A MIND for Reasoning: Meta-learning for In-context Deduction

Anonymous ACL submission

Abstract

Large language models (LLMs) are increasingly evaluated on formal tasks, where strong reasoning abilities define the state of the art. However, their ability to generalize to out-of-distribution problems remains limited. In this paper, we investigate how LLMs can achieve a systematic understanding of deductive rules. Our focus is on the task of identifying the appropriate subset of premises within a knowledge base needed to derive a given hypothesis. To tackle this challenge, we propose **M**eta-learning for **IN**-context **D**eduction (MIND), a novel few-shot meta-learning fine-tuning approach. The goal of MIND is to enable models to generalize more effectively to unseen knowledge bases and to systematically apply inference rules. Our results show that MIND significantly improves generalization in small LMs ranging from 1.5B to 7B parameters. The benefits are especially pronounced in smaller models and low-data settings. Remarkably, small models fine-tuned with MIND outperform state-of-the-art LLMs, such as GPT-4o and o3-mini, on this task.

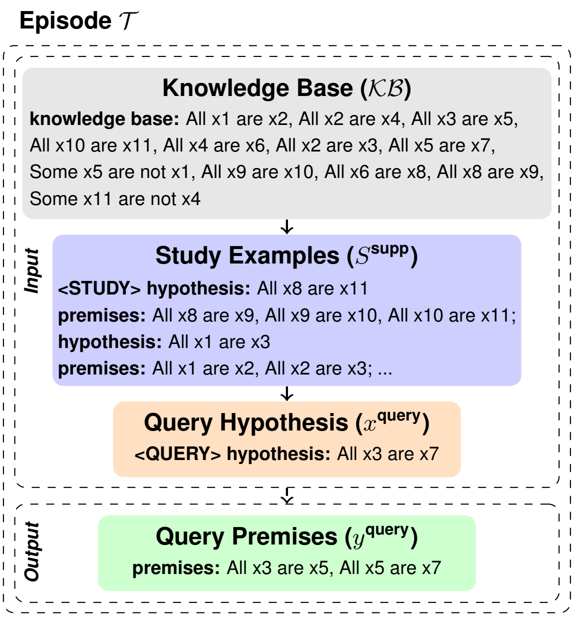


Figure 1: **Overview of a MIND episode**. Given a set of premises (the knowledge base, $\mathcal{KB}$), a set of task demonstrations (or study examples, denoted by the <STUDY> tag), and a query hypothesis x^{query} (denoted by the <QUERY> tag) that is entailed from $\mathcal{KB}$, models must generate the *minimal* subset of premises y^{query} from which x^{query} can be derived. During each MIND episode, models can *practice* on hypothesis-premise pairs before processing the main query hypothesis. The examples show how we frame syllogistic inferences as a premise selection task.

1 Introduction

Reasoning refers to a broad set of abilities that are applied not only in formal domains, such as mathematics and logic, but also in goal-directed scenarios involving problem-solving and decision-making (Leighton, 2004). All types of reasoning share a common foundation: the capacity to reach an abstract understanding of the problem at hand. With the advent of increasingly capable large language models (LLMs), reasoning has become a central domain for evaluating and comparing these systems (Huang and Chang, 2023; Mondorf and Plank, 2024).

Despite extensive training on mathematical, programming, and STEM-related data, LLMs continue to struggle in out-of-distribution (OOD) reasoning scenarios. Their performance often deteriorates on longer inference chains than those seen during training (Clark et al., 2021; Saparov et al., 2023), and they exhibit variability when evaluated with perturbed versions of the same problems (Mirzadeh et al., 2025; Gulati et al., 2024; Huang et al., 2025). In particular, LLMs can get distracted by irrelevant context, becoming unable to solve problems they could otherwise solve (Shi et al., 2023; Yoran et al., 2024). These challenges relate to broader debates surrounding generalization versus memorization in LLMs (Balloccu et al., 2024; Singh et al., 2024).

Few-shot meta-learning approaches (Irie and

Lake, 2024) have emerged as promising methods for inducing OOD generalization and rapid domain adaptation in LLMs. Specifically, this class of methods has proven effective in few-shot task generalization (Min et al., 2022; Chen et al., 2022), systematic generalization (Lake and Baroni, 2023), and mitigating catastrophic forgetting (Irie et al., 2025).

In this work, we propose **M**eta-learning for **IN**-context **D**eduction (MIND), a new few-shot meta-learning fine-tuning approach for deductive reasoning. As illustrated in Figure 1, we evaluate the effectiveness of this approach using a logical reasoning task grounded in syllogistic logic (Smiley, 1973; Vargas Guzmán et al., 2024). Each problem presents a knowledge base of atomic logical statements. Models are tasked with identifying the *minimal* subset of premises that logically entail a given test hypothesis. This premise selection task captures a core aspect of deductive reasoning: determining which known facts are necessary and sufficient to justify a conclusion. We apply MIND to small LMs from the Qwen-2.5 family (Qwen Team, 2025), ranging from 1.5B to 7B parameters. Specifically, we assess the generalization capabilities induced by MIND, such as systematically performing inferences over unseen sets of premises, as well as over more complex (longer) or simpler (shorter) sets of premises than those encountered during training.[1]

Our main contributions are as follows:

- We introduce a new synthetic dataset based on syllogistic logic to study reasoning generalization in LLMs.
- We show that MIND enables LMs to better generalize in OOD reasoning problems with particularly strong performance in smaller models and low-data regimes.
- We demonstrate that small LMs fine-tuned with MIND can outperform state-of-the-art LLMs such as GPT-4o and o3-mini, on our premise selection task.

2 Background

2.1 Syllogistic Logic

In our experiments, we focus on the syllogistic fragment of first-order logic. Originally, syllogisms have been studied by Aristotle as arguments composed of two premises and a conclusion, such as: "*All dogs are mammals; some pets are not mammals; therefore, some pets are not dogs.*" This basic form can be extended to include inferences involving more than two premises (see Łukasiewicz 1951; Smiley 1973).

[1]Code and data will be made available, CC BY license, upon acceptance.

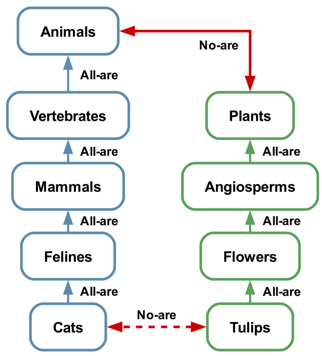


Figure 2: **Example inference.** Edges labeled "All-are" denote universal affirmatives (e.g., *All cats are felines*). The solid red edge is a universal negative (*No animals are plants*). From these "*atomic facts*" we infer *No cats are tulips* (dashed red edge). Formally, this is expressed as $\{Aa-b,\ Ac-d,\ Ebd\} \vDash Eac$ (Smiley, 1973).

Syntax and semantics. The language of syllogistic logic comprises a finite set of atomic terms $\{a, b, c, \ldots\}$ and four quantifier labels A, E, I, and O. Well-formed formulas consists of Aab ("All a are b"), Eab ("No a are b"), Iab ("Some a are b"), and Oab ("Some a are not b"). Finally, an *A-chain*, denoted as $Aa-b$ represents the single formula Aab or a sequence of formulas Aac_1, Ac_1c_2, $\ldots$, $Ac_{n-1}c_n$, Ac_nb for $n \geq 1$. A *knowledge base* ($\mathcal{KB}$) is defined as a finite set of formulas (premises).

An *inference* $\mathcal{F} \vDash F$ (i.e., deriving a conclusion from a set of premises) holds when the conclusion F is true in every interpretation (an assignment of non-empty sets to terms) where all formulas in $\mathcal{F}$ are true. A set of formulas is *consistent* if there exists at least one interpretation in which all formulas are simultaneously true.

Minimal inferences. We aim for models to identify the minimal set of premises in a knowledge base to derive a given hypothesis. Formally, we are interested in inferences $\mathcal{F} \vDash F$ such that $\mathcal{F}' \nvDash F$ for any proper subset $\mathcal{F}' \subsetneq \mathcal{F}$. For example, $\{Abc, Abd\} \vDash Icd$ is minimal, while $\{Aab, Abc, Abd\} \vDash Icd$ is not because Aab is not needed to infer the conclusion.

There are seven types of minimal syllogistic inferences.[2] To facilitate understanding, Figure 2 provides an intuitive representation of a type 6 inference. Further details about the syllogistic logic can be found in Appendix A.

2.2 Meta-learning in Autoregressive Models

Meta-learning, or "learning to learn", is a paradigm that aims to enable machine learning models to acquire transferable knowledge across multiple tasks, allowing rapid adaptation to new tasks with minimal data. Among the numerous existing meta-learning frameworks (Hospedales et al., 2022), MIND is mainly inspired by Meta-learning Sequence Learners (MSL) (Irie and Lake, 2024).

Data organization. In standard supervised learning, data consists of a static dataset $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$ where inputs x_i are mapped to targets y_i under a fixed distribution $p(x, y)$. By contrast, meta-learning organizes data into tasks (or episodes) $\mathcal{T} = (S^{\text{supp}}, S^{\text{query}})$ drawn from $p(\mathcal{T})$, where $S^{\text{supp}} = \{(x_i, y_i)\}_{i=1}^{K}$ is the *support set* containing task demonstrations, or study examples, and $S^{\text{query}} = \{(x_j, y_j)\}_{j=1}^{M}$ is the *query set* for evaluation. We consider the simplest scenario where $|S^{\text{query}}| = 1$, containing a single example $(x^{\text{query}}, y^{\text{query}})$. We adapt this episodic formulation to our task, as shown in Figure 1.

Optimization. The fundamental difference between the two paradigms appears in their optimization objectives. Standard supervised learning finds parameters θ^* that maximize the likelihood:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{(x,y)\in\mathcal{D}_{\text{train}}} \log p_\theta(y \mid x) \quad (1)$$

while meta-learning finds parameters θ^* that maximize the expected likelihood across tasks:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \mathbb{E}_{\mathcal{T}} \left[\log p_\theta(y^{\text{query}} \mid x^{\text{query}}, S^{\text{supp}})\right] \quad (2)$$

For autoregressive models, the probability $p_\theta(y^{\text{query}} \mid x^{\text{query}}, S^{\text{supp}})$ is computed by conditioning on the support set S^{supp} as part of the input context, formatted as a sequence of input-output pairs preceding the query. This approach forces the model to develop the capabilities of recognizing and applying task patterns from the support examples to generate appropriate query outputs.

[2]See the full list in Table 4 in Appendix A.

3 Method

3.1 Data Generation

In this section, we describe the methodology employed to construct textual datasets designed for the task of logical premise selection. The process begins with the random generation of graph-like structures representing $\mathcal{KB}s$. These are then translated into text using fixed syntactic templates and assigning pseudowords to nodes.

Abstract representation. To avoid ambiguity in premise selection, we use only *non-redundant* $\mathcal{KB}s$, where for each derivable hypothesis F, there is a unique $\mathcal{F} \subseteq \mathcal{KB}$ such that $\mathcal{F} \vDash F$ is minimal. We represent $\mathcal{KB}s$ as graphs, with constants as nodes and quantifiers as edges.[3] Synthetic $\mathcal{KB}s$ are generated by constructing such graphs. To ensure non-redundancy, A-formulas form disjoint subgraphs with at most one path between any two nodes. We created three independent sets of consistent $\mathcal{KB}s$ for training, validation, and testing to ensure diversity across splits.[4]

Textual translation. To translate a given $\mathcal{KB}_i$ into a textual string, we: (1) assign a unique identifier $x_1, \ldots, x_n$ to each node within $\mathcal{KB}_i$; (2) map each edge to a fixed template connecting nodes x_i and x_j based on the quantifier represented by the edge (e.g., Ax_ix_j becomes "All x_i are x_j"); and (3) assign each unique node identifier $x_1, \ldots, x_n$ to a random English-like pseudoword (e.g., x_1 = wug, x_2 = blump).[5]

As illustrated in Figure 1, we structured each datapoint in the three splits to begin with the token "`knowledge base:`", followed by the full sequence of premises, separated by commas. This is immediately followed by the special tag `<QUERY>`, and then the token "`hypothesis:`", which introduces the target hypothesis. Next comes the token "`premises:`", followed by the specific comma-separated premises that entail the hypothesis. To increase variability, we applied ten random pseudoword assignments and three random permutations of premise order for each $\mathcal{KB}$, resulting in multiple variants per datapoint.

Within each $\mathcal{KB}$, valid hypotheses can be inferred by minimal sets of premises of varying

[3]A visual representation of $\mathcal{KB}s$ and the seven types of inferences as graphs can be seen in Appendix B.2.

[4]See Appendix B.1 for the exact algorithms used to generate $\mathcal{KB}s$

[5]Further details on the vocabulary of pseudowords we used are provided in Appendix B.3.

lengths. We define the **length** of a inference as the total length of all A-chains it contains, which corresponds to the total number of A-formulas among its premises. For a given inference type t, we denote the maximum and minimum lengths as $\mu(t)$ and $\sigma(t)$, respectively.

We generated enough $\mathcal{KB}$s to obtain 1000 training, 5 validation, and 100 test examples for each inference type and length combination in the range from 0 to 19.[6] This range was chosen to allow experiments with generalization to both unseen shorter and longer inferences. Full dataset statistics, including the number of generated $\mathcal{KB}$s per split, are reported in Appendix B.4.

3.2 MIND

When applying meta-learning principles to the framework of syllogistic logic, we conceptualize the premises within a $\mathcal{KB}$ as *atomic facts*. The seven types of syllogism (as detailed in Table 4) are treated as *arguments*, constructed using these atomic facts, and the model's task is to extract the minimal set of facts within a $\mathcal{KB}$ to produce a valid argument that proves the query hypothesis.

The type of systematic generalization MIND addresses involves applying the seven fixed syllogistic inferences to new, unseen sets of atomic facts. This is central to logical reasoning because logical rules are, by definition, formal: conclusions follow from premises based solely on the *structure* of the arguments, regardless of their specific content. Thus, successfully applying an inference type to a novel, unseen $\mathcal{KB}$ requires the model to recognize and instantiate the same formal structure with different premises. This generalization also includes variations in the number of atomic facts needed to instantiate an argument. Specifically, handling A-chains of varying lengths requires applying the learned inference patterns to longer or shorter instances of the same formal type.

Episodes organization. To induce meta-learning of inference types, MIND uses a set of episodes where each episode $\mathcal{T} = (\mathcal{KB}, S^{\text{supp}}, x^{\text{query}}, y^{\text{query}})$. Here, $\mathcal{KB}$ is a knowledge base, S^{supp} is a set of study valid hypothesis-premises pairs, x^{query} is a valid query hypothesis, and y^{query} is the minimal set of premises entailing x^{query}. Figure 1 shows a full MIND episode using indexed variables in place of pseudowords for improved readability. Importantly, we consider study examples with inferences of the same type as the query. The number of study examples we set, i.e. valid hypothesis–premise pairs, is three. In their textual translation, we add the special tags <STUDY> to indicate the beginning of the sequence of study examples. During MIND fine-tuning, models are trained to minimize the cross-entropy loss of the tokens in y^{query} given the input tokens from the context $(\mathcal{KB}, S^{\text{supp}}, x^{\text{query}})$.

Baseline. Similarly to Lake and Baroni (2023), we consider a baseline where models are not fine-tuned on episodes but on single input-output pairs $(x^{\text{query}}, y^{\text{query}})$ preceded by a $\mathcal{KB}$. The baseline is fine-tuned to minimize the cross-entropy loss of the tokens in y^{query} given the input tokens from the context $(\mathcal{KB}, x^{\text{query}})$. To ensure a fair comparison between the meta-learning model and the baseline, we ensured that both models were fine-tuned on the exact same aggregate set of unique hypothesis-premises pairs. Specifically, the baseline was fine-tuned using a set $\mathcal{D}_{\text{baseline}}$ consisting of $(x^{\text{query}}, y^{\text{query}})$ unique pairs. For the meta-learning approach, the corresponding set of all unique hypothesis-premises pairs encountered across all N episodes $\mathcal{T}_i = (\mathcal{KB}_i, S_i^{\text{supp}}, x_i^{\text{query}}, y_i^{\text{query}})$ is given by $\mathcal{D}_{\text{meta}} = \bigcup_{i=1}^{N}(S_i^{\text{supp}} \cup \{(x_i^{\text{query}}, y_i^{\text{query}})\})$. We verified that $\mathcal{D}_{\text{baseline}} = \mathcal{D}_{\text{meta}}$. Moreover, since the meta-learning model processes more hypothesis-premises pairs within each episode (due to S_i^{supp}), we counterbalanced this by training the baseline model for a proportionally larger number of epochs.[7]

4 Experimental Setup

4.1 Models

We run experiments using the Qwen 2.5 family of decoder-only LMs (Qwen Team, 2025). More specifically, we test three sizes: 1.5B, 3B, and 7B parameters. This family of models is selected because it allows us to experiment with varying small sizes (ranging from 1.5 to 7 billion parameters) and achieves a better size vs. performance trade-off compared to other open weights model families.

In addition to the Qwen 2.5 family, we also evalu-

[6] Note that some inference types (e.g., type 3) span the full range of lengths from 0 to 19, while others span only a subrange (e.g., type 2 spans from 1 to 10). See all type-length combinations within the generated $\mathcal{KB}$s in Figure 6 in Appendix B.4.

[7] Further details on the training regime and number of epochs for each approach are provided in Appendix C.2.

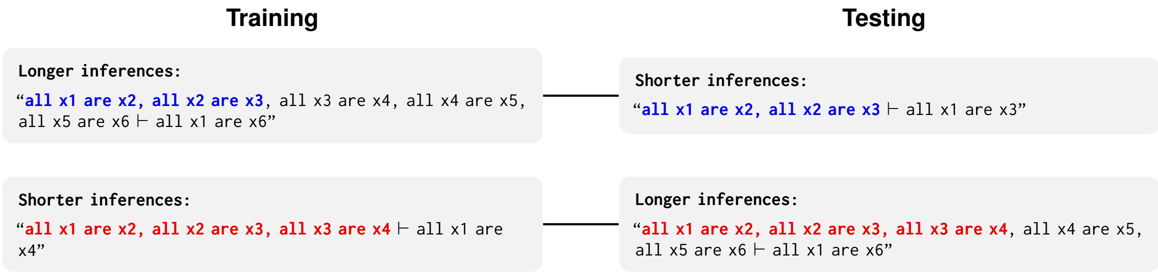


Figure 3: **Length generalization**. We evaluate models on two types of length generalization: models trained on more complex (i.e., longer) inferences are tested on simpler (i.e., shorter) ones (Top) and vice versa (Bottom). The examples illustrate type 2 inferences.

ate the closed-source LLM GPT-4o (OpenAI, 2024) and the Large Reasoning Model (LRM) o3-mini (OpenAI, 2025) on the logical premise selection task.[8] We conduct the evaluation both in a zero-shot setting and in a few-shot setting, using the S^{supp} study pairs as examples.[9]

4.2 Experiments

We design experiments to evaluate the ability of MIND to teach pretrained small LMs to systematically apply inferences to new, unseen sets of premises —that is, to reason in a formal way by recognizing and instantiating the same underlying structure independently of the $\mathcal{KB}$s' content.

To ensure consistency, both MIND and the baseline receive inputs at test time in the same format as during training. MIND models are provided as context $(\mathcal{KB}, S^{\text{supp}}, x^{\text{query}})$, and are tasked to generate y^{query}, while the baseline receives $(\mathcal{KB}, x^{\text{query}})$.

Generalization. In the first experiment, models are evaluated on their ability to generalize to unseen $\mathcal{KB}s$, while all inference lengths are seen. The training and testing sets contain inferences of all lengths for each of the seven types. Since this is the simplest form of systematic application of syllogistic inference, we refer to it as **core generalization**.

We then consider two more challenging generalizations involving inferences of **unseen length**. As illustrated in Figure 3, we examine the case of generalizing to **longer inferences** when the model has only learned from shorter ones (as studied in Saparov et al. 2023), and vice versa —generalizing to **shorter inferences** after seeing only longer ones. In the logic literature, they are respectively known as recursiveness and compositionality (Vargas Guzmán et al., 2024). To test this, we train exclusively on inferences whose lengths x are $\sigma(t) \leq x \leq \mu(t) - 5$, and test on the five longest inferences for each type, i.e., those whose length is $\mu(t) - 5 < x \leq \mu(t)$. In the second case, we train on inferences with length $\sigma(t) + 5 \leq x \leq \mu(t)$, and test only on the five shortest inference lengths for each type, i.e., those with length $\sigma(t) \leq x < \sigma(t) + 5$. An intuitive representation of these generalizations is provided in Figure 3. Notably, within the MIND approach, we consider two varying types of study examples S^{supp}: the **aligned** and **disaligned** sets of study examples, in which each $(x^{\text{supp}}, y^{\text{supp}})$ either falls within or outside the range of inference lengths used for testing, respectively.[10]

Figure 6, in the Appendix, shows all inference type-length combinations within training and test split in the core and in the length generalization settings. These datasets contain 1,000 and 100 datapoints for each training and testing type–length combination, respectively. To further investigate the performance of MIND in a **limited data regime**, we also consider the case where only 100 datapoints are available for each training type–length combination.

4.3 Prediction Accuracy

We consider a model prediction to be correct if the set of premises extracted from the generated text matches the ground truth set of *minimal* premises.

[8] Note that LRMs are also LLMs, but post-trained to generate longer intermediate chains of thought, improving performance on complex reasoning tasks (Xu et al., 2025).

[9] See the API details and the exact prompts used to evaluate closed models in Appendix C.3.

[10] More precisely, the meanings of **aligned** and **disaligned** depend on whether we are evaluating models on unseen shorter or longer inferences. For longer inferences, **disaligned** includes inferences with lengths $\sigma(t) \leq x \leq \mu(t) - 5$, and **aligned** includes those with lengths $\mu(t) - 5 < x \leq \mu(t)$. For shorter ones, instead, **aligned** includes inferences with lengths $\sigma(t) \leq x < \sigma(t) + 5$, and **disaligned** includes those with lengths $\sigma(t) + 5 \leq x \leq \mu(t)$.

	Model	Method	All	Short	Long
Fine-tuning	Qwen-2.5 1.5B	MIND	93.11 ± 0.61	94.28 ± 0.61	91.76 ± 0.27
		Baseline	85.56 ± 1.24	91.42 ± 0.82	80.56 ± 1.78
	Qwen-2.5 3B	MIND	96.16 ± 0.44	96.24 ± 0.56	95.55 ± 0.43
		Baseline	93.03 ± 1.15	95.34 ± 1.18	90.92 ± 1.27
	Qwen-2.5 7B	MIND	98.13 ± 0.98	98.26 ± 0.82	97.69 ± 1.40
		Baseline	95.76 ± 1.10	97.27 ± 1.22	94.13 ± 0.90
Prompting	GPT-4o	Few-shot	39.76	52.91	33.51
		Zero-shot	15.90	28.97	9.89
	o3-mini	Few-shot	88.45	87.91	88.51
		Zero-shot	67.98	73.29	64.54

Table 1: **Core generalization.** Accuracy (mean ± std) on test inferences across all type-length combinations (All), plus breakdown into the five shortest (Short) and longest (Long) inferences for each of the seven types of inference. Fine-tuned Qwen models use MIND vs. Baseline; GPT-4o and o3-mini use few-shot vs. zero-shot prompting.

Using this criterion, we measure accuracy both in aggregate, i.e., across an entire test set, and decomposed by each test type-length combination. All models (1.5B, 3B, and 7B) are fine-tuned three times and with different random seeds, thus we report mean and standard deviation of each accuracy.

5 Results

5.1 Core Generalization

We first examine the performance of meta-learning versus the baseline on core generalization (Table 1), with models trained and tested on all inference type-length combinations. The “Short” and “Long” columns report aggregated accuracy on the sets of the five shortest and longest inferences, respectively, for each type. We hypothesize that longer inferences are harder because, to be correct, models must select *all* premises belonging to a larger minimal set of premises.

Across all Qwen-2.5 model sizes (1.5B, 3B, 7B), the meta-learning approach consistently yields higher accuracy than the baseline. Performance improves with model scale in both approaches. For example, MIND accuracy increases from 93.11% (1.5B) to 98.13% (7B) on all type-length combinations, with accuracy on shortest inferences rising from 94.28% to 98.26%, and on the longest ones increasing from 91.76% to 97.69%. In contrast, baseline performance rises more slowly —from 85.56% (1.5B) to 95.76% (7B) —and shows a wider drop on the longest inferences, falling as low as 80.56% for the smallest model. Notably, the performance gap between MIND and the baseline narrows as model size increases, suggesting that larger models achieve better core generalization even without meta-learning. It is worth noting that with limited data, MIND’s advantage over the baseline becomes much wider at all sizes, as shown in Appendix D.3.

The closed-source models GPT-4o and o3-mini still underperform compared to Qwen-2.5 models fine-tuned with MIND. Both models perform poorly in the zero-shot setting but improve with few-shot prompting: GPT-4o reaches 39.76% on all type-length combinations (with 52.91% on shortest and 33.51% on longest inferences), while o3-mini performs substantially better (88.45% all combination, 87.91% on shorters, and 88.51% on longest). As expected, performance on the longest inferences is worse than that on the shortest ones for GPT-4o, while o3-mini maintains a more robust performance across inference lengths.

5.2 Length Generalization

Table 2 shows that MIND models consistently outperform baseline models in generalizing to both longer and shorter inferences than those seen during training. In core generalization, we observed that longer inferences are more challenging than shorter ones. Instead, in the case of unseen lengths, an interesting and somewhat counterintuitive pattern emerges: it is generally easier for models to generalize to longer inferences than to shorter ones. This is true across all model sizes and in both approaches; For instance, the largest model, Qwen-2.5 7B, achieved 90.03% accuracy on longer inferences (disaligned) compared to 76.23% on shorter

Model	Method	Short → Long		Long → Short	
		Disaligned	Aligned	Disaligned	Aligned
Qwen-2.5 1.5B	MIND	76.42 ± 2.95	91.75 ± 1.10	70.94 ± 2.27	71.13 ± 1.83
	Baseline	63.53 ± 1.16	63.53 ± 1.16	56.67 ± 1.22	56.67 ± 1.22
Qwen-2.5 3B	MIND	87.61 ± 1.97	95.86 ± 0.70	77.19 ± 3.53	78.53 ± 1.71
	Baseline	76.78 ± 1.63	76.78 ± 1.63	71.88 ± 1.49	71.88 ± 1.49
Qwen-2.5 7B	MIND	90.03 ± 1.09	96.84 ± 0.15	76.23 ± 2.91	83.41 ± 1.63
	Baseline	80.76 ± 2.65	80.76 ± 2.65	71.08 ± 1.55	71.08 ± 1.55

Table 2: **Generalization to unseen lengths.** Accuracy (mean ± std) of meta-learning and baseline models when trained on short inferences and tested on longer ones or vice versa. In both cases, we compare the settings in which the inferences in the study examples either falls within (Aligned) or outside (Disaligned) the range of inference lengths used for testing. Baseline models have no study examples, hence such difference does not hold for them.

ones (disaligned).

Aligning study example lengths with the test condition (aligned) proves moderately to highly effective for unseen short inferences across all MIND model sizes. For example, Qwen-2.5 1.5B improved from 76.42% to 91.75%, and Qwen-2.5 3B improved from 87.61% to 95.86%. For unseen long inferences, this alignment is moderately effective in larger models: Qwen-2.5 7B improved from 76.23% to 83.41%, while the 1.5B and 3B models showed smaller gains (70.94% to 71.13% and 77.19% to 78.53%, respectively). These results indicate that MIND enables models in the aligned condition to exploit abstract patterns in the study examples (unseen inference lengths), allowing them to more effectively answer query hypotheses requiring length generalization.

Again, MIND's better performance in length generalization is especially noticeable with limited training data, where the difference between MIND and baseline models grows significantly (see Appendix D.3 for more details).

6 Error Analysis

Beyond simply measuring the accuracy of MIND and the baseline, we additionally focus on two main types of errors models make when evaluated on unseen lengths. First, among all errors, we consider the proportion of *non-minimal valid set of premises* (NVM). This means that the correct minimal set was generated by the model, but together with unnecessary premises; for this case, we also measure how many unnecessary premises, on average, the models generate. Alternatively, models may fail to provide the complete A-chain within the correct minimal set of premises, meaning that at least one *necessary A premise is missing* (MAP); here, we also track the average number of missing necessary A-formulas in erroneous answers. NVM and MAP are mutually exclusive. Furthermore, we consider an additional type of error that can occur simultaneously with either NVM or MAP: models may *hallucinate* premises —referred to as *hallucinated premises* (HP) —and output a formula that is not contained in the $\mathcal{KB}$.

Table 3 presents the error analysis for Qwen-2.5 7B[11] on the challenging length generalization settings.[12] HP is a common error type across both settings (often >50%). The baseline model has the highest HP rate in long to short (72.78%), while MIND models are generally better.

When generalizing to shorter inferences, a substantial portion of errors (28-43%) are NVM, indicating models indeed find logical solutions but include unnecessary premises. In this context, a lower number of unnecessary premises is better, as it is closer to the minimal set. The baseline model adds the most unnecessary premises (6.19 average), compared to MIND (disaligned) (4.9) and MIND (aligned) (3.72).

For generalizations to longer inferences, errors show different patterns, with few NVM errors (4-14%) and predominantly MAP errors (81-90%). The average number of missing premises is higher in short to long (3.65-6.66) than in long to short (1.76-2.1), suggesting models struggle to provide the complete set of premises when evaluated on

[11]Each model was fine-tuned three times with different random seeds, we selected the best model for each approach for this analysis.

[12]See Appendix D.4 for further error analysis results.

	Method	NVM [%]	Avg. NVM	MAP [%]	Avg. MAP	HP [%]
L $\rightarrow$ S	MIND (aligned)	42.94	4.9	36.68	2.1	57.5
	MIND (disaligned)	28.31	3.72	52.81	1.76	66.06
	Baseline	28.21	6.19	23.38	2.1	72.78
S $\rightarrow$ L	MIND (aligned)	9.76	1.66	87.54	5.08	60.94
	MIND (disaligned)	14.14	6.14	81.82	3.65	35.35
	Baseline	3.87	2.36	89.79	6.66	66.9

Table 3: **Error analysis.** Error analysis comparing MIND and baseline on long to short (L $\rightarrow$ S) and short to long (S $\rightarrow$ L) generalization. The table shows percentages and averages for non-minimal valid sets of premises (NVM) and missing necessary A premises (MAP), and the percentage of hallucinated premises (HP).

longer inferences than seen during training. The baseline model struggles most with longer inferences, with a high MAP error rate (89.79%) and a large number of missing premises (6.66) contributing to its lower accuracy compared to MIND.

7 Related Work

7.1 LLMs' Logical Capabilities

Recent work has highlighted weaknesses in LLMs' logical reasoning. LLMs often struggle with OOD generalization (Clark et al., 2021; Saparov et al., 2023; Vargas Guzmán et al., 2024), multi-step inference (Creswell et al., 2023), and consistency across formal reasoning patterns (Parmar et al., 2024; Hong et al., 2024). Neuro-symbolic methods address these gaps by integrating logic modules or symbolic solvers, improving both performance and interpretability (Pan et al., 2023; Olausson et al., 2023; Kambhampati et al., 2024). In a different direction, LRMs have shown strong gains in reasoning and planning tasks (Xu et al., 2025). Our proposed meta-learning approach offers a complementary alternative by enabling LLMs to adapt across logical tasks without relying on symbolic modules, as our results demonstrate.

7.2 Meta-learning

Meta-learning enables models to rapidly adapt to new tasks by leveraging prior experiences across tasks (Thrun and Pratt, 1998; Hospedales et al., 2022). Foundational approaches include memory-augmented neural networks (Santoro et al., 2016), prototypical networks (Snell et al., 2017), and model-agnostic meta-learning (MAML) (Finn et al., 2017). In the context of LLMs, meta-learning has been explored through techniques such as meta-in-context learning (Coda-Forno et al., 2023), in-context tuning (Chen et al., 2022), and MetaICL (Min et al., 2022), which either train for or exploit the in-context learning abilities of models to adapt to new tasks using few-shot examples. Our proposed method draws inspiration from the MSL framework (Irie and Lake, 2024), which we adapt and extend to solve the logical premise selection task.

8 Conclusion

In this work, we introduced MIND, a meta-learning fine-tuning approach to improve deductive reasoning in LLMs, explicitly targeting the logical premise selection task. Our results show that MIND significantly enhances generalization compared to the baseline, especially in small-scale and low-data scenarios. Remarkably, our fine-tuned small models outperform state-of-the-art LLMs on this task. This demonstrates the potential of MIND to advance the development of more robust and reliable AI systems.

Future work should explore several potential avenues. First, we should investigate not only systematic generalization using fixed inference rules, as we have done here, but also extend our research to learning the composition of multiple logical inferences. This approach aligns with ideas proposed in other meta-learning research, such as Meta-Learning for Compositionality(Lake and Baroni, 2023). Additionally, we should examine increasingly complex fragments of language, where the interactions among various inference-building blocks and reasoning forms become more intricate, and assess the effectiveness of MIND in helping LLMs to generalize in such contexts.

9 Limitations

Despite demonstrating meaningful progress in enhancing the deductive reasoning capabilities of language models through the MIND approach, this study has several limitations that future research could address.

Model selection. The evaluation primarily targets small to mid-sized language models (1.5B to 7B parameters), largely due to computational constraints. This focus leaves open the question of whether the observed improvements from MIND generalize to larger-scale models.

Meta-learning trade-offs. The gains in reasoning ability achieved by MIND come with associated costs. The meta-learning strategy adopted involves incorporating multiple study examples into the input context during fine-tuning. This leads to longer input sequences, which in turn increase memory usage and computational demands compared to standard fine-tuning approaches.

Focus on a logic fragment. This work is constrained to the syllogistic fragment of first-order logic. Future research should investigate whether our conclusions extend to more expressive logical systems or to real-world scenarios where reasoning tasks are less structured. However, syllogistic logic is a restricted domain that allows for precise control over variables such as the type of inference considered, inference length, and the structure of knowledge bases. In the context of this study, it serves as a valuable testbed for investigating logical generalization in LLMs.

References

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 67–93, St. Julian's, Malta. Association for Computational Linguistics.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022. Meta-learning via language model in-context tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 719–730, Dublin, Ireland. Association for Computational Linguistics.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2021. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20.

Julian Coda-Forno, Marcel Binz, Zeynep Akata, Matt Botvinick, Jane Wang, and Eric Schulz. 2023. Meta-in-context learning in large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 65189–65201. Curran Associates, Inc.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2023. Selection-inference: Exploiting large language models for interpretable logical reasoning. In *The Eleventh International Conference on Learning Representations*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: efficient finetuning of quantized llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1126–1135. JMLR.org.

Aryan Gulati, Brando Miranda, Eric Chen, Emily Xia, Kai Fronsdal, Bruno de Moraes Dumont, and Sanmi Koyejo. 2024. Putnam-AXIOM: A functional and static benchmark for measuring higher level mathematical reasoning. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

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.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2022. Meta-Learning in Neural Networks: A Survey . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 44(09):5149–5169.

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*.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada. Association for Computational Linguistics.

Kaixuan Huang, Jiacheng Guo, Zihao Li, Xiang Ji, Jiawei Ge, Wenzhe Li, Yingqing Guo, Tianle Cai, Hui Yuan, Runzhe Wang, Yue Wu, Ming Yin, Shange Tang, Yangsibo Huang, Chi Jin, Xinyun Chen, Chiyuan Zhang, and Mengdi Wang. 2025.

MATH-Perturb: Benchmarking LLMs' math reasoning abilities against hard perturbations. *arXiv preprint arXiv:2502.06453*.

Kazuki Irie, Róbert Csordás, and Jürgen Schmidhuber. 2025. Metalearning continual learning algorithms. *Transactions on Machine Learning Research*.

Kazuki Irie and Brenden M. Lake. 2024. Neural networks that overcome classic challenges through practice. *Preprint*, arXiv:2410.10596.

Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Paul Saldyt, and Anil B Murthy. 2024. Position: LLMs can't plan, but can help planning in LLM-modulo frameworks. In *Forty-first International Conference on Machine Learning*.

Emmanuel Keuleers and Marc Brysbaert. 2010. Wuggy: A multilingual pseudoword generator. *Behavior research methods*, 42:627–633.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization.

Brenden M. Lake and Marco Baroni. 2023. Human-like systematic generalization through a meta-learning neural network. *Nature*, 623:115–121.

Jacqueline P. Leighton. 2004. Defining and describing reason. In Jacqueline P. Leighton and Robert J. Sternberg, editors, *The Nature of Reasoning*. Cambridge University Press.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States. Association for Computational Linguistics.

Seyed Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2025. GSM-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *The Thirteenth International Conference on Learning Representations*.

Philipp Mondorf and Barbara Plank. 2024. Beyond accuracy: Evaluating the reasoning behavior of large language models - a survey. In *First Conference on Language Modeling*.

Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. 2023. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5153–5176, Singapore. Association for Computational Linguistics.

OpenAI. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

OpenAI. 2025. Openai o3-mini. `https://openai.com/index/openai-o3-mini/`. Accessed: 2025-05-08.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, Singapore. Association for Computational Linguistics.

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.

Qwen Team. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1842–1850. JMLR.org.

Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Mehran Kazemi, Najoung Kim, and He He. 2023. Testing the general deductive reasoning capacity of large language models using OOD examples. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Aaditya K. Singh, Muhammed Yusuf Kocyigit, Andrew Poulton, David Esiobu, Maria Lomeli, Gergely Szilvasy, and Dieuwke Hupkes. 2024. Evaluation data contamination in llms: how do we measure it and (when) does it matter? *Preprint*, arXiv:2411.03923.

Timothy J. Smiley. 1973. What is a syllogism? *Journal of Philosophical Logic*, 2(1):136–154.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4080–4090, Red Hook, NY, USA. Curran Associates Inc.

Sebastian Thrun and Lorien Pratt. 1998. *Learning to Learn: Introduction and Overview*, pages 3–17. Springer US, Boston, MA.

Manuel Vargas Guzmán, Jakub Szymanik, and Maciej Malicki. 2024. Testing the limits of logical reasoning in neural and hybrid models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2267–2279, Mexico City, Mexico. Association for Computational Linguistics.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, Chenyang Shao, Yuwei Yan, Qinglong Yang, Yiwen Song, Sijian Ren, Xinyuan Hu, Yu Li, Jie Feng, Chen Gao, and Yong Li. 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models. *Preprint*, arXiv:2501.09686.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*.

Jan Łukasiewicz. 1951. *Aristotle's Syllogistic From the Standpoint of Modern Formal Logic*. Oxford, England: Garland.

A Formal Semantics and Syllogistic Inference Patterns

In this section, we formally define the semantics of syllogistic logic by translating syllogistic formulas into first-order logic. We also specify a consistent set of such formulas and formalize a valid inference within this framework. Let $\mathcal{A} = \{a, b, c, \ldots\}$ be a set of atomic terms, and let $\mathcal{R} = \{R, S, T, \ldots\}$ be a set of unary relational symbols. We bijectively assign to every atomic term $a \in \mathcal{A}$ a relational symbol $R_a \in \mathcal{R}$, and interpret syllogistic formulas as first-order logic sentences: Aab as $\forall x\, [R_a(x) \rightarrow R_b(x)]$, Eab as $\forall x\, [R_a(x) \rightarrow \neg R_b(x)]$, Iab as $\exists x\, [R_a(x) \wedge R_b(x)]$, and Oab as $\exists x\, [R_a(x) \wedge \neg R_b(x)]$. We say that a set $\mathcal{F}$ of syllogistic formulas is *consistent* if there exists a structure M in signature $\mathcal{R}$ such that every relation R^M is non-empty, and the interpretation of every sentence in $\mathcal{F}$ holds in M, denoted by $M \vDash \mathcal{F}$. For a syllogistic formula F, the pair $(\mathcal{F}, F)$ is an *inference*, denoted by $\mathcal{F} \vDash F$, if $M \vDash \{F\}$, whenever $M \vDash \mathcal{F}$ for a structure M in signature $\mathcal{R}$.

B Dataset

B.1 $\mathcal{KB}$s' Generation

Knowledge bases can be modeled as edge-labeled graphs, in which nodes correspond to atomic terms and edges are labeled with quantifiers. Our graph generation algorithm comprises two principal stages: (1) We first construct all *A-chains* of the knowledge base, which is used as its structural backbone, by generating disjoint trees—directed acyclic graphs that ensure a unique path exists between any pair of nodes. (2) Subsequently, we incorporate additional label edges corresponding to E, I, and O formulas, while maintaining the overall consistency of the knowledge base.

Type	Inference
1	$\{Aa-b, Ac-d, Oad\} \vDash Obc$
2	$\{Aa-b\} \vDash Aab$
3	$\{Aa-b, Ac-d, Aa-e, Ede\} \vDash Obc$
4	$\{Aa-b, Aa-c\} \vDash Ibc$
5	$\{Aa-b, Ac-d, Ae-f, Iae, Edf\} \vDash Obc$
6	$\{Aa-b, Ac-d, Ebd\} \vDash Eac$
7	$\{Aa-b, Ac-d, Iac\} \vDash Ibd$

Table 4: **Syllogistic inference types.** Each row shows a distinct logical inference pattern. Notation follows traditional categorical logic: Aab denotes a universal affirmative ("All a are b"), Eab a universal negative ("No a are b"), Iac a existential affirmative ("Some a are c"), and Oad a existential negative ("Some a are not d"). Formulas of the form $Aa-b$ denote a sequence of n A-formulas relating a and b.

To construct all possible valid syllogisms from each artificially generated knowledge base, we employ antillogisms—minimal inconsistent set of syllogistic formulas. For example, consider the set $\{Aab, Aac, Ebc\}$, which forms an antilogism. By negating the inconsistent formula Ebc, we obtain a valid inference in which the remaining formulas $\{Aab, Aac\}$ entail its negation, i.e., $\{Aab, Aac\} \vDash Ibc$. This corresponds to an inference of type 4. More generally, any syllogism can be derived from an antilogism of the form $\mathcal{F} \cup \{\neg F\}$ by inferring the conclusion F from the consistent set $\mathcal{F}$, that is, $\mathcal{F} \vDash F$. This result was formally established by (Smiley, 1973), who also demonstrated that there exist only three distinct types of antilogisms. Furthermore, as shown by (Vargas Guzmán et al., 2024), all valid syllogistic inferences can be systematically derived from these three canonical forms of antilogism (see Table 4).

B.2 $\mathcal{KB}$s' Visualization

To provide an intuitive understanding of the various types of inferences and their derivation from the knowledge bases employed in our framework, we represent syllogistic formulas as graphs. These graphs encompass the knowledge base, the corresponding hypothesis, and the minimal inference—defined as the smallest subset of premises required to derive the hypothesis.

Experiment	Split	Size	# KBs	# Premises (Min–Max)
Core Generalization	Train	97,000	100	26–35
	Validation	485	15	26–36
	Test	9,700	200	26–38
Short $\to$ Long	Train	62,000	100	26–35
	Validation	310	15	26–36
	Test	3,500	194	26–38
Long $\to$ Short	Train	62,000	100	26–35
	Validation	310	15	26–36
	Test	3,500	200	26–38

Table 5: **Dataset statistics across experiments.** For each experiment and split, the table reports the number of unique query hypothesis-premises pairs (Size), the number of $\mathcal{KB}$s from which the pairs are generated (# KBs), and the range of total premises within $\mathcal{KB}$s (# Premises). In the additional experiment with limited training data, the total training size is reduced by a factor of ten.

Figure 19 illustrates a type 2 inference, characterized by a conclusion in the form of a universal affirmative (A-formula). The premises consist of a single sequence of A-formulas. This represents the most elementary form of syllogistic inference, whose structural pattern is embedded within all other types. Inferences of types 1, 3, and 5, which yield particular negative conclusions (O-formulas), are presented in Figures 18, 20, and 22, respectively. Syllogisms corresponding to types 4 and 7, both concluding with particular affirmative statements (I-formulas), are shown in Figures 21 and 24. Finally, the type 6 inference, which concludes with a universal negative (E-formula), is depicted in Figure 23.

B.3 Term Vocabulary

To train and evaluate our models, we artificially generated 5000 unique pseudowords by randomly concatenating two syllables selected from a set of approximately 300 of the most commonly used English syllables. Although these pseudowords are semantically meaningless, they remain phonologically plausible and are generally pronounceable. On occasion, the generation process may yield actual English words.

Additionally, we constructed two substitution sets to support our lexical generalization evaluation (see Appendix D.2). The first set comprises 5000 pseudowords generated using the Wuggy pseudoword generator (Keuleers and Brysbaert, 2010). We selected 500 English two-syllable nouns and, for each, produced 10 distinct pseudowords using Wuggy's default parameters. The second set consists of symbolic constants, each formed by the character "X" followed by an integers ranging from 1 to 5000.

B.4 Data Statistics

As described in Section 3.1, we generated as many KBs as necessary to obtain at least 1000 training, 5 validation, and 100 test examples for each inference type and length combination in the range from 0 to 19 (see all the combinations in Figure 6). Table 5 summarizes dataset statistics for the core generalization experiment, as well as for the length generalization ones ("Short $\to$ Long" and "Long $\to$ Short"). For each experiment and split, the table provides the total number of examples, the number of $\mathcal{KB}$s used to generate them, and the range of premises across $\mathcal{KB}$s. In the additional experiment with limited training data described in Appendix D.3, the total training size is reduced by a factor of ten in each setting.

C Experiment Details

C.1 Implementation Details

All experiments were conducted using the PyTorch and Hugging Face Transformers libraries. We used NVIDIA A100 80GB GPUs. Due to the relatively small size of the models used in the experiments, each fine-tuning run, both for MIND and the baseline, was able to fit on a single GPU. We estimate a total compute usage of approximately 500 GPU hours across all experiments. Additionally, GitHub Copilot was used as an assistant tool for parts of the project's source code development.

```
You are tasked with logical premise selection. Given:
1. A knowledge base consisting of premises.
2. A query hypothesis to solve, preceded by the token <QUERY>.
Your task is to identify the unique minimal set of premises from the
knowledge base that logically proves the query hypothesis. Since the
knowledge base is non-redundant, every valid hypothesis has exactly one
minimal set of premises that proves it.
Provide your answer in exactly this format:
### Answer: premise1, premise2, ..., premiseN
```

Figure 4: **Zero-shot system prompt.** The zero-shot system prompt used with the closed models GPT-4o and o3-mini. The query hypothesis is subsequently provided as the first user interaction. We then extract the set of premises returned by the model using regular expressions.

```
You are tasked with logical premise selection. Given:
1. A knowledge base consisting of premises.
2. Example hypotheses along with their correct minimal premise sets,
preceded by the token <STUDY>.
3. A query hypothesis to solve, preceded by the token <QUERY>.
Your task is to identify the unique minimal set of premises from the
knowledge base that logically proves the query hypothesis. Since the
knowledge base is non-redundant, every valid hypothesis has exactly one
minimal set of premises that proves it.
Examine the provided examples carefully to understand how to select
the correct minimal set of premises. The examples demonstrate correct
premise selections for various hypotheses.
Provide your answer in exactly this format:
### Answer: premise1, premise2, ..., premiseN
```

Figure 5: **Few-shot system prompt.** The Few-shot system prompt used with the closed models GPT-4o and o3-mini. The set of study examples provided as few-shot examples, along with the query hypothesis are provided as the first user interaction. We then extract the set of premises returned by the model using regular expressions.

C.2 Fine-tuning Details

All models were fine-tuned using Low-Rank Adaptation (LoRA) (Hu et al., 2022) with a rank $r = 64$, alpha value $\alpha = 128$, and dropout probability $p = 0.05$. The adaptation was applied to all attention and linear weight matrices, excluding the embedding and unembedding layers. Baseline models were loaded in `bfloat16` precision, while MIND fine-tuned models employed QLoRA (Dettmers et al., 2023) with 4-bit quantization to accommodate memory constraints from longer sequences. Despite the lower precision, the meta-learning models outperformed the baseline.

Training hyperparameters included a learning rate of 5×10^{-5}, zero weight decay, and no learning rate warmup (steps=0, ratio=0.0). Batch sizes were 4 (training), 8 (validation), and 32 (testing). We used the AdamW optimizer (Kingma and Ba, 2015) with a linear learning rate scheduler. Although we experimented with a range of other hyperparameter configurations, we found this setup to be the most stable across tasks and random seeds. Baseline models were trained for 4 epochs, whereas meta-learning models were trained for only 1 epoch to account for differences in per-sample data exposure (see Section 3.2). We performed 10 validations per epoch and selected the model with the highest validation accuracy. Each fine-tuning run was repeated with three different random seeds: 1048, 512, and 1056.

C.3 Closed Source Models

API details. We accessed OpenAI's closed-source models GPT-4o (OpenAI, 2024) and o3-mini (OpenAI, 2025) through the Azure OpenAI Service's Batch API. The API version used was `2025-03-01-preview`, and the specific model versions were `gpt-4o-2024-08-06` and `o3-mini-2025-01-31`. The total cost of the experiments was approximately 250 USD. For both models, we employed the default API settings. In the case of o3-mini, this corresponds to a "medium" reasoning effort. We did not experiment with a high reasoning effort in order to limit API usage costs.

Prompts. We provide the exact system prompts used in the experiments involving GPT-4o and o3-mini in both the zero-shot (Figure 4) and few-shot (Figure 5) settings. In both cases, the system prompt instructs the models on how to perform the task and specifies the exact format of the answer they should provide. This format facilitates the extraction of the set of premises generated by the models. We then present the query hypothesis as the first user interaction. In the few-shot setting, example interactions are included in the user message prior to the query.

D Additional Results

D.1 Accuracies by Type and Length

In this section, we present the complete set of accuracies broken down by type and length for both MIND and baseline models, as well as closed source models.

MIND and baseline. We report the average accuracy for each inference type and length combination in both the core and length generalization settings for the Qwen-2.5 models. Figures 7, 8, and 9 show the accuracies for core generalization for the 1.5B, 3B, and 7B models, respectively, in both the MIND and baseline settings. Figures 13, 14, and 15 show the accuracies for short to long generalization, while Figures 10, 11, and 12 show the accuracies for long to short generalization for the same models, again in both the MIND and baseline settings.

Across model sizes and approaches, the easiest types of inferences are type 2 and type 6. In contrast, types 1, 3, and 4 are typically the most challenging. A notable difference between the MIND and baseline models is that the latter consistently struggle with type 5 inferences, whereas the former show stronger performance. However, apart from type 5 inferences, MIND models generally perform better but still tend to struggle or excel in similar type and length combinations as the baseline models.

These patterns also hold in the length generalization setting, with the additional observation that performance tends to degrade as the distance between the lengths used for training and those used for testing increases.

Closed models. Figures 16 and 17 show the accuracies for zero-shot and few-shot prompting of GPT-4o and o3-mini, respectively. Both models show substantial improvement in the few-shot setting. GPT-4o is the lowest-performing model according to Table 1, a result further supported by the detailed breakdown in this section. It consistently achieves high accuracy only on type 2 inferences, which are the easiest and rely primarily on simple transitivity. o3-mini struggles more with types 3 and 4. Additionally, a clear difference in performance on type 5 inferences is observed between the zero-shot and few-shot settings. This resembles the difference seen in Qwen-2.5 models between MIND and baseline. These results show that even pretrained models tend to struggle with the same types of syllogistic inferences as fine-tuned models, with a few exceptions, such as type 5 inferences.

D.2 Lexical Generalization

In the main body of the paper, we evaluated core and length generalization. Here, we report an additional set of results related to **lexical generalization**. By *lexical* generalization, we mean the manipulation of the vocabulary assigned to each of the terms appearing in the formulas within $\mathcal{KB}$s.

Section 5.1 presents results using the same vocabulary of pseudowords employed during training, tested on unseen $\mathcal{KB}$s. Here, we explore two more challenging settings: one using a new vocabulary of pseudowords, and another using abstract symbols (e.g., `x2435`) in place of pseudowords. This latter setting is distributionally the most distant from the training data.

Table 6 presents the results of this lexical generalization experiment. Across all Qwen-2.5 model sizes (1.5B, 3B, 7B) and conditions, the MIND approach consistently yields higher accuracy than the baseline, with performance improving with model scale for both approaches. Notably, for both known and unseen pseudowords, performance is similar in

Model	Type	Core	Unseen Pseudowords	Unseen Constants
Qwen-2.5 1.5B	MIND	93.11 ± 0.61	93.15 ± 0.11	74.24 ± 1.07
	Baseline	85.56 ± 1.24	83.34 ± 1.90	38.49 ± 1.06
Qwen-2.5 3B	MIND	96.16 ± 0.44	96.09 ± 0.30	83.21 ± 1.19
	Baseline	93.03 ± 1.15	91.49 ± 0.68	53.12 ± 2.03
Qwen-2.5 7B	MIND	98.13 ± 0.98	98.03 ± 1.19	86.87 ± 0.31
	Baseline	95.76 ± 1.10	94.89 ± 1.55	57.81 ± 2.17

Table 6: **Lexical generalization.** Accuracy (mean ± std) of MIND and Baseline models in core generalization as in the main paper (Core) and with novel unseen terms (Unseen Pseudowords, Unseen Constants).

Model	Type	Core	Long $\rightarrow$ Short	Short $\rightarrow$ Long
Qwen-2.5 1.5B	MIND	76.67 ± 0.38	50.40 ± 3.45	45.81 ± 1.13
	Baseline	55.14 ± 0.53	29.37 ± 1.85	30.22 ± 1.52
Qwen-2.5 3B	MIND	84.68 ± 0.54	64.77 ± 0.73	53.95 ± 3.46
	Baseline	66.51 ± 0.19	43.66 ± 1.93	43.67 ± 2.05
Qwen-2.5 7B	MIND	88.01 ± 1.11	69.24 ± 9.79	60.90 ± 2.94
	Baseline	68.54 ± 2.25	45.27 ± 0.95	43.94 ± 2.82

Table 7: **Generalization in limited data regime.** Accuracy (mean ± std) of meta-learning and baseline models trained and tested on all inference types and lengths (Core), as well as tested for longer or shorter inferences than those seen during training. The models are trained on only 100 examples for each combination of inference type and inference length.

both the MIND and baseline settings, that is, changing the pseudoword vocabulary has little impact on model performance.

In contrast, for the most challenging generalization setting—unseen constants—both approaches exhibit a significant drop in performance, but the performance gap between MIND and the baseline becomes more pronounced: MIND achieves 86.87% at 7B, compared to just 57.81% for the baseline.

D.3 Generalization with Limited Data

Table 7 presents the performance of the models when trained in a low data regime, using only 100 examples for each combination of inference type and length. Consistent with the findings in Table 6 and Table 2, MIND significantly outperforms the baseline across all model sizes and evaluation metrics. For the core generalization performance, the MIND models achieve substantially higher accuracy (e.g., 88.01% for Qwen-2.5 7B MIND vs. 68.54% for baseline). Similarly, when evaluating generalization to shorter and longer inferences than seen during training, MIND models demonstrate a clear advantage.

Crucially, the performance gap between the meta-learning and baseline approaches is notably wider in this limited data setting compared to the standard data setting. This highlights the enhanced generalization capabilities on limited data induced by meta-learning.

D.4 Additional Error Analysis

In this section, we present the additional error analysis results for Qwen-2.5 7B both in MIND and baseline setting on the core generalization experiment. Additionally, we also show the error analysis results for GPT-4o and o3-mini. The detailed breakdown of these errors is presented in Table 8.

MIND and baseline. For the Qwen-2.5 7B model, MIND shows a higher percentage of non-minimal valid set of premises (NVM) errors (17.86%) compared to the baseline (6.67%) on core generalization. However, when these NVM errors occur, MIND includes fewer unnecessary premises on average (Avg. NVM of 2.80) than the baseline (Avg. NVM of 5.19). Conversely, the baseline model exhibits a higher proportion of errors due to missing necessary A premises (MAP) at 91.43%, with an average of 5.39 missing premises.

Model	Method	NVM [%]	Avg. NVM	MAP [%]	Avg. MAP	HP [%]
Qwen-2.5 7B	MIND	17.86	2.80	80.36	3.32	75.00
	Baseline	6.67	5.19	91.43	5.39	80.95
GPT-4o	Few-shot	28.13	2.92	70.54	5.76	22.76
	Zero-shot	14.46	3.50	83.01	6.45	17.15
o3-mini	Few-shot	84.57	2.38	14.23	2.65	7.21
	Zero-shot	76.60	2.61	22.55	7.09	2.62

Table 8: **Error analysis.** Error analysis on core generalization in Qwen-2.5 7B, and the closed models GPT-4o and o3-mini. The table shows percentages and averages for non-minimal valid sets of premises (NVM) and missing necessary A premises (MAP), and the percentage of hallucinated premises (HP).

This is higher than MIND, which has a MAP percentage of 80.36% and an average of 3.32 missing premises. Both methods show high rates of hallucinated premises (HP), with MIND at 75.00% and the baseline slightly higher at 80.95%. These results suggest that not only MIND has generally a higher core generalization performance than the baseline, but also that MIND errors tend to be closer to the correct set of premises.

Closed models. The error analysis for closed models reveals distinct patterns for GPT-4o and o3-mini. For GPT-4o, MAP errors are predominant in both few-shot (70.54%) and zero-shot (83.01%) settings. The average number of missing A premises is also high (5.76 for few-shot and 6.45 for zero-shot) and indicates that the model struggles to provide all the necessary premises to derive hypotheses.

In contrast, o3-mini primarily struggles with NVM errors, which constitute 84.57% of errors in the few-shot setting and 76.60% in the zero-shot setting. The average number of unnecessary premises is relatively low and similar in both settings (2.38 for few-shot, 2.61 for zero-shot). This shows that the model is capable of providing logically valid set of premises from which hypotheses can be derived but, on the other hand, struggles with the concept of minimality. An interesting characteristic of o3-mini is its very low HP rate, at 7.21% for few-shot and an even lower 2.62% for zero-shot, which is considerably better than both Qwen-2.5 7B and GPT-4o.

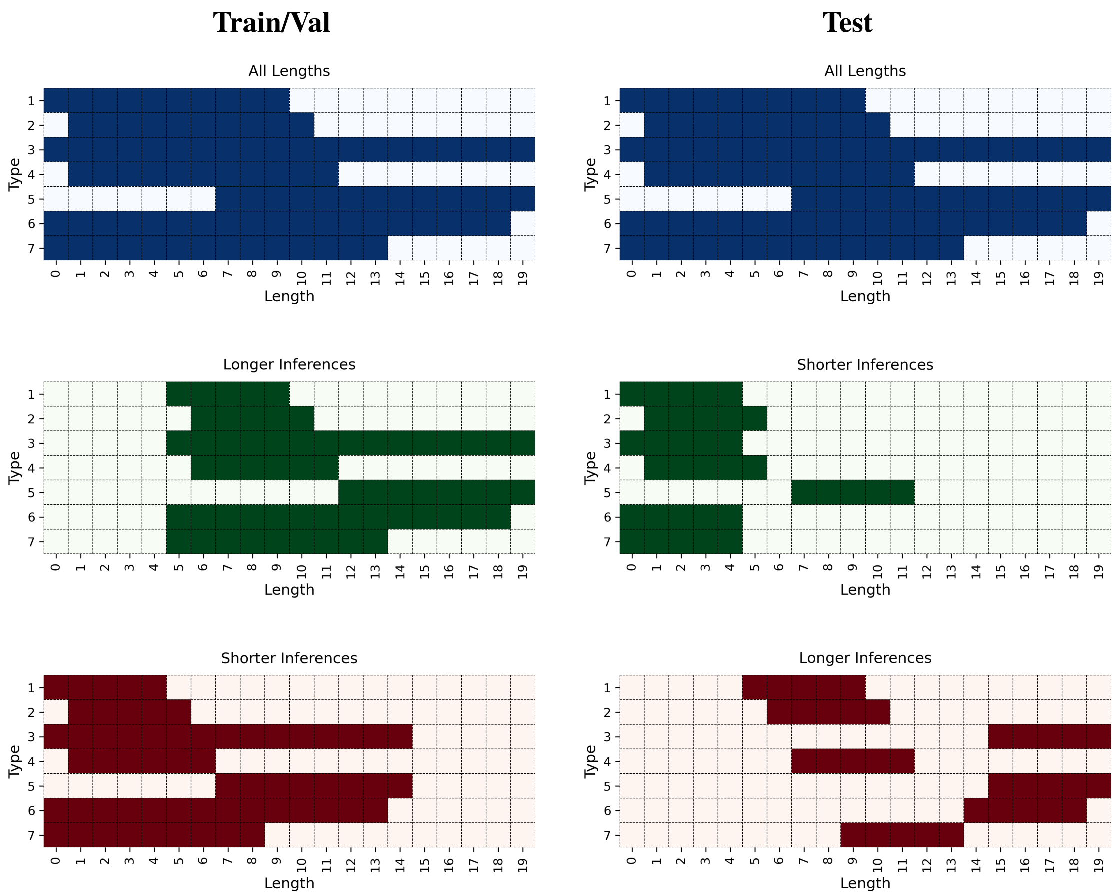


Figure 6: **Combination of inference type and length within generated $\mathcal{KB}$s.** In each heatmap, rows represent Inference Types (1–7), while columns represent Lengths (0–19). The train, validation, and test splits use fixed values of 1000 or 100, 5, and 100 samples respectively for all non-zero entries (Colored). Entries with values equal to 0 indicate non-existing combinations of length and type within the split that is considered (White).

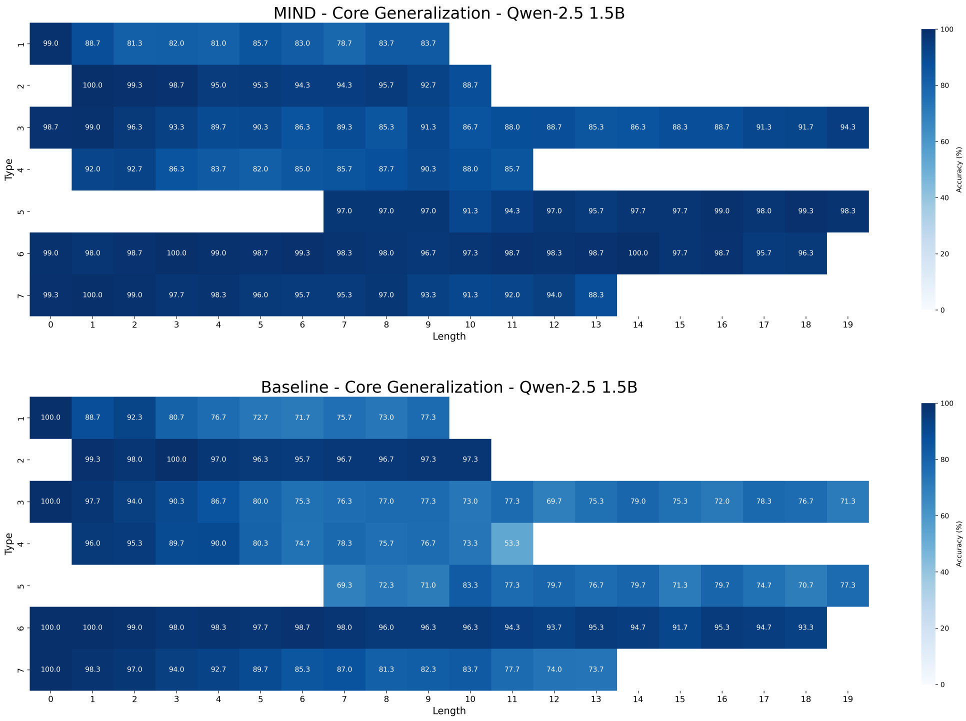


Figure 7: Accuracy of MIND (Top) and Baseline (Bottom) Qwen-2.5 1.5B on core generalization decomposed by inference type and length.

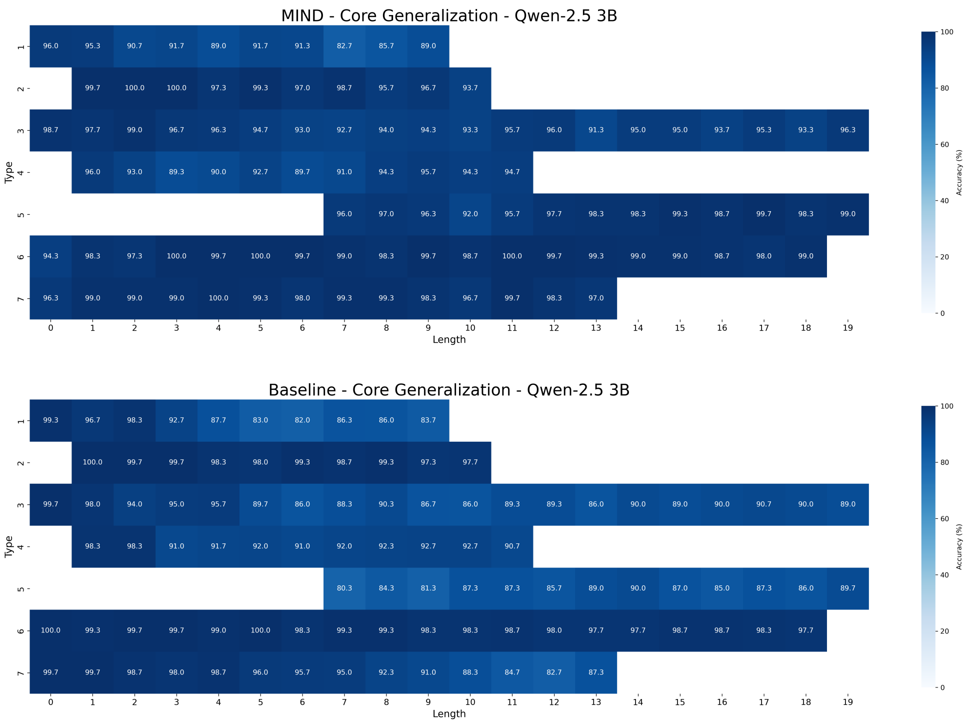


Figure 8: Accuracy of MIND (Top) and Baseline (Bottom) Qwen-2.5 3B on core generalization decomposed by inference type and length.

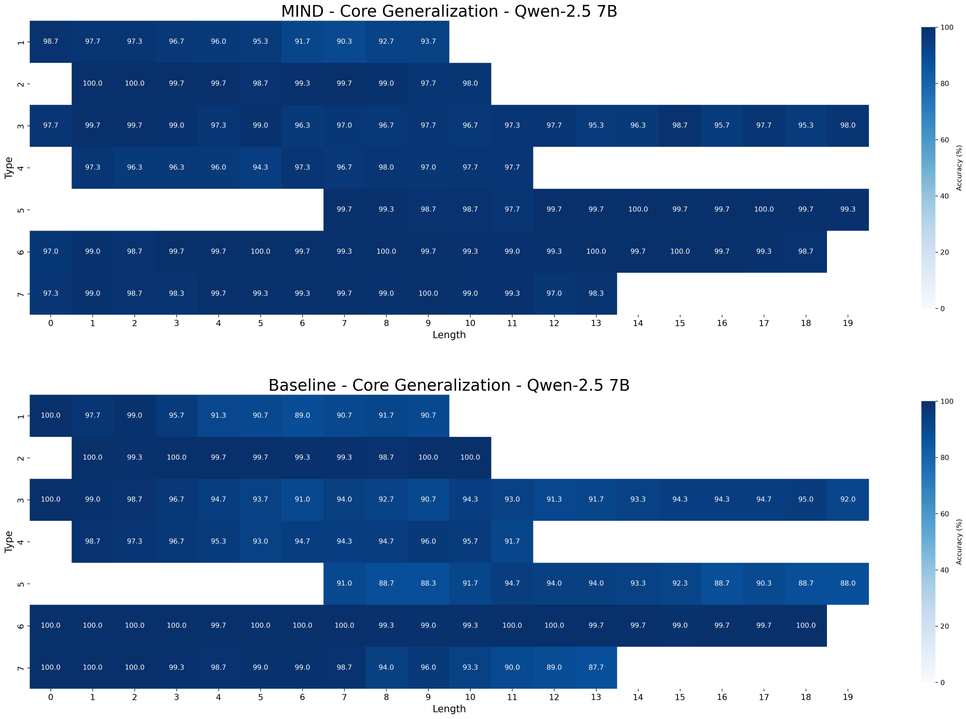


Figure 9: Accuracy of MIND (Top) and Baseline (Bottom) Qwen-2.5 7B on core generalization decomposed by inference type and length.

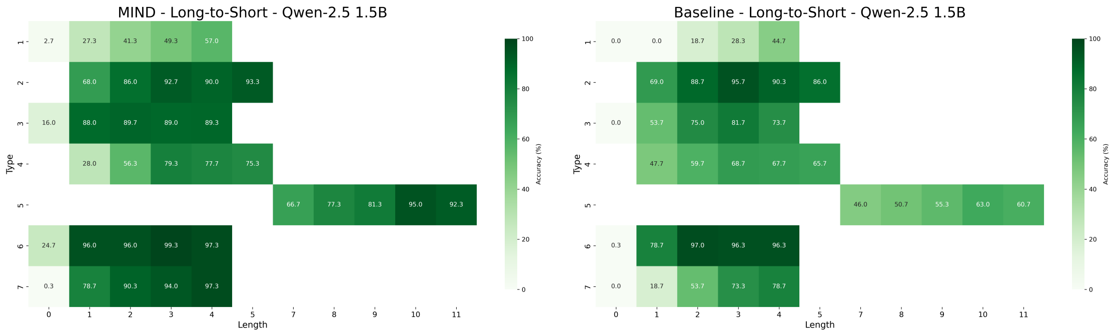


Figure 10: Accuracy of MIND (Left) and Baseline (Right) Qwen-2.5 1.5B on long to short generalization decomposed by inference type and length.

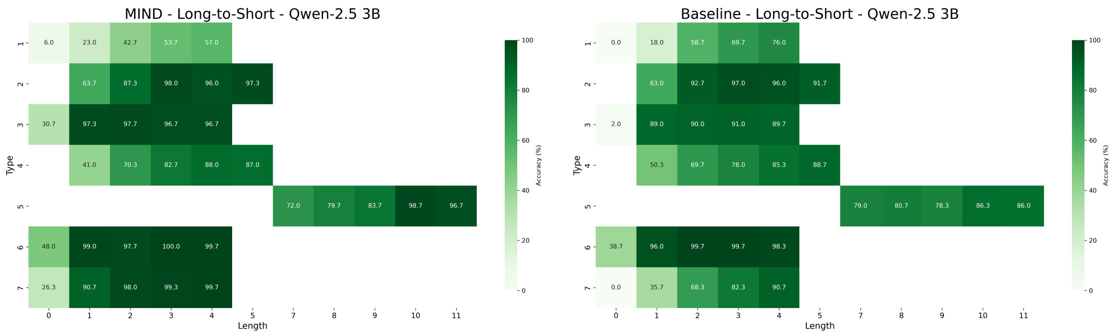


Figure 11: Accuracy of MIND (Left) and Baseline (Right) Qwen-2.5 3B on long to short generalization decomposed by inference type and length.

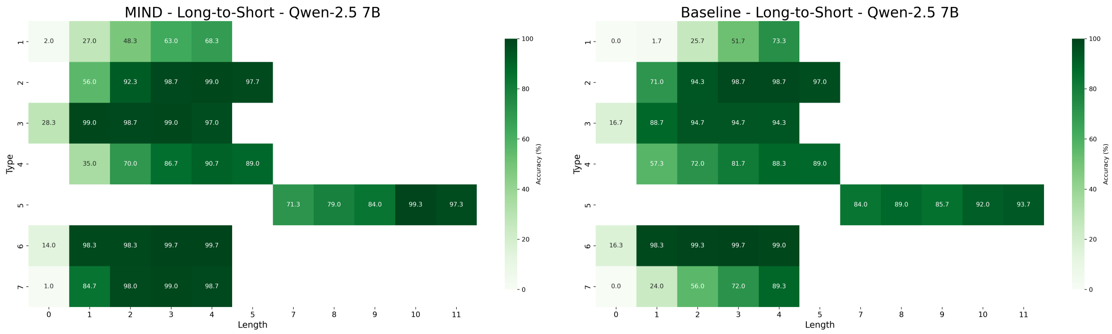


Figure 12: Accuracy of MIND (Left) and Baseline (Right) Qwen-2.5 7B on long to short generalization decomposed by inference type and length.

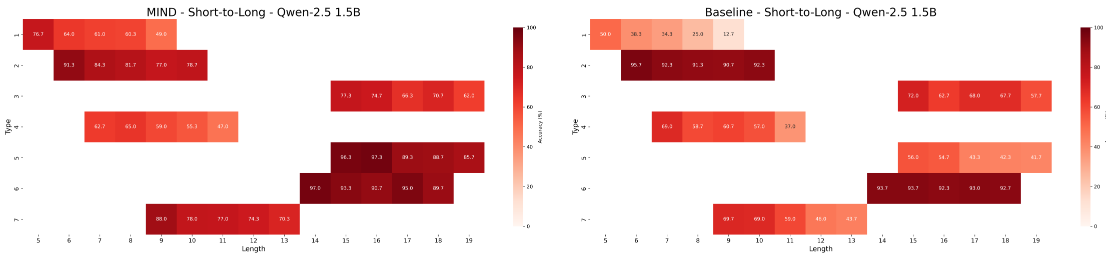


Figure 13: Accuracy of MIND (Left) and Baseline (Right) Qwen-2.5 1.5B on short to long generalization decomposed by inference type and length.

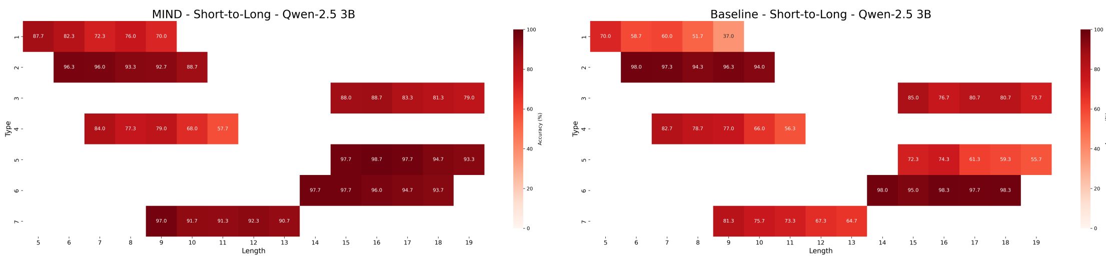


Figure 14: Accuracy of MIND (Left) and Baseline (Right) Qwen-2.5 3B on short to long generalization decomposed by inference type and length.

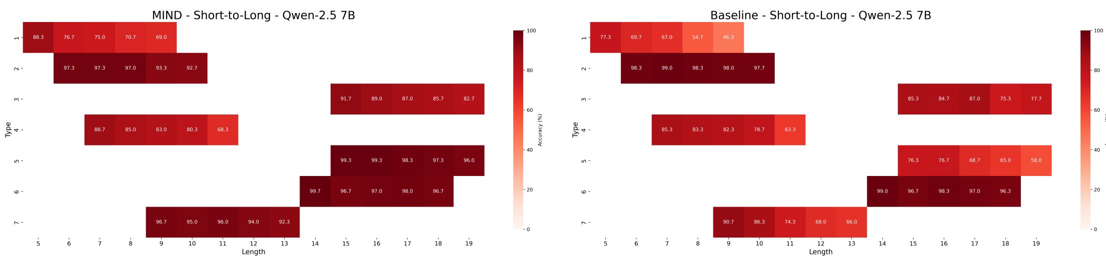


Figure 15: Accuracy of MIND (Left) and Baseline (Right) Qwen-2.5 7B on short to long generalization decomposed by inference type and length.

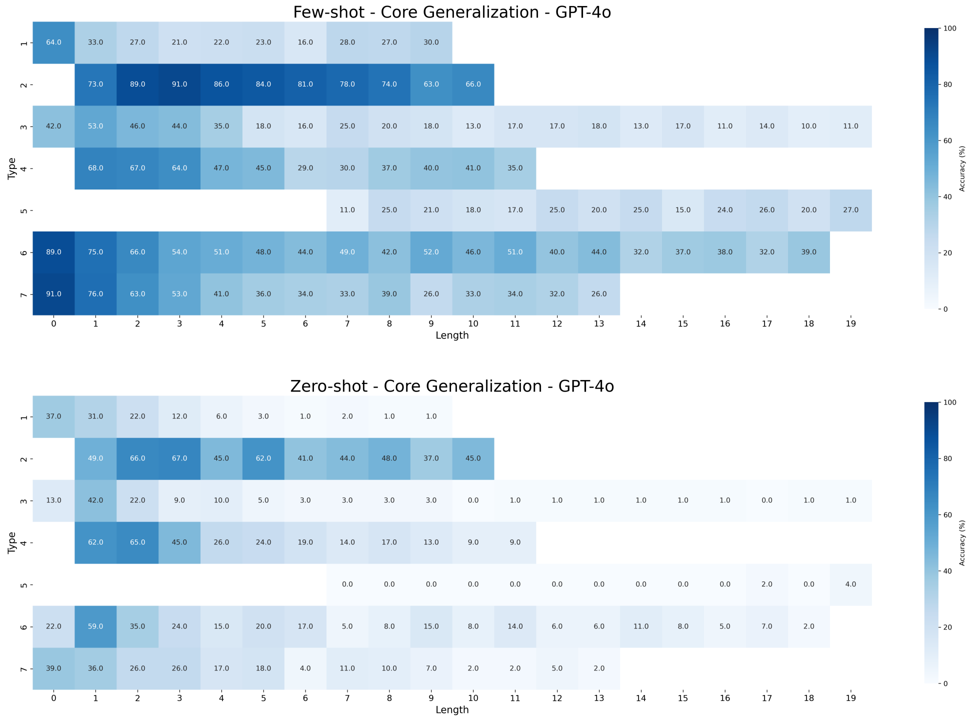


Figure 16: Accuracy of Few-shot (Top) and Zero-shot (Bottom) GPT-4o on core generalization decomposed by inference type and length.

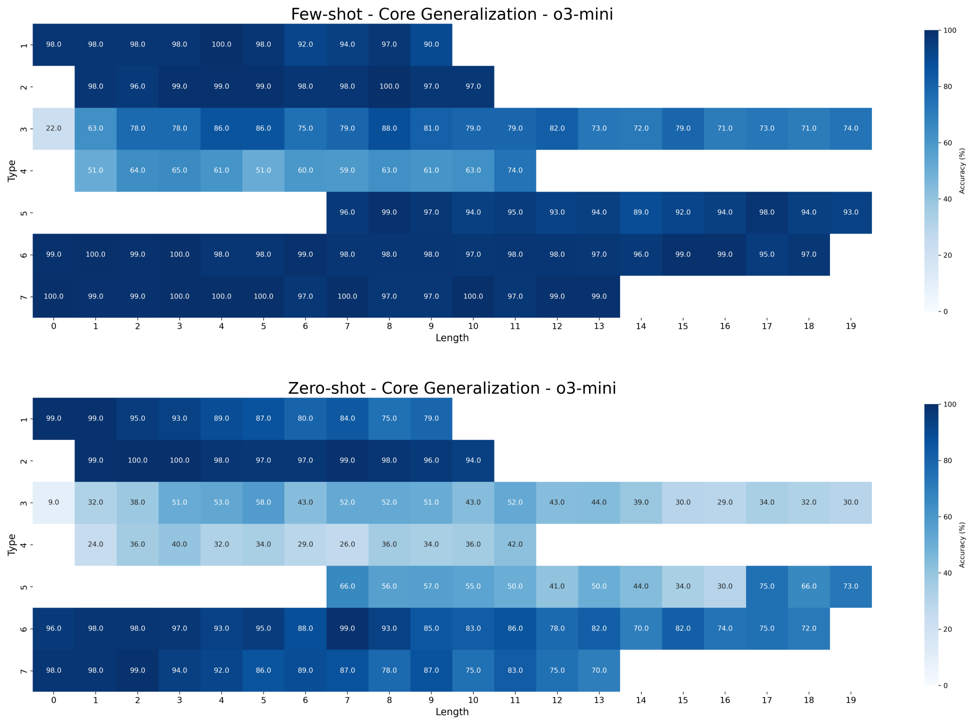
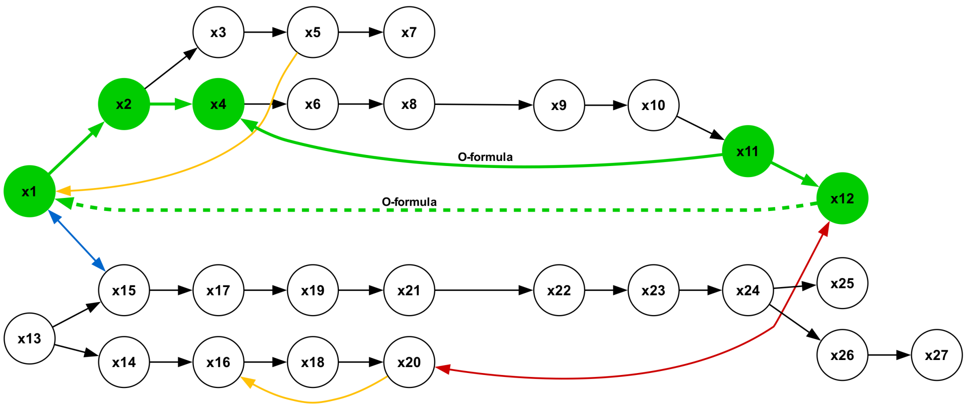


Figure 17: Accuracy of Few-shot (Top) and Zero-shot (Bottom) o3-mini on core generalization decomposed by inference type and length.

KB with Query Hypothesis and Type 1 Inference:



Textual Translation:

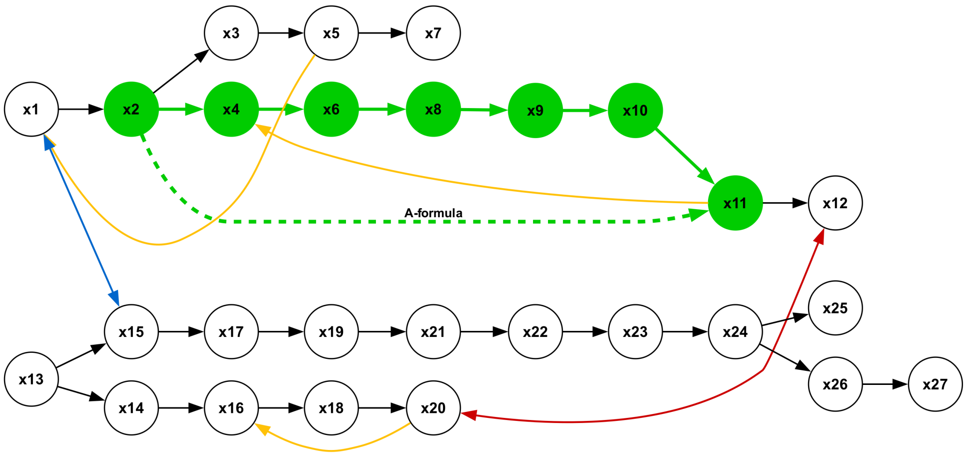
```
knowledge base: All x1 are x2, All x2 are x3, All x2 are x4, All x3 are x5, All x4 are x6, All
x5 are x7, All x6 are x8, All x8 are x9, All x9 are x10, All x10 are x11, All x11 are x12, All
x13 are x14, All x13 are x15, All x14 are x16, All x15 are x17, All x16 are x18, All x17 are
x19, All x18 are x20, All x19 are x21, All x21 are x22, All x22 are x23, All x23 are x24, All
x24 are x25, All x24 are x26, All x26 are x27, No x20 are x12, Some x15 are x1, Some x11 are not
x4, Some x5 are not x1, Some x20 are not x16

hypothesis: Some x12 are not x1

premises: All x1 are x2, All x2 are x4, All x11 are x12, Some x11 are not x4
```

Figure 18: **Type 1 syllogistic inference on graphs**. Visualization of a type 1 syllogistic inference using a graph representation of an example $\mathcal{KB}$, alongside the corresponding textual translation. In the graph (top), nodes represent predicates. Black edges indicate A-formulas ("All As are Bs"), blue edges indicate I-formulas ("Some As are Bs"), red edges indicate E-formulas ("No As are Bs"), and yellow edges indicate O-formulas ("Some As are not Bs"). The query hypothesis is represented by a dashed green edge, and the edges that prove the hypothesis are highlighted in green. The text translation illustrates how the abstract graph representation is converted into a text format suitable for LM processing by applying fixed templates that represent logical formulas.

KB with Query Hypothesis and Type 2 Inference:



Textual Translation:

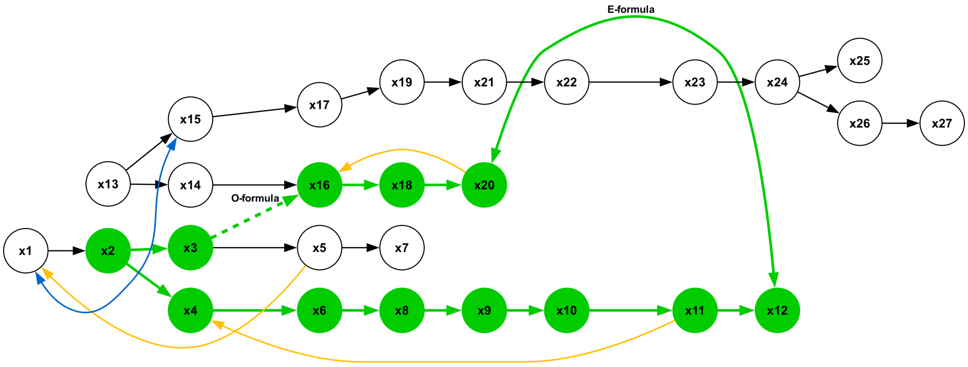
knowledge base: All x1 are x2, All x2 are x3, **All x2 are x4**, All x3 are x5, **All x4 are x6**, All x5 are x7, **All x6 are x8**, **All x8 are x9**, **All x9 are x10**, **All x10 are x11**, All x11 are x12, All x13 are x14, All x13 are x15, All x14 are x16, All x15 are x17, All x16 are x18, All x17 are x19, All x18 are x20, All x19 are x21, All x21 are x22, All x22 are x23, All x23 are x24, All x24 are x25, All x24 are x26, All x26 are x27, No x20 are x12, Some x15 are x1, Some x11 are not x4, Some x5 are not x1, Some x20 are not x16

hypothesis: All x2 are x11

premises: **All x2 are x4**, **All x4 are x6**, **All x6 are x8**, **All x8 are x9**, **All x9 are x10**, **All x10 are x11**

Figure 19: **Type 2 syllogistic inference on graphs**. Visualization of a type 2 syllogistic inference using a graph representation of an example $\mathcal{KB}$, alongside the corresponding textual translation. In the graph (top), nodes represent predicates. Black edges indicate A-formulas (“All As are Bs”), blue edges indicate I-formulas (“Some As are Bs”), red edges indicate E-formulas (“No As are Bs”), and yellow edges indicate O-formulas (“Some As are not Bs”). The query hypothesis is represented by a dashed green edge, and the edges that prove the hypothesis are highlighted in green. The text translation illustrates how the abstract graph representation is converted into a text format suitable for LM processing by applying fixed templates that represent logical formulas.

KB with Query Hypothesis and Type 3 Inference:



Textual Translation:

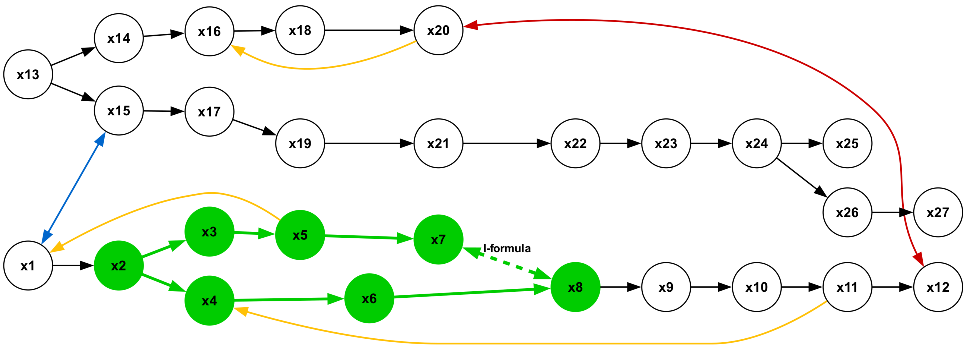
```
knowledge base: All x1 are x2, All x2 are x3, All x2 are x4, All x3 are x5, All x4 are x6, All
x5 are x7, All x6 are x8, All x8 are x9, All x9 are x10, All x10 are x11, All x11 are x12, All
x13 are x14, All x13 are x15, All x14 are x16, All x15 are x17, All x16 are x18, All x17 are
x19, All x18 are x20, All x19 are x21, All x21 are x22, All x22 are x23, All x23 are x24, All
x24 are x25, All x24 are x26, All x26 are x27, No x20 are x12, Some x15 are x1, Some x11 are not
x4, Some x5 are not x1, Some x20 are not x16

hypothesis: Some x3 are not x16

premises: All x2 are x3, All x2 are x4, All x4 are x6, All x6 are x8, All x8 are x9, All x9 are
x10, All x10 are x11, All x11 are x12, All x16 are x18, All x18 are x20, No x20 are x12
```

Figure 20: **Type 3 syllogistic inference on graphs**. Visualization of a type 3 syllogistic inference using a graph representation of an example $\mathcal{KB}$, alongside the corresponding textual translation. In the graph (top), nodes represent predicates. Black edges indicate A-formulas ("All As are Bs"), blue edges indicate I-formulas ("Some As are Bs"), red edges indicate E-formulas ("No As are Bs"), and yellow edges indicate O-formulas ("Some As are not Bs"). The query hypothesis is represented by a dashed green edge, and the edges that prove the hypothesis are highlighted in green. The text translation illustrates how the abstract graph representation is converted into a text format suitable for LM processing by applying fixed templates that represent logical formulas.

KB with Query Hypothesis and Type 4 Inference:



Textual Translation:

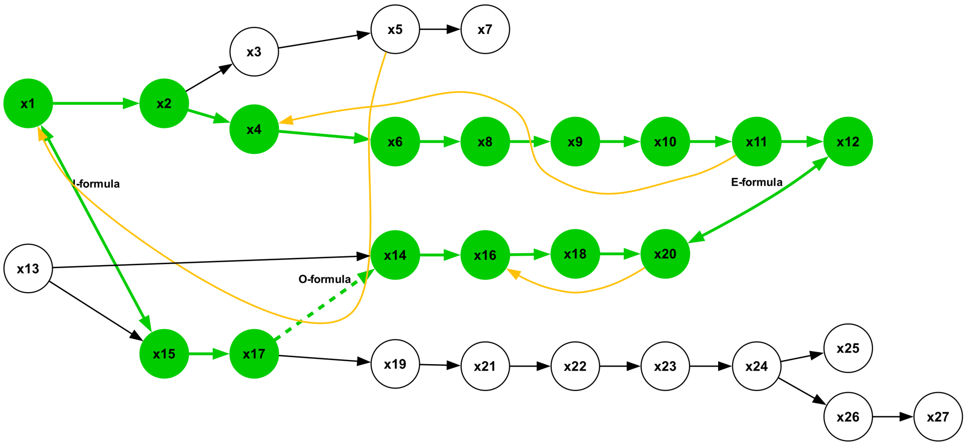
```
knowledge base: All x1 are x2, All x2 are x3, All x2 are x4, All x3 are x5, All x4 are x6, All
x5 are x7, All x6 are x8, All x8 are x9, All x9 are x10, All x10 are x11, All x11 are x12, All
x13 are x14, All x13 are x15, All x14 are x16, All x15 are x17, All x16 are x18, All x17 are
x19, All x18 are x20, All x19 are x21, All x21 are x22, All x22 are x23, All x23 are x24, All
x24 are x25, All x24 are x26, All x26 are x27, No x20 are x12, Some x15 are x1, Some x11 are not
x4, Some x5 are not x1, Some x20 are not x16

hypothesis: Some x7 are x8

premises: All x2 are x4, All x2 are x3, All x4 are x6, All x6 are x8, All x3 are x5, All x5 are
x7
```

Figure 21: **Type 4 syllogistic inference on graphs**. Visualization of a type 4 syllogistic inference using a graph representation of an example $\mathcal{KB}$, alongside the corresponding textual translation. In the graph (top), nodes represent predicates. Black edges indicate A-formulas (“All As are Bs”), blue edges indicate I-formulas (“Some As are Bs”), red edges indicate E-formulas (“No As are Bs”), and yellow edges indicate O-formulas (“Some As are not Bs”). The query hypothesis is represented by a dashed green edge, and the edges that prove the hypothesis are highlighted in green. The text translation illustrates how the abstract graph representation is converted into a text format suitable for LM processing by applying fixed templates that represent logical formulas.

KB with Query Hypothesis and Type 5 Inference:



Textual Translation:

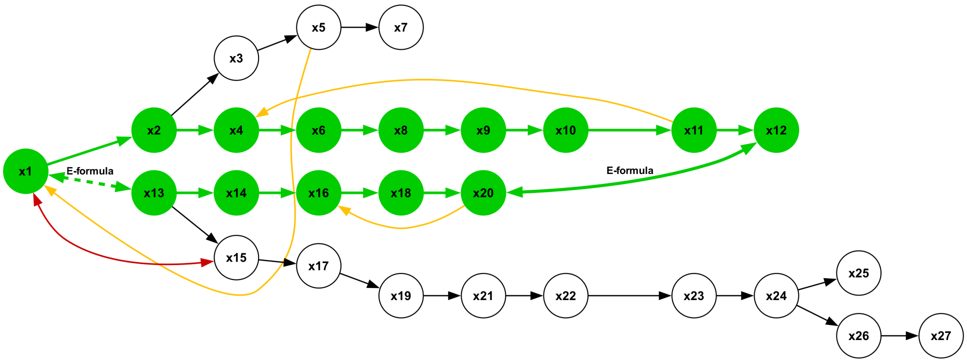
```
knowledge base: All x1 are x2, All x2 are x3, All x2 are x4, All x4 are x6, All x3 are x5, All
x6 are x8, All x5 are x7, All x8 are x9, All x9 are x10, All x10 are x11, All x11 are x12, All
x13 are x14, All x13 are x15, All x14 are x16, All x15 are x17, All x16 are x18, All x17 are
x19, All x18 are x20, All x19 are x21, All x21 are x22, All x22 are x23, All x23 are x24, All
x24 are x25, All x24 are x26, All x26 are x27, No x20 are x12, Some x15 are x1, Some x11 are not
x4, Some x5 are not x1, Some x20 are not x16

hypothesis: Some x17 are not x14

premises: All x1 are x2, All x2 are x4, All x4 are x6, All x6 are x8, All x8 are x9, All x9 are
x10, All x10 are x11, All x11 are x12, All x14 are x16, All x15 are x17, All x16 are x18, All
x18 are x20, No x20 are x12, Some x15 are x1,
```

Figure 22: **Type 5 syllogistic inference on graphs**. Visualization of a type 5 syllogistic inference using a graph representation of an example $\mathcal{KB}$, alongside the corresponding textual translation. In the graph (top), nodes represent predicates. Black edges indicate A-formulas (“All As are Bs”), blue edges indicate I-formulas (“Some As are Bs”), red edges indicate E-formulas (“No As are Bs”), and yellow edges indicate O-formulas (“Some As are not Bs”). The query hypothesis is represented by a dashed green edge, and the edges that prove the hypothesis are highlighted in green. The text translation illustrates how the abstract graph representation is converted into a text format suitable for LM processing by applying fixed templates that represent logical formulas.

KB with Query Hypothesis and Type 6 Inference:



Textual Translation:

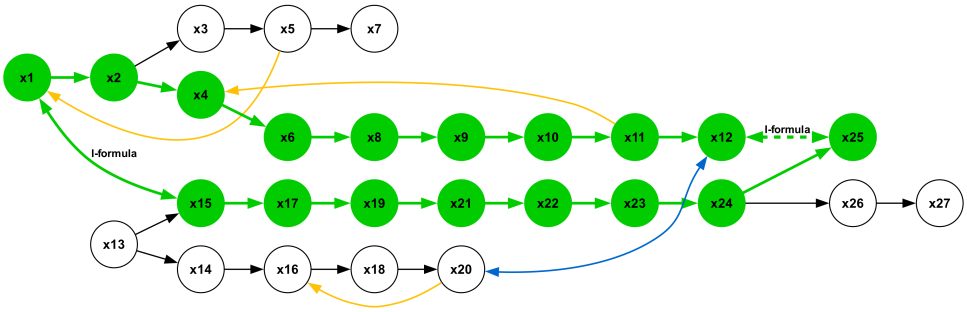
```
knowledge base: All x1 are x2, All x2 are x3, All x2 are x4, All x3 are x5, All x4 are x6, All
x5 are x7, All x6 are x8, All x8 are x9, All x9 are x10, All x10 are x11, All x11 are x12, All
x13 are x14, All x13 are x15, All x14 are x16, All x15 are x17, All x16 are x18, All x17 are
x19, All x18 are x20, All x19 are x21, All x21 are x22, All x22 are x23, All x23 are x24, All
x24 are x25, All x24 are x26, All x26 are x27, No x20 are x12, Some x15 are x1, Some x11 are not
x4, Some x5 are not x1, Some x20 are not x16

hypothesis: No x1 are x13

premises: All x1 are x2, All x2 are x4, All x4 are x6, All x6 are x8, All x8 are x9, All x9 are
x10, All x10 are x11, All x11 are x12, All x13 are x14, All x14 are x16, All x16 are x18, All
x18 are x20, No x20 are x12,
```

Figure 23: **Type 6 syllogistic inference on graphs**. Visualization of a type 6 syllogistic inference using a graph representation of an example $\mathcal{KB}$, alongside the corresponding textual translation. In the graph (top), nodes represent predicates. Black edges indicate A-formulas ("All As are Bs"), blue edges indicate I-formulas ("Some As are Bs"), red edges indicate E-formulas ("No As are Bs"), and yellow edges indicate O-formulas ("Some As are not Bs"). The query hypothesis is represented by a dashed green edge, and the edges that prove the hypothesis are highlighted in green. The text translation illustrates how the abstract graph representation is converted into a text format suitable for LM processing by applying fixed templates that represent logical formulas.

KB with Query Hypothesis and Type 7 Inference:



Textual Translation:

```
knowledge base: All x1 are x2, All x2 are x3, All x2 are x4, All x3 are x5, All x4 are x6, All
x5 are x7, All x6 are x8, All x8 are x9, All x9 are x10, All x10 are x11, All x11 are x12, All
x13 are x14, All x13 are x15, All x14 are x16, All x15 are x17, All x16 are x18, All x17 are
x19, All x18 are x20, All x19 are x21, All x21 are x22, All x22 are x23, All x23 are x24, All
x24 are x25, All x24 are x26, All x26 are x27, No x20 are x12, Some x15 are x1, Some x11 are not
x4, Some x5 are not x1, Some x20 are not x16

hypothesis: Some x25 are x12

premises: All x1 are x2, All x2 are x4, All x4 are x6, All x6 are x8, All x8 are x9, All x9 are
x10, All x10 are x11, All x11 are x12, All x15 are x17, All x17 are x19, All x19 are x21, All
x21 are x22, All x22 are x23, All x23 are x24, All x24 are x25, Some x15 are x1
```

Figure 24: **Type 7 syllogistic inference on graphs**. Visualization of a type 7 syllogistic inference using a graph representation of an example $\mathcal{KB}$, alongside the corresponding textual translation. In the graph (top), nodes represent predicates. Black edges indicate A-formulas ("All As are Bs"), blue edges indicate I-formulas ("Some As are Bs"), red edges indicate E-formulas ("No As are Bs"), and yellow edges indicate O-formulas ("Some As are not Bs"). The query hypothesis is represented by a dashed green edge, and the edges that prove the hypothesis are highlighted in green. The text translation illustrates how the abstract graph representation is converted into a text format suitable for LM processing by applying fixed templates that represent logical formulas.