Meta-reasoning Induced Generalization in Language Models via Parameter-Efficient Fine-Tuning

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have demonstrated strong logical reasoning capabilities. However, prior work has largely underexplored the impact that distinct meta-reasoning paradigms (e.g. abduction) can have on generalizability. This leaves a crucial gap in understanding how training on trajectories of different meta-reasoning types can affect an LLM's reasoning capabilities on real-world tasks. In this study, we systematically investigate this problem, focusing on various Parameter-Efficient Fine-Tuning (PEFT) domain adaptation strategies across three meta-reasoning types: *deductive*, *inductive*, and *abductive*. Specifically, our methodology involves generating high-quality meta-reasoning trajectories using a larger teacher model, based on procedurally-generated logical questions. We then apply PEFT to smaller models, using these trajectories as the fine-tuning data. Extensive experiments reveal that targeted fine-tuning on synthetic meta-reasoning tasks offers strong generalizability with substantial performance gains across a comprehensive set of out-of-domain reasoning benchmarks, even when the target task does not explicitly align with the fine-tuning meta-reasoning paradigm in logical structure and world knowledge.[1]

1 Introduction

Large Language Models (LLMs) have demonstrated strong reasoning performance (Wei et al., 2022; Kojima et al., 2022; Yao et al., 2023). However, while they can generate plausible reasoning for tasks like mathematics and coding, their performance gains are often limited across a wider spectrum of reasoning scenarios (Sprague et al., 2025).

[1]Our code is available at https://anonymous.4open.science/r/MrIG.

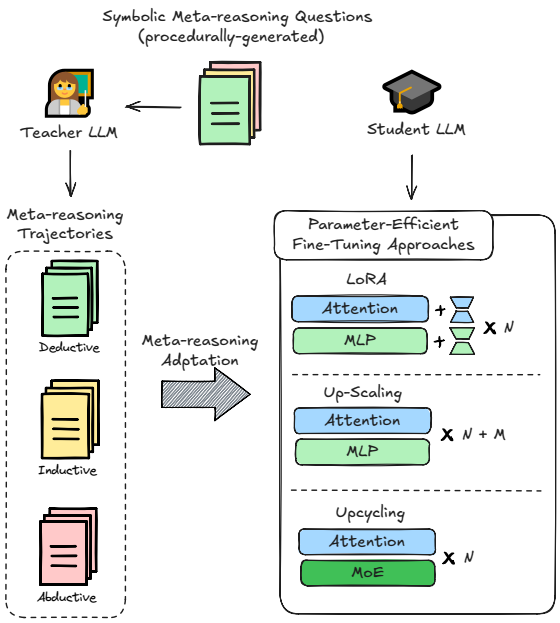

Figure 1: We apply PEFT on meta-reasoning trajectories to elicit deductive, inductive, and abductive reasoning.

It is still poorly understood, in fact, how fundamental reasoning paradigms, including deduction, induction, and abduction, and their associated logical structures, can form the foundation for complex and generalizable reasoning behavior in LLMs (Hu et al., 2025). Dougrez-Lewis et al. (2025) define abduction, deduction and induction as fundamental atomic reasoning types, the building blocks of reasoning. These are intrinsically linked: (1) *deduction* infers conclusions based on premises, rules and observations; (2)*abduction* infers the most plausible conclusion from a set of candidate hypotheses, rules and observations; and (3) *induction* abstracts rules based on patterns in observations and hypotheses; As such, they can constitute the basis to investigate how reasoning generalization can be obtained from LLMs. We will refer to them throughout as meta-reasoning types or paradigms.

Our work fills a crucial gap in understanding

how training on different trajectories, at a varying level of abstraction, can affect reasoning capabilities of LLMs on real-world tasks. To this end, we systematically evaluate a battery of Parameter-Efficient Fine-Tuning (PEFT) domain adaptation methods (Rebuffi et al., 2017; Houlsby et al., 2019; Pfeiffer et al., 2021) for imbuing LLMs with these three meta-reasoning paradigms. Our methodology involves two main steps: 1) we create a high-quality dataset for fine-tuning by sampling trajectories from a larger, powerful model such as *Llama-3.3-70B-Instruct* (Grattafiori et al., 2024), using procedurally-generated, symbolic meta-reasoning questions; 2) we apply a range of PEFT methods to two smaller base models, *Llama-3.1-8B-Instruct* (Grattafiori et al., 2024) and *Qwen3-8B* (Yang et al., 2025a), to rigorously evaluate their in-domain and out-of-domain (OOD) transfer performance on a comprehensive suite of reasoning benchmarks.

Our experimental results show that fine-tuning on synthetic, symbolic questions targeting meta-reasoning paradigms provides substantial performance gains on OOD reasoning benchmarks. Such OOD generalization is still observable even when the model is fine-tuned on meta-reasoning tasks that differ from the test set in logical structure and concrete world knowledge.

Regarding PEFT strategies, we find them to be model-dependent: for *Llama-3.1-8B-Instruct*, a model expansion method (Wu et al., 2024) yields better generalization compared to others, while an upcycling Mixture-of-Experts (MoE) (Komatsuzaki et al., 2023) approach outperforms others for *Qwen3-8B*. Our in-depth qualitative analysis suggests that training on meta-reasoning trajectories is the key to enabling such a generalizable behavior across different methods.

Overall, our contributions are as follows:

- We construct a fine-tuning dataset of reasoning trajectories spanning three meta-reasoning types: deductive, inductive, and abductive.
- We perform the first systematic study of PEFT domain adaptation methods applied specifically to the three meta-reasoning types.
- We provide an extensive evaluation of the models' OOD transfer performance across reasoning benchmarks, demonstrating that fine-tuning on meta-reasoning trajectories yields substantial generalized performance gains and identifying the optimal, model-dependent PEFT strategy for transferring these abilities.

2 Related Work

2.1 Logical Reasoning with LLMs

Chain-of-Thought (CoT) enables Large Language Models (LLMs) to perform multi-step reasoning, yielding substantial performance gains across various benchmarks, particularly in the mathematical domain (Wei et al., 2022; Kojima et al., 2022; Yao et al., 2023). Large Reasoning Models (LRMs), such as DeepSeek-R1 and QwQ, integrate multi-step reasoning as a built-in behavior through extensive post-training (Qwen-Team, 2025; DeepSeek-AI, 2025). However, the efficacy of CoT reasoning is not guaranteed across a wider spectrum of reasoning settings (Sprague et al., 2025; Dougrez-Lewis et al., 2025; Lewis-Lim et al., 2025). In fact Dougrez-Lewis et al. (2025) showed that LLMs perform poorly on abductive reasoning.

Recent work has investigated how learning from certain reasoning types impacts the effectiveness and efficiency of reasoning LLMs. Hu et al. (2025) merge three LLMs, which are fine-tuned on corresponding "meta-abilities" via reinforcement learning, but their evaluation predominantly focuses on deductive math and code reasoning, leaving the comprehensive acquisition of "meta reasoning abilities" inconclusive. Tan et al. (2025) propose to enhance logical reasoning by fine-tuning on trajectories generated from Symbolic-ReAct, demonstrating improved OOD performance on questions requiring both deductive and abductive reasoning. However, they only use synthetic deductive reasoning data for fine-tuning, without exploring the influence of training on other reasoning types. In this study, we investigate various domain adaptation approaches for three reasoning types and systematically evaluate them on a comprehensive set of reasoning datasets.

2.2 PEFT for Domain Adaptation

Adapting LLMs on data for different reasoning types with thinking trajectories is challenging. Direct fine-tuning with target domain data often leads to catastrophic forgetting (Ramasesh et al., 2022; Kotha et al., 2024; Luo et al., 2025). Although originally designed for efficiency, Parameter-Efficient Fine-Tuning (PEFT) methods demonstrate effectiveness in domain adaptation (Rebuffi et al., 2017; Houlsby et al., 2019; Pfeiffer et al., 2021). Hu et al.

(2022) proposed Low-Rank Adaptation (LoRA) to update on a pair of low-rank matrices instead of full-rank linear layer, which substantially reduces the number of trainable parameters. However, recent studies suggest that LoRA's limited capacity hinders its effectiveness in domain adaptation (Biderman et al., 2024; Pletenev et al., 2025). A growing trend in recent PEFT work focuses on depth up-scaling and upcycling dense models to Mixture-of-Experts (MoE).

Progressive Depth Up-scaling. Depth up-scaling involves inserting new layers into a pre-trained LLM to reduce training time and computational overhead (Chen et al., 2022; Wang et al., 2024; Yao et al., 2024). Recent work on depth up-scaling only updates the newly-added layers, (i.e., progressive depth up-scaling). It preserves the parametric knowledge of base layers while allowing new knowledge to be learned in the added layers (Choudhury et al., 2025; Hennara et al., 2025). The new layers are typically initialized using information from existing layers (Wu et al., 2024; Yang et al., 2025b; Cao et al., 2025). Progressive depth up-scaling, exemplified by LLaMAPRO (Wu et al., 2024), is widely used in cross-lingual domain adaptation (Dou et al., 2025; Lee et al., 2025; Lim et al., 2025).

Upcycling. Upcycling refers to transferring a pre-trained dense layer into an MoE layer. During upcycling, the vanilla Multi-Layer Perceptron (MLP) is expanded into multiple experts and a router. The newly-added experts can be initialized by random initialization(Fedus et al., 2022; Nakamura et al., 2025) or duplication from the existing MLP (Komatsuzaki et al., 2023). Recent work shows that training individual experts with domain-specific data while freezing other experts can mitigate catastrophic forgetting in both multi-lingual (Zhou et al., 2025; Zhang et al., 2025) and multi-domain settings (Li et al., 2025b).

3 Synthetic Data Collection

Our main aim is to systematically investigate how to effectively adapt LLMs across three meta-reasoning types: deductive, inductive, and abductive. For that purpose, we first sample high-quality meta-reasoning trajectories from a larger teacher LLM for each of these types, using procedurally-generated logical reasoning questions as input.

3.1 Meta-reasoning Tasks

We adapt three types of meta-reasoning tasks from Hu et al. (2025), which elicit deductive, inductive, and abductive reasoning respectively. The questions and gold answers in these tasks are procedurally-generated. Our modification to these tasks is mainly instruction and answer descriptions refinement to minimize ambiguity and enable automated evaluation (Figure 2). We opt for using these procedurally-generated tasks because they represent the fundamental, symbolic structure of their respective logical reasoning types. By eliminating semantically loaded text and presenting reasoning problems in their most basic form, we prevent models from relying on context bias and, instead, maximize their capacity for generalization to OOD tasks.

Deductive Task. The deductive questions are constructed by randomly generating Boolean Satisfiability (SAT) problems. These problems feature a Boolean formula in Conjunctive Normal Form (CNF), where a random set of variables are linked using logical connectives such as implication, conjunction, and disjunction. The objective for the model is to provide a truth value to the variables that satisfy the formula. All generated questions in this dataset are guaranteed to be satisfiable and possess a unique satisfying assignment. SAT is highly relevant to deductive reasoning as traditional automated reasoning research often transforms theorem proving or logical entailment into SAT problems (Brown, 2013; Richardson and Sabharwal, 2021). To solve a SAT question, the model needs to strictly follow logical inference rules to find a consistent assignment under the assumption that the entire formula is true. The original questions by Hu et al. (2025) list the formulas separately without expressing them in a CNF, causing confusion for LLMs. Furthermore, their gold answers are presented as plain text descriptions, which complicates automated evaluation. We restructure the formulas into CNFs and express the gold answers in JSON.

Inductive Task. Inductive questions are generated by constructing number sequences based on a hidden, repeating pattern. This pattern involves randomly applying addition, subtraction, and multiplication operations to the preceding number in the sequence. The sequence of operations repeats after a randomly determined period. The model needs to predict the next number in the sequence after ob-

Deductive Task

This is a <Deductive> reasoning task. Below are some formulas connected by conjunctions:
$\neg(((\neg A \wedge A) \vee \neg(F)))$
$\wedge((\neg(\neg E) \vee \neg(\neg B)) \oplus ((H \wedge F) \rightarrow (H \leftrightarrow F)))$
$\wedge\neg(((H \rightarrow A) \vee (\neg G \oplus D)))$
$\wedge(((F \wedge C) \wedge (G \oplus \neg G)) \leftrightarrow ((D \leftrightarrow A) \wedge (F \wedge G)))$
$\wedge((\neg(C) \wedge (\neg F \rightarrow D)) \oplus \neg((\neg B \leftrightarrow \neg F)))$
Please list the truth value of each variable to make the whole conjunction true using a JSON dictionary, which maps variable names to their truth values, then enclose the answer in <answer><answer>. Please put all the intermediate reasoning steps in <think><think>.

Gold Answer:
{ "A": "False",
"B": "False",
"C": "True",
"D": "False",
"E": "False",
"F": "True",
"G": "True",
"H": "True"
}

Inductive Task

This is a <Inductive> reasoning task. Given the following sequence, ['5', '10', '6', '12', '15', '30', '26', '52', '55', '110', '?']
What is the value at the question mark? Please enclose the answer in <answer><answer>, and put all the intermediate reasoning steps in <think><think>.

Gold Answer: 106

Abductive Task

Premises:
((((NOT O) OR ((NOT O) OR M)) AND (((NOT I) OR O) OR O))) => I,
(M) => O,
((((I OR (NOT I)) AND ((NOT O) OR (NOT I))) AND ((M AND M) AND ((NOT I) AND (NOT O))))) => M,
.....
Known Atoms: [J, O, K, M, F, I]
Goals: [D, M, C]
Instruction: This is a <Abductive> reasoning task. For each goal, identify which premises directly lead to the goal. Then, trace back what the true value of the atoms must be to make each of the goal true. Only the atoms in the 'known atoms' are known but their values are not shown. Finally, return the reachable goals with the true values of the known atoms that make it true. Please enclose the final answer with <answer><answer>. All the intermediate thinking steps should be enclosed in <think><think> tags.

Gold Answer:
Goal 'M' is reachable. Here are the possible assignments to make it true:
Solution 1: {"F": true, "I": true, "J": false, "K": true, "O": true}
Solution 2: {"F": true, "I": true, "J": true, "K": true, "O": true}

Figure 2: Questions and golden answers in three meta-reasoning tasks.

serving the initial sequence of the array. This task reflects a basic form of inductive reasoning as the model must infer the underlying arithmetic rules (the pattern) from the given examples to predict the next value. Compared to the original questions, we only add instructions to make LLMs enclose the answer in "*<answer>*" tags for easier automated evaluation.

Abductive Task. To generate abductive questions, we use a set of logic formulas and known atoms whose truth values are hidden from the model. The objective is to first identify all reachable goals from the atoms by checking whether there are formulas related to the goals, and then trace backward to establish the truth values of the known atoms for the goals to be true. This setup embodies abductive reasoning because the model must work backward from an assumed conclusion (that the goals are true) to hypothesize the most likely set of premises (the antecedent truth values) that would lead to it. However, the original task instructions by Hu et al. (2025) fail to clarify that the known atoms do not necessarily equate to true atoms, leading to misinterpretations by LLMs. Moreover, their gold answers only check goal reachability despite asking LLMs to identify related premises and trace back the truth values of the known atoms. This oversimplifies the task and makes it no longer about abduction, because the model can simply check whether the goals appear in the premises. By contrast, we find the possible truth values of the known atoms that makes the reachable goal true using Prolog (Wielemaker et al., 2012) and add them to the gold answer.

3.2 Reasoning Trajectories Sampling

We sample reasoning trajectories for each of the inductive, deductive and abductive taks using a large *Llama-3.3-70B-Instruct* model to build our fine-tuning dataset. As this model substantially exceeds the size of the smaller models targeted for fine-tuning, its generated trajectories are assumed to provide valuable, high-quality insights for the smaller models to learn. The original dataset in Hu et al. (2025) has 3,600 deductive questions, 4,500 inductive questions, and 9,000 abductive questions. We take 100 questions from each task as test sets and another 100 as development sets, and use the rest for fine-tuning. After adding clearer instructions and answer descriptions, we sample 5 trajectories for each question. The questions are input to the LLM in their original dataset form (details in Appendix C) without supplementary instructions or examples. For efficiency, we used the vLLM offline inference API (Kwon et al., 2023). Table 1 shows the statistics of the sampled trajectories.

Data	# Quest.	# Traject.	# Tokens	Avg. Tokens
Deductive	3,400	17,000	18.6M	1,093
Inductive	4,300	21,500	11.7M	544
Abductive	8,800	44,000	51.2M	1,164

Table 1: Synthetic meta-reasoning data statistics.

4 Parameter Efficient Meta-reasoning Adaptation

For meta-reasoning adaptation, we employ LoRA, Depth Up-scaling and Upcycling MoE for PEFT following related work (Wu et al., 2024; Zhou et al., 2025; Zhang et al., 2025). We keep the trainable parameters of each method at a similar size for a fair comparison, updating only the newly-added parameters.

LoRA (Hu et al., 2022). For a weight matrix $W \in \mathbb{R}^{d\times k}$, LoRA injects a low-rank decomposition $\Delta W = BA$, where $A \in \mathbb{R}^{r\times k}$ and $B \in \mathbb{R}^{d\times r}$.

Depth Up-scaling. Let $\mathcal{M}$ be a LLM composed of n Transformer layers f, parameterized by θ. Depth Up-scaling inserts $k - n$ of new layers f' into $\mathcal{M}$, resulting in $\mathcal{M}'$ with k layers:

$$\mathcal{M}'(x; \theta, \phi) = f_\theta^{(k)} \circ f_\phi'^{(k-1)} \circ \cdots \circ f_\theta^{(1)}(x)$$

We adopt LLaMAPro (Wu et al., 2024) for depth up-scaling. $\mathcal{M}$ is divided into g groups and inserts p new layers atop each group. f'_ϕ are initialized by copying weights from the top-p base layers within respective group, but set Attention and MLP output projection matrices to zero.

Upcycling. Upcycling converts the MLP block of a Transformer layer into a MoE block consisting of a router R and K experts MLP blocks:

$$f_{\text{MLP}}(h) \;\Rightarrow\; f_{\text{MoE}}(h) = \sum_{k=1}^{K} R_k(h) f_{\text{MLP},k}(h)$$

We employ Sparse-Upcycling (Komatsuzaki et al., 2023), initializing the new expert MLPs by copying the original pre-trained MLP weights.

5 Experimental Setup

5.1 Models

We conduct extensive experiments using the 32-layer *Llama-3.1-8B-Instruct*, aligning with prior domain adaptation studies (Hoffmann et al., 2025; Choudhury et al., 2025; Lim et al., 2025). We also include the 36-layer *Qwen3-8B* to investigate performance across different architectures.

5.2 Baselines

We compare PEFT methods to (1) the vanilla base model without fine-tuning; and (2) a full fine-tuning baseline that directly updates all model parameters.

5.3 Evaluation

In-Domain. We evaluate the test set of the meta-reasoning data (Section 3.1) to investigate these skills are acquired during training. Each of the three tasks have 100 test questions.

Out-of-Domain. We also include the following datasets to evaluate how well the meta-reasoning abilities generalize. These datasets differ from the fine-tuned meta-reasoning tasks, but require certain meta-reasoning abilities (in square brackets).

- RECV (Dougrez-Lewis et al., 2025) **[deductive+abductive]** contains claims from twitter or Wikipedia and requires the model to determine whether the claim is true or false.
- FOLIO (Han et al., 2024a) **[deductive]** contains logical reasoning questions in natural language. Given the context, which is equivalent to a set of first-order logic formulas, the model need to determine a statement is true, false, or uncertain (cannot be proved or disproved).
- αNLI (Zhao et al., 2023) **[abductive]** consists of multi-choice question for choosing the most probable explanation given the context.
- True detective (Del and Fishel, 2023) **[abductive]** contains detective puzzles with multi-choice questions.
- WinoWhy (Zhang et al., 2020) **[abductive]** includes explanations for the questions in the Winograd Schema Challenge, and requires the model to distinguish plausible reasons from very similar but wrong reasons.

Due to the complex answer format, models sometimes fail to generate in the exact format that our parser can process. To address this issue, we ask *Qwen3-30B-A3B-Instruct-2507* (Yang et al., 2025a) to compare the generated answer with the gold answer. We simply instruct it to check whether the generation is consistent with the gold label and return *TRUE* or *FALSE*. The full version of the prompt is detailed in Appendix **??**.

5.4 Implementation details

We apply LoRA to all linear layers, with ranks set to 1,024 for *Llama-3.1-8B-Instruct* and 512 for *Qwen3-8B*. We set $g = 4$ and $p = 3$ for both *Llama-3.1-8B-Instruct* and *Qwen3-8B*. The expanded models have 44 and 48 layers, respectively. We upcycle dense layers with odd indices into MoE layers, adding one expert MLP to each. During training the global batch size and sequence length are set to 64 and 2,048, respectively. We use a maximum learning rate of $5e-6$ (degraded to $5e-7$) for training all methods. We fine-tune models using TRL (von Werra et al., 2020) on a single NVIDIA A100 (80GB). We employ Flash-Attention 2 (Dao, 2024) and mixed-precision `bf16` for accelerated training. During evaluation, the maximum number

of generated tokens is set to 4,096. We use Transformers (Wolf et al., 2020) for reproducible output generation and vLLM (Kwon et al., 2023) for efficient evaluation. The generation of the training data used about 280 hours of an AMD MI300X.

6 Results

Table 2 presents the in- and out-of-domain results obtained by applying PEFT methods on different base models for meta-reasoning ability adaptation.

6.1 In-Domain

Meta-reasoning trajectories benefit instruct models. The vanilla *Llama-3.1-8B-Instruct* exhibits limited in-domain capacity across inductive, deductive and the abductive tasks (13.00/32.00/2.00). However, fine-tuning the model substantially enhances its performance. For example, full FT on corresponding meta-reasoning tasks improves all in-domain tasks (25.00/57.00/6.00). In 28 out of 36 cases, the FT model outperforms the vanilla baseline.

Meta-reasoning trajectories harm thinking model. *Qwen3-8B* exhibits strong in-domain performance (82.00/68.00/36.00) largely due to its thinking mode. Surprisingly, fine-tuning the model leads to substantial in-domain performance degradation. For instance, full FT on the inductive task decreases performance across all in-domain tasks (69.00/58.00/24.00). We hypothesize this is because the training trajectories, sampled from Llama-3.3-70B-Instruct, conflicts with its native reasoning style. This observation aligns with recent work documenting cross-model trajectory mismatch (Wen et al., 2025) and the fragility of trajectory-based reasoning (Li et al., 2025a).

Meta-reasoning abilities correlate. We observe a strong correlation among different meta-reasoning tasks. For *Llama-3.1-8B-Instruct*, fine-tuning on one task generally results in an overall performance increase across tasks. For example, deductive fine-tuning boosts all three tasks, regardless of the adaptation method. Conversely, *Qwen3-8B* model exhibits the opposite pattern: fine-tuning on a single task causes overall in-domain performance degradation in most cases. This behavior highlights a strong interconnection between distinct meta-reasoning abilities.

6.2 Out-of-Domain

Meta-reasoning trajectories offer OOD generalization regardless of in-domain performance. We observe consistent OOD performance improvements after fine-tuning *Llama-3.1-8B-Instruct*. For example, Depth Up-scaling with deductive fine-tuning wins four out of five OOD benchmarks compare to the vanilla model (αNLI: 76.33 vs 73.98, Winowhy: 59.13 vs 49.01, Folio: 58.62 vs 53.20, RECV: 75.07 vs 71.93). The same performance gains also occur when using LoRA for deductive fine-tuning and Depth Up-scaling for inductive fine-tuning. Compared to the vanilla model, eight out of 12 fine-tuned models show average performance gains in OOD benchmarks. These results verify that the improved performance of in-domain meta-reasoning tasks can help the model generalize effectively to the OOD benchmarks.

Interestingly, although *Qwen3-8B* suffers from an in-domain performance degradation, eight out of 12 fine-tuned models show a performance gain in certain OOD benchmarks (i.e. αNLI and RECV). For instance, upcycling with inductive fine-tuning wins four out of five OOD benchmarks compare to the vanilla model (Detect: 43.98 vs 39.27, αNLI: 81.86 vs 81.30, Winowhy: 75.74 vs 71.83, Folio: 72.20 vs 71.40). Notably, upcycling consistently outperforms the vanilla model on average performance of OOD benchmarks when fine-tuning on meta-reasoning tasks (70.32/69.75/69.38 vs 68.62). This observation highlights that even when in-domain performance degrades, adapting meta-reasoning abilities using reasoning trajectories can still benefit OOD logical reasoning.

No consistent best-performing PEFT method across settings. We observe that Depth Up-scaling achieves the best overall OOD performance for *Llama-3.1-8B-Instruct* with both inductive (58.03) and deductive fine-tuning (58.53). It achieves the second-best performance with abductive fine-tuning (56.31) among all methods (best: 56.89). Regarding *Qwen3-8B*, the best-performing method is Upcycling, achieving the highest overall OOD performance (Inductive: 70.32, Deductive: 69.75, Abductive: 69.38). These observations demonstrate that the effectiveness of PEFT methods varies across settings and models. This is supported by recent work that shows the performance difference when applying PEFT for different tasks and model structures (Han et al., 2024b).

Method	FT	Train/Total	In-Domain			Out-of-Domain					
			Deduct	Induct	Abduct	Detect	αNLI	Winowhy	Folio	RECV	Avg
Llama-3.1-8B-IT	-	0/8.03	13.00	32.00	2.00	28.80	73.98	49.01	53.20	71.93	55.38
Full FT	Induct	8.03/8.03	**16.00**	57.00	6.00	19.37	79.41	25.34	**60.10**	78.53	52.55
LoRA		3.73/10.71	11.00	**59.00**	**15.00**	17.80	**79.73**	34.03	55.17	**80.07**	53.36
Up-Scaling		3.67/10.65	13.00	56.00	11.00	**24.61**	79.11	**49.21**	59.61	77.60	**58.03**
Upcycling		3.87/10.85	12.00	53.00	8.00	24.08	78.59	43.35	55.17	76.80	55.60
Full FT	Deduct	8.03/8.03	25.00	**39.00**	**10.00**	**26.18**	**77.38**	46.67	**58.62**	**78.60**	57.49
LoRA		3.73/10.71	33.00	37.00	3.00	24.00	77.18	53.26	57.64	78.33	58.00
Up-Scaling		3.67/10.65	**36.00**	35.00	4.00	23.50	76.33	**59.13**	**58.62**	75.07	**58.53**
Upcycling		3.87/10.85	34.00	33.00	8.00	25.13	70.19	21.08	54.68	73.27	48.87
Full FT	Abduct	8.03/8.03	13.00	28.00	6.00	**21.47**	75.74	**47.40**	**58.13**	**81.73**	**56.89**
LoRA		3.73/10.71	10.00	**36.00**	**10.00**	18.85	75.91	47.12	56.65	79.87	55.68
Up-Scaling		3.67/10.65	13.00	**36.00**	6.00	**21.47**	77.21	47.16	56.65	79.07	56.31
Upcycling		3.87/10.85	**14.00**	29.00	9.00	19.90	**77.35**	31.87	54.19	78.20	52.30
Qwen3-8B	-	0/8.19	82.00	68.00	36.00	39.27	81.30	71.83	79.31	71.40	68.62
Full FT	Induct	8.19/8.19	69.00	58.00	24.00	36.13	78.88	57.35	75.37	71.87	63.92
LoRA		3.88/10.83	62.00	60.00	15.00	31.94	78.23	40.91	67.49	72.13	58.14
Up-Scaling		3.56/10.51	32.00	57.00	**40.00**	39.79	**82.09**	68.69	73.40	**72.67**	67.33
Upcycling		3.96/10.91	**74.00**	**72.00**	31.00	**43.98**	81.86	**75.74**	**77.83**	72.20	**70.32**
Full FT	Deduct	8.19/8.19	36.00	65.00	**19.00**	30.37	81.53	67.68	71.92	70.53	64.41
LoRA		3.88/10.83	28.00	57.00	18.00	30.37	**83.23**	56.54	80.30	71.00	64.29
Up-Scaling		3.56/10.51	28.00	62.00	12.00	30.89	74.40	53.72	**82.76**	70.00	62.35
Upcycling		3.96/10.91	**72.00**	**71.00**	13.00	**40.31**	82.80	**74.35**	79.80	**71.47**	**69.75**
Full FT	Abduct	8.19/8.19	34.00	**69.00**	**18.00**	40.31	**83.92**	68.55	70.94	74.00	67.54
LoRA		3.88/10.83	44.00	64.00	**18.00**	38.74	82.05	54.66	71.92	**74.07**	64.29
Up-Scaling		3.56/10.51	44.00	60.00	13.00	36.65	81.92	45.10	71.92	73.47	61.81
Upcycling		3.96/10.91	**53.00**	68.00	14.00	**47.12**	83.72	**69.08**	**75.37**	71.60	**69.38**

Table 2: Accuracy of the fine-tuned models on the meta-reasoning tasks. **Method** indicates the base model and the applied PEFT methods. **FT** shows the meta-reasoning task used for fine-tuning. **Train/Total** denotes the number of trainable and total parameters. The best and second-best methods for each meta-reasoning task are indicated in **bold** and underlined, respectively. Scores that exceed the *base* model performance are shown in green.

PEFT methods contribute to OOD generalization. Compared to *Llama-3.1-8B-Instruct* (39 out of 60 cases), OOD generalization is less frequent when fine-tuning *Qwen3-8B* (27 out of 60 cases). However, a clear pattern emerges: OOD generalization occurs more frequently when applying Upcycling to *Qwen3-8B*. Upcycling wins at least three out of five OOD benchmarks compared to other methods, offering more substantial improvements compared to other methods. For example, when conducting inducitve fine-tuning and deductive fine-tuning, Upcycling shows substantial performance gains on Detect (43.98/40.31, second-best: 39.79/30.89) and Winowhy (75.74/74.35, second-best: 68.69/67.68). These findings highlight that carefully selecting a PEFT method can yield better OOD generalization.

OOD generalization is task-related. Fine-tuning *Llama-3.1-8B-Instruct* leads to OOD generalization on specific benchmarks. Regardless of the PEFT methods, the fine-tuned models offer substantial performance gains in αNLI, Folio, and RECV compared to the vanilla model (11 out of 12 cases). A similar observation presents in *Qwen3-8B*, even though degradation occurs in-domain. Fine-tuned variants offer OOD generalization in 9 out of 12 cases on αNLI and Folio, and 6 out of 12 cases in Detective. We hypothesize this is because different logical meta-reasoning tasks offer a more diverse reasoning skill set. This hypothesis is supported by work documenting that task diversity enables transferable generalization when underlying representations are shared (Tripuraneni et al., 2020), and the overlapping task-specific neurons across tasks leads to better generalization (Li et al., 2025a).

7 Qualitive Analysis

In-Domain Examples. Example 1 displays the output of the inductive full FT *Llama-3.1-8B-Instruct* for an inductive problem that the vanilla model fails to answer. It answers correctly and capable of multi-step reasoning. Interestingly, we

observe an "AHA! moment": the model performs self-reflection to identify the correct pattern. After two attempts, the model realizes the previous approach is too complex and derives the correct pattern on its third attempt. While previous work highlights the role of reinforcement learning in the emergence of "AHA!" moments (DeepSeek-AI, 2025; Hu et al., 2025), our observation suggests they may also arise even when reinforcement learning is not explicitly involved.

Example 1: Inductive-Tuned Llama

This is a <Inductive> reasoning task. Given the following sequence, ['2', '5', '20', '23', '26', '104', '107', '110', '440', '443', '?']
What is the value at the question mark? Please enclose the answer in <answer><answer>, and put all the intermediate reasoning steps in <think><think>.
Generated Output:
<think> To find the pattern in the sequence, let's first examine the differences between consecutive numbers:
5 - 2 = 3, 20 - 5 = 15, 23 - 20 = 3, 26 - 23 = 3, 104 - 26 = 78, 107 - 104 = 3, 110 - 107 = 3, 440 - 110 = 330, 443 - 440 = 3

From the differences, we can observe a pattern where the difference alternates between 3 and a larger number. The larger numbers are increasing (15, 78, 330), which suggests a pattern of multiplication or addition that increases each time.

Looking closer at the larger differences:
- 15 * 5.2 = 78 (approximately)
- 78 * 4.23 = 330 (approximately)

However, a more precise pattern can be observed:
15 = 3 * 5, 78 = 15 * 5 + 3, 330 = 78 * 4 + 6

But a simpler pattern can be observed if we consider the sequence of operations:
Start with 2, Add 3 to get 5, Multiply by 4 to get 20, Add 3 to get 23, Add 3 to get 26, Multiply by 4 to get 104, Add 3 to get 107, Add 3 to get 110, Multiply by 4 to get 440, Add 3 to get 443

Following this pattern, the next operation should be to add 3 again, as the pattern alternates between multiplying by 4 and adding 3.

So, the next number in the sequence after 443 would be 443 + 3 = 446. </think>
<answer>446</answer>

Example 2 displays the output of the deductive full FT *Llama-3.1-8B-Instruct* that correctly answering the same question. Interestingly, the reasoning perspective differs: Example 1 relies on a pattern-observing, data-driven approach, while Example 2 demonstrates a rule-driven, hypothesis-testing process. This difference in perspective is consistent with the meta-reasoning task used for fine-tuning. Furthermore, an "AHA! moment" is also observed, where the model re-checks its previous reasoning and derives the correct answer. These observations further support meta-reasoning abilities are highly correlated with each other.

OOD Examples. Examples 3–5 in Appendix A show output for a FOLIO question. Example 3 (vanilla *Llama-3.1-8B-Instruct*) and Example 4 (deductive full FT *Llama-3.1-8B-Instruct*) fail because they incorrectly reverse the logical implication by treating a sufficient condition as a necessary one. However, deductive Depth Up-scaling *Llama-3.1-8B-Instruct* produces the correct Example 5. This further supports that using specific PEFT methods can yield better OOD generalization.

Example 2: Deductive-Tuned Llama

Generated Output:
<think> To solve this, let's analyze the given sequence for any patterns or rules that could help us determine the next value. The sequence is: '2', '5', '20', '23', '26', '104', '107', '110', '440', '443', '?'.

Looking at the sequence, we can see that it starts with '2' and '5', which are both prime numbers. Then, '20' and '23' are not prime, but '26' is not a prime either, which makes us think that the sequence might not follow a simple pattern of prime numbers.

However, if we look closer, we can see that the sequence seems to be formed by adding a certain number to the previous term to get the next term. Let's try to find this pattern.

From '2' to '5', the difference is 3. From '5' to '20', the difference is 15. From '20' to '23', the difference is 3. From '23' to '26', the difference is 3. From '26' to '104', the difference is 78. From '104' to '107', the difference is 3. From '107' to '110', the difference is 3. From '110' to '440', the difference is 330. From '440' to '443', the difference is 3.

We can see that the differences are not consistent except for the differences between '2' and '5', '23' and '26', '104' and '107', '107' and '110', and '440' and '443', which are all 3. However, the other differences are not consistent.

But if we look at the sequence again, we can see that the numbers '20', '104', and '440' are all multiples of 4, and '23', '26', '107', and '110' are all 3 more than a multiple of 4.

So, the pattern seems to be that the sequence is formed by adding 3 to the previous term when the previous term is 3 more than a multiple of 4, and the term before that is a multiple of 4.

Let's apply this pattern to find the next term:
- The last term before the question mark is '443', which is 3 more than a multiple of 4 (440).
- So, the next term should be '443 + 3 = 446'.

Therefore, the value at the question mark is 446.
</think>
<answer>446</answer>

8 Conclusion

We have systematically investigated the efficacy of PEFT domain adaptation methods for endowing LLMs with meta-reasoning abilities. We addressed a critical gap in existing research by moving beyond the typical focus on deductive reasoning, instead constructing a fine-tuning dataset using high-quality reasoning trajectories sampled from a teacher model based on procedurally-generated logical reasoning questions. The results demonstrate that targeted fine-tuning on these synthetic meta-reasoning tasks yields substantial and highly transferable generalized performance gains on OOD benchmarks.

Limitations

All data and evaluations are in English given the availability and diversity of reasoning datasets. We acknowledge that the linguistic expression and underlying cognitive strategies for meta-reasoning may differ across languages which is an interesting avenue for future work.

References

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John Patrick Cunningham. 2024. LoRA learns less and forgets less. *Transactions on Machine Learning Research*. Featured Certification.

Chad E Brown. 2013. Reducing higher-order theorem proving to a sequence of sat problems. *Journal of Automated Reasoning*, 51(1):57–77.

Mingzi Cao, Xi Wang, and Nikolaos Aletras. 2025. Progressive depth up-scaling via optimal transport. *Preprint*, arXiv:2508.08011.

Cheng Chen, Yichun Yin, Lifeng Shang, Xin Jiang, Yujia Qin, Fengyu Wang, Zhi Wang, Xiao Chen, Zhiyuan Liu, and Qun Liu. 2022. bert2BERT: Towards reusable pretrained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2134–2148, Dublin, Ireland. Association for Computational Linguistics.

Monojit Choudhury, Shivam Chauhan, Rocktim Jyoti Das, Dhruv Sahnan, Xudong Han, Haonan Li, Aaryamonvikram Singh, Alok Anil Jadhav, Utkarsh Agarwal, Mukund Choudhary, Debopriyo Banerjee, Fajri Koto, Junaid Bhat, Awantika Shukla, Samujjwal Ghosh, Samta Kamboj, Onkar Pandit, Lalit Pradhan, Rahul Pal, and 11 others. 2025. Llama-3-nanda-10b-chat: An open generative large language model for hindi. *Preprint*, arXiv:2504.06011.

Tri Dao. 2024. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Maksym Del and Mark Fishel. 2023. True detective: A deep abductive reasoning benchmark undoable for GPT-3 and challenging for GPT-4. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 314–322, Toronto, Canada. Association for Computational Linguistics.

Longxu Dou, Qian Liu, Fan Zhou, Changyu Chen, Zili Wang, Ziqi Jin, Zichen Liu, Tongyao Zhu, Cunxiao Du, Penghui Yang, Haonan Wang, Jiaheng Liu, Yongchi Zhao, Xiachong Feng, Xin Mao, Man Tsung Yeung, Kunat Pipatanakul, Fajri Koto, Min Si Thu, and 22 others. 2025. Sailor2: Sailing in southeast asia with inclusive multilingual llms. *Preprint*, arXiv:2502.12982.

John Dougrez-Lewis, Mahmud Elahi Akhter, Federico Ruggeri, Sebastian Löbbers, Yulan He, and Maria Liakata. 2025. Assessing the reasoning capabilities of LLMs in the context of evidence-based claim verification. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20604–20628, Vienna, Austria. Association for Computational Linguistics.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, and 16 others. 2024a. FOLIO: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22017–22031, Miami, Florida, USA. Association for Computational Linguistics.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024b. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Transactions on Machine Learning Research*.

Khalil Hennara, Sara Chrouf, Mohamed Motaism Hamed, Zeina Aldallal, Omar Hadid, and Safwan AlModhayan. 2025. Kuwain 1.5b: An arabic slm via language injection. *Preprint*, arXiv:2504.15120.

Michael Hoffmann, Jophin John, Stefan Schweter, Gokul Ramakrishnan, Hoi-Fong Mak, Alice Zhang, Dmitry Gaynullin, and Nicolay J. Hammer. 2025. Llama-genba-10b: A trilingual large language model for german, english and bavarian. *Preprint*, arXiv:2509.05668.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In

Proceedings of the 36th International Conference on Machine Learning, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Zhiyuan Hu, Yibo Wang, Hanze Dong, Yuhui Xu, Amrita Saha, Caiming Xiong, Bryan Hooi, and Junnan Li. 2025. Beyond 'aha!': Toward systematic meta-abilities alignment in large reasoning models. *Preprint*, arXiv:2505.10554.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213.

Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. 2023. Sparse upcycling: Training mixture-of-experts from dense checkpoints. In *The Eleventh International Conference on Learning Representations*.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. 2024. Understanding catastrophic forgetting in language models via implicit inference. In *The Twelfth International Conference on Learning Representations*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Wooseong Lee, Minseo Kim, Taeil Hur, Gyeong Hwan Jang, Woncheol Lee, Maro Na, and Taeuk Kim. 2025. ENGinius: A bilingual LLM optimized for plant construction engineering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 1350–1364, Vienna, Austria. Association for Computational Linguistics.

Samuel Lewis-Lim, Xingwei Tan, Zhixue Zhao, and Nikolaos Aletras. 2025. Analysing chain of thought dynamics: Active guidance or unfaithful post-hoc rationalisation? *Preprint*, arXiv:2508.19827.

Chunyang Li, Weiqi Wang, Tianshi Zheng, and Yangqiu Song. 2025a. Patterns over principles: The fragility of inductive reasoning in LLMs under noisy observations. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19608–19626, Vienna, Austria. Association for Computational Linguistics.

Junzhuo Li, Bo Wang, Xiuze Zhou, and Xuming Hu. 2025b. Dynamic expert specialization: Towards catastrophic forgetting-free multi-domain moe adaptation. *Preprint*, arXiv:2509.16882.

Junghwan Lim, Gangwon Jo, Sungmin Lee, Jiyoung Park, Dongseok Kim, Jihwan Kim, Junhyeok Lee, Wai Ting Cheung, Dahye Choi, Kibong Choi, Jaeyeon Huh, Beomgyu Kim, Jangwoong Kim, Taehyun Kim, Haesol Lee, Jeesoo Lee, Dongpin Oh, Changseok Song, and Daewon Suh. 2025. Expanding foundational language capabilities in open-source llms through a korean case study. *Preprint*, arXiv:2509.03972.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2025. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *Preprint*, arXiv:2308.08747.

Taishi Nakamura, Takuya Akiba, Kazuki Fujii, Yusuke Oda, Rio Yokota, and Jun Suzuki. 2025. Drop-upcycling: Training sparse mixture of experts with partial re-initialization. In *The Thirteenth International Conference on Learning Representations*.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Sergey Pletenev, Maria Marina, Daniil Moskovskiy, Vasily Konovalov, Pavel Braslavski, Alexander Panchenko, and Mikhail Salnikov. 2025. How much knowledge can you pack into a LoRA adapter without harming LLM? In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4309–4322, Albuquerque, New Mexico. Association for Computational Linguistics.

Qwen-Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. 2022. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 506–516, Red Hook, NY, USA. Curran Associates Inc.

Kyle Richardson and Ashish Sabharwal. 2021. Pushing the limits of rule reasoning in transformers through natural language satisfiability. In *AAAI Conference on Artificial Intelligence*.

Zayne Rea Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann

Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2025. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. In *The Thirteenth International Conference on Learning Representations*.

Xingwei Tan, Marco Valentino, Mahmud Akhter, Maria Liakata, and Nikolaos Aletras. 2025. Enhancing logical reasoning in language models via symbolically-guided monte carlo process supervision. *Preprint*, arXiv:2505.20415.

Nilesh Tripuraneni, Michael I. Jordan, and Chi Jin. 2020. On the theory of transfer learning: the importance of task diversity. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl.

Yite Wang, Jiahao Su, Hanlin Lu, Cong Xie, Tianyi Liu, Jianbo Yuan, Haibin Lin, Ruoyu Sun, and Hongxia Yang. 2024. LEMON: Lossless model expansion. In *The Twelfth International Conference on Learning Representations*.

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 *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Xiangyu Wen, Junhua Huang, Zeju Li, Min Li, Jianyuan Zhong, Zhijian Xu, Mingxuan Yuan, Yongxiang Huang, and Qiang Xu. 2025. Reasoning scaffolding: Distilling the flow of thought from llms. *Preprint*, arXiv:2509.23619.

Jan Wielemaker, Tom Schrijvers, Markus Triska, and Torbjörn Lager. 2012. SWI-Prolog. *Theory and Practice of Logic Programming*, 12(1-2):67–96.

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, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Chengyue Wu, Yukang Gan, Yixiao Ge, Zeyu Lu, Jiahao Wang, Ye Feng, Ying Shan, and Ping Luo. 2024. LLaMA pro: Progressive LLaMA with block expansion. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6518–6537, Bangkok, Thailand. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Yifei Yang, Zouying Cao, Xinbei Ma, Yao Yao, Zhi Chen, Libo Qin, and Hai Zhao. 2025b. LESA: Learnable LLM layer scaling-up. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22463–22476, Vienna, Austria. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Yiqun Yao, Zheng Zhang, Jing Li, and Yequan Wang. 2024. Masked Structural Growth for 2x Faster Language Model Pre-training. In *The Twelfth International Conference on Learning Representations*.

Hongming Zhang, Xinran Zhao, and Yangqiu Song. 2020. WinoWhy: A deep diagnosis of essential commonsense knowledge for answering Winograd schema challenge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5736–5745, Online. Association for Computational Linguistics.

Xue Zhang, Yunlong Liang, Fandong Meng, Songming Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2025. Less, but better: Efficient multilingual expansion for LLMs via layer-wise mixture-of-experts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17948–17963, Vienna, Austria. Association for Computational Linguistics.

Wenting Zhao, Justin Chiu, Claire Cardie, and Alexander Rush. 2023. Abductive commonsense reasoning exploiting mutually exclusive explanations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14883–14896, Toronto, Canada. Association for Computational Linguistics.

Hao Zhou, Zhijun Wang, Shujian Huang, Xin Huang, Xue Han, Junlan Feng, Chao Deng, Weihua Luo, and Jiajun Chen. 2025. Moe-lpr: multilingual extension of large language models through mixture-of-experts with language priors routing. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in*

Artificial Intelligence, AAAI'25/IAAI'25/EAAI'25. AAAI Press.

A Out-of-Domain Generation Example

The following examples shows the generated output of answering a question from Folio. The vanilla and Deductive Full FT *Llama-3.1-8B-Instruct* failed to answer the question, while Deductive fine-tuned Depth Up-scaling *Llama-3.1-8B-Instruct* can generate the correct answer.

Example 5: Deductive-Tuned Depth Up-scaling

Generated Output:
in order to determine if the conclusion is correct, we need to analyze the premises:

1. People in this club who perform in school talent shows often attend and are very engaged with school events.
2. People in this club either perform in school talent shows often or are inactive and disinterested community members.
3. People in this club who chaperone high school dances are not students who attend the school.
4. All people in this club who are inactive and disinterested members of their community chaperone high school dances.
5. All young children and teenagers in this club who wish to further their academic careers and educational opportunities are students who attend the school.
6. Bonnie is in this club and she either both attends and is very engaged with school events and is a student who attends the school or is not someone who both attends and is very engaged with school events and is not a student who attends the school.

From premise 6, we know that Bonnie is in this club and has two possible scenarios:
- She both attends and is very engaged with school events and is a student who attends the school.
- She is not someone who both attends and is very engaged with school events and is not a student who attends the school.

From premise 1, if Bonnie is someone who both attends and is very engaged with school events, then she must perform in school talent shows often.

From premise 3, if Bonnie is a student who attends the school, then she cannot chaperone high school dances.

From premise 4, if Bonnie is an inactive and disinterested member of her community, then she must chaperone high school dances.

From premise 5, if Bonnie is a young child or teenager in this club who wishes to further her academic careers and educational opportunities, then she must be a student who attends the school
.
Considering all premises, we cannot directly conclude that Bonnie performs in school talent shows often from the given information. However, we can deduce that if Bonnie is a student who attends the school and is very engaged with school events, then she must perform in school talent shows often.

But premise 6 does not directly state that Bonnie is a student or that she is not a student. It only gives two possible scenarios for Bonnie's situation.

Therefore, the conclusion that Bonnie performs in school talent shows often is not directly supported by the premises, and we cannot be certain of it.

<answer>Uncertain</answer>

Example 4: Deductive-Tuned Llama

Generated Output:
in order to determine if the conclusion is correct, we need to analyze the premises and apply them to Bonnie's situation.

From the premises, we know:
1. People in this club who perform in school talent shows often attend and are very engaged with school events.
2. People in this club either perform in school talent shows often or are inactive and disinterested community members.
3. People in this club who chaperone high school dances are not students who attend the school.
4. All people in this club who are inactive and disinterested members of their community chaperone high school dances.
5. All young children and teenagers in this club who wish to further their academic careers and educational opportunities are students who attend the school.

Given that Bonnie is in this club, we can start by looking at the last premise:
All young children and teenagers in this club who wish to further their academic careers and educational opportunities are students who attend the school.
This implies that if Bonnie is a young child or teenager and wishes to further her academic career and educational opportunities, then she must be a student who attends the school.

However, we also know from premise 3 that people in this club who chaperone high school dances are not students who attend the school.
If Bonnie is a student who attends the school, then she cannot be a chaperone for high school dances.

Now, looking at premise 2:
People in this club either perform in school talent shows often or are inactive and disinterested community members.
If Bonnie is a student who attends the school, then from premise 1, she must perform in school talent shows often because she is very engaged with school events.

However, premise 4 states that all people in this club who are inactive and disinterested members of their community chaperone high school dances.
This means that if Bonnie is not a student who attends the school, then she must be an inactive and disinterested member of her community, which would make her a chaperone for high school dances, which contradicts our earlier deduction that if she is a student, she cannot be a chaperone.

Considering all premises, we can conclude that if Bonnie is a student who attends the school, then she must perform in school talent shows often.
But we also know from premise 3 that if she is a chaperone, she cannot be a student.
So, if we consider the possibility that Bonnie is not a student, then she would be an inactive and disinterested member of her community, which would make her a chaperone, but this contradicts our initial assumption that she is not a student.

Given the premises, the most consistent conclusion we can draw about Bonnie is that if she is a student who attends the school, then she must perform in school talent shows often.
But since we cannot definitively conclude that Bonnie is a student or not based on the given information, we cannot definitively say if she performs in school talent shows often.

However, looking closely at premise 5 and the structure of the premises, we can see that if Bonnie is a student who attends the school and is very engaged with school events, then she must perform in school talent shows often.
But we cannot conclude this with absolute certainty without more information about Bonnie's status as a student or a non-student.

Therefore, based on the given premises, we cannot definitively conclude that Bonnie performs in school talent shows often without assuming that she is a student who attends the school.
But since the conclusion is about whether Bonnie performs in school talent shows often, and we have a premise that supports this conclusion if she is a student, we can say that the conclusion is True if we assume Bonnie is a student who attends the school and is very engaged with school events.

<answer>True</answer>

Example 3: Vanilla Llama

Based on the given premises, is the conclusion correct? Please respond with True, False, or Uncertain enclosing in <answer><answer>.
Premises: People in this club who perform in school talent shows often attend and are very engaged with school events.
People in this club either perform in school talent shows often or are inactive and disinterested community members.
People in this club who chaperone high school dances are not students who attend the school.
All people in this club who are inactive and disinterested members of their community chaperone high school dances.
All young children and teenagers in this club who wish to further their academic careers and educational opportunities are students who attend the school.
Bonnie is in this club and she either both attends and is very engaged with school events and is a student who attends the school or is not someone who both attends and is very engaged with school events and is not a student who attends the school.
Conclusion: Bonnie performs in school talent shows often.

Generated Output:
in this case, we can break down the information as follows:

1. If Bonnie is a student who attends the school, then she must perform in school talent shows often (from the first premise).
2. If Bonnie is not a student who attends the school, then she must be an inactive and disinterested community member (from the second premise).
3. If Bonnie is an inactive and disinterested community member, then she must chaperone high school dances (from the fourth premise).
4. However, we know that people who chaperone high school dances are not students who attend the school (from the third premise).
5. This means that if Bonnie is not a student who attends the school, then she must chaperone high school dances, which contradicts the fact that she is not a student who attends the school.
6. Therefore, Bonnie must be a student who attends the school.
7. Since Bonnie is a student who attends the school, she must perform in school talent shows often (from the first premise).

So, the conclusion is correct: <answer>True</answer>.

B Prompt Examples

Here is an example of the prompt we use for comparing the generated answers and the gold answers.

Flexible Match

Instruction: Please check whether the generation results is consistent with the gold label.

Generation Results:{output}

Gold Label:{answer}

Please output TRUE if they are consistent, otherwise output FALSE.

C Reasoning Task Examples

The following boxes show the questions and gold answers in our synthetic meta-reasoning tasks.

Deductive Task

This is a <Deductive> reasoning task. Below are some formulas connected by conjunctions:
$\neg(((\neg A \wedge A) \vee \neg(F)))$
$\wedge((\neg(\neg E) \vee \neg(\neg B)) \oplus ((H \wedge F) \rightarrow (H \leftrightarrow F)))$
$\wedge\neg(((H \rightarrow A) \vee (\neg G \oplus D)))$
$\wedge(((F \wedge C) \wedge (G \oplus \neg G)) \leftrightarrow ((D \leftrightarrow A) \wedge (F \wedge G)))$
$\wedge((\neg(C) \wedge (\neg F \rightarrow D)) \oplus \neg((\neg B \leftrightarrow \neg F)))$
Please list the truth value of each variable to make the whole conjunction true using a JSON dictionary, which maps variable names to their truth values, then enclose the answer in <answer><answer>.
Please put all the intermediate reasoning steps in <think><think>.
Gold Answer:
{ "A": "False",
"B": "False",
"C": "True",
"D": "False",
"E": "False",
"F": "True",
"G": "True",
"H": "True"
}

Inductive Task

This is a <Inductive> reasoning task. Given the following sequence, [‘5’, ‘10’, ‘6’, ‘12’, ‘15’, ‘30’, ‘26’, ‘52’, ‘55’, ‘110’, ‘?’]
What is the value at the question mark? Please enclose the answer in <answer><answer>, and put all the intermediate reasoning steps in <think><think>.
Gold Answer: 106

Abductive Task

Premises:
(((NOT O) OR ((NOT O) OR M)) AND (((NOT I) OR O) OR O))) => I,
(M) => O,
((((I OR (NOT I)) AND ((NOT O) OR (NOT I))) AND ((M AND M) AND ((NOT I) AND (NOT O))))) => M,
(((NOT O) AND ((NOT F) OR (I AND (NOT I))))) => F,
((M OR ((NOT J) OR (J AND M)))) => K,
(F) => O,
(((O AND (NOT M)) OR ((I AND M) AND (NOT I)))) => M,
(J) => GB,
((((K OR I) AND (F AND M)) OR ((I AND (NOT K)) AND (J OR (NOT M))))) => M,
((((J OR M) AND (K OR (NOT J))) AND F)) => K,
((((I AND I) AND ((NOT O) OR M)) AND ((M AND O) AND (NOT F)))) => O,
((F AND (F OR (K AND K)))) => O,
(I) => O,
((((O OR K) OR (O OR (NOT J))) OR ((O OR K) AND (F AND (NOT F))))) => I,
(((M OR ((NOT I) OR (NOT I))) AND ((J AND M) OR (NOT M)))) => F,
(M) => F,
(F) => F
Known Atoms: [J, O, K, M, F, I]
Goals: [D, M, C]
Instruction: This is a <Abductive> reasoning task. For each goal, identify which premises directly lead to the goal. Then, trace back what the true value of the atoms must be to make each of the goal true. Only the atoms in the ‘known atoms’ are known but their values are not shown. Finally, return the reachable goals with the true values of the known atoms that make it true. Please enclose the final answer with <answer><answer>. All the intermediate thinking steps should be enclosed in <think><think> tags.
Gold Answer:
Goal ‘M’ is reachable. Here are the possible assignments to make it true:
Solution 1: {"F": true, "I": true, "J": false, "K": true, "O": true}
Solution 2: {"F": true, "I": true, "J": true, "K": true, "O": true}