

Improving Chain-of-Thought for Logical Reasoning via Attention-Aware Intervention

Phuong Minh Nguyen and Tien Huu Dang and Naoya Inoue

Japan Advanced Institute of Science and Technology

{phuongnm, tiendh, naoya-i}@jaist.ac.jp

Abstract

Modern logical reasoning with LLMs primarily relies on employing complex interactive frameworks that decompose the reasoning process into subtasks solved through carefully designed prompts or requiring external resources (*e.g.*, symbolic solvers) to exploit their strong logical structures. While interactive approaches introduce additional overhead or depend on external components, which limit their scalability. In this work, we introduce a *non-interactive, end-to-end* framework for reasoning tasks, enabling reasoning to emerge within the model itself—improving generalization while preserving analyzability without any external resources. We show that introducing *structural information into the few-shot prompt activates a subset of attention heads that patterns aligned with logical reasoning operators*. Building on this insight, we propose Attention-Aware Intervention (AAI), an inference-time intervention method that reweights attention scores across selected heads identified by their logical patterns. AAI offers an efficient way to steer the model’s reasoning toward leveraging prior knowledge through attention modulation. Extensive experiments show that AAI enhances logical reasoning performance across diverse benchmarks, and model architectures, while incurring negligible additional computational overhead. Code is available at https://github.com/phuongnm94/aai_for_logical_reasoning.

1 Introduction

With the rapid pace of recent LLM developments (Achiam et al., 2023; Dubey et al., 2024; Yang et al., 2025), logical reasoning has become a crucial capability for many real-world applications that leverage the reasoning abilities of LLMs (Olausson et al., 2023; Gao et al., 2023a; Pan et al., 2023). In this task, a question is accompanied by a context consisting of predefined *facts*

and *inference rules*. The system is required to perform reasoning over these facts and rules in order to produce the correct answer to the question.

The challenge of logical reasoning tasks lies in the complexity of predefined rules and the large number of reasoning steps that must be processed sequentially. In general, the reasoning process can be divided into three sub-steps: assess whether the available information (premises or facts) is sufficient to answer the question; if not, select the appropriate rule(s) to apply; and derive new premises from the given facts and selected rules (Sun et al., 2024). These sub-steps are repeated until the question is answered. Fig. 1 (left) shows a sample from ProofWriter dataset (Tafjord et al., 2021), where each reasoning step is annotated with numbered tags. Although each step of reasoning *i.e.*, matching conditions and generating new premises is relatively trivial for humans, LLMs often make mistakes such as selecting the incorrect rules for the given step or failing to determine when the reasoning process should terminate (Zhang et al., 2025).

To address this challenge, prior work has investigated two main approaches. The first approach integrates external symbolic solvers (Fig. 1 middle), delegating the reasoning process to automated theorem provers or programming languages (Ye et al., 2023; Pan et al., 2023; Gao et al., 2023b; Xu et al., 2024a). While symbolic solvers can achieve strong performance, they often rely on highly capable LLMs (*e.g.*, ChatGPT) to reliably translate natural language descriptions into formal logical representations (*e.g.*, first-order logic) (Ye et al., 2023; Gao et al., 2023b). The second approach constructs frameworks that systematically decompose complex reasoning into sub-tasks such as rule selection, premise inference, and scoring to improve reliability in solving individual steps (Zhang et al., 2025; Feng et al., 2024; Sun et al., 2024; Xu et al., 2024b). This approach highlights sub-task specialization as a key factor in improving the faith-

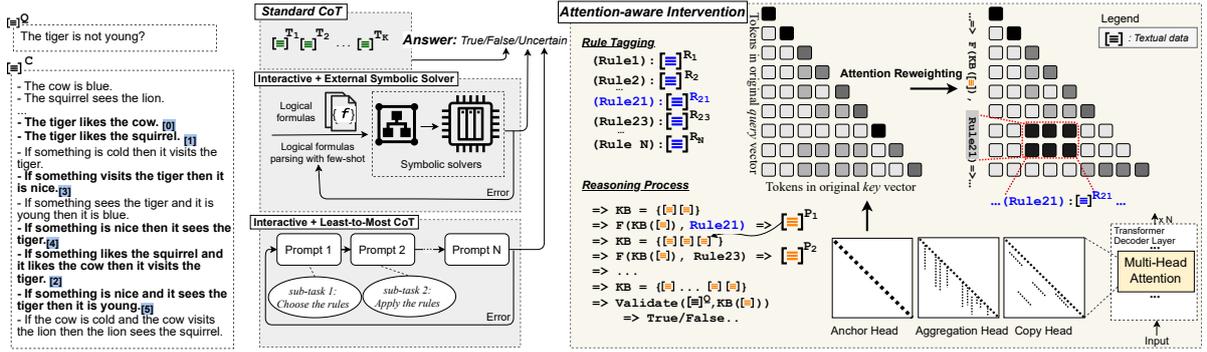


Figure 1: Overview of the comparison between the proposed method and existing approaches.

fulness of the reasoning process. However, such methods do not directly evaluate the intrinsic reasoning ability of LLMs, that is, their capacity to *solve the entire problem step by step in a single pass (non-interactive)*. Nguyen et al. (2025) introduced Symbolic-Aided CoT prompting, which incorporates a lightweight symbolic structure into the input. This structure abstractly represents fundamental reasoning operators, including rule selection, inference, and knowledge updating. However, its efficiency has so far been empirically validated, hindering a better understanding of how and why the injected structural information enhances LLMs’ logical reasoning capabilities. In this paper, we pose a general question:

RQ: How do LLMs encode and utilize structural information in their latent representations for logical reasoning?

We begin with a preliminary analysis of attention heads in LLM for logical reasoning. We observe that injecting symbolic structures into few-shot prompts induces structured activation patterns in certain attention heads, aligning with specific logical reasoning operations. We categorize attention heads into three functional types. These include *anchor heads*, which are responsible for memorizing information; *aggregation heads*, which integrate newly derived premises and propagate information to the subsequent words of the generation process, and *copy heads*, which replicate newly generated premises when updating the knowledge base or applying matching rules.

Inspired by recent findings (Sun et al., 2024; Xu et al., 2024b), which demonstrate that each logical reasoning sub-task e.g., rule selection or new premise generation can be individually inferred by specific few-shot prompts, we introduce

Attention-Aware Intervention (AAI), a lightweight and model-agnostic intervention method that enables the model to *focus on focal rules being considered at each inference step while reducing noise from unrelated rules*—as a form of prior knowledge. Extensive experiments demonstrate that AAI enhances logical reasoning performance on well-known benchmarks, including ProofWriter, PrOntoQA, Logical Deduction, and FOLIO across model architectures such as Qwen-3, Phi-4, OLMo-2, and sizes ranging from 1.7B to 32B while introducing negligible additional computational overhead. Moreover, with the Qwen-3 32B model, our method achieves competitive performance with state-of-the-art approaches, even compared to those using more powerful closed-source LLMs (e.g., gpt-4) with external interactive modules or symbolic solvers.

2 Preliminary Experiments

Problem. Logical reasoning requires deriving an answer A to a question Q from a set of N contextual rules $\mathcal{R} = \{r_i\}_i^N$ where only a relevant subset $\mathcal{R}^* \subset \mathcal{R}$ supports the correct inference graph (Weston et al., 2015; Tafjord et al., 2021).

Experimental setting. In this section, we conduct preliminary experiments to examine the effect of injecting structural information into few-shot CoT prompting on the distribution of attention heads in the Transformer’s multi-head layers. Specifically, we reproduce the Symbolic-Aided CoT method (Nguyen et al., 2025) and analyze the typical patterns of attention scores to explain why Symbolic-Aided CoT achieves significant performance on logical reasoning tasks.

In Symbolic-Aided CoT prompting (Nguyen et al., 2025), the authors first tag the rules and facts in the logical reasoning problem as variables (e.g.,

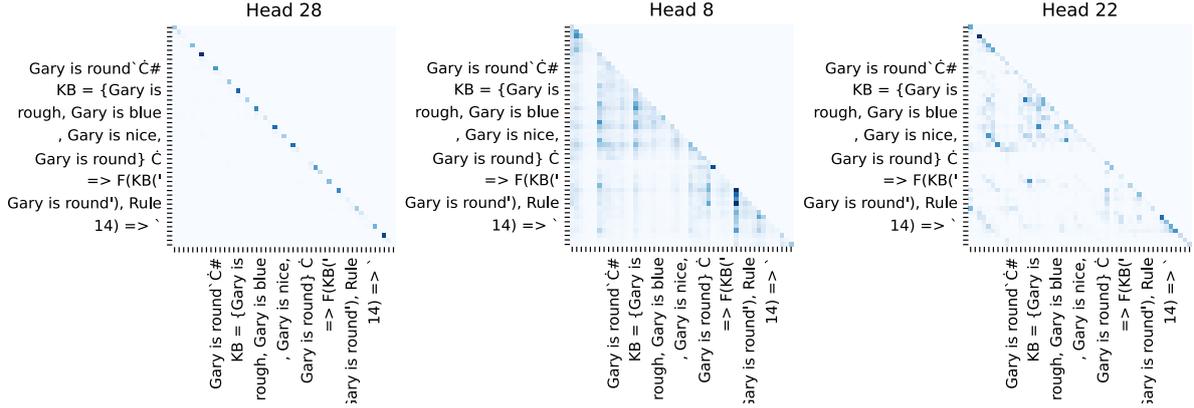


Figure 2: Visualization of attention heads in the Qwen3-8B model on an example (a generated reasoning output) from the ProofWriter dataset. Higher attention scores are represented by stronger colors. For ease of interpretation, every five subword tokens are grouped along both the x- and y-axes in the figure caption. A more detailed (or full-scale) version of this figure is provided in Fig. 7).

Rule1). A set of symbolic tokens (e.g., =>, F) is then injected into the prompt to represent the structure of the reasoning flow. The inference process is constructed using reasoning operators, including *rule selection*, *new premise inference*, and *knowledge base (KB) updating*.

The experiments are performed using two-shot Symbolic-Aided CoT on the ProofWriter dataset (Table 1 shows a generated reasoning output produced by Qwen3-8B LLM). We visualize the attention matrix values in Fig. 2. These values are computed as $\text{softmax}(qk^T/\sqrt{d})$, where q and k denote the query and key matrices in the last transformer decoder layer, and d is their dimensionality (Vaswani et al., 2017). The x- and y-axes correspond to the key and query tokens, respectively.

Observations. *Attention structuring.* Firstly, consistent with the observations of Nguyen et al. (2025), we found that these logical symbols are represented separately in the hidden semantic space. In the self-attention representations, these symbols receive significantly less attention than informative words such as “Gary” or “round”. This discrepancy causes discontinuities in the self-attention matrix (shown as white or colorless points in Fig. 2).

Attention pattern. Secondly, we found that a subset of attention heads in the self-attention layer reflects the reasoning operators, as shown by the highlighted patterns corresponding to strong word-pair dependencies. In the Transformer decoder’s self-attention layer, the information of a query token vector is aggregated from previous token vectors, with the proportion of contribution represented in the attention matrix (further detail in Appendix B).

Table 1: *Symbolic-Aided CoT* prompting. The colored text spans were used for attention visualization.

```

...
# (Rule8): Rough things are blue.
# (Rule9): Red, quiet things are blue.
# (Rule10): All rough, blue things are nice.
# (Rule11): If something is quiet and round then it is not nice.
# (Rule12): If Harry is red then Harry is rough.
# (Rule13): If something is nice then it is round.
# (Rule14): If something is round then it is red.
# (Rule15): Red things are not quiet.
# (Question): Based on the above information, is the following
statement true, false, or unknown? Gary is red.
# (Answer): Start from the object and their condition mentioned in
the question to collect relevant facts: Gary, is red
# KB = {}
=> Rule6 = "Gary is rough"
# KB = {Gary is rough}
=> F(KB("Gary is rough"), Rule8) => "Gary is blue"
# KB = {Gary is rough, Gary is blue}
=> F(KB("Gary is blue"), Rule10) => "Gary is nice"
# KB = {Gary is rough, Gary is blue, Gary is nice}
=> F(KB("Gary is nice"), Rule13) => "Gary is round"
# KB = {Gary is rough, Gary is blue, Gary is nice, Gary is round}
=> F(KB("Gary is round"), Rule14) => "Gary is red"
# KB = {Gary is rough, Gary is blue, Gary is nice, Gary is round,
Gary is red}
# valid the question with current inferred premises
=> Validate(Gary is red, KB("Gary is red")) = True.

```

In Fig. 2 - Head 28, each token primarily attends to itself, without incorporating information from other tokens. The highlighted points mainly correspond to informative words (e.g., “Gary”, “round”). We identify this type of head as an *anchor head*, which serves to retain premise-related information. In the second image of Fig. 2 (Head 8), the newly generated premise tends to distribute its information to subsequent inference steps. For example, the premise “Gary is round” contributes to the representations of many following tokens, corresponding to the vertical highlights observed in the at-

tention map. We identify this type of head as an *aggregation head*, which supports matching the generated premises with the applicable rules in the logical inference process. The last image (Head 22) presents another attention pattern characterized by short diagonal alignments, reflecting the copying of an entire information span to another span. For instance, the premise ‘‘Gary is round’’ is copied to the knowledge base (KB) variable after it is generated. We identify this type of head as a *copy head*.

Motivation. Building on the previous observations, we address the second research question (RQ2): *how can we modify the attention matrix to effectively inject prior knowledge into LLMs?* In the Symbolic-Aided CoT framework, each rule is tagged with a symbolic identifier (Rule<index>), requiring the model to retrieve and maintain the corresponding rule information throughout the reasoning process. However, this design can lead to information decay in long reasoning chains, as the identifiers and their associated rule contents often appear in distant spans (e.g., Rule14 in Table 1). Furthermore, in copy heads, the copying behavior typically occurs within the same or paraphrased token spans, whereas the original rule content should instead be propagated to the representation of its identifier during reasoning. Similarly, in anchor heads, attention tends to focus on the identifiers themselves rather than capturing the semantics of the underlying rule content.

To address this issue, we propose a mechanism that first categorizes *anchor heads* and *copy heads* and then adaptively reweights them to reinforce the linkage between rule identifiers and their original rule representations. In essence, this process injects structural prior knowledge into the attention mechanism, ensuring that *each rule reference remains grounded in its semantic content during reasoning*.

3 Methodology

In this section, we present the details of our proposed method for analyzing attention head patterns to select attention head patterns (e.g., *anchor heads*), and our *Attention-Aware Intervention* mechanism for reweighting their attention scores to inject prior knowledge into LLMs.

Transformer decoder-only architectures are the dominant design for recent LLMs (Radford et al., 2019; Dubey et al., 2024; Yang et al., 2025). In these models, semantic features representing the meaning of an input sequence are computed within

the multi-head attention layers. Specifically, the self-attention mechanism captures dependencies among word pairs. For generation tasks, a causal mask ($M = M^{\text{Causal}}$) is applied to prevent tokens from attending to future positions (Vaswani et al., 2017). Given, query and key matrices, $q, k \in \mathbb{R}^{L \times d}$ general formulation of attention is:

$$S = \frac{qk^\top}{\sqrt{d}} \quad (1)$$

$$A = \text{softmax}(S + M) \quad (2)$$

$$h^{\text{attn}} = A \cdot v \quad (3)$$

Beyond this standard use, masked matrices have also been explored as a means of injecting prior knowledge into Transformer architectures. For example, in neural machine translation, masks have been employed to encode syntactic dependency structures (Bugliarello and Okazaki, 2020; Wang et al., 2019) or to model localness (Yang et al., 2018). Similarly, in emotion recognition tasks, masks have been utilized to represent intra- and inter-utterance relationships (Liu et al., 2022; Zhao et al., 2022). However, in the context of in-context learning with LLMs, no training is performed, which makes customizing masks for prior knowledge injection significantly more challenging. Motivated by these observations, we propose an efficient method to inject *reference information* into the decoder-only Transformer architecture by reweighting the masked matrix M .

3.1 Attention Head Analysis

In this step, our objective is to identify attention patterns by analyzing the relative positional relationships between token pairs exhibiting strong dependencies. Given an attention score matrix $A \in \mathbb{R}^{L \times L}$ (as defined in Eq. 2), where L denotes the length of the input sequence, each element (A_{ij}) represents the dependency score between the i -th query token and the j -th key token.

$$H_{ij} = \begin{cases} 1 & \text{if } (A_{i,j} > \text{threshold}) \wedge (i \geq j) \\ 0 & \text{otherwise} \end{cases}$$

$$s^{\text{diagonal}} = \frac{\sum_{i,j|i,j < L-1} (H_{i,j} \wedge H_{i+1,j+1})}{\sum_{i,j} H_{i,j}} \quad (4)$$

$$s^{\text{vertical}} = \frac{\sum_{i,j|j < L-1} (H_{i,j} \wedge H_{i,j+1})}{\sum_{i,j} H_{i,j}} \quad (5)$$

$$s^{\text{horizontal}} = \frac{\sum_{i,j|i < L-1} (H_{i,j} \wedge H_{i+1,j})}{\sum_{i,j} H_{i,j}} \quad (6)$$

$$\text{pattern}(A) = (s^{\text{diagonal}}, s^{\text{vertical}}, s^{\text{horizontal}}) \quad (7)$$

where *threshold* is a hyperparameter used for binarization. The head’s pattern is characterized by directional scores: s^{diagonal} , s^{vertical} , and $s^{\text{horizontal}}$, which quantify the relative density of attention edges along the diagonal, vertical, and horizontal directions, respectively. Aggregation heads exhibit high vertical scores, reflecting strong dependencies between informative tokens. For instance, in Head 8 of Fig. 2, the token “round” receives concentrated attention from subsequent query tokens. In contrast, anchor heads generally display high diagonal but low vertical and horizontal scores, indicating self-focused attention patterns.

3.2 Attention Head Reweighting

This component is designed to inject human prior knowledge into LLMs through a masked matrix (M). By default, this matrix is defined as the causal mask (M^{Causal}), which prevents information flow from future tokens during generation. In Symbolic-Aided CoT prompting for logical reasoning tasks, the rule identifier tokens (Rule<number>) should be semantically linked with their corresponding rule content. To enable this, we construct two subsets of token pairs: \mathcal{D}^{Ref} , which includes pairs connecting a rule identifier to tokens within its own rule span, and $\mathcal{D}^{\text{NoRef}}$, which includes pairs connecting a rule identifier to tokens’ index in other rules. Intuitively, dependencies in \mathcal{D}^{Ref} should be encouraged, whereas those in $\mathcal{D}^{\text{NoRef}}$ should be suppressed. For instance, in Fig. 1, Rule21 illustrates such relationships within the \mathcal{D}^{Ref} set.

Given the scaled dot-product values S computed by Eq. 1, and considering that the range of hidden-state values differs across attention heads, we adaptively compute median-based normalization for each head during the reweighting process. Accordingly, we define two masking matrices, $M_{i,j}^{\text{Ref}}$ and $M_{i,j}^{\text{NoRef}}$, to reinforce the representation of rule identifiers throughout the reasoning process. These matrices can be applied independently or combined to integrate both effects, and are used to re-weight selected attention heads in the previous step.

$$M_{i,j}^{\text{Causal}} = \begin{cases} -\infty & \text{if } i < j \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

$$M_{i,j}^{\text{NoRef}} = \begin{cases} -\infty & \text{if } (i, j) \in \mathcal{D}^{\text{NoRef}} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

$$M_{i,j}^{\text{Ref}} = \begin{cases} c \times \text{median}(S) + b & \text{if } (i, j) \in \mathcal{D}^{\text{Ref}} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

$$M^{\text{Final}} = M^{\text{Ref}} + M^{\text{NoRef}} + M^{\text{Causal}} \quad (11)$$

Here, c (reweighting coefficient) and b (reweighting bias) are configuration parameters that control the degree of reweighting (Eq. 10). Larger values of c strengthen the connections within the \mathcal{D}^{Ref} set in alignment with the original attention matrix values. In contrast, increasing the constant b introduces a uniform bias that does not depend on the original attention values.

4 Experiments and Result Analysis

In this section, we present the experimental settings, datasets, and evaluation metrics used to assess our proposed method. We then report the main results compared with several strong baselines across multiple logical reasoning benchmarks. Finally, we provide an in-depth analysis and ablation studies to investigate the effectiveness of each component in our approach.

4.1 Experimental Setup

Datasets. We evaluate our approach on four widely used benchmark datasets for logical reasoning and one mathematical reasoning dataset. (1) **ProofWriter** (Tafjord et al., 2021): we adopt the subset constructed under the open-world assumption, where each instance can take one of three possible labels—true, false, or unknown. Following the experimental setup of Pan et al. (2023), we focus on the subset with the highest reasoning complexity (five-hop reasoning), which comprises 600 examples. (2) **ProntoQA** (Saparov and He, 2023): similar to ProofWriter, this dataset contains multi-hop logical reasoning problems. Consistent with prior studies (Qi et al., 2025; Nguyen et al., 2025), we use the most challenging subset that requires five-hop reasoning, containing 500 evaluation samples. (3) **LogicalDeduction** (Srivastava et al., 2023): this dataset evaluates the ability to deduce the correct ordering of entities based on a set of logical constraints. Following the configuration of Sun et al. (2024), we use 300 samples that include subsets of three, five, and seven objects, where an increased number of objects corresponds to greater reasoning complexity. (4) **FOLIO** (Han et al., 2024): this dataset is constructed by domain experts, in which logical rules are compiled using real-world knowledge. Following the experimental settings of previous work (Sun et al., 2024), we use the validation subset of this dataset, comprising 204 examples, for evaluation. (5) **GSM8k** (Cobbe et al., 2021): GSM8k is a collection of high-quality, linguisti-

cally diverse grade-school math word problems. We evaluate our method on the test subset, consisting of 1319 examples, to assess our method’s generalization capability.

Evaluation metric. We employ accuracy as the evaluation metric, which is a standard measure in prior work (Sun et al., 2024; Qi et al., 2025; Nguyen et al., 2025). This choice enables direct and consistent comparison of our results with existing methods on the same benchmarks.

Experimental settings. We conducted our experiments primarily on open-source LLMs, including Qwen-3 (Yang et al., 2025), OLMo (Groeneveld et al., 2024), and Phi-4 (Abdin et al., 2024). The primary objective of this study is to evaluate the effectiveness of the proposed Attention-Aware Intervention (AAI) in comparison with the baseline *Symbolic-Aided CoT* (Nguyen et al., 2025) and standard CoT prompting. To further validate the generalizability of the AAI mechanism, we introduce a new structured reasoning prompt, termed *Compact Symbolic-Aided CoT*, which employs symbolic representations to provide a minimal structure for the reasoning process. The prompting templates used in both the original *Symbolic-Aided CoT* and the proposed *Compact* version are provided in Appendix E.

We design two experimental settings: (1) Main AAI setting – we perform attention head analysis to select *anchor heads* and *copy heads* with $s^{\text{diagonal}} > 0.3$, then apply the Attention Head Reweighting mechanism to these heads; (2) AAI^{agg.} setting – we apply the same reweighting mechanism to *aggregation heads*, selected under the conditions $s^{\text{vertical}} > 0.6$ and $s^{\text{horizontal}}, s^{\text{diagonal}} < 0.3$. In addition, we conducted extensive experiments to analyze the sensitivity of model performance to hyperparameters, including the binarization, diagonal score thresholds, reweighting coefficient, and bias.

Furthermore, we performed a detailed scalability analysis on Qwen-3 models of different sizes—1.7, 4, 8, and 32B—to assess the consistency of our method across model scales. Our experiments were conducted on NVIDIA A40 (46 GB memory) and H200 (140 GB memory) GPUs for larger LLMs. For simplicity of implementation, AAI was applied only to reweight attention scores during the input-prompting (prefill) phase. We employed greedy decoding, selecting the highest-logit token at each generation step to produce deterministic outputs.

4.2 Experimental Results

Main results. We evaluate the performance of the proposed method from two perspectives: (1) scalability with respect to model size (Fig. 4), and (2) generalization across different LLMs (Table 2). In Table 2, the proposed method demonstrates the generalization capability. Specifically, it yields improvements of +2.5 and +3.0 accuracy on the ProofWriter and LogicalDeduction datasets, respectively, when applied to the OLMo-2 32B model. Moreover, when using Phi-4, AAI achieves an additional +0.66 accuracy gain on the LogicalDeduction dataset.

As shown in Fig. 4, by using Qwen-3 models, our method (AAI – represented by the red diamond line) exhibits increasingly stronger gains as the model size grows—particularly on the ProofWriter and Logical Deduction datasets—achieving notable improvements of +2.83 and +1.67 accuracy, respectively, over the baseline (SymbolA.CoT). For the ProntoQA dataset, the performance reaches near saturation with the 32B model (over 99% accuracy); hence, the improvement gains are more pronounced for the 8B and 14B model sizes.

Our AAI mechanism also demonstrates strong generalization across diverse structured prompting strategies and domains. It consistently improves performance across datasets and LLMs when combined with the *Compact Symbolic-Aided CoT* prompting and remains effective on GSM8k, a mathematical reasoning benchmark (Table 2). Notably, under the GSM8k prompting setting (Table 12 in Appendix E), few-shot prompting enables LLMs to autonomously decompose the input context into rule-like structures. The key principle underlying the application of our AAI mechanism is to decompose the overall problem into subproblems and to appropriately steer the attention of LLMs to address each subproblem through a structured reasoning process. These results highlight the potential applicability of our approach to other reasoning tasks, such as commonsense and multi-hop reasoning.

Ablation studies. To evaluate the contribution of each sub-component in our proposed method, including the effects of Attention Head selection and reweighting, we conducted ablation studies on two datasets that are not affected by saturation performance: ProofWriter and LogicalDeduction (Fig. 3). In comparison, the AAI method tends to select copy heads and anchor heads, while AAI^{agg.}

Table 2: Performance comparison between our method and existing approaches. *E.Solver* denotes systems that incorporate an external symbolic solver module. Results marked with * are reproduced or obtained in this work. The **best** performance in each category is highlighted.

Methods + LLMs	Interaction mode	ProofWriter	ProntoQA	L.Deduction	FOLIO	GSM8k
<i>Closed-source LLM: GPT-4</i>						
CoT (Sun et al., 2024)	Non-Interactive	67.41	91.00	73.33	67.65	–
Logic-LM (Pan et al., 2023)	Interactive +E.Solver	79.66	83.20	87.63	78.92	–
DetermLR (Sun et al., 2024)	Interactive +Program	79.17	98.60	85.00	75.49	–
SymbCoT (Xu et al., 2024b)	Interactive +Program	82.50	99.60	93.00	83.33	–
Symbolic-Aided CoT (Nguyen et al., 2025)	Non-Interactive	77.09	100.00	86.33	74.51	–
<i>Qwen3-32B</i>						
Symbolic-Aided CoT (Nguyen et al., 2025)*	Non-Interactive	79.67	99.20	88.00	69.61	–
+AAI (ours)*	Non-Interactive	82.50	99.40	89.67	71.09	–
Compact Symbolic-Aided CoT (ours)*	Non-Interactive	74.00	91.20	74.67	70.59	91.81
+AAI (ours)*	Non-Interactive	74.83	95.20	75.33	75.00	92.27
<i>OLMo-2-0325-32B-Instruct</i>						
Symbolic-Aided CoT (Nguyen et al., 2025)*	Non-Interactive	65.00	92.60	51.33	64.22	–
+AAI (ours)*	Non-Interactive	67.50	93.00	54.33	64.22	–
Compact Symbolic-Aided CoT (ours)*	Non-Interactive	55.17	73.60	45.00	52.94	82.94
+AAI (ours)*	Non-Interactive	56.00	72.80	46.00	53.92	83.02
<i>Phi-4 (14B)</i>						
Symbolic-Aided CoT (Nguyen et al., 2025)*	Non-Interactive	76.33	99.40	91.67	70.59	–
+AAI (ours)*	Non-Interactive	76.33	99.00	92.33	72.06	–
Compact Symbolic-Aided CoT (ours)*	Non-Interactive	66.00	97.00	79.33	64.21	87.41
+AAI (ours)*	Non-Interactive	67.00	97.80	80.00	64.71	87.57

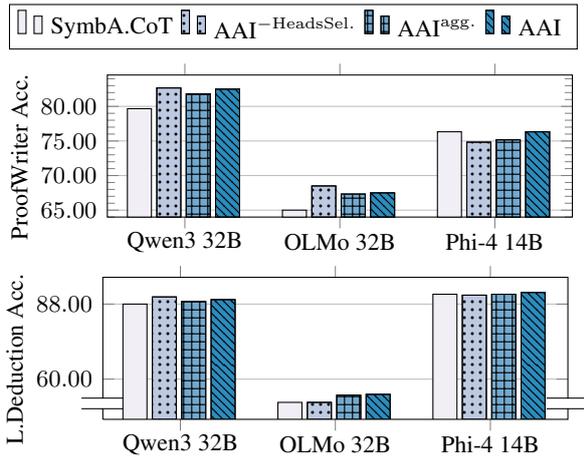


Figure 3: Ablation study on our AAI method.

tends to select aggregation heads. Meanwhile, AAI^{-HeadsSel} applies attention reweighting to all heads without selection. Notably, in the case of the Qwen-3 and OLMo 32B model, our methods attained consistently higher performance across all three configurations, highlighting the effectiveness of our attention reweighting mechanism in embedding prior knowledge within LLMs.

We found that using copy heads and anchor heads leads to more consistent performance improvements over the baseline model across various LLMs and datasets. In the AAI^{agg} variant, the proportion of selected heads typically ranges from 30-40% of all attention heads, whereas the standard AAI selects 7-20%. Because both AAI^{-HeadsSel}

and AAI^{agg} affect a substantially larger portion of aggregation-related attention heads, they are more likely to alter the LLM’s decoding states. This broader intervention can yield noticeable performance gains (e.g., the 8–14B region in Fig. 4), but it also entails a higher risk of degrading the model’s overall behavior. In contrast, our main AAI mechanism targets anchor or copy heads, which are responsible for storing information, as the attention reweighting strategy is designed to relocate referencing information. This targeted intervention offers a more controlled and interpretable adjustment, whereas reweighting a large fraction of aggregation attention heads may introduce instability.

Hyperparameter sensitivity analysis. To assess the robustness of our AAI method with respect to configuration hyperparameters, we conducted extensive experiments on the LogicalDeduction and ProofWriter datasets using two LLMs, Qwen3-32B and Phi-4 (Fig. 5). The evaluated hyperparameters include the binarization threshold, diagonal criterion, reweighting coefficient, and bias values.

The results demonstrate that our AAI method is robust to variations in configuration hyperparameters. In addition, both Qwen3-32B and Phi-4 show consistent performance improvements on the LogicalDeduction dataset compared to the baseline. In the rightmost column of the figure, by setting the *coef* value to 0, a constant term (*bias*), which is not adaptive to the original attention val-

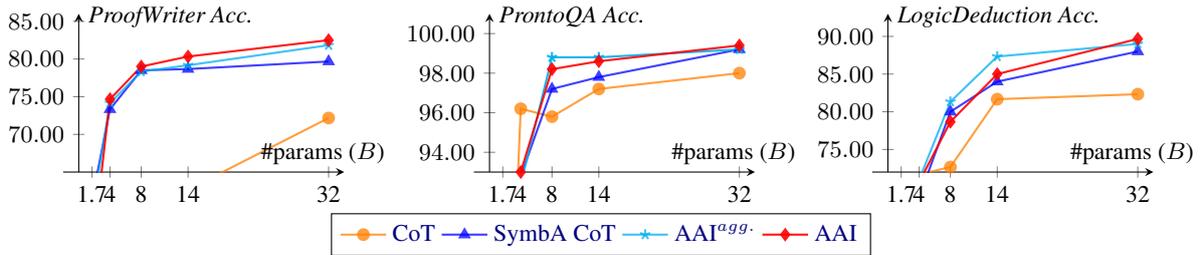


Figure 4: Performance across different model sizes of Qwen-3 with three prompting techniques on three datasets.

ues, is added to the attention matrix. As the magnitude of this constant increases, the performance of the LLMs can be adversely affected, particularly on datasets with strong rule dependencies such as ProofWriter (the red lines of the right most column). For experiments with the Phi-4 14B model on the ProofWriter dataset (the bottom row), only a small difference is observed compared with the baseline model. Our observations indicate that a large portion of the errors stem from incorrect rule selection or mismatches between the question and the premises in the KB set during the final answer step (Nguyen et al., 2025). Consequently, overall performance improvements remain limited, despite our AAI method supporting more faithful reasoning through improved rule representations. We leave addressing these issues to future work.

Semantic representation. Fig. 6 presents a t-SNE visualization of the last-layer hidden states of the final token in the input prompt, obtained using the Qwen3-8B model on the ProofWriter dataset, comparing standard CoT, Symbolic-Aided CoT, and our proposed AAI method. We observe that both Symbolic-Aided CoT and AAI exhibit similar distribution patterns, segmenting the dataset into finer-grained sub-clusters compared to standard CoT. Moreover, misclassified samples (red points) tend to appear in regions where clusters overlap. These results indicate that both Symbolic-Aided CoT and AAI effectively incorporate structural information into their representations. When comparing Symbolic-Aided CoT with our AAI method, AAI produces more distinct and compact clusters, as reflected by the wider range of values along both the x- and y-axes and the denser grouping of points within each cluster.

Attention head pattern. In Fig. 11 Appendix D, we present the attention head pattern scores for all heads across 36 layers in the Qwen3-8B LLM. Each attention head is evaluated using three metrics: di-

agonal, vertical, and horizontal scores. The red zoomed-in region on the left highlights a head with a high diagonal score. The top-right blue zoomed-in region illustrates heads exhibiting high values across all three scores. In contrast, the bottom blue zoomed-in region shows heads characterized by a high vertical score but relatively low horizontal and diagonal scores. The results show that a large proportion of heads in this model exhibit a high rate of vertical patterns. In contrast, the lower layers typically show high scores across all three pattern types—vertical, horizontal, and diagonal—indicating that information in these layers is beginning to be integrated and features are being extracted based on more complex dependencies.

5 Related Work

Logical reasoning. Existing research on logical reasoning with LLMs generally follows two directions. The first leverages external symbolic solvers, where LLMs translate natural language into formal logic and delegate inference to symbolic reasoners (Pan et al., 2023; Ye et al., 2023; Olausson et al., 2023; Gao et al., 2023b; Yao et al., 2023; Xu et al., 2024a). In contrast, our approach embeds structure directly into the reasoning process, enabling the LLM itself to act as a symbolic reasoner without relying on external solvers. The second direction focuses on interactive prompting frameworks, such as least-to-most prompting, which decompose complex reasoning into subtasks (e.g., rule selection, premise derivation, and scoring) to improve accuracy and faithfulness (Zhou et al., 2023; Zhang et al., 2025; Kazemi et al., 2023; Feng et al., 2024; Sun et al., 2024; Lee and Hwang, 2025). Unlike these methods, our approach is non-interactive, producing the final answer in a single forward pass through the LLM. Compared to Nguyen et al. (2025), we further enhance the explainability of Symbolic-aided CoT prompting and introduce an Attention-Aware Intervention mechanism to inject

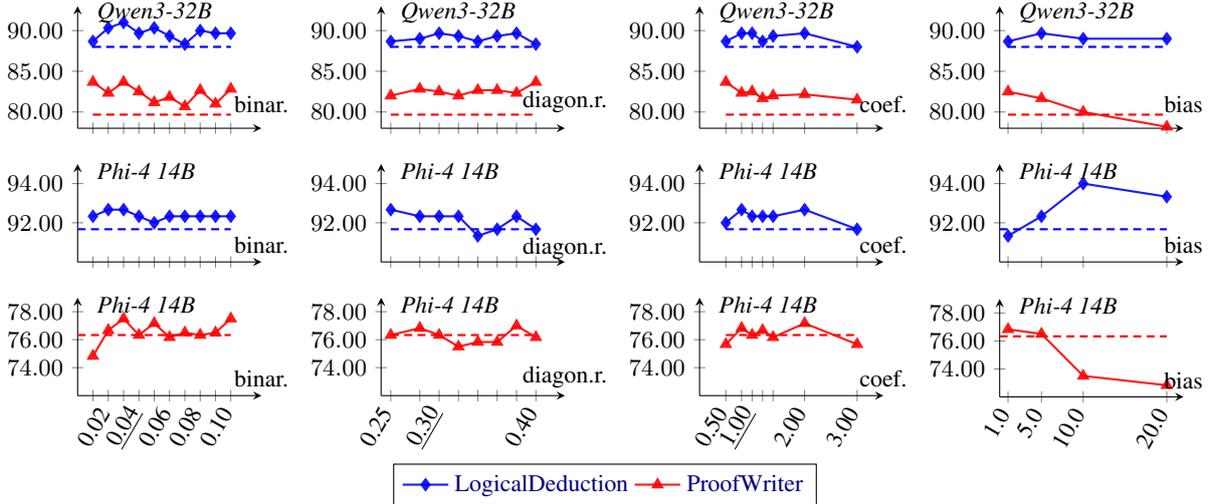


Figure 5: Performance (accuracy) changes with respect to different hyperparameter values (attention binarization threshold, $s^{diagonal}$, reweighting coefficient (c), and bias constant (b) in Eq. 10) on two benchmark datasets, Logical Deduction and ProofWriter. The dashed line denotes the baseline system, which does not use our AAI mechanism. Values marked with underscores indicate the default settings used in Table 2 ($binar. = 0.04$, $diagon.r. = 0.3$, $coef. = 1.0$, $bias = 0$). In the last column, the coefficient is set to $coef. = 0$ and the bias value is varied.

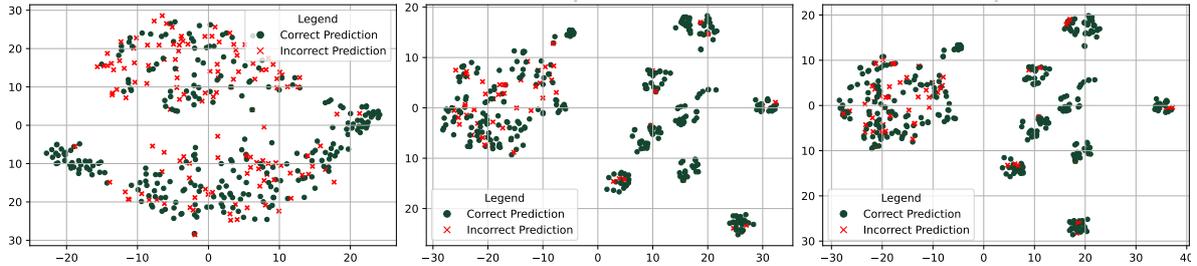


Figure 6: Visualization of the last hidden states of a sample encoded by Qwen3-8B LLM from the ProofWriter dataset under three reasoning settings: standard CoT, Symbolic-Aided CoT, and our proposed Attention-Aware Intervention (AAI), shown from left to right, respectively.

prior knowledge into LLMs.

Attention analysis. Recent work has examined the interpretability of Transformer attention heads by analyzing their functional roles and activation patterns. Prior studies have identified causally relevant or reasoning-specialized heads through interventions (Nam et al., 2025; Park et al., 2025). Elhelo and Geva (2025) and Zhao et al. (2025) examined how attention parameters or activation stability relate to task generalization. Bogdan et al. (2025) explore sentences as thought anchors and analyze patterns of attention heads with high relevance. Other approaches steer attention to emphasize user-specified instruction components (Zhang et al., 2024; Venkateswaran and Contractor, 2025). In contrast, our work investigates attention behavior under structured reasoning prompts with explicit symbolic operators. We introduce a fine-grained taxonomy of anchor, aggregation, and copy

heads, and propose a dynamic attention reweighting mechanism to inject prior knowledge, bridging attention-level interpretability with controllable symbolic reasoning in non-interactive settings.

6 Conclusion

We presented Attention-Aware Intervention, a non-interactive and end-to-end framework for enhancing logical reasoning with LLMs. By incorporating symbolic structural information into the few-shot prompt, we revealed that special attention heads exhibit activation patterns aligned with logical reasoning operators. Leveraging this insight, AAI modulates attention scores across these heads to guide reasoning without relying on external components or complex multi-step interactions. Our findings shed light on the development of scalable, interpretable reasoning systems with LLMs.

Limitations

While our proposed Attention-Aware Intervention (AAI) framework demonstrates strong performance and improved interpretability across multiple logical reasoning benchmarks, it also has several limitations. First, our method has been primarily evaluated on synthetic and small-to-medium-scale datasets, which may not fully capture the complexity of real-world reasoning scenarios. Second, although AAI effectively injects symbolic structure into LLMs, the intervention depends on model-specific attention patterns, which may limit its generalizability to architectures that do not employ the self-attention mechanism. Finally, our current work focuses exclusively on logical reasoning tasks; extending AAI to other domains, such as common-sense reasoning, remains an important direction for future research.

Acknowledgments

This work is partially supported by the Nakajima Foundation and JSPS KAKENHI Grant Number 22H00524. We used ABCI 3.0 provided by AIST and AIST Solutions with support from “ABCI 3.0 Development Acceleration Use”.

References

- Marah Abidin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. [Phi-4 technical report](#). *Preprint*, arXiv:2412.08905.
- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. [Gpt-4 technical report](#). *arXiv preprint arXiv:2303.08774*.
- Paul C. Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. 2025. [Thought anchors: Which llm reasoning steps matter?](#) *Preprint*, arXiv:2506.19143.
- Emanuele Bugliarello and Naoaki Okazaki. 2020. [Enhancing machine translation with dependency-aware self-attention](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1618–1627, Online. Association for Computational Linguistics.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *CoRR*, abs/2110.14168.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 82 others. 2024. [The llama 3 herd of models](#). *CoRR*, abs/2407.21783.
- Amit Elhelo and Mor Geva. 2025. [Inferring functionality of attention heads from their parameters](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17701–17733, Vienna, Austria. Association for Computational Linguistics.
- Jiazhan Feng, Ruochen Xu, Junheng Hao, Hiteshi Sharma, Yelong Shen, Dongyan Zhao, and Weizhu Chen. 2024. [Language models can be deductive solvers](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4026–4042, Mexico City, Mexico. Association for Computational Linguistics.
- Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023a. [Pal: program-aided language models](#). In *Proceedings of the 40th International Conference on Machine Learning, ICML’23*. JMLR.org.
- Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. [PAL: Program-aided language models](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10764–10799. PMLR.
- Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, and 24 others. 2024. [Olmo: Accelerating the science of language models](#). *Preprint*, arXiv:2402.00838.
- Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenyong 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. 2024. [FOLIO: Natural language reasoning with first-order logic](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22017–22031, Miami, Florida, USA. Association for Computational Linguistics.
- Mehran Kazemi, Najoung Kim, Deepti Bhatia, Xin Xu, and Deepak Ramachandran. 2023. [LAMBADA](#):

- Backward chaining for automated reasoning in natural language. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6547–6568, Toronto, Canada. Association for Computational Linguistics.
- Jinu Lee and Wonseok Hwang. 2025. **SymBa: Symbolic backward chaining for structured natural language reasoning**. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2468–2484, Albuquerque, New Mexico. Association for Computational Linguistics.
- Yuchen Liu, Jinming Zhao, Jingwen Hu, Ruichen Li, and Qin Jin. 2022. **DialogueEIN: Emotion interaction network for dialogue affective analysis**. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 684–693, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Andrew Nam, Henry Conklin, Yukang Yang, Thomas Griffiths, Jonathan Cohen, and Sarah-Jane Leslie. 2025. **Causal head gating: A framework for interpreting roles of attention heads in transformers**. *Preprint*, arXiv:2505.13737.
- Phuong Minh Nguyen, Tien Dang, and Naoya Inoue. 2025. **Non-interactive symbolic-aided chain-of-thought for logical reasoning**. In *Proceedings of the 39th Pacific Asia Conference on Language, Information and Computation*, pages 332–343, Hanoi, Vietnam. Association for Computational Linguistics.
- 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.
- 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.
- Yein Park, Minbyul Jeong, and Jaewoo Kang. 2025. **Thinking sparks!: Emergent attention heads in reasoning models during post training**. *Preprint*, arXiv:2509.25758.
- Chengwen Qi, Ren Ma, Bowen Li, He Du, Binyuan Hui, Jinwang Wu, Yuanjun Laili, and Conghui He. 2025. **Large language models meet symbolic provers for logical reasoning evaluation**. In *The Thirteenth International Conference on Learning Representations*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.
- Abulhair Saparov and He He. 2023. **Language models are greedy reasoners: A systematic formal analysis of chain-of-thought**. In *The Eleventh International Conference on Learning Representations*.
- Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, and 431 others. 2023. **Beyond the imitation game: Quantifying and extrapolating the capabilities of language models**. *Transactions on Machine Learning Research*. Featured Certification.
- Hongda Sun, Weikai Xu, Wei Liu, Jian Luan, Bin Wang, Shuo Shang, Ji-Rong Wen, and Rui Yan. 2024. **De-termLR: Augmenting LLM-based logical reasoning from indeterminacy to determinacy**. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9828–9862, Bangkok, Thailand. Association for Computational Linguistics.
- Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. **ProofWriter: Generating implications, proofs, and abductive statements over natural language**. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634, Online. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. **Attention is all you need**. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Praveen Venkateswaran and Danish Contractor. 2025. **Spotlight your instructions: Instruction-following with dynamic attention steering**. *Preprint*, arXiv:2505.12025.
- Yaoshian Wang, Hung-Yi Lee, and Yun-Nung Chen. 2019. **Tree transformer: Integrating tree structures into self-attention**. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1061–1070, Hong Kong, China. Association for Computational Linguistics.
- Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart Van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. **Towards ai-complete question answering: A set of prerequisite toy tasks**. *arXiv preprint arXiv:1502.05698*.
- Fangzhi Xu, Zhiyong Wu, Qiushi Sun, Siyu Ren, Fei Yuan, Shuai Yuan, Qika Lin, Yu Qiao, and Jun Liu.

- 2024a. [Symbol-LLM: Towards foundational symbol-centric interface for large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13091–13116, Bangkok, Thailand. Association for Computational Linguistics.
- Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. 2024b. [Faithful logical reasoning via symbolic chain-of-thought](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13326–13365, 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, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Baosong Yang, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. 2018. [Modeling localness for self-attention networks](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4449–4458, Brussels, Belgium. 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*.
- Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. 2023. [SatLM: Satisfiability-aided language models using declarative prompting](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Qingru Zhang, Chandan Singh, Liyuan Liu, Xiaodong Liu, Bin Yu, Jianfeng Gao, and Tuo Zhao. 2024. [Tell your model where to attend: Post-hoc attention steering for LLMs](#). In *The Twelfth International Conference on Learning Representations*.
- Yifan Zhang, Jingqin Yang, Yang Yuan, and Andrew C Yao. 2025. [Cumulative reasoning with large language models](#). *Transactions on Machine Learning Research*.
- Weixiang Zhao, Yanyan Zhao, and Xin Lu. 2022. [Cauain: Causal aware interaction network for emotion recognition in conversations](#). In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4524–4530. International Joint Conferences on Artificial Intelligence Organization. Main Track.
- Yang Zhao, Li Du, Xiao Ding, Kai Xiong, Ting Liu, and Bing Qin. 2025. [Analyzing the rapid generalization of SFT via the perspective of attention head activation patterns](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16980–16992, Vienna, Austria. Association for Computational Linguistics.
- Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. [Least-to-most prompting enables complex reasoning in large language models](#). In *The Eleventh International Conference on Learning Representations*.

A AI Usage Declaration

AI tools were used for grammar checking and formatting of tables and figures, and polish writing. All technical content and implementations were written by the authors.

B Attention Head Observation

In the self-attention layer of the Transformer architecture, for a token at position t in a sentence of length L , we have the corresponding query ($q_t \in \mathbb{R}^{1 \times d_k}$), key ($k_t \in \mathbb{R}^{1 \times d_k}$) and value ($v_t \in \mathbb{R}^{1 \times d_v}$) representations. Intuitively, the representation of the token at step t (h_t) is computed from the value vectors of previous tokens ($v_{\leq t} \in \mathbb{R}^{t \times d_v}$), weighted by their similarity to the current token as determined by the softmax function:

$$h_t = \text{softmax} \left(\frac{q_t k_{\leq t}^\top}{\sqrt{d_k}} \right) v_{\leq t} \quad (12)$$

Fig. 7 (full-scale version) visualizes the attention heads of the Qwen3-8B model on a sample from the ProofWriter dataset. Higher attention scores are indicated by darker colors. Based on this mechanism, we identify three characteristic attention patterns: (**Anchor Head**): When a token at position t attends almost exclusively to itself (close to 100%), only the main diagonal elements of the attention map are activated. This pattern is observed in token–attribute representations in Head28 (Fig.2), for example, “blue” and “rough”. (**Aggregation Head**): When a sequence of tokens attends strongly to a specific earlier token in order to incorporate information introduced by that token, an aggregation pattern emerges. For example, after the token “round” is generated, this information propagates to many subsequent tokens (Head8, Fig.2). In the attention map, this behavior appears as a *vertical line*. Conversely, when a query token needs to gather information from multiple previous tokens, the pattern appears as *horizontal lines*. (**Copy Head**): A diagonal pattern emerges when a span of generated tokens is copied forward. Specifically, if a span of three tokens starting at position k is copied to position h ($h > k$), the attention scores are high

at the token pairs $(k \rightarrow h)$, $(k + 1 \rightarrow h + 1)$, $(k + 2 \rightarrow h + 2)$. This pattern is observed in Head22 (Fig.2), where the span “Gary is round” is copied from its original generation location to the *KB variable set* region in Symbolic-Aided CoT prompting.

C Improving examples

In this section, we present several representative cases in which our AAI mechanism improves the baseline system on the ProofWriter dataset (Figures 8 and 9) and the ProntoQA dataset (Figure 10). The raw prompting outputs shown in Table 3 are aligned with the corresponding proof tree in Figure 8. The improvements achieved by our AAI mechanism typically arise from correcting incorrect rule selection and improper rule application during the reasoning process.

Table 3: Our Symbolic-Aided CoT + AAI prompting output for an example from the ProofWriter dataset (ID: ProofWriter_AttNoneg-OWA-D5-1189_Q7).

```

...
# (Rule6): Harry is kind.
# (Rule7): Harry is smart.
# (Rule8): If someone is kind and big then they are nice.
# (Rule9): If Gary is quiet and Gary is big then Gary is smart.
# (Rule10): If someone is cold then they are green.
# (Rule11): If Harry is smart and Harry is cold then Harry is green.
# (Rule12): Quiet people are big.
# (Rule13): If someone is nice and big then they are cold.
# (Rule14): All nice, quiet people are cold.
# (Rule15): If Gary is nice then Gary is kind.
# (Rule16): Smart people are quiet.
# (Question): Based on the above information, is the following
statement true, false, or unknown? Harry is nice.
# (Answer): Start from the object and their condition mentioned in
the question to collect relevant facts: Harry, is nice
# KB = {}
=> Rule6 = 'Harry is kind'
=> Rule7 = 'Harry is smart'
# KB = {Harry is kind, Harry is smart}
=> F(KB['Harry is smart'], Rule16) => 'Harry is quiet'
# KB = {Harry is kind, Harry is smart, Harry is quiet}
=> F(KB['Harry is quiet'], Rule12) => 'Harry is big'
# KB = {Harry is kind, Harry is smart, Harry is quiet, Harry is big}
=> F(KB['Harry is kind', 'Harry is big'], Rule8) => 'Harry is nice'
# KB = {Harry is kind, Harry is smart, Harry is quiet, Harry is big,
Harry is nice}
# valid the question with current infered premises
=> Validate(Question='Harry is nice', KB('Harry is nice')) = True.

```

D Attention Head Pattern

In Fig. 11, we present the attention head pattern scores for all heads across 36 layers (each consisting of 32 heads) in the Qwen3-8B LLM. Each attention head is evaluated using three metrics: diagonal, vertical, and horizontal scores. The red zoomed-in region on the left highlights a head with a high diagonal score. The top-right blue zoomed-in region illustrates heads exhibiting high values across all

three scores—diagonal, vertical, and horizontal. In contrast, the bottom blue zoomed-in region shows heads characterized by a high vertical score but relatively low horizontal and diagonal scores.

E Prompting Template

The contents of *Symbolic-Aided CoT* prompting (Nguyen et al., 2025) and the *Compact Symbolic-Aided CoT* for the four datasets – ProofWriter, LogicalDeduction, ProntoQA, FOLIO, and GSM8k – are presented in Tables 4, 5, 6, 7, and Tables 8, 9, 10, 11, 12, respectively.

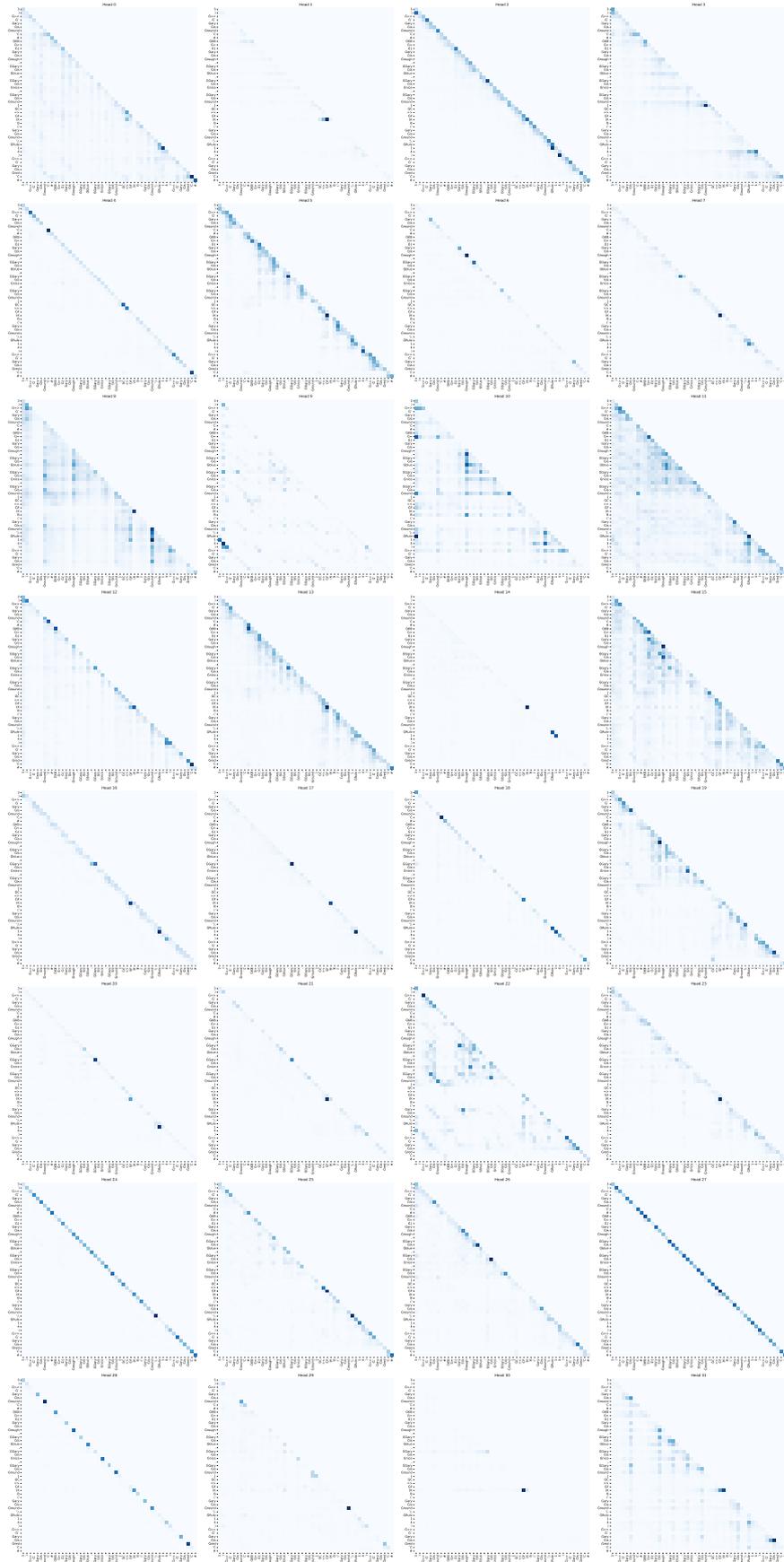


Figure 7: Visualization of attention heads in the Qwen3-8B model on an example from the ProofWriter dataset.

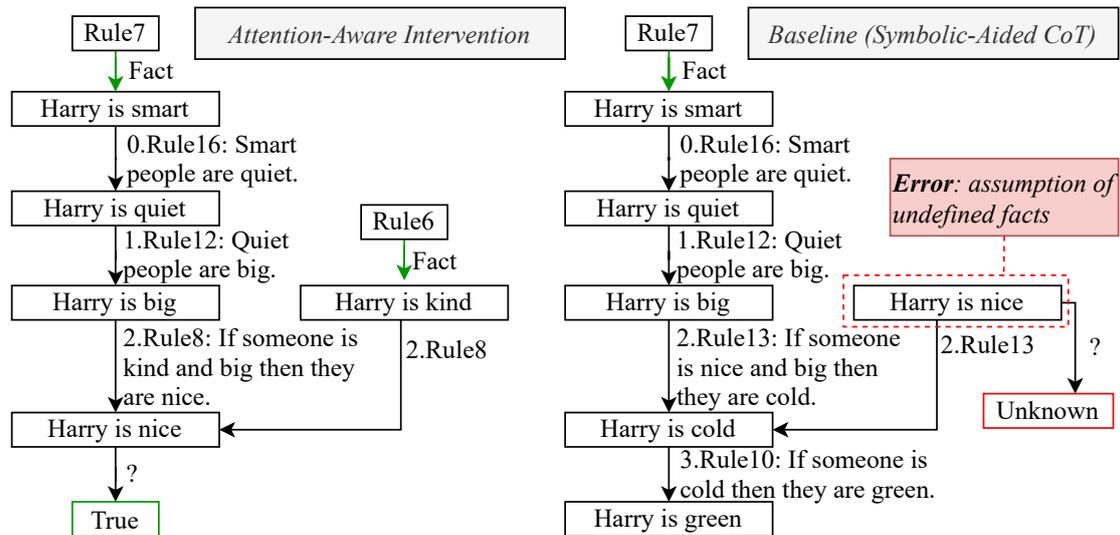


Figure 8: An example illustrating the improvement achieved by our AAI (left) compared to the baseline method (right) on the ProofWriter dataset.

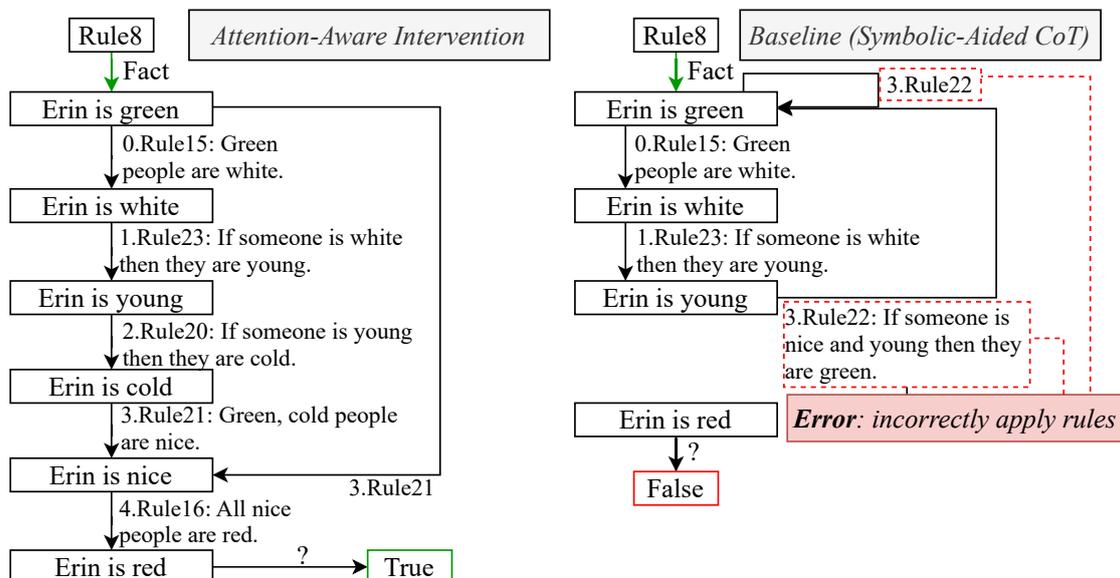


Figure 9: An example illustrating the improvement achieved by our AAI (left) compared to the baseline method (right) on the ProofWriter dataset.

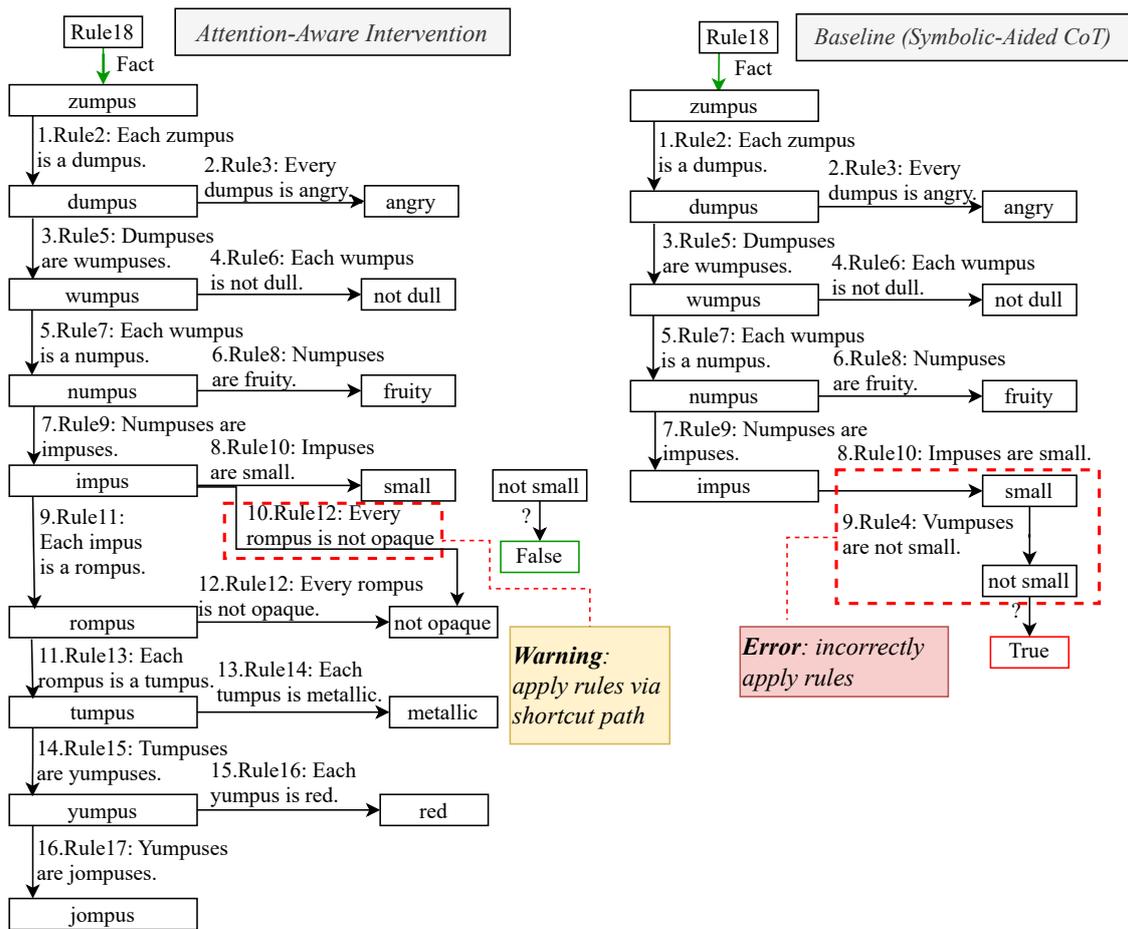


Figure 10: An example illustrating the improvement achieved by our AAI (left) compared to the baseline method (right) on the ProntoQA dataset.

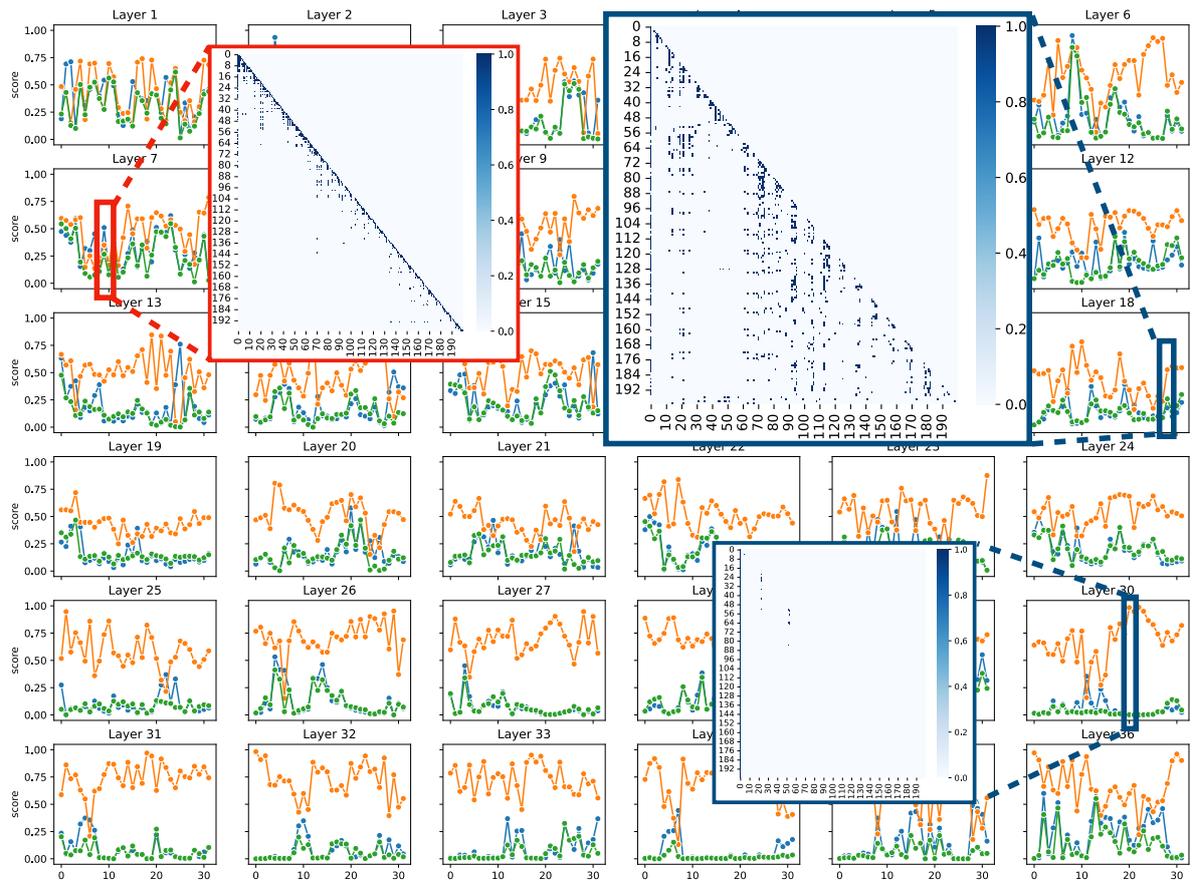


Figure 11: Visualization of the distribution of Attention Head Patterns in the Qwen3-8B LLM. The blue, orange, and green lines respectively represent the three scores—diagonal, vertical, and horizontal—computed by the pattern function defined in Eq. 7.

Table 4: *Symbolic-Aided CoT* prompting for ProofWriter dataset.

```

### Let us define F as a function that infers new premises based on a given list of facts and rules. Using these facts and rules, provide a reasoning path that leads to one of the values of a Validate function: True, False, or Uncertain.

### Example1: Given list of facts and rules:
# (Rule1): The cow is blue.
# (Rule2): The cow is round.
# (Rule3): The cow likes the lion.
# (Rule4): The cow visits the tiger.
# (Rule5): The lion is cold.
# (Rule6): The lion is nice.
# (Rule7): The lion likes the squirrel.
# (Rule8): The squirrel is round.
# (Rule9): The squirrel sees the lion.
# (Rule10): The squirrel visits the cow.
# (Rule11): The tiger likes the cow.
# (Rule12): The tiger likes the squirrel.
# (Rule13): If something is cold then it visits the tiger.
# (Rule14): If something visits the tiger then it is nice.
# (Rule15): If something sees the tiger and it is young then it is blue.
# (Rule16): If something is nice then it sees the tiger.
# (Rule17): If something likes the squirrel and it likes the cow then it visits the tiger.
# (Rule18): If something is nice and it sees the tiger then it is young.
# (Rule19): If the cow is cold and the cow visits the lion then the lion sees the squirrel.
# (Question): Based on the above information, is the following statement true, false, or uncertain? The tiger is not young.
# (Answer): Start from the object and their condition mentioned in the question to collect relevant facts: The tiger
# KB = {}
=> Rule11 = 'The tiger likes the cow'
=> Rule12 = 'The tiger likes the squirrel'
# KB = {The tiger likes the cow, The tiger likes the squirrel}
=> F(KB('The tiger likes the cow', 'The tiger likes the squirrel'), Rule17) => 'The tiger visits the tiger'
# KB = {The tiger likes the cow, The tiger likes the squirrel, The tiger visits the tiger}
=> F(KB('The tiger visits the tiger'), Rule14) => 'The tiger is nice'
# KB = {The tiger likes the cow, The tiger likes the squirrel, The tiger visits the tiger, The tiger is nice}
=> F(KB('The tiger is nice'), Rule16) => 'The tiger sees the tiger'
# KB = {The tiger likes the cow, The tiger likes the squirrel, The tiger visits the tiger, The tiger is nice, The tiger sees the tiger}
=> F(KB('The tiger is nice', 'The tiger sees the tiger'), Rule18) => 'The tiger is young'
# KB = {The tiger likes the cow, The tiger likes the squirrel, The tiger visits the tiger, The tiger is nice, The tiger sees the tiger, The tiger is young}
# valid the question with current inferred premises
=> Validate(Question='The tiger is not young', KB('The tiger is young')) = False.

### Example2: Given list of facts and rules:
# (Rule1): The dog sees the rabbit.
# (Rule2): The dog sees the squirrel.
# (Rule3): The dog sees the tiger.
# (Rule4): The rabbit eats the dog.
# (Rule5): The rabbit does not eat the tiger.
# (Rule6): The rabbit does not like the tiger.
# (Rule7): The squirrel does not see the rabbit.
# (Rule8): The tiger does not eat the rabbit.
# (Rule9): The tiger is not kind.
# (Rule10): The tiger likes the dog.
# (Rule11): The tiger sees the dog.
# (Rule12): If something is cold then it likes the rabbit.
# (Rule13): If something eats the tiger and it is nice then it likes the rabbit.
# (Rule14): If something likes the squirrel then the squirrel likes the rabbit.
# (Rule15): If something likes the rabbit and the rabbit is kind then it sees the tiger.
# (Rule16): If something likes the tiger then the tiger is young.
# (Rule17): If something is young and it eats the rabbit then it likes the tiger.
# (Rule18): If something sees the rabbit then the rabbit is cold.
# (Rule19): If something likes the rabbit then it likes the squirrel.
# (Rule20): If something likes the squirrel then the squirrel is cold.
# (Question): Based on the above information, is the following statement true, false, or uncertain? The rabbit is cold.
# (Answer): Start from the object and their condition mentioned in the question to collect relevant facts: The rabbit, is cold
# KB = {}
=> Rule4 = 'The rabbit eats the dog'
=> Rule5 = 'The rabbit does not eat the tiger'
=> Rule6 = 'The rabbit does not like the tiger'
# KB = {The rabbit eats the dog, The rabbit does not eat the tiger, The rabbit does not like the tiger}
=> F(KB(Rule1), Rule18) => 'The rabbit is cold'
# KB = {eats the dog, not eat the tiger, not like the tiger, The rabbit is cold}
# valid the question with current inferred premises
=> Validate(Question='The rabbit is cold', KB('The rabbit is cold')) = True.

### Example3: Given list of facts and rules:
{{RULE CONTENT}}
# (Question): {{QUESTION}}
# (Answer):

```

Table 5: *Symbolic-Aided CoT* prompting for LogicalDeduction dataset.

```

### Let us define F as a function that infers new premises based on a given list of facts and rules. Using these facts and rules, provide a reasoning path that leads to the answer of the question.

### Example1: Given list of rules:
# (Rule1): The following paragraphs each describe a set of seven objects arranged in a fixed order.
# (Rule2): The statements are logically consistent within each paragraph.
# (Rule3): On a branch, there are seven birds: a falcon, an owl, a quail, a raven, a hummingbird, a blue jay, and a crow.
# (Rule4): The quail is to the left of the owl.
# (Rule5): The owl is the second from the left.
# (Rule6): The crow is to the right of the falcon.
# (Rule7): The blue jay is to the right of the hummingbird.
# (Rule8): The hummingbird is to the right of the raven.
# (Rule9): The blue jay is to the left of the falcon.
# (Question): Which of the following is true?
# (Options):
A) The falcon is the second from the left.
B) The owl is the second from the left.
C) The quail is the second from the left.
D) The raven is the second from the left.
E) The hummingbird is the second from the left.
F) The blue jay is the second from the left.
G) The crow is the second from the left.
# (Answer): Since the options ask which object is the second from the left, the objects are ordered from the left:
# KB = []
=> F('ordered from the left',Rule1) => KB = [1., 2., 3., 4., 5., 6., 7.]
=> F('ordered from the left',Rule5) => absolute position of second from the left = 2 => KB = [1., 2.owl, 3., 4., 5., 6., 7.]
=> F('ordered from the left',Rule4) => Draft relative position: quail ->owl
=> F('ordered from the left',Rule8) => Draft relative position: raven ->hummingbird => Combine bridge relative: raven ->hummingbird; quail ->owl
=> F('ordered from the left',Rule7) => Draft relative position: hummingbird ->blue jay => Combine bridge relative: raven ->hummingbird ->blue jay; quail ->owl
=> F('ordered from the left',Rule9) => Draft relative position: blue jay ->falcon
=> Combine bridge relative: raven ->hummingbird ->blue jay ->falcon; quail ->owl
=> F('ordered from the left',Rule6) => Draft relative position: falcon ->crow
=> Combine bridge relative: raven ->hummingbird ->blue jay ->falcon ->crow; quail ->owl
=> Combine (KB=[1., 2.owl, 3., 4., 5., 6., 7.], Draft) => with respect to absolute positions in KB: [1.quail, 2.owl, 3.raven, 4.hummingbird, 5.blue jay, 6.falcon, 7.crow]
# Since the options ask which object is the second from the left => absolute position is 2, the correct answer is 'owl'. Check all the above options => answer: B

### Example2: Given list of rules:
# (Rule1): The following paragraphs each describe a set of five objects arranged in a fixed order.
# (Rule2): The statements are logically consistent within each paragraph.
# (Rule3): A fruit stand sells five fruits: watermelons, apples, kiwis, cantaloupes, and mangoes.
# (Rule4): The cantaloupes are the second-cheapest.
# (Rule5): The mangoes are more expensive than the watermelons.
# (Rule6): The kiwis are less expensive than the apples.
# (Rule7): The kiwis are more expensive than the mangoes.
# (Question): Which of the following is true?
# (Options):
A) The watermelons are the cheapest.
B) The apples are the cheapest.
C) The kiwis are the cheapest.
D) The cantaloupes are the cheapest.
E) The mangoes are the cheapest.
# (Answer): Since the options ask which fruit is the cheapest, the objects are ordered from cheapest:
# KB = []
=> F('ordered from cheapest',Rule1) => KB = [1., 2., 3., 4., 5.]
=> F('ordered from cheapest',Rule4) => absolute position of second-cheapest = 2 => KB = [1., 2.cantaloupes, 3., 4., 5.]
=> F('ordered from cheapest',Rule5) => Draft relative position: watermelons ->mangoes
=> F('ordered from cheapest',Rule7) => Draft relative position: mangoes ->kiwis => Combine bridge relative: watermelons ->mangoes ->kiwis ->apples
=> F('ordered from cheapest',Rule6) => Draft relative position: kiwis ->apples => Combine bridge relative: watermelons ->mangoes ->kiwis ->apples
=> Combine (KB=[1., 2.cantaloupes, 3., 4., 5.], Draft) => with respect to absolute positions in KB: [1.watermelons, 2.cantaloupes, 3.mangoes, 4.kiwis, 5.apples]
# Since the options ask which fruit is the cheapest => absolute position is 1, the correct answer is 'watermelons'. Check all the above options => answer: A

### Example3: Given list of rules:
{{RULE CONTENT}}
# (Question): {{QUESTION}}
# (Options): {{OPTIONS}}
# (Answer): Since the options ask

```

Table 6: *Symbolic-Aided CoT* prompting for ProntoQA dataset.

```

### Let us define F as a function that determines whether a statement is True or False, based on
a given list of facts and rules. Using these facts and rules, provide a reasoning path that leads to
one of the values: 'True' or 'False'.

### Example1: Given list of facts and rules:
# (Rule1): If K then P;
# (Rule2): Q is true;
# (Rule3): If Q then K.
# (Question): truth value of P?
# (Answer): Start from the object mentioned in the question: P
# KB = {Q}
=> F(KB['Q'], Rule3) => 'K'
# KB = {Q, K}
=> F(KB['K'], Rule1) => 'P'
# KB = {Q, K, P}
=> Validate(KB, Question='P') = True.

### Example2: Given list of facts and rules:
# (Rule1): Jompuses are not shy.
# (Rule2): Jompuses are yumpuses.
# (Rule3): Zumpuses are rompuses.
# (Rule4): Each yumpus is a dumpus.
# (Rule5): Dumpuses are not wooden.
# (Rule6): Dumpuses are wumpuses.
# (Rule7): Wumpuses are red.
# (Rule8): Every wumpus is an impus.
# (Rule9): Each impus is opaque.
# (Rule10): Impuses are tumpuses.
# (Rule11): Numpuses are sour.
# (Rule12): Tumpuses are not sour.
# (Rule13): Tumpuses are vumpuses.
# (Rule14): Vumpuses are earthy.
# (Rule15): Every vumpus is a zumpus.
# (Rule16): Zumpuses are small.
# (Rule17): Each yumpus is aggressive.
# (Rule18): Max is a yumpus.
# (Question): Is the following statement true or false? Max is sour.
# (Answer): Start from the object mentioned in the question: Max
# KB = {}
=> F(KB, Rule17) => 'Max is a yumpus'
# KB = {yumpus}
=> F(KB['yumpus'], Rule17) => 'Max is aggressive'
=> F(KB['yumpus'], Rule4) => 'Max is dumpus'
# KB = {yumpus, aggressive, dumpus}
=> F(KB['dumpus'], Rule5) => 'Max is not wooden'
=> F(KB['dumpus'], Rule6) => 'Max is wumpus'
# KB = {yumpus, aggressive, dumpus, not wooden, wumpus}
=> F(KB['wumpus'], Rule7) => 'Max is an red'
=> F(KB['wumpus'], Rule8) => 'Max is an impus'
# KB = {yumpus, aggressive, dumpus, not wooden, wumpus, red, impus}
=> F(KB['impus'], Rule9) => 'Max is a opaque'
=> F(KB['impus'], Rule10) => 'Max is a tumpus'
# KB = {yumpus, aggressive, dumpus, not wooden, wumpus, red, impus, opaque, tumpus}
=> F(KB['tumpus'], Rule12) => 'Max is a not sour'
=> F(KB['tumpus'], Rule13) => 'Max is a vumpus'
# KB = {yumpus, aggressive, dumpus, not wooden, wumpus, red, impus, opaque, tumpus, not
sour, vumpus}
# valid the question with current infered premises
=> Validate(KB['not sour'], Question='Max is sour') = False.

### Example3: Given list of facts and rules:
# (Rule1): Every tumpus is not angry.
# (Rule2): Tumpuses are rompuses.
# (Rule3): Every numpus is not bright.
# (Rule4): Rompuses are not luminous.
# (Rule5): Rompuses are yumpuses.
# (Rule6): Yumpuses are transparent.
# (Rule7): Yumpuses are zumpuses.
# (Rule8): Each zumpus is not bitter.
# (Rule9): Zumpuses are impuses.
# (Rule10): Impuses are red.
# (Rule11): Each impus is a dumpus.
# (Rule12): Every dumpus is happy.
# (Rule13): Each dumpus is a vumpus.
# (Rule14): Vumpuses are bright.
# (Rule15): Every vumpus is a jompus.
# (Rule16): Jompuses are large.
# (Rule17): Each jompus is a wumpus.
# (Rule18): Stella is a yumpus.
# (Question): Is the following statement true or false? Stella is bright.
# (Answer): Start from the object mentioned in the question: Stella
# KB = {}
=> F(KB, Rule17) => 'Stella is a yumpus'
# KB = {yumpus}
=> F(KB['yumpus'], Rule6) => 'Stella is transparent'
=> F(KB['yumpus'], Rule7) => 'Stella is zumpus'
# KB = {yumpus, transparent, zumpus}
=> F(KB['zumpus'], Rule9) => 'Stella is impus'
=> F(KB['zumpus'], Rule8) => 'Stella is not bitter'
# KB = {yumpus, transparent, zumpus, impus, not bitter}
=> F(KB['impus'], Rule10) => 'Stella is red'
=> F(KB['impus'], Rule11) => 'Stella is dumpus'
# KB = {yumpus, transparent, zumpus, not bitter, impus, red, dumpus}
=> F(KB['dumpus'], Rule13) => 'Stella is a vumpus'
=> F(KB['dumpus'], Rule12) => 'Stella is happy'
# KB = {yumpus, transparent, zumpus, not bitter, impus, red, dumpus, vumpus, happy}
=> F(KB['vumpus'], Rule14) => 'Stella is bright'
# KB = {yumpus, transparent, zumpus, not bitter, impus, red, dumpus, vumpus, happy, bright}
# valid the question with current infered premises
=> Validate(KB['bright'], Question='Stella is bright') = True.

### Example4: Given list of facts and rules:
{{RULE_CONTENT}}
# (Question): {{QUESTION}}
# (Answer):

```

Table 7: Symbolic-Aided CoT prompting for FOLIO dataset.

```

#### Let us define F as a function that infers new premises based on a given list of facts and rules. Using these facts and rules, provide a reasoning path that leads to one of the values of a Validate function: True, False, or Uncertain. (Guidelines: True – If the hypothesis can be logically proven correct using only the provided facts and rules. False – If the hypothesis can be logically disproven (the facts or rules directly contradict the hypothesis). Uncertain – If there is not enough information to prove or disprove the hypothesis. Important Rules: Only use the provided facts and rules as your knowledge base (KB); Do not use outside knowledge or make assumptions not supported by the KB; Avoid circular or infinite reasoning. Each rule may only be applied once to prevent loops; If the reasoning process leads to conflicting conclusions, classify as Uncertain. Conditions in if-then rules are not confirmed facts. When rules reference vague entities (e.g., ‘an object exists’ or ‘some objects’), the outcome is often uncertain. Output Format: Return only one word: True, False, or Uncertain)
-----
#### Example1: Given list of facts and rules:
# (Rule1): The Blake McFall Company Building is a commercial warehouse listed on the National Register of Historic Places.
# (Rule2): The Blake McFall Company Building was added to the National Register of Historic Places in 1990.
# (Rule3): The Emmet Building is a five-story building in Portland, Oregon.
# (Rule4): The Emmet Building was built in 1915.
# (Rule5): The Emmet Building is another name for the Blake McFall Company Building.
# (Rule6): John works at the Emmet Building.
# (Question): Based on the above information, is the following statement true, false, or uncertain? The Blake McFall Company Building is located in Portland, Oregon.
# (Answer): Start with the object and its condition mentioned in the question to collect relevant facts. In this case, the main object is ‘Blake McFall Company Building’. To verify the statement — ‘The Blake McFall Company Building is located in Portland, Oregon’ - is True, False, or Uncertain (Contradiction or not enough information to conclusion).
=> KB = {}
# collect all the facts or assumption conditions
=> Rule1 = KB(‘The Blake McFall Company Building is a commercial warehouse listed on the National Register of Historic Places’)
=> Rule2 = KB(‘The Blake McFall Company Building was added to the National Register of Historic Places in 1990’)
=> Rule5 = KB(‘The Emmet Building is another name for the Blake McFall Company Building’)
=> F(KB(‘The Emmet Building is another name for the Blake McFall Company Building’), Rule3) => KB(‘The Blake McFall Company Building is a five-story building in Portland, Oregon’)
# valid the question with current inferred premises
=> Validate(Question=‘The Blake McFall Company Building is located in Portland, Oregon’, KB(‘The Blake McFall Company Building is a five-story building in Portland, Oregon’)) = True.
-----
#### Example2: Given list of facts and rules:
# (Rule1): People eat meat regularly or are vegetarian.
# (Rule2): If people eat meat regularly, then they enjoy eating hamburgers and steaks.
# (Rule3): All people who are vegetarian are conscious of the environment or their health.
# (Rule4): If people are conscious about the environment or their health, then they do not go to fast food places often.
# (Rule5): If people have busy schedules without time to cook, then they go to fast food places often.
# (Rule6): If Jeremy does not both go to fast food places often and is conscious about the environment or their health, then he goes to fast food places often.
# (Question): Based on the above information, is the following statement true, false, or uncertain? If Jeremy has a busy schedule without time to cook, then Jeremy does not enjoy eating hamburgers and steaks.
# (Answer): Start with the object and its condition mentioned in the question to collect relevant facts. In this case, the main object is ‘Jeremy’. To verify the assumption — if ‘Jeremy has a busy schedule without time to cook’ — determine whether the conclusion ‘Jeremy does not enjoy eating hamburgers and steaks’ is True, False, or Uncertain (Contradiction or not enough information to conclusion).
=> KB = {}
# collect all the facts or assumption conditions
=> KB (‘Jeremy has a busy schedule without time to cook’)
=> F(KB(‘Jeremy has a busy schedule without time to cook’), Rule5) => KB(‘Jeremy goes to fast food places often’)
=> F(KB(‘Jeremy goes to fast food places often’), Rule4) => KB(‘Jeremy is not conscious about the environment or his health’)
=> F(KB(‘Jeremy is not conscious about the environment or his health’), Rule3) => KB(‘Jeremy is not vegetarian’)
=> F(KB(‘Jeremy is not vegetarian’), Rule1) => KB(‘Jeremy eats meat regularly’)
=> F(KB(‘Jeremy eats meat regularly’), Rule2) => KB(‘Jeremy enjoys eating hamburgers and steaks’)
# valid the question with current inferred premises
=> Validate(Question=‘If Jeremy has a busy schedule without time to cook, then Jeremy does not enjoy eating hamburgers and steaks’, KB(‘Jeremy enjoys eating hamburgers and steaks’)) = False.
-----
#### Example3: Given list of facts and rules:
# (Rule1): Pasifika New Zealanders are a pan-ethnic group of New Zealanders.
# (Rule2): Asian New Zealanders are a pan-ethnic group of New Zealanders.
# (Rule3): Pasifika New Zealanders are not Asian New Zealanders.
# (Rule4): Pasifika New Zealanders speak Samoan.
# (Rule5): Joe is a Pasifika New Zealander.
# (Rule6): Amy speaks Samoan.
# (Question): Based on the above information, is the following statement true, false, or uncertain? Amy is an Pasifika New Zealander.
# (Answer): Start with the object and its condition mentioned in the question to collect relevant facts. In this case, the main object is ‘Amy’. To verify the statement — ‘Amy is an Pasifika New Zealander’ - is True, False, or Uncertain (Contradiction or not enough information to conclusion).
=> KB = {}
# collect all the facts or assumption conditions
=> KB (‘Amy speaks Samoan’)
=> F(KB(‘Amy speaks Samoan’), Rule4) => Uncertain(‘Amy is a Pasifika New Zealander’)
# valid the question with current inferred premises
=> Validate(Question=‘Amy is a Pasifika New Zealander’, KB) = Uncertain.
-----
#### Example4: Given list of facts and rules:
{{RULE CONTENT}}
# (Question): Based on the above information, is the following statement true, false, or uncertain? {{QUESTION}}
# (Answer):

```

Table 8: *Compact Symbolic-Aided CoT* prompting for ProofWriter dataset.

```

### Let us define F as a function that determines whether a statement is True, False, or Uncertain, based on a given list of facts and rules. Using these facts and rules, provide a reasoning path that leads to one of the values: 'True', 'False', or 'Uncertain'.

### Example1: Given list of facts and rules:
# (Rule1): If K then P;
# (Rule2): Q is true;
# (Rule3): If Q then K.
# (Question): truth value of P?
# (Answer): F(Rule2) => F(Rule3, Output1='K') => F(Output1, Rule1, Output2='P') => F(Question) = True.

### Example2: Given list of facts and rules:
# (Rule1): The cow is blue.
# (Rule2): The cow is round.
# (Rule3): The cow likes the lion.
# (Rule4): The cow visits the tiger.
# (Rule5): The lion is cold.
# (Rule6): The lion is nice.
# (Rule7): The lion likes the squirrel.
# (Rule8): The squirrel is round.
# (Rule9): The squirrel sees the lion.
# (Rule10): The squirrel visits the cow.
# (Rule11): The tiger likes the cow.
# (Rule12): The tiger likes the squirrel.
# (Rule13): If something is cold then it visits the tiger.
# (Rule14): If something visits the tiger then it is nice.
# (Rule15): If something sees the tiger and it is young then it is blue.
# (Rule16): If something is nice then it sees the tiger.
# (Rule17): If something likes the squirrel and it likes the cow then it visits the tiger.
# (Rule18): If something is nice and it sees the tiger then it is young.
# (Rule19): If the cow is cold and the cow visits the lion then the lion sees the squirrel.
# (Question): Based on the above information, is the following statement true, false, or uncertain? The tiger is not young.
# (Answer): F((Rule11, Rule12), Rule17, Output1='The tiger visits the tiger') => F(Output1, Rule14, Output2='The tiger is nice') => F(Output2, Rule18, Output='The tiger is young') => F(Question) = False.

### Example3: Given list of facts and rules:
# (Rule1): The dog sees the rabbit.
# (Rule2): The dog sees the squirrel.
# (Rule3): The dog sees the tiger.
# (Rule4): The rabbit eats the dog.
# (Rule5): The rabbit does not eat the tiger.
# (Rule6): The rabbit does not like the tiger.
# (Rule7): The squirrel does not see the rabbit.
# (Rule8): The tiger does not eat the rabbit.
# (Rule9): The tiger is not kind.
# (Rule10): The tiger likes the dog.
# (Rule11): The tiger sees the dog.
# (Rule12): If something is cold then it likes the rabbit.
# (Rule13): If something eats the tiger and it is nice then it likes the rabbit.
# (Rule14): If something likes the squirrel then the squirrel likes the rabbit.
# (Rule15): If something likes the rabbit and the rabbit is kind then it sees the tiger.
# (Rule16): If something likes the tiger then the tiger is young.
# (Rule17): If something is young and it eats the rabbit then it likes the tiger.
# (Rule18): If something sees the rabbit then the rabbit is cold.
# (Rule19): If something likes the rabbit then it likes the squirrel.
# (Rule20): If something likes the squirrel then the squirrel is cold.
# (Question): Based on the above information, is the following statement true, false, or uncertain? The rabbit is cold.
# (Answer): F(Rule1, Rule18, Output1='The rabbit is cold') => F(Question) = True.

### Example3: Given list of facts and rules:
{{RULE CONTENT}}
# (Question): Based on the above information, is the following statement true, false, or unknown? {{QUESTION}}
# (Answer):

```

Table 9: Compact Symbolic-Aided CoT prompting for LogicalDeduction dataset.

Let us define F as a function that infers new premises based on a given list of facts and rules. Using these facts and rules, provide a reasoning path that leads to the answer of the question.

Example1: Given list of rules:

- # (Rule1): The following paragraphs each describe a set of seven objects arranged in a fixed order.
- # (Rule2): The statements are logically consistent within each paragraph.
- # (Rule3): On a branch, there are seven birds: a falcon, an owl, a quail, a raven, a hummingbird, a blue jay, and a crow.
- # (Rule4): The quail is to the left of the owl.
- # (Rule5): The owl is the second from the left.
- # (Rule6): The crow is to the right of the falcon.
- # (Rule7): The blue jay is to the right of the hummingbird.
- # (Rule8): The hummingbird is to the right of the raven.
- # (Rule9): The blue jay is to the left of the falcon.
- # (Question): Which of the following is true?
- # (Options):
 - A) The falcon is the second from the left.
 - B) The owl is the second from the left.
 - C) The quail is the second from the left.
 - D) The raven is the second from the left.
 - E) The hummingbird is the second from the left.
 - F) The blue jay is the second from the left.
 - G) The crow is the second from the left.

(Answer): Since the options ask which object is the second from the left, the objects are ordered from the left:
 # => F(Rule1, Output='[1., 2., 3., 4., 5., 6., 7.]') => F(Rule5, Output='[1., 2.owl, 3., 4., 5., 6., 7.]') => F(Rule4, Output='quail -> owl') => F(Rule8, Output='raven -> hummingbird; quail -> owl') => F(Rule7, Output='raven -> hummingbird -> blue jay; quail -> owl') => F(Rule9, Output='raven -> hummingbird -> blue jay -> falcon; quail -> owl') => F(Rule6, Output='raven -> hummingbird -> blue jay -> falcon -> crow; quail -> owl') => Combination(Output='[1.quail, 2.owl, 3.raven, 4.hummingbird, 5.blue jay, 6.falcon, 7.crow]') => Validate(Question='which object is the second from the left', Output='owl') => answer: B

Example2: Given list of rules:

- # (Rule1): The following paragraphs each describe a set of five objects arranged in a fixed order.
- # (Rule2): The statements are logically consistent within each paragraph.
- # (Rule3): A fruit stand sells five fruits: watermelons, apples, kiwis, cantaloupes, and mangoes.
- # (Rule4): The cantaloupes are the second-cheapest.
- # (Rule5): The mangoes are more expensive than the watermelons.
- # (Rule6): The kiwis are less expensive than the apples.
- # (Rule7): The kiwis are more expensive than the mangoes.
- # (Question): Which of the following is true?
- # (Options):
 - A) The watermelons are the cheapest.
 - B) The apples are the cheapest.
 - C) The kiwis are the cheapest.
 - D) The cantaloupes are the cheapest.
 - E) The mangoes are the cheapest.

(Answer): Since the options ask which fruit is the cheapest, the objects are ordered from cheapest:
 # => F(Rule1, Output='[1., 2., 3., 4., 5.]') => F(Rule4, Output='[1., 2.cantaloupes, 3., 4., 5.]') => F(Rule5, Output='watermelons -> mangoes') => F(Rule7, Output='watermelons -> mangoes -> kiwis') => F(Rule6, Output='watermelons -> mangoes -> kiwis -> apples') => Combination(Output='[1.watermelons, 2.cantaloupes, 3.mangoes, 4.kiwis, 5.apples]') => Validate(Question='which fruit is the cheapest', Output='watermelons') => answer: A

Example3: Given list of rules:

- {{RULE CONTENT}}
- # (Question): {{QUESTION}}
- # (Options): {{OPTIONS}}
- # (Answer): Since the options ask

Table 10: *Compact Symbolic-Aided CoT* prompting for ProntoQA dataset.

<pre> ### Let us define F as a function that determines whether a statement is True or False, based on a given list of facts and rules. Using these facts and rules, provide a reasoning path that leads to one of the values: 'True' or 'False'. ### Example1: Given list of facts and rules: # (Rule1): If K then P; # (Rule2): Q is true; # (Rule3): If Q then K. # (Question): truth value of P? # (Answer): Start from the object mentioned in the question: P # F(Rule2) => F(Rule2, Rule3, Output1='K') => F(Output1, Rule1, Output2='P') => Validate(Output2, Question='P') = True. </pre>
<pre> ### Example2: Given list of facts and rules: # (Rule1): Jompuses are not shy. # (Rule2): Jompuses are yumpuses. # (Rule3): Each yumpus is aggressive. # (Rule4): Each yumpus is a dumpus. # (Rule5): Dumpuses are not wooden. # (Rule6): Dumpuses are wumpuses. # (Rule7): Wumpuses are red. # (Rule8): Every wumpus is an impus. # (Rule9): Each impus is opaque. # (Rule10): Impuses are tumpuses. # (Rule11): Numpuses are sour. # (Rule12): Tumpuses are not sour. # (Rule13): Tumpuses are vumpuses. # (Rule14): Vumpuses are earthy. # (Rule15): Every vumpus is a zumpus. # (Rule16): Zumpuses are small. # (Rule17): Zumpuses are rompuses. # (Rule18): Max is a yumpus. # (Question): Is the following statement true or false? Max is sour. # (Answer): Start from the object mentioned in the question: Max # F(Rule18, Rule4, Output1='Max is dumpus') => F(Output1, Rule6, Output2='Max is wumpus') => F(Output2, Rule8, Output3='Max is an impus') => F(Output3, Rule10, Output4='Max is a tumpus') => F(Output4, Rule12, Output5='Max is not sour') => Validate(Output5, Question='Max is sour') = False. </pre>
<pre> ### Example3: Given list of facts and rules: # (Rule1): Every tumpus is not angry. # (Rule2): Tumpuses are rompuses. # (Rule3): Every numpus is not bright. # (Rule4): Rompuses are not luminous. # (Rule5): Rompuses are yumpuses. # (Rule6): Yumpuses are transparent. # (Rule7): Yumpuses are zumpuses. # (Rule8): Each zumpus is not bitter. # (Rule9): Zumpuses are impuses. # (Rule10): Impuses are red. # (Rule11): Each impus is a dumpus. # (Rule12): Every dumpus is happy. # (Rule13): Each dumpus is a vumpus. # (Rule14): Vumpuses are bright. # (Rule15): Every vumpus is a jompus. # (Rule16): Jompuses are large. # (Rule17): Each jompus is a wumpus. # (Rule18): Stella is a yumpus. # (Question): Is the following statement true or false? Stella is bright. # (Answer): Start from the object mentioned in the question: Stella # F(Rule18, Rule7, Output1='Stella is a zumpus') => F(Output1, Rule9, Output2='Stella is an impus') => F(Output2, Rule11, Output3='Stella is a dumpus') => F(Output3, Rule13, Output4='Stella is a vumpus') => F(Output4, Rule14, Output5='Stella is bright') => Validate(Output5, Question='Stella is bright') = True. </pre>
<pre> ### Example3: Given list of rules: {{RULE CONTENT}} # (Question): {{QUESTION}} # (Answer): </pre>

Table 11: *Compact Symbolic-Aided CoT* prompting for FOLIO dataset.

<p>### Let us define F as a function that infers new premises based on a given list of facts and rules. Using these facts and rules, provide a reasoning path that leads to one of the values of a Validate function: True, False, or Uncertain.</p> <hr/> <p>### Example1: Given list of facts and rules: # (Rule1): The Blake McFall Company Building is a commercial warehouse listed on the National Register of Historic Places. # (Rule2): The Blake McFall Company Building was added to the National Register of Historic Places in 1990. # (Rule3): The Emmet Building is a five-story building in Portland, Oregon. # (Rule4): The Emmet Building was built in 1915. # (Rule5): The Emmet Building is another name for the Blake McFall Company Building. # (Rule6): John works at the Emmet Building. # (Question): Based on the above information, is the following statement true, false, or uncertain? The Blake McFall Company Building is located in Portland, Oregon.</p> <p># (Answer): Start with the object and its condition mentioned in the question to collect relevant facts. # => F(Rule5, Rule3, Output1='The Blake McFall Company Building is a five-story building in Portland, Oregon') => Validate(Output1, Question='The Blake McFall Company Building is located in Portland, Oregon') = True.</p> <hr/> <p>### Example2: Given list of facts and rules: # (Rule1): People eat meat regularly or are vegetarian. # (Rule2): If people eat meat regularly, then they enjoy eating hamburgers and steaks. # (Rule3): All people who are vegetarian are conscious of the environment or their health. # (Rule4): If people are conscious about the environment or their health, then they do not go to fast food places often. # (Rule5): If people have busy schedules without time to cook, then they go to fast food places often. # (Rule6): If Jeremy does not both go to fast food places often and is conscious about the environment or their health, then he goes to fast food places often. # (Question): Based on the above information, is the following statement true, false, or uncertain? If Jeremy has a busy schedule without time to cook, then Jeremy does not enjoy eating hamburgers and steaks.</p> <p># (Answer): Start with the object and its condition mentioned in the question to collect relevant facts. # => F(Assume('Jeremy has a busy schedule without time to cook'), Rule5, Output1='Jeremy goes to fast food places often') => F(Output1, Rule4, Output2='Jeremy is not conscious about the environment or his health') => F(Output2, Rule3, Output3='Jeremy is not vegetarian') => F(Output3, Rule1, Output4='Jeremy eats meat regularly') => F(Output4, Rule2, Output5='Jeremy enjoys eating hamburgers and steaks') => Validate(Output5, Question='If Jeremy has a busy schedule without time to cook, then Jeremy does not enjoy eating hamburgers and steaks') = False.</p> <hr/> <p>### Example3: Given list of facts and rules: # (Rule1): Pasifika New Zealanders are a pan-ethnic group of New Zealanders. # (Rule2): Asian New Zealanders are a pan-ethnic group of New Zealanders. # (Rule3): Pasifika New Zealanders are not Asian New Zealanders. # (Rule4): Pasifika New Zealanders speak Samoan. # (Rule5): Joe is a Pasifika New Zealander. # (Rule6): Amy speaks Samoan. # (Question): Based on the above information, is the following statement true, false, or uncertain? Amy is an Pasifika New Zealander.</p> <p># (Answer): Start with the object and its condition mentioned in the question to collect relevant facts. # => F(Rule6, Rule4, Output1=Uncertain('Amy is a Pasifika New Zealander') => Validate(Output1, Question='Amy is a Pasifika New Zealander') = Uncertain.</p> <hr/> <p>### Example4: Given list of facts and rules: {{RULE_CONTENT}} # (Question): Based on the above information, is the following statement true, false, or uncertain? {{QUESTION}} # (Answer):</p>

Table 12: Compact Symbolic-Aided CoT prompting for GSM8k dataset.

<pre> ### Let us define F as a function to infer solution from rules (mathematical conditions). Context: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today? # (Rule1): There are 15 trees in the grove # (Rule2): Grove workers will plant trees in the grove today. After they are done, there will be 21 trees # (Question): How many trees did the grove workers plant today? # (Answer): Variables = {original_trees, trees_after_some_more_were_planted, trees_were_planted} => F(Rule1) => original_trees = 15 => F(Rule2) => trees_after_some_more_were_planted = 21 => F(Question) => trees_were_planted = trees_after_some_more_were_planted - original_trees = 21 - 15 = 6. </pre>
<pre> Context: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot? # (Rule1): there are 3 cars in the parking lot # (Rule2): 2 more cars arrive # (Question): how many cars are in the parking lot? # (Answer): Variables = {original_cars, new_cars_arrive, current_cars} => F(Rule1) => original_cars = 3 => F(Rule2) => new_cars_arrive = 2 => F(Question) => current_cars = original_cars + new_cars_arrive = 5. </pre>
<pre> Context: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room? # (Rule1): There were nine computers in the server room # (Rule2): Five more computers were installed each day, from monday to thursday # (Question): How many computers are now in the server room? # (Answer): Variables = {original_computers, computers_installed_each_day, number_days, number_computer_now} => F(Rule1) => original_computers = 9 => F(Rule2) => computers_installed_each_day = 5; number_days = 4 # (from monday to thursday) => number_new_computers_installed = 5 * 4 = 20 => F(Question) => number_computer_now = original_computers + number_new_computers_installed = 9 + 20 = 29. </pre>
<pre> Context: Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On wednesday, he lost 2 more. How many golf balls did he have at the end of wednesday? # (Rule1): Michael had 58 golf balls # (Rule2): On tuesday, he lost 23 golf balls # (Rule3): On wednesday, he lost 2 more # (Question): How many golf balls did he have at the end of wednesday? # (Answer): Variables = {num_balls_michael_started, num_balls_tuesday_michael_lost, num_balls_wednesday_michael_lost, num_balls_wednesday_michael_have} => F(Rule1) => num_balls_michael_started = 58 => F(Rule2) => num_balls_tuesday_michael_lost = 23 => F(Rule3) => num_balls_wednesday_michael_lost = 2 => total_num_balls_michael_lost = num_balls_tuesday_michael_lost + num_balls_wednesday_michael_lost = 25 => F(Question) => num_balls_wednesday_michael_have = num_balls_michael_started - total_num_balls_michael_lost = 58 - 25 = 33. </pre>
<pre> Context: {{CONTEXT}} </pre>