been examined for small LMs trained on small pre-training corpora such as T5 and RoBERTa; it has been highly questionable whether they can still benefit state-of-the-art LLMs trained on a huge pre-training corpus. Furthermore, even if their benefits were verified, it remains unclear which design of synthetic logic samples yields the largest benefits due to the lack of systematic discussions on sample designs and empirical verification of these designs. We aimed to answer these questions in this paper and demonstrate the potential of synthetic logic corpora.

A.3 Distilling Reasoning Traces from Very Large LLMs

Recent approaches Ho et al. (2023); Magister et al. (2023); Li et al. (2022, 2023); Shridhar et al. (2023); Wang et al. (2023); Mitra et al. (2023); Liu et al. (2023c); Ben Allal et al. (2024); Lu et al. (2024) utilize very large LLMs, such as GPT-4, to prepare synthetic reasoning datasets to train smaller LLMs. A typical procedure is as follows: (i) prepare existing reasoning problems, (ii) prompt large LLMs to generate reasoning traces to solve these problems using techniques such as chain-of-thought prompting Wei et al. (2022), and (iii) train smaller LLMs on these reasoning traces.

The distillation approach and the synthetic logic corpora approach examined in this paper have specific advantages and disadvantages, as follows.

The advantage of the distillation approach is its immediate practical effect, as it directly teaches LLMs solutions to various existing problems. The disadvantages could be that (i) it is non-trivial for specific solutions to specific problems to generalize to other problems, (ii) the number of training samples is limited to existing problems in nature, (iii) the correctness and faithfulness of the reasoning traces are not guaranteed; indeed, some studies Turpin et al. (2023); Lanham et al. (2023) suggest that large LLMs do not always faithfully follow the "reasoning traces" they themselves generate, and (iv) it cannot enhance the very large LLMs themselves by nature.

The advantages of synthetic logic corpus approaches are that (i) since they teach the fundamentals of reasoning, such as deductive reasoning, they have the potential to generalize to various problems, (ii) they can generate an unlimited number of new samples, and (iii) the correctness of the reasoning traces is guaranteed by nature. The disadvantage of this approach is that, as it only teaches the basics of reasoning, additional training may be needed to solve more complex real-world problems, as suggested in Section 6.4.

We hypothesize that integrating both approaches could be promising. That is, we first train LLMs using ALT to make them understand the fundamentals of reasoning through high-quality samples and then train them using more realistic reasoning traces to solve complex real-world problems.

B Limitations

- We only used deductive reasoning samples for ALT. Future work should examine other reasoning samples, e.g., abductive and inductive reasoning.
- We only examined the first-order predicate logic system. Future work should examine other logic systems, such as modal and linear logic.

C Ethics and Social Impacts

The ultimate goal of the direction of this study is to develop an AI capable of reasoning logically step by step. If AI can make a decision one logical step at a time, it would be highly explainable and transparent to users. Furthermore, the user would be able to trace the AI's errors. We believe that our study is a step towards such AI that will positively impact society.

D Details of Formal Logic Deduction Diverse

Figure D.3 shows a real sample from FLD$^{\times 2}$. Below, We briefly explain our sample generator. Please refer to Morishita et al. (2023) for the details.

D.1 Answer Labels

In addition to the logical steps, the samples of FLD$^{\times 2}$ and previous corpora include *answer labels* (Figure D.3): "proved" indicating that the hypothesis can be proved by the logical steps, "disproved" indicating that the hypothesis can be disproved, and "unknown" indicating that the given facts are insufficient for either proving or disproving the hypothesis. For samples with "unknown" labels, the logical steps are "None.". FLD$^{\times 2}$ have a uniform distribution over the labels.

D.2 Splits

FLD$^{\times 2}$ includes 100k/5k/5k samples for train/valid/test splits.

D.3 Generation of Multistep Deduction

Our sample generator first randomly generates examples of multistep deduction by forward- and backward random deduction, using the deduction rules specified by a user.

The forward random deduction is done as follows. The generator first chooses a deduction rule randomly and forms the initial tree where the root node is the conclusion of the chosen deduction rules and the child nodes are the premises of the chosen deduction rule. The generator next randomly chooses another deduction rule that can be "jointed" to the root note of the tree. A deduction rule can be jointed to the root node of a tree if one of the premises of that deduction rule can be identified with the root node. Then, the generator updates the tree by jointing this chosen deduction rule. The generator continues this step multiple times until the tree achieves the required depth.

The backward random deduction is done as follows. For each step, the generator randomly chooses a leaf node of the tree. Then, the generator randomly chooses a deduction rule that can be jointed to the leaf node. Here, a deduction rule can be jointed to the leaf node if the deduction rule's conclusion can be identified with the leaf node. Then, the generator updates the tree by jointing this chosen deduction rule. The generator continues this step multiple times until the complexity of branches achieves the required level.

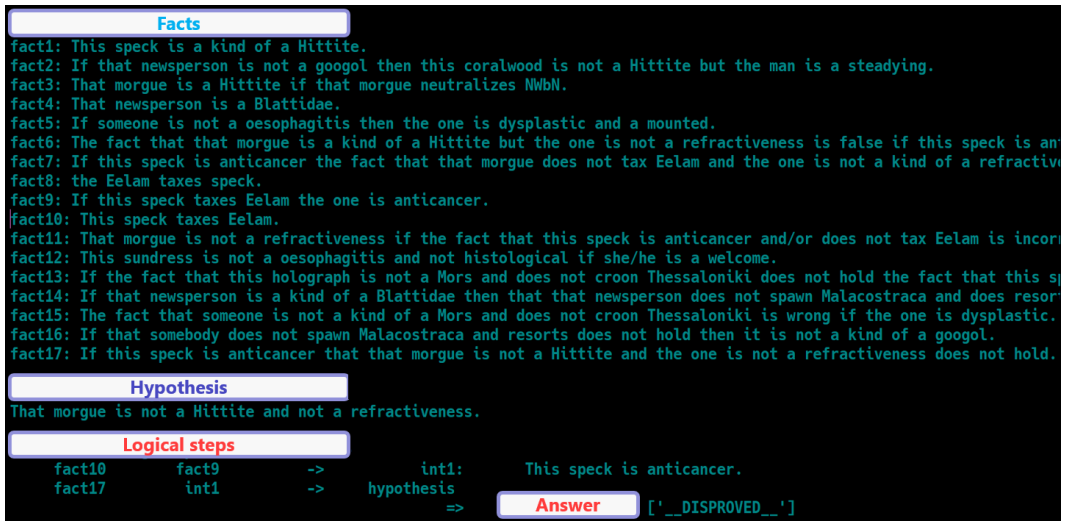

Figure D.3: A real deduction sample included in Formal Logic Deduction Diverse. **Facts** and **hypothesis** are given to LLMs, then the LLMs are required to generate **logical steps** to (dis-)prove the hypothesis based on the facts, and an **answer** label (see Appendix D.2).

D.4 Linguistic Expressions

We prepared linguistic templates for each logical formula, exemplified as follows:

$$
\begin{aligned}
\langle (A \wedge B) \rightarrow C \rangle &: \text{If } \langle (A \wedge B).\text{predicate\_phrase} \rangle, \text{ then } \langle C.\text{predicate\_phrase} \rangle. \\
&: \langle (A \wedge B).\text{noun\_pharse} \rangle \ \langle \text{cause\_synonyms} \rangle \ \langle C.\text{noun\_phrase} \rangle. \\
&: (\ldots) \\
\langle (A \wedge B).\text{predicate\_phrase} \rangle &: A \ \langle \text{occur\_synonyms} \rangle \text{ and also } B \ \langle \text{occur\_synonyms} \rangle. \\
&: A \text{ and also } B \ \langle \text{occur\_synonyms} \rangle. \\
&: \text{Both } A \text{ and } B \ \langle \text{occur\_synonyms} \rangle. \\
&: (\ldots) \\
\langle C.\text{predicate\_phrase} \rangle &: C \ \langle \text{occur\_synonyms} \rangle. \\
&: (\ldots) \\
\langle \text{occur\_synonyms} \rangle &: \text{occur} \\
&: \text{happen} \\
&: \text{take place} \\
&: (\ldots) \\
\langle (A \wedge B).\text{noun\_pharse} \rangle &: A \text{ and } B \\
&: A \text{ and also } B \\
&: \text{Both } A \text{ and } B \\
&: \text{That } A \text{ and } B \ \langle \text{occur\_synonyms} \rangle \\
&: (\ldots) \\
\langle \text{cause\_synonyms} \rangle &: \text{cause} \\
&: \text{result in} \\
&: \text{lead to} \\
&: \text{bring about} \\
&: (\ldots) \\
&(\ldots)
\end{aligned}
\tag{D.1}
$$

As can be seen, the templates can be nested deeply, yielding combinatorially diverse linguistic expressions.

Expanding these templates beforehand is intractable due to the combinatorial explosion, so we expand these templates on the fly to randomly sample a single expression at a time. Estimating the exact number of expressions is intractable for the same reason.

We manually crafted several additional English templates per logical formula (i.e., the left-hand sides of (D.1)) compared to those used in FLD, which yield combinatorially more diverse English

expressions. We observed that at least dozens of expressions, including minor variations, are yielded for each formula.

E Details of Experimental Setup

E.0.1 Prevention of Knowledge Forgetting by Recall Adam Optimizer

We employed the Recall Adam (RecAdam) optimizer (Chen et al., 2020), which regularizes parameter updates to prevent them from being too far from the pre-training parameters. Recall Adam stands out for LLM training as it does not require access to the pre-training corpus, which is often inaccessible or too huge to handle, nor does it require changes to the model architecture, and it has a proven track record of usage in language models such as BERT.

E.1 Benchmarks

Table E.7 details the benchmarks used in the experiments.

E.2 Experimental Runs

We show the average and standard deviations over three seeds.

E.3 Computational Resources

The entire experiment, including preliminary ones, took about 1 week x 128 NVIDIA H100 GPUs of our own.

F Results without using Recall Adam

Table F.8 shows the results of LLMs trained without using Recall Adam.

Table E.7: 31 benchmarks used in the experiments. These benchmarks cover a wide range of tasks and are prominent for LLM evaluation. We also show the form of reasoning and the type of knowledge required to solve the problems in each benchmark.

task	benchmark	reasoning form	required knowledge
Logic	bAbi deduction Weston et al. (2015),	deduction	-
	FOLIO Han et al. (2022)		(not required)
	LogicNLI Tian et al. (2021)		
	RobustLR Sanyal et al. (2022a)		
	AR-LSAT Zhong et al. (2021)		commonsense
	LogiQA2 Liu et al. (2023a)		
	ReClor Yu et al. (2020)		
	AbductionRules Young et al. (2022)	abduction	
	ART Bhagavatula et al. (2019)		commonsense
NLI	HELP Yanaka et al. (2019)	validate a conclusion based on given premises	commonsense
	MultiNLI Williams et al. (2018)		
	RTE Dagan et al. (2005); Giampiccolo et al. (2007); Bentivogli et al. (2009)		
	SNLI Bowman et al.		
Math	GSM8k Cobbe et al. (2021)	Math	Math
	MATH Hendrycks et al. (2021b)		
	MathQA Amini et al. (2019)		
Coding	HumanEval Chen et al. (2021)	Coding	Coding
	MBPP Austin et al. (2021)		
	MBPP+ Liu et al. (2023d)		
	MultiPL-E (cpp/go) Cassano et al. (2023)		
Others	CommonsenseQA Talmor et al. (2018)	complicated procedures	commonsense
	HellaSWAG Zellers et al. (2019)		
	SQuAD2 Rajpurkar et al. (2018)		
	WinoGrande Sakaguchi et al. (2021)		
	ARC (easy/challenge) Clark et al. (2018)		science
	GPQA Rein et al. (2023)		
	OpenBookQA Mihaylov et al. (2018)		
	SciQ Welbl et al. (2017)		
aggregated	MMLU Hendrycks et al. (2021a)	various	various
	MMLU-Pro Wang et al. (2024c)		
	BBH Suzgun et al. (2022)		

Table F.8: 5-shot performance of LLMs before and after ALT. ⊕**ALT**-x denotes the LLM trained with ALT on the synthetic logic corpus x from Table 1. Color shows the rank in each column (darker is better). "Logic", "Math", "Code", and "Others" each comprises various benchmarks (see Table E.7). "Avg." represents the micro-average of all the benchmarks. "w/o RecAdam" denotes that LLM was trained without knowledge forgetting prevention by Recall Adam optimizer.

(a) LLaMA-3.1-8B.

	Avg.	Logic	Math	Code	NLI	Others	BBH (3-shot)		BBH (0-shot)		MMLU	
								CoT		CoT		Pro
LLaMA-3.1-8B	47.9	42.8$_{\pm0.4}$	39.6$_{\pm0.5}$	35.4	65.4$_{\pm0.3}$	60.7$_{\pm0.3}$	44.9$_{\pm0.4}$	61.9$_{\pm0.4}$	8.2$_{\pm0.2}$	36.5$_{\pm0.4}$	65.3$_{\pm0.4}$	35.8$_{\pm0.4}$
⊕**ALT**-PRP w/o RecAdam	43.5	39.5$_{\pm0.2}$	29.1$_{\pm0.3}$	35.3	57.8$_{\pm0.2}$	61.0$_{\pm0.2}$	40.5$_{\pm0.2}$	47.0$_{\pm0.2}$	3.9$_{\pm0.1}$	6.3$_{\pm0.1}$	64.9$_{\pm0.2}$	34.0$_{\pm0.2}$
⊕**ALT**-PRP	48.1	43.7$_{\pm0.2}$	39.2$_{\pm0.3}$	35.7	65.6$_{\pm0.2}$	60.8$_{\pm0.2}$	44.9$_{\pm0.2}$	61.8$_{\pm0.2}$	8.2$_{\pm0.1}$	36.4$_{\pm0.2}$	65.3$_{\pm0.2}$	35.3$_{\pm0.2}$
⊕**ALT**-RT	50.1	46.6$_{\pm0.2}$	42.3$_{\pm0.2}$	36.4	68.5$_{\pm0.1}$	61.3$_{\pm0.1}$	46.7$_{\pm0.2}$	63.7$_{\pm0.2}$	13.7$_{\pm0.1}$	38.4$_{\pm0.2}$	65.2$_{\pm0.2}$	35.7$_{\pm0.2}$
⊕**ALT**-FLD	51.7	51.3$_{\pm0.2}$	43.4$_{\pm0.2}$	37.9	69.9$_{\pm0.1}$	61.5$_{\pm0.1}$	46.9$_{\pm0.2}$	64.8$_{\pm0.2}$	12.1$_{\pm0.1}$	39.5$_{\pm0.2}$	65.4$_{\pm0.2}$	36.3$_{\pm0.2}$
⊕**ALT**-FLD$^{\times2}$	52.1	52.3$_{\pm0.2}$	43.2$_{\pm0.2}$	38.1	70.7$_{\pm0.1}$	61.5$_{\pm0.1}$	46.6$_{\pm0.3}$	65.4$_{\pm0.2}$	11.4$_{\pm0.1}$	39.0$_{\pm0.3}$	65.5$_{\pm0.2}$	36.4$_{\pm0.2}$

(b) LLaMA-3.1-70B.

	Avg.	Logic	Math	Code	NLI	Others	BBH (3-shot)		BBH (0-shot)		MMLU	
								CoT		CoT		Pro
LLaMA-3.1-70B	60.0	57.4$_{\pm0.4}$	60.0$_{\pm0.5}$	46.2	73.7$_{\pm0.3}$	67.7$_{\pm0.3}$	60.4$_{\pm0.3}$	82.1$_{\pm0.2}$	6.5$_{\pm0.1}$	50.1$_{\pm0.3}$	78.7$_{\pm0.3}$	50.7$_{\pm0.4}$
⊕**ALT**-PRP$_{\text{w/o RecAdam}}$	58.8	54.3$_{\pm0.4}$	59.2$_{\pm0.5}$	48.2	72.7$_{\pm0.3}$	65.9$_{\pm0.3}$	60.4$_{\pm0.4}$	81.5$_{\pm0.3}$	6.1$_{\pm0.2}$	48.3$_{\pm0.4}$	78.5$_{\pm0.3}$	50.7$_{\pm0.4}$
⊕**ALT**-PRP	60.4	57.7$_{\pm0.4}$	59.8$_{\pm0.5}$	49.2	73.5$_{\pm0.3}$	67.6$_{\pm0.3}$	60.4$_{\pm0.4}$	82.2$_{\pm0.3}$	6.0$_{\pm0.2}$	50.1$_{\pm0.4}$	78.7$_{\pm0.3}$	50.9$_{\pm0.4}$
⊕**ALT**-RT	62.7	61.4$_{\pm0.2}$	62.1$_{\pm0.3}$	50.8	75.4$_{\pm0.2}$	68.4$_{\pm0.2}$	64.1$_{\pm0.3}$	82.5$_{\pm0.2}$	11.5$_{\pm0.2}$	59.2$_{\pm0.3}$	79.0$_{\pm0.2}$	52.4$_{\pm0.3}$
⊕**ALT**-FLD	64.2	65.7$_{\pm0.1}$	63.6$_{\pm0.2}$	52.0	75.3$_{\pm0.1}$	68.5$_{\pm0.1}$	65.0$_{\pm0.2}$	83.6$_{\pm0.1}$	12.1$_{\pm0.1}$	59.9$_{\pm0.2}$	79.3$_{\pm0.1}$	54.4$_{\pm0.2}$
⊕**ALT**-FLD$^{\times2}$	64.4	66.1$_{\pm0.1}$	63.3$_{\pm0.2}$	52.4	76.1$_{\pm0.1}$	68.5$_{\pm0.1}$	65.4$_{\pm0.2}$	83.6$_{\pm0.2}$	11.4$_{\pm0.1}$	60.8$_{\pm0.2}$	79.5$_{\pm0.1}$	54.4$_{\pm0.2}$

Table F.9: Benchmark-wise 5-shot performance of LLaMA-3.1-8B before and **after** ALT on FLD$^{\times2}$.

(a) Logic.

	bAbiD	FOLIO	LogicNLI	RobustLR	AR-LSAT	LogiQA	ReClor	AbductionR	ART
LLaMA-3.1-8B	48.7$_{\pm1.6}$	50.0$_{\pm1.6}$	28.5$_{\pm1.0}$	43.2$_{\pm0.9}$	20.7$_{\pm1.0}$	39.6$_{\pm1.2}$	28.7$_{\pm0.7}$	52.4$_{\pm0.9}$	73.4$_{\pm1.1}$
⊕**ALT**-FLD$^{\times2}$	**55.3**$_{\pm0.7}$	**54.5**$_{\pm0.7}$	**42.2**$_{\pm0.5}$	**63.5**$_{\pm0.4}$	**21.1**$_{\pm0.5}$	**42.8**$_{\pm0.6}$	**29.4**$_{\pm0.3}$	**85.9**$_{\pm0.3}$	**76.1**$_{\pm0.5}$

(b) Math.

	GSM8k			MATH	MathQA
	CoT	CoT (0-shot)		-	-
LLaMA-3.1-8B	50.2$_{\pm1.4}$	51.5$_{\pm1.4}$	39.5$_{\pm1.3}$	14.1$_{\pm0.5}$	42.8$_{\pm0.9}$
⊕**ALT**-FLD$^{\times2}$	**53.4**$_{\pm0.6}$	**56.4**$_{\pm0.6}$	**48.8**$_{\pm0.6}$	**14.3**$_{\pm0.2}$	**43.2**$_{\pm0.4}$

(c) Coding.

	HumanEval	MBPP	MBPP+	MultiPL-E (cpp)	MultiPL-E (go)
LLaMA-3.1-8B	22.6	31.6	38.1	21.7	63.0
⊕**ALT**-FLD$^{\times2}$	**25.0**	**33.8**	**39.8**	**23.0**	**68.8**

(d) Natural language inference (NLI).

	HELP	MNLI	RTE	SNLI
LLaMA-3.1-8B	46.4$_{\pm0.5}$	68.1$_{\pm0.5}$	74.6$_{\pm0.9}$	72.6$_{\pm0.4}$
⊕**ALT**-FLD$^{\times2}$	**47.5**$_{\pm0.2}$	**75.2**$_{\pm0.2}$	**83.4**$_{\pm0.3}$	**76.8**$_{\pm0.2}$

(e) Others.

	CommonsenseQA	HellaSwag	SQuAD	WinoGrande	ARCe	ARCc	GPQA	OpenBookQA	SciQ
LLaMA-3.1-8B	73.9$_{\pm1.3}$	61.2$_{\pm0.5}$	30.8$_{\pm0.0}$	77.4$_{\pm1.2}$	84.2$_{\pm0.7}$	54.7$_{\pm1.5}$	31.1$_{\pm1.3}$	35.3$_{\pm0.7}$	**97.7**$_{\pm0.5}$
⊕**ALT**-FLD$^{\times2}$	**75.0**$_{\pm0.6}$	**61.5**$_{\pm0.2}$	**33.5**$_{\pm0.0}$	**78.1**$_{\pm0.5}$	**85.0**$_{\pm0.3}$	**55.6**$_{\pm0.6}$	**31.2**$_{\pm0.6}$	**36.3**$_{\pm0.3}$	97.6$_{\pm0.2}$

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist principles carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and principles below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Claims stated in Section 1 is supported by the experimental results in Sections 5, 6.

   principles:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: Appendix B

principles:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: Our paper does not include theoretical results.

   principles:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4, appendix E. Further, we release all the resources, including (i) the corpus, (ii) the trained model, and (iii) code for corpus generation, LLM training, and LLM evaluation [3].

[3] https://anonymous.4open.science/r/ALT/README.md

principles:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: we release the code, data, and model.

   principles:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission principles (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission principles (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4, appendix E.

   principles:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: As stated in Appendix E.

   principles:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96 % CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix E.3.

   principles:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/Ethicsprinciples`?

   Answer: [Yes]

   Justification:

   principles:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix C

    principles:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification:

principles:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage principles or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the corpora and benchmarks used in the experiments properly state their licenses.

    principles:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We will release our corpus.

    principles:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human objects.

principles:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: THis paper does not involve crowdsourcing nor research with human objects.

    principles:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the principles for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.