Awakening LLMs' Reasoning Potential: A Fine-Grained Pipeline to Evaluate and Mitigate Vague Perception

Anonymous ACL submission

Abstract

Large language models (LLMs) are increasingly trained to abstain on difficult questions by answering *unknown*. However, we observe that LLMs often misuse this option: they output *unknown* even when LLMs can actually solve the questions, or they fail to understand why questions are truly unsolvable. We formalize this mismatch between potential ability and the inclination of *abstention* as the *Vague Perception* phenomenon. We introduce the WAKENLLM pipeline that (1) Extracts *Vague Perception* samples and (2) Measures how many of them can be converted to correct answers under stimulation. Based on stage-wise metrics (TCR, OCR, etc.) and the upper-bound accuracy Acc(WAKENLLM), we quantify LLMs' reasoning potential beyond one-shot accuracy. Experiments on six LLMs suggest that, without further training or parameter revisions, LLMs can achieve up to a 68.53% increase in accuracy on *Vague Perception* samples through our designed pipeline. We further analyze how *Vague Perception*, *Conformity* and *Degradation* vary from model families and parameter sizes, and offer model selection strategies in multi-stage reasoning workflows. Finally, by comparing WAKENLLM against mainstream reasoning baselines, both training and non-training ones, we show that existing baselines only activate a small portion of LLMs' reasoning potential, pointing to perception-aware reasoning as a promising direction for future LLM designing. Code and datasets are available at https://anonymous.4open.science/r/WakenLLM-toolkit-018B.

1 Introduction

LLMs such as GPT-4, Gemini-2.5 and Claude-3 have shown superior reasoning performance on datasets like BIG-Bench, MMLU, and GSM8K (Brown et al., 2020; OpenAI, 2023; Chowdhery et al., 2022; Srivastava and et al., 2022; Hendrycks and et al., 2021; Cobbe et al., 2021). With stronger performance, many papers focus on the trustworthiness and transparent reasoning to better understand the reasoning paradigms(Jiang et al., 2024; Bubeck, 2023). Specifically, solving Knowledge Boundary problems can help LLMs to know when to answer and when to abstain(Li et al., 2025b; Wen et al., 2024). However, most prior work focuses on whether and when LLMs should refuse to answer when facing difficult problems. In contrast, we ask a complementary yet opposite question: Are there situations that LLMs give up too early, despite being capable of solving the input questions? If so, inclinations towards *abstention* would harm LLMs' ability and performance.

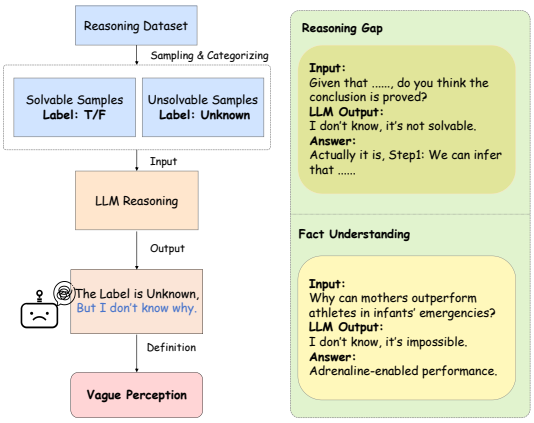


Figure 1: Definition of *Vague Perception*: LLMs misjudge solvable samples as *unknown*, or unable to understand unsolvable ones (Left). We attribute this phenomenon to two representative causes: *Fact Understanding* and *Reasoning Gap*—highlighting 1. Limited ability to logically solve problems. 2. Limited understanding of facts (Right).

We dive deeper into questions that contain uncertainties, as there are two possible situations regarding LLMs outputting *unknown*: (1) Problems are objectively unsolvable, or (2) LLMs subjectively and wrongly consider the problems unsolvable. The inclination of outputting *unknown* can

Table 1: Comparison of different LLM reasoning–guidance frameworks. WAKENLLM integrates all eight dimensions as the first holistic solution for diagnosing and mitigating the *Vague Perception* phenomenon. For used symbols, ✓: Fully covered, ●: Partially covered, ✗: Not covered. Detailed Related Work section can be found in Appendix A.

Paradigm	Guidance	Questioning	Stimulation	Dynamic Samples Selection	Reflection	Stage-wise Examining	Fine-Grained Decomposing	Diversity
CoT (Wei et al., 2022b)	✓	✗	✗	✗	✗	✗	✗	✗
ToT (Yao et al., 2023b)	✓	✓	✗	●	●	✓	✓	✓
Faithful-CoT (Lyu et al., 2023)	✓	✓	✓	✗	✗	✓	✗	✓
Plan&Solve (Wang et al., 2023b)	✓	✗	●	✗	✗	✓	✓	✗
Self-Refine (Madaan et al., 2023)	✓	✗	✓	✗	✓	✓	✗	✓
Reflexion (Shinn et al., 2023)	✓	✗	✓	✗	✓	✓	✗	✓
Decomposed Prompting (Khot et al., 2023)	✓	✓	✗	✗	✗	✗	✓	✗
Active Prompt (Diao et al., 2023)	✓	✗	✓	✓	✗	✗	✗	✓
SCORE (Zhang et al., 2024)	✓	✗	✓	✗	✓	✓	✗	✓
ProCo (Wu et al., 2024)	✓	✓	✓	✗	✓	✓	✗	✓
CoTAL (Cohn et al., 2025)	✓	✗	✓	✓	✗	✓	✗	✗
Cochain (Zhao et al., 2025)	✓	✓	✗	✓	✗	✓	✓	✓
ISP^2 (Zhu et al., 2025a)	✓	✓	✓	✗	✗	✗	✗	✓
Trace-of-Thought (McDonald and Emami, 2025)	✓	✓	✗	✗	✗	✓	✓	✗
MAPS (de Souza Loureiro et al., 2025)	✓	✗	✓	●	✓	✓	✗	✓
CoD (Xu et al., 2025)	✓	✗	✗	✗	✗	✓	✗	✗
ReSearch (Chen et al., 2025)	✓	✓	✗	✓	✓	✓	✓	✗
WAKENLLM (Ours)	✓	✓	✓	✓	✓	✓	✓	✓

largely harm LLMs' performance, as this situation could be potentially corrected, this paper formalizes it as the*Vague Perception* phenomenon, as in Figure 1. Since potential reasoning ability remain under-explored, understanding how LLMs express uncertainty and the potential of overcoming this phenomenon can be rather important for improving trustworthiness and reliable LLM reasoning.

To address this, we propose WAKENLLM as a fine-grained pipeline to explore potential reasoning abilities of LLMs on *Vague Perception* samples, yielding substantial improvements in accuracy across six LLMs, both open-source and closed-source families, which further proves that LLMs' reasoning ability is harmed by inclinations of outputting *unknown* and *abstention*.

Dataset samples are input into LLMs for inference, and we extract the corresponding *Vague Perception* samples that meet the conditions in Figure 1. Subsequently, the samples are processed via a two-stage pipeline aiming at "awakening" LLMs' reasoning potential against the *Vague Perception* phenomenon, as shown in Figure 3. By measuring the "awakening ability", we present a different perspective on evaluating and mitigating this phenomenon, one that considers not only LLMs one-shot performance, but also the potential ability of solving problems instead of *abstaining*.

Apart from evaluations and problem mitigation, we offer model-selection strategies in multi-stage reasoning workflows, as our pipeline shows that placing specific models at certain sequences matters much. Besides, we dive deeper into the *Vague Perception* phenomenon to explore the root causes.

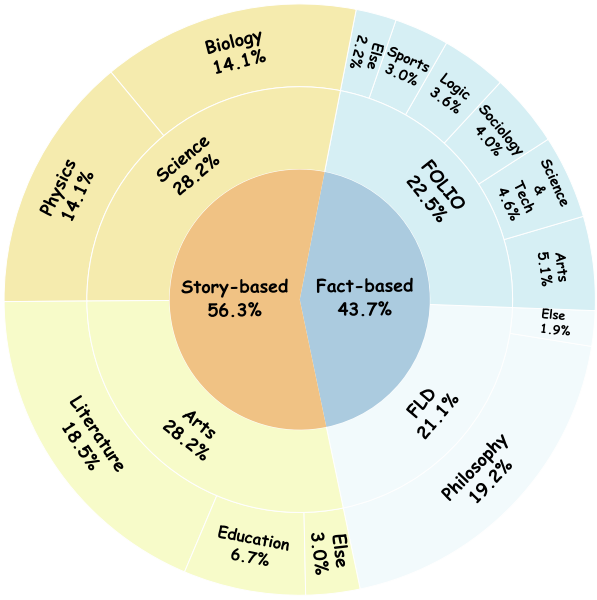
To summarize, our contributions are fourfold:

- We formulate the *Vague Perception* phenomenon, where models output *unknown* despite having the potential to solve problems correctly. We propose WAKENLLM, a fine-grained pipeline that evaluates the percentage of *Vague Perception* samples, and mitigate this phenomenon by eliciting LLMs' reasoning potential.
- We introduce multiple stage-wise converting metrics (TCR, OCR, etc.) and the overall upper-bound accuracy Acc(WAKENLLM) that quantify the ratio of *Vague Perception* samples can be converted to correct answers without any further training or parameters revisions, evaluating and eliciting LLMs' reasoning potential.
- Across six LLMs and four datasets (*Fact-based* and *Story-based*) from different fields, we analyze how *Vague Perception*, *Conformity*, and *Degradation* vary from model families and parameter sizes, thus we offer LLM selection strategies for placing models at specific sequences in multi-stage reasoning workflows.
- By comparing WAKENLLM with existing reasoning baselines (both Training and Non-Training), we measure how effectively each framework elicits LLMs' potential. Our work offers a new perspective on future developing LLM reasoning frameworks with more focus on their under-explored potential.

2 Datasets Composition

To explore the effects of the *Vague Perception* phenomenon on different types of samples, we classify

Figure 2: Overall dataset distribution, with samples grouped into two forms: **story-based** (56.3%) and **fact-based** (43.7%). Details of topics ("Else" category included) is in Appendix C.



them by their labels: (1) *solvable samples*, where problems have *True/False* labels that can be solved with valid reasoning; and (2) *unsolvable samples*, which are labeled *unknown* and objectively can not be solved. Furthermore, as in Figure 2, to better understand how different context styles influence, we categorize datasets as follows:

Fact-based Datasets The information in contexts is stated atomically, like "A is X, B is Y". **FLD** (Morishita et al., 2024) supplies three kinds of sample labels annotated as *True*, *False*, or *Unknown*. We randomly sample 600 samples: 300 solvable (*True*/*False*) and 300 unsolvable *Unknown* ones. And we draw 640 samples—320 solvable samples with *T/F* labels and 320 *Unknown* samples from **FOLIO** (Han et al., 2022), which has the same three-way labeling.

Story-based Datasets The contexts are long passages, more aligned with humans' reading habits. We extract samples from four fields: *Physics*, *Biology*, *Figurative-language* and *Writing-strategies* from ScienceQA (Lu et al., 2022), and applied the same sampling strategy. The former two are categorized as Science datasets, the latter two are Arts datasets. Dataset details are in Appendix C.

3 Proposed Pipeline: WAKENLLM

The WAKENLLM pipeline is illustrated in Figure 3. It first extracts *Vague Perception* samples from input datasets, later the two-stage stimulation is implemented to elicit model's potential on these samples. For the Vanilla Setting, the stage 1 deals with the initial *Vague Perception* samples, and the stage 2 receives falsely answered samples from stage 1 (FC^1). Via the two-stage stimulation pipeline, we can get the percentage of *Vague Perception* samples that are eventually corrected by LLMs. For **Remind-then-Guide (RtG)** settings, in addition to reminding LLMs of prior mistakes, we either randomly assign labels, or provide previous reasoning processes in the prompt. We examine whether LLMs still show conformity after being reminded of answering problems wrong previously; or they would conform to previous reasoning processes and make the same mistakes. Detailed metrics and pseudo-code visualizations are available in Appendix D and M.

Table 2: Information of datasets used in experiments.

Dataset	Samples	Form	Writing	Source
FLD	600	**Conclusion**	Abstract	Gen
FOLIO	640	**Conclusion**	Plain	Real
Science	800	**Q&A**	Plain	Real
Arts	800	**Q&A**	Plain	Real

3.1 Samples Extraction

Our pipeline begins with reasoning on preprocessed datasets with 50% *solvable* (Label:True/False) and 50% *unsolvable* (Label:Unknown) samples, we acquire tasks accuracy and extract corresponding *Vague Perception* samples—where LLMs output *unknown* when samples are solvable, or fail to explain why samples are truly unsolvable. Since this is performed separately for each model, *Vague Perception* samples differs from each too. Details are in Appendix J and Table 8.

3.2 Vanilla Setting

After extracting *Vague Perception* samples, we input them into stage 1. The conversion results fall into three categories: (1) LLMs output correct labels under stimulation, called **True Converting (TC)**. (2) LLMs output an incorrect *T/F* label for *solvable* samples, denoted as **False Converting (FC)**. (3) LLMs fail to comprehend and remain output *Unknown* for *solvable* samples, which is **Unknown Converting (UC)**. To quantify these proportions, we use **UCR**, **FCR**, and **TCR** to represent **rates of each type converting.** Detailed for-

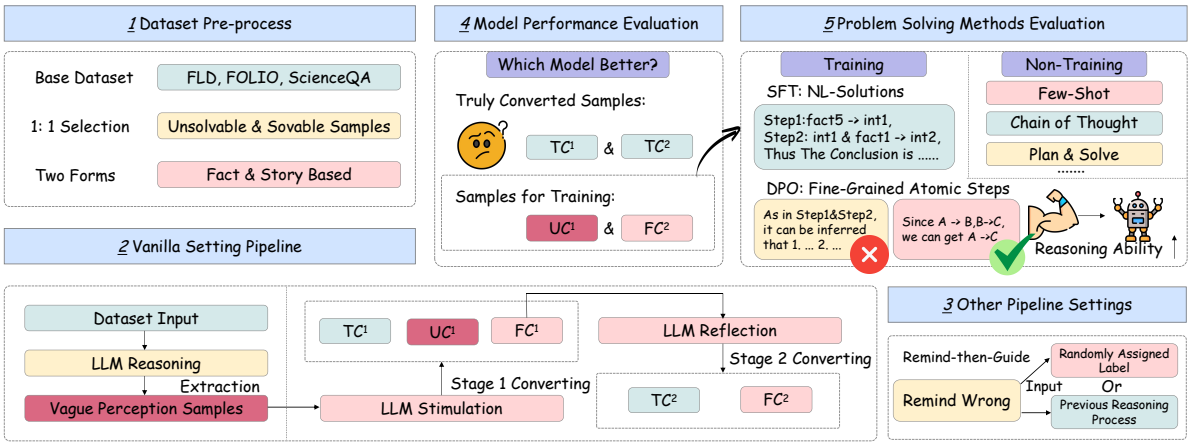


Figure 3: Overview of our proposed pipeline. We begin with Fact and Story-based datasets, sampling from solvable (T/F) and unsolvable (Unknown) samples at 1 : 1 ratio. The pipeline then (1) Extracts *Vague Perception* samples during LLMs' reasoning and (2) Applies a two-stage pipeline: stage 1 uses stimulation to re-evaluate the mislabeled samples, while stage 2 gathers the remaining *False Converting* samples from stage 1 for model self-reflection. Stimulation types vary from different dataset forms.

mulas are in Appendix F. Given our focus on the reasoning correction of LLMs, we define them as:

$$TC_t^n = \sum_{y_t \in D_{\text{VP}}^n} \mathbb{1}(\hat{y}_t^n = y_t), \quad TCR_t^n = \frac{|TC_t^n|}{|D_{\text{VP}}^n|}$$

$$FC_t^n = \sum_{y_t \in D_{\text{VP}}^n} \mathbb{1}(\hat{y}_t^n \neq y_t)$$

$$UC_t^n = \sum_{y_t \in D_{\text{VP}}^n} \mathbb{1}(\hat{y}_t^n \in U,\ y_t \notin U)$$

In the above equations, $t \in \{s, u\}$ denotes sample types (solvable or unsolvable), and $n \in \{1, 2\}$ indicates the stimulation stage. D_{VP}^n represents the input of *Vague Perception* samples in the n-th stage. First stage input D_{VP}^1 was extracted from the input datasets. For the second stage, we collect the **False Converting** samples as the input, from which we can obtain: $D_{\text{VP}}^2 = FC^1$.

Omitting superscripts like TCR_s represent the sum of results from stage 1 and 2, and TCR^1 denotes the converting rate of both sample forms in the first stage. To be specific, $TCR_s = TCR_s^1 + TCR_s^2$, standing for the TCR value of solvable samples in both stages; Likewise, $TCR^1 = TCR_s^1 + TCR_u^1$, indicating TCR for stage 1 that sums up both solvable and unsolvable samples.

To get a comprehensive view of LLMs' reasoning potential, we define **Overall Converting Rate (OCR)** as follows:

$$OCR = \frac{|TC^1 \cup TC^2|}{|D_{\text{vp}}|}, \quad TC^1 \cap TC^2 = \emptyset$$

OCR refers to the portion of *Vague Perception* samples that are eventually corrected by LLMs. By calculating this, we can obtain the potential accuracy harmed by inclinations of *abstention*.

3.3 Remind-then-Guide (RtG) Setting

Label Conformity We test how LLMs display conformity facing *Vague Perception* samples. We firstly inform LLMs of inputting *Vague Perception* samples, then provide randomly assigned labels to test whether misguiding affects outputs. We compare TCR changes with and without the RtG setting, and average results on two stages:

$$Conf_t = \frac{1}{2}(\underbrace{TCR_t^1 - TCR_t^{1'}}_{\text{Stage 1 Conformity}} + \underbrace{TCR_t^2 - TCR_t^{2'}}_{\text{Stage 2 Conformity}})$$

As indicated in Section 3.2, the subscript t represents two sample types s or u, separating conformity analysis by solvable and unsolvable samples. $TCR_t^{1\prime}, TCR_t^{2\prime}$ indicates stage-wise True Converting Rate given all assigned labels are wrong.

Reasoning Process (RP) Conformity We also examine whether LLMs, after being reminded of previous mistakes, can reflect on previous reasoning processes to avoid making same mistakes. To the best of our knowledge, we are the first to study conformity on sentence-level answers.

Reasoning Process Coherence (RPC) Including LLMs' reasoning processes in prompts, and we rerun stage 1 to assess its stability. Similarly, RPC is calculated by changes of converting rate with and

Table 3: Comparison of six LLMs across four datasets (FLD, FOLIO, Science, Arts), grouped by evaluation settings. **RP** refers to reasoning process under Remind-then-Guide setting. The highest value is in bold per column.

Model	Dataset	Vanilla			RtG (Label)		RtG (RP)	
		TCR^1	TCR^2	OCR	Conf$_s$	Conf$_u$	CGR	RPC
GPT-3.5	**FLD**	12.61	14.43	25.22	**64.25**	3.33	-12.30	10.84
	FOLIO	14.41	42.10	50.44	16.81	31.97	-42.10	-6.30
	Science	21.05	3.33	21.63	19.22	1.96	-3.33	0.59
	Arts	26.04	4.00	27.10	14.24	-3.88	0.00	3.12
GPT-4o	**FLD**	18.88	28.19	41.65	8.17	37.38	-23.50	-6.48
	FOLIO	37.34	44.40	65.16	9.63	51.90	-44.77	0.00
	Science	24.73	7.76	28.96	11.74	4.58	-7.16	-3.68
	Arts	18.60	8.24	24.17	8.07	23.31	-8.24	1.14
LLaMA-3.1-8B	**FLD**	41.97	26.03	48.90	50.04	11.04	-18.56	1.09
	FOLIO	45.86	11.40	50.84	38.73	13.61	-3.07	5.52
	Science	45.94	5.95	50.61	38.60	51.44	-5.95	12.91
	Arts	24.26	8.57	25.75	30.99	9.02	-4.77	11.30
LLaMA-3.1-405B	**FLD**	20.65	**53.73**	41.50	9.35	**57.63**	-48.28	-17.66
	FOLIO	28.76	29.92	49.22	6.11	23.01	-17.69	0.00
	Science	32.85	2.50	33.03	9.02	18.50	**0.19**	0.71
	Arts	4.40	0.00	4.40	0.81	28.91	0.00	-0.77
Qwen-2.5-7B	**FLD**	**48.31**	40.00	**68.53**	22.06	42.34	-31.67	-5.61
	FOLIO	19.73	36.84	47.36	4.92	38.71	-36.84	14.48
	Science	15.83	10.00	24.08	55.21	0.00	-2.98	**14.59**
	Arts	29.34	25.37	36.44	43.39	7.20	-13.27	8.47
Qwen-2.5-72B	**FLD**	48.20	9.80	51.21	33.08	9.14	-9.80	1.21
	FOLIO	31.51	11.37	38.36	10.65	2.20	-6.82	-1.37
	Science	10.81	3.12	11.20	33.42	22.07	0.00	-0.38
	Arts	25.00	9.76	25.56	12.52	24.12	0.00	-1.95

without reasoning processes. Values below zero indicates that the previous reasoning process is misguiding and thus results in declined accuracy, while above zero suggests that it is helpful and can guide, or help reflect reasoning towards the correct way.

Correct Guiding Rate (CGR) We save reasoning processes of FC^1 samples in stage 1, and run stage 2 by including them in prompts. Similarly, we remind LLMs these reasoning may contains flaws since the samples come from the FC^1. CGR accesses whether LLMs can better reflect on their reasoning, is calculated by the changes of converting rate with and without reasoning processes.

$$RPC = \frac{|TC^1| - |TC^{1\prime\prime}|}{|D^1_{VP}|}, CGR = \frac{|TC^2| - |TC^{2\prime\prime}|}{|D^2_{VP}|}$$

$TC^{1\prime\prime}, TC^{2\prime\prime}$ are obtained with previous reasoning processes included in prompts. Detailed summary of metrics is available in Table 7.

4 Experiments

4.1 Experimental Setup

Our experiment covers both open and closed-source models: GPT-3.5 and GPT-4o (OpenAI, 2023); LLaMA-3.1 series (Touvron et al., 2023), and Qwen-2.5 series (Bai et al., 2023). For each model, two stimulating strategies are implemented:
▷ **Concise Stimulation**: Basic error notifications without correction guidance.
▷ **Detailed Stimulation**: Detailed error notifications with structural and step-by-step guidance.

The experimental design considers different characteristics of two datasets forms. For Fact-based ones, which LLMs are less trained on, providing *Detailed Stimulation* can enhance performance; For Story-based ones, the information is embedded in long stories, which LLMs are largely trained on, providing *Detailed Stimulation* can induce hallucinations (Huang et al., 2025). Experiments of different stimulation styles are in Section 5.4.

To achieve a balance between randomness and dynamicity, we set all Temperatures to 0.5, except for 1.0 in Ablation Study as we want to examine the quantitative influence between prompts designing and LLMs setup.

In our results evaluation procedure, we apply a two-step output checking: the output answering is first examined for specific keywords matching. When this method fails to find one matched keyword, GPT-4o (Temperature=0.0) is deployed as a LLM judge and assign appropriate labels.

4.2 Results on the Vanilla Setting

As shown in Table 3, we report detailed experimental results, including three metrics addressed before: True Converting Rate (TCR) for both stages, and Overall Converting Rate (OCR). The LLaMA and Qwen series show superior OCR on small-parameter series, whereas the GPT series shows

the opposite trend. We infer that, in general, higher parameter sizes correlate with lower OCR.

4.3 Results on the RtG Setting

RtG settings contain two modes: Label and Reasoning Process (RP) Conformity: We remind LLMs that they incorrectly predicted the sample before, then add randomly assigned labels, or previous reasoning processes. A wrap-up of findings can be found in Appendix L. During our experiment, we adjust rates of assigned wrong labels at four given probabilities (100%, 66.7%, 50%, 0%). Detailed Remind-then-Guide conformity tests are available in the Table 9, 10, 11, 12 and Appendix H.

Label Conformity Judging from Table 3, we observe that even when LLMs are reminded of their previous incorrect answers, they still demonstrate a strong tendency towards conformity. One key influencing factor appears to be the parameter sizes: models with smaller parameter sizes, such as GPT-3.5 and LLaMA-3.1-8B, are more easily misled when reasoning solvable samples with *T/F* labels. In contrast, models with larger parameter sizes, such as GPT-4o and LLaMA-3.1-405B, strongly conforming to unsolvable samples with *Unknown* label, which is a trade-off bias regarding different sample types. Meanwhile, Qwen series show distinct yet related features, it may be attributed to smaller parameters gap between 7B and 72B compared to other model families.

Reasoning Process (RP) Conformity We input previous reasoning processes and explicitly require LLMs to reflect on their mistakes. However, the resulting *CGR* values are mostly below zero, indicating that after being reminded of a past error and provided with its own flawed reasoning processes, LLMs still tend to repeat previous incorrect conclusions, leading to even larger accuracy declines.

Results on both CGR and RPC reveal that previous reasoning processes are highly unstable. Changes in accuracy can be attributed to the divergence in how problems are reasoned at different times. Notably, for *unsolvable samples*, reasoning processes tend to be more unreliable — reflecting on them typically leads to a higher drop in accuracy, which exposes a hidden issue: In large-scale LLM-annotated "problem-solving solutions" with own reasoning processes, unsolvable samples are particularly susceptible to unreliability. For *solvable samples*, the reasoning remains relatively unstable too. As a result, **the precision and reliability of LLM-annotated reasoning steps for dataset samples remain to be a problem, which strongly needs manual verifications.**

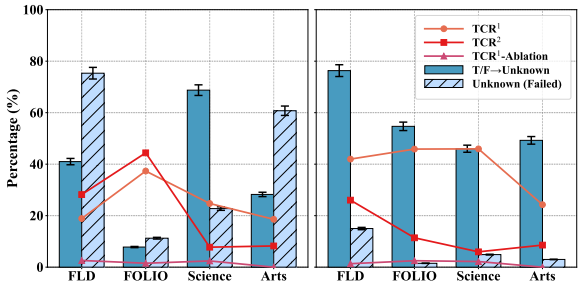


Figure 4: Percentage of GPT-4o (left) and LLaMA-3.1-8B (Right) exhibit *Vague Perception*. "T/F→Unknown" indicates solvable samples are predicted as "*Unknown*". "Unknown (Failed)" denotes cases predicted as "*Unknown*" because of limited understandings.

5 Detailed Analysis

5.1 TCR Analysis & Insights

As shown in the line chart of Figure 4, we observe clear divergences between TCR^1 and TCR^2 across different datasets. This indicates that many samples mis-predicted in stage 1 can be corrected in stage 2, where the model refines its initial output. In contrast, Story-based TCR^1 shows a substantially lower TCR^2, whereas LLaMA-3.1-8B demonstrates much higher TCR^1.

Thus, we introduce **model selection strategies for multi-stage reasoning workflows**: (i) For GPT-4o, when facing *Fact-based* datasets, we can place the model at a latter order, since stage 2 offers a huge accuracy gain; However, for *Story-based* datasets, the improvement of stage 2 accuracy is limited, so being placed in the front should be better. (ii) For LLaMA-3.1-8B, it is always better to "do it right the first time", as being placed in the front always leads to better performance. By deploying LLMs at proper sequences, we can design better multi-stage reasoning workflows. Details are available in Appendix L.4.

5.2 Vague Perception Phenomenon Statistics

As LLMs differ in reasoning ability, we first quantify the percentage of samples that models show *Vague Perception* on. Bar charts in blue in Fig 4 demonstrate the percentage of samples that GPT-4o and LLaMA-3.1-8b showing this phenomenon, while complete results for all models are in Table 8.

We conclude that **the *Vague Perception* phenomenon is prevalent across LLM Reasoning**.

For GPT-4o, FOLIO exhibits a relatively lower percentage. For FLD and Arts—which focus on story understanding, LLMs exhibit greater variability in understanding why samples are labeled as *unknown*; In sharp contrast, *Science Dataset* faces significant challenges even when the samples are solvable; For LLaMA-3.1-8B, there is a significant rise in the percentage of *solvable samples* predicted as *unknown*, but the ability to comprehend *unknown* samples is much worse.

5.3 Ablation Study

From the stage 1 output, we isolate the Unknown Converting samples (UR^1)–those LLMs have *abstained* on twice, and re-input them back to stage 1 stimulation. Both LLaMA-3.1-8b and GPT-4o are set with 1.0 temperature to ensure dynamicity.

As Figure 4 shows, this procedure produces virtually few converting samples: more than 95% of the samples remain unchanged. This finding demonstrates that *Vague Perception* is largely immune to repeated n-shot prompting; applying the same stimulation twice is therefore futile. **Consequently, any effective correction must rely on a distinct stage 2 rather than reiterating stage 1, adjusting temperature alone without revising stimulation style can be limited, and all LLMs perform stably when exposed to the same prompt content multiple times.**

5.4 Stimulating Styles Comparison

As indicated in Section 4.1, we apply different prompt styles to different datasets. To evaluate the effectiveness, we swapped the two styles of stimulation within our pipeline, with GPT-4o as the base model. As shown in Table 4, **switching stimulation styles leads to a drop in TCR^1_s**, indicating weakened understanding on *solvable samples* with *T/F* labels. In contrast, TCR^1_u increases for unsolvable samples. Although the overall converting rate (OCR) improves, this gain is unstable, as TCR^1_s remains more consistent. **Reliability** indicates the results are more reliable based on higher TCR^1_s.

Furthermore, we observe a trade-off: **as the reasoning ability on solvable samples declines, the model becomes better at reasoning unsolvable ones**, a shift in LLM reasoning biases.

5.5 Baseline Comparisons

In our work, we primarily analyze the "awakening ability" regarding *Vague Perception* samples being answered correctly based on TC^1 and TC^2 samples, and this is denoted as LLMs' "Potential Ability". We combine this with the "Explicit Ability" *Acc(Direct)*, as in Section 3.1, our pipeline starts with LLM reasoning on four datasets, acquiring the corresponding accuracy *Acc(Direct)* and extracting *Vague Perception* samples. By adding "Potential" and "Explicit" accuracy together, we gain the LLMs' potential ability can be achieved without additional training or parameter revisions. Here, *D* represents the complete dataset, as we calculate the proportion of TC^1 and TC^2 samples over the whole dataset to acquire the accuracy. The formula is written as below:

$$\text{Acc}(\textsc{WakenLLM}) = \underbrace{\text{Acc}(\text{Direct})}_{\text{Explicit Ability}} + \underbrace{\frac{|TC^1 \cup TC^2|}{|D|}}_{\text{Potential Ability}}$$

As illustrated in the line chart of the left picture in Figure 5, most mainstream reasoning baselines exhibit substantial performance gaps compared to WAKENLLM, indicating current reasoning frameworks are far from eliciting LLMs' full potential, pointing to future perception-aware LLM designing. Details are in Table 6.

Table 4: Converting rate regarding different stimulating methods *CS*, *DS* refer to *concise* and *detailed Stimulation*. Rows shaded gray are used in our pipeline. Higher values of TCR^1_s are associated with better Reliability.

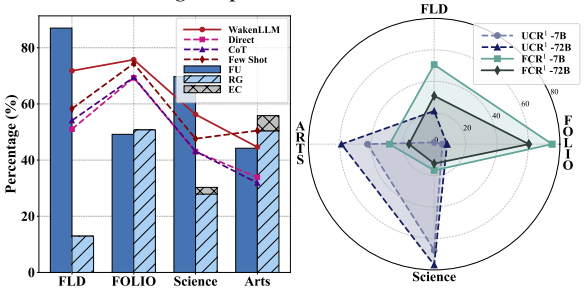
Metrics(%)	TCR^1	TCR^1_s	TCR^1_u	Reliability
FLD (CS)	37.27	10.65	26.62	↓
FLD (DS)	18.88	**13.64**	5.24	↑
Arts (CS)	18.60	**16.06**	2.54	↑
Arts (DS)	22.54	14.65	7.89	↓

5.6 Phenomenon Exploration

To explore root causes of *Vague Perception*, we deploy GPT-4o as an LLM judge (Temperature=0.0) to identify problems based on their reasoning processes, problems mainly fall into three categories:
▷ **Fact Understanding (FU)** – LLMs do not comprehend facts in the passage.
▷ **Reasoning Gap (RG)** – LLMs fail to perform logical operations that reach the hypothesis.
▷ **Excessive Caution (EC)** – LLMs demonstrate cautiousness over outputting their answering.

Causes Distribution As shown in Figure 5, EC is a minor cause compared to FU and RG. Our experiments reveal that root causes LLMs inclined to output *unknown* result from both objective and

Figure 5: Reasons of samples demonstrating *Vague Perception* errors for GPT-4o (left). Qwen-2.5-7B and 72B False and Unexcited Converting Rate (FCR, UCR) comparison (Right). "FU": *Fact Understanding*, "RG": *Reasoning Gap*, and "EC": excessive caution.



subjective reasons. For the former part, FU and RG represent whether LLMs can understand given facts, or successfully inference based on comprehension, which are the main reasons; for EC, the inner biases, is a minor problem that only occurs in *Story-based* datasets. We attribute this to the narrative complexity of *Story-based* samples: long stories in **Arts** highlight comprehension requirements. Detailed analysis is available in Appendix J.

5.7 Degradation Analysis

In the stage 2 of WAKENLLM pipeline, we don't exclude the possibility that incorrectly predicted samples FC^1 from stage 1 may be further converted to *unknown* in stage 2, which should be denoted as UC^2. It is reasonable that LLMs initially generate incorrect answers, but because of the limited ability, they fail to output correct answers after further stimulation, and instead output *unknown*. We elaborate this degradation behavior as below:

$$Deg = \frac{\left|FC^1 \setminus \left(FC^2 \cup TC^2\right)\right|}{|FC^1|},\ FC^2 \cap TC^2 = \emptyset$$

As shown in Table 5, GPT-4o achieves higher TCR^2 among all datasets than GPT-3.5. And problems regarding *Degradation* can be minor, since only GPT-4o demonstrates serious effects on Science dataset. Coupled with higher *Deg*, this suggests that **when facing problems LLMs previously answered incorrectly—especially those beyond their ability—GPT-4o tends to output *Unknown*. In contrast, GPT-3.5 is much more likely to provide a definitive but incorrect answer instead of admitting uncertainty.** One factor attributed to this phenomenon is LLMs' parameter sizes.

This is further proved by the uniform distributions of UCR and FCR for Qwen-7b and 72b series.

Table 5: Degradation Analysis of GPT-4o and GPT-3.5. Exceptionally high *Deg* values of GPT-4o is in bold.

Models	**GPT-4o**		**GPT-3.5**	
Metrics(%)	**TCR^2**	**Deg**	**TCR^2**	**Deg**
FLD	28.19	3.08	14.43	3.09
FOLIO	44.40	5.22	42.10	6.32
Science	7.76	**26.86**	3.33	3.33
Arts	8.24	1.13	4.00	0.00

The formulas are addressed in Appendix F, calculated similarly to TCR. We compare the answering of the Qwen-2.5 series (7B and 72B model), as shown in the right radar-chart in Figure 5: (1) Models show different yet uniform output distribution among four datasets. For FLD and FOLIO, **LLMs are more willing to output a false answer rather than outputting *unknown***; However, for **Story-based datasets**, like Science and Arts, LLMs behave **exactly the opposite**. (2) Higher the parameter sizes, the more likely models are to output *unknown* on *Vague Perception* samples.

6 Conclusion

In this work, we formulate the *Vague Perception* phenomenon: the inclinations of LLMs outputting *unknown* despite being potentially able to solve input questions, and such *abstention* behavior strongly influence LLM performances. We present WAKENLLM, a fine-grained pipeline for evaluating and mitigating the *Vague Perception* phenomenon in LLM reasoning. Our pipeline elicits LLMs' potential reasoning ability, substantially improving the accuracy among tested datasets. Experiments show that while LLMs give up and answer *unknown* on the first try, a well-designed pipeline can stimulate them to answer correctly. These insights suggest evaluation metrics that value LLMs' ability beyond one-shot performances alone. By researching *Vague Perception*, *Conformity* and *Degradation*, we better understand reasoning behaviors vary from model families and parameter sizes. Although significant leaps in LLM architectures may be limited in the near future, our findings suggest that current reasoning baselines remain far from eliciting LLMs' full potential, paving the way for future perception-aware LLM designing. Our research offers high values for trustworthiness and reliable LLM reasoning development.

7 Limitations

Even though our work has helped elicit LLMs' reasoning potential, our final metrics are calculated with respect to output labels, and producing the correct label while providing a flawed reasoning chain is common and remains an open problem for better reinforcement learning approaches in the future. How to fundamentally quantify their true ability with true reasoning process remains a problem in the future. Besides, specifying whether the output *Unknown* is due to understanding the unverifiable elements or inability to fully comprehend still remains a problem, currently we just clarify demands within prompts.

8 Ethical Considerations

Our research utilizes publicly available datasets (FLD, FOLIO and ScienceQA) containing synthetic and artificial reasoning samples without personal information. We acknowledge potential biases in our SFT and DPO training processes and implement atomic reasoning criteria to ensure fair evaluation. All training scripts, evaluation code, and prompt templates are made available for reproducibility. We focus on improving reasoning quality rather than enabling harmful applications. Our work adheres to academic integrity standards and contributes to responsible AI development through more interpretable reasoning.

References

Alibaba DAMO Academy. 2024. Qwen2: The next-generation large language model series. https://huggingface.co/Qwen. Accessed: 2025-07-27.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Sébastien et al. Bubeck. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, and 1 others. 2025. Research: Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, and 1 others. 2022. Palm: Scaling language modeling with pathways. In *Proceedings of ICML*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2924–2936. Association for Computational Linguistics.

Karl Cobbe and 1 others. 2021. Training a helpful and harmless assistant with reinforcement learning. *arXiv preprint arXiv:2104.08718*.

Clayton Cohn, Nicole Hutchins, Ashwin T. S., and Gautam Biswas. 2025. Cotal: Human-in-the-loop prompt engineering, chain-of-thought reasoning, and active learning for generalizable formative assessment scoring. *Preprint*, arXiv:2504.02323.

André de Souza Loureiro, Jorge Valverde-Rebaza, Julieta Noguez, David Escarcega, and Ricardo Marcacini. 2025. Advancing multi-step mathematical reasoning in large language models through multi-layered self-reflection with auto-prompting. *arXiv preprint arXiv:2506.23888*.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xiang Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3563–3578. Association for Computational Linguistics.

Shizhe Diao, Pei Wang, Yujia Lin, Ruili Pan, Xuefeng Liu, and Tong Zhang. 2024. Active prompting with chain-of-thought for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, pages 1330–1350. Association for Computational Linguistics.

Shizhe Diao, Pengcheng Wang, Yong Lin, Rui Pan, Xiang Liu, and Tong Zhang. 2023. Active prompting with chain-of-thought for large language models. arXiv preprint arXiv:2302.12246. Cs.CL.

Yilun Du, Sherry Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, pages 11733–11763. PMLR.

Luyu Gao, Aman Madaan, Shuning Zhou, Uri Alon, Pengfei Liu, Yinhan Yang, Jamie Callan, and Graham Neubig. 2022. PAL: Program-aided language models. *arXiv preprint arXiv:2211.10435*.

Zhipeng Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yang Yang, Nan Duan, and Weizhu Chen. 2024. Critic: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, and 1 others. 2022. Folio: Natural language reasoning with first-order logic. *arXiv preprint arXiv:2209.00840*.

Dan Hendrycks and et al. 2021. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Boyi Hou, Yucheng Liu, Kun Qian, Jacob Andreas, Shiyu Chang, and Yiming Zhang. 2024. Decomposing uncertainty for large language models through input clarification ensembling. In *Proceedings of the 41st International Conference on Machine Learning*, pages 19023–19042. PMLR.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Xinyan Huang, Shuo Li, Mo Yu, Maurizio Sesia, Hadi Hassani, Insu Lee, Osbert Bastani, and Edgar Dobriban. 2024. Uncertainty in language models: Assessment through rank-calibration. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 284–312. Association for Computational Linguistics.

Jiahao Huo, Shuliang Liu, Bin Wang, Junyan Zhang, Yibo Yan, Aiwei Liu, Xuming Hu, and Mingxun Zhou. 2025. Pmark: Towards robust and distortion-free semantic-level watermarking with channel constraints. *arXiv preprint arXiv:2509.21057*.

Chujie Jiang, Runzhe Dong, Xin Liu, Diyi Xiong, and Yiming Yang. 2024. Honestllm: Measuring and improving truthfulness in large language models. In *Findings of the Association for Computational Linguistics*, pages 1234–1250.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed prompting: A modular approach for solving complex tasks. *Preprint*, arXiv:2210.02406.

Jiaxin Li, Geng Zhao, and Xiaoci Zhang. 2025a. Multilingual federated low-rank adaptation for collaborative content anomaly detection across multilingual social media participants. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15253–15273, Suzhou, China. Association for Computational Linguistics.

Moxin Li, Yong Zhao, Wenxuan Zhang, Shuaiyi Li, Wenya Xie, See-Kiong Ng, Tat-Seng Chua, and Yang Deng. 2025b. Knowledge boundary of large language models: A survey. *Preprint*, arXiv:2412.12472.

Zipeng Ling, Yuehao Tang, Chen Huang, Shuliang Liu, Gaoyang Jiang, Shenghong Fu, Junqi Yang, Yao Wan, Jiawan Zhang, Kejia Huang, and Xuming Hu. 2025. Instruction boundary: Quantifying biases in llm reasoning under various coverage. *Preprint*, arXiv:2509.20278.

Shuliang Liu, Xuming Hu, Chenwei Zhang, Shu'ang Li, Lijie Wen, and Philip Yu. 2022. HiURE: Hierarchical exemplar contrastive learning for unsupervised relation extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5970–5980, Seattle, United States. Association for Computational Linguistics.

Shuliang Liu, Hongyi Liu, Aiwei Liu, Bingchen Duan, Qi Zheng, Yibo Yan, He Geng, Peijie Jiang, Jia Liu, and Xuming Hu. 2025a. A survey on proactive defense strategies against misinformation in large language models. *Preprint*, arXiv:2507.05288.

Shuliang Liu, Qi Zheng, Jesse Jiaxi Xu, Yibo Yan, Junyan Zhang, He Geng, Aiwei Liu, Peijie Jiang, Jia Liu, Yik-Cheung Tam, and Xuming Hu. 2025b. Vla-mark: A cross modal watermark for large vision-language alignment model. *Preprint*, arXiv:2507.14067.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. arXiv preprint arXiv:2301.13379. Cs.AI.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Peter Clark, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*.

Tyler McDonald and Ali Emami. 2025. Trace-of-thought prompting: Investigating prompt-based knowledge distillation through question decomposition. *arXiv preprint arXiv:2504.20946*.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*. Association for Computational Linguistics.

Terufumi Morishita, Gaku Morio, Atsuki Yamaguchi, and Yasuhiro Sogawa. 2024. Enhancing reasoning capabilities of llms via principled synthetic logic corpus. *Advances in Neural Information Processing Systems*, 37:73572–73604.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, and et al. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

Tobias Olausson, Albert Gu, Brandon Lipkin, Chiyuan 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. Association for Computational Linguistics.

OpenAI. 2022. ChatGPT: `gpt-3.5-turbo`. https://openai.com/blog/chatgpt. Accessed 19 Jun 2025.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Long Ouyang, Jeff Wu, Xu Jiang, and *et al.* 2022. Training language models to follow instructions with human feedback. *arXiv:2203.02155*.

Rafael Rafailov, Usama Chughtai, Yining Zhang, Xuechen Li, Colin Raffel, and Tatsunori Hashimoto. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*. Association for Computational Linguistics.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*.

Aarohi Srivastava and et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. In *International Conference on Learning Representations*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: A large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2018)*. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron and Meta AI LLM Team. 2025. Llama 3: Open foundation and instruction models. https://ai.meta.com/llama. Accessed 19 Jun 2025.

Miles Turpin, Stephanie Lin, Ethan Perez, and Samuel R Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Bohan Wang, Xiang Zhang, Cheng Liu, and Pan Zhou. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023b. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*.

Xuezhi Wang, Ethan Zhou, Abulhair Saparov, Quoc Le, and Hsien-Yu Liu. 2023c. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*.

Yujia Wang, Zhe Zhang, Junxian He, and Lianhui Li. 2024. Reinforced self-training for reasoning-enhanced language models. *arXiv preprint arXiv:2402.10270*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, and et al. 2022a. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Zhou, Jay Li, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Zhihua Wen, Zhiliang Tian, Zexin Jian, Zhen Huang, Pei Ke, Yifu Gao, Minlie Huang, and Dongsheng Li. 2024. Perception of knowledge boundary for large language models through semi-open-ended question answering. In *Advances in Neural Information Processing Systems (NeurIPS) 37*.

Zhiyuan Weng, Guikun Chen, and Wenguan Wang. 2025. Do as we do, not as you think: the conformity of large language models. *Preprint*, arXiv:2501.13381.

Zhenyu Wu, Qingkai Zeng, Zhihan Zhang, Zhaoxuan Tan, Chao Shen, and Meng Jiang. 2024. Large language models can self-correct with key condition verification. *Preprint*, arXiv:2405.14092.

Jing Xu, Hengrui Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. 2024. Faithful logical reasoning via symbolic chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, pages 13326–13365. Association for Computational Linguistics.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*. Association for Computational Linguistics.

Shinn Yao, Jeffrey Zhao, Dian Yu, Sherry Zhao, Nathan Wang, Bill Wang, and Denny Zhou. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Shunyu Yao, Shiyu Liang, Jeffrey Zhao, and Karthik Narasimhan. 2023b. Tree of thoughts: Deliberate reasoning via tree search. *arXiv preprint arXiv:2305.10601*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Junyan Zhang, Shuliang Liu, Aiwei Liu, Yubo Gao, Jungang Li, Xiaojie Gu, and Xuming Hu. 2025a. Cohemark: A novel sentence-level watermark for enhanced text quality. *Preprint*, arXiv:2504.17309.

Yu Zhang, Shuliang Liu, Xu Yang, and Xuming Hu. 2025b. Catmark: A context-aware thresholding framework for robust cross-task watermarking in large language models. *arXiv preprint arXiv:2510.02342*.

Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Jaekyeom Kim, Moontae Lee, Honglak Lee, and Lu Wang. 2024. Small language models need strong verifiers to self-correct reasoning. *Preprint*, arXiv:2404.17140.

Jiaxing Zhao, Hongbin Xie, Yuzhen Lei, Xuan Song, Zhuoran Shi, Lianxin Li, Shuangxue Liu, and Haoran Zhang. 2025. Connecting the dots: A chain-of-collaboration prompting framework for llm agents. *Preprint*, arXiv:2505.10936.

Lexin Zhou, Wout Schellaert, Fernando Martínez-Plumed, Yael Moros-Daval, Cèsar Ferri, and José Hernández-Orallo. 2024. Larger and more instructable language models become less reliable. *Nat.*, 634(8032):61–68.

Shunyu Zhou, Surya Pal, Silvio Savarese, and Amir Zamir. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

Dong-Hai Zhu, Yu-Jie Xiong, Jia-Chen Zhang, Xi-Jiong Xie, and Chun-Ming Xia. 2025a. Understanding before reasoning: Enhancing chain-of-thought with iterative summarization pre-prompting. *Preprint*, arXiv:2501.04341.

Xiaochen Zhu, Caiqi Zhang, Tom Stafford, Nigel Collier, and Andreas Vlachos. 2025b. Conformity in large language models. *Preprint*, arXiv:2410.12428.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul F Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. In *Advances in Neural Information Processing Systems*.

A Related Work Details

Chain-of-Thought (CoT) prompting (Wei et al., 2022a) first showed that furnishing step-wise exemplars markedly improves LLM performance on arithmetic and commonsense problems. Follow-up work refines this paradigm: *scratchpads* (Nye et al., 2021), *least-to-most prompting* (Zhou et al., 2022), and *self-consistency* (Wang et al., 2023c). More recently, multi-agent and symbolic reasoning approaches like *multiagent debate* (Du et al., 2024), *symbolic chain-of-thought* (Xu et al., 2024), and neurosymbolic reasoning with logic provers (Olausson et al., 2023) further strengthen factuality and logical consistency.

Tool-Augmented Reasoning Program-Aided LMs (Gao et al., 2022) delegate arithmetic or logical computation to Python. **ReAct** (Yao et al., 2022) interleaves reasoning and external tools. *CRITIC* (Gou et al., 2024) validates intermediate answers with execution tools. *Chain-of-Verification* (Dhuliawala et al., 2024) reduces hallucinations by verifying sub-claims against evidence.

Iterative Self-Refinement and Critique Self-Refine (Madaan et al., 2023) establishes a generate–criticise–rewrite loop. Ambiguity handling and uncertainty decomposition are also explored: *AmbigQA* (Min et al., 2020), *Decomposing Uncertainty* (Hou et al., 2024), and rank-based calibration (Huang et al., 2024).

Datasets for Reasoning and Verification Classic benchmarks like *BoolQ* (Clark et al., 2019), *SQuAD 2.0* (Rajpurkar et al., 2018), *FEVER* (Thorne et al., 2018), and *HotpotQA* (Yang

et al., 2018) stress-test fact verification, ambiguity, and multi-hop reasoning. Meanwhile, multilingual and cross-domain robustness is increasingly crucial for real-world deployment. Recent work such as *MuLA-F* (Li et al., 2025a) highlights the importance of handling multilingual heterogeneity and collaborative anomaly detection across diverse social media environments.

Active Prompting Active-Prompt (Diao et al., 2024) selects informative exemplars to boost performance under budget constraints. Meanwhile, Ling et al. (2025) denotes different prompt coverage leads to various effects.

Faithfulness and Safety Beyond correctness, methods ensure interpretability and reliability, such as *Faithful-CoT* (Lyu et al., 2023), symbolic verification (Xu et al., 2024), and multi-agent debate (Du et al., 2024). Recent research on trustworthy language models has focused on two major directions: mitigating misinformation and developing watermarking techniques for tracing generated content. Liu et al. (Liu et al., 2025a) survey proactive defenses against misinformation in LLMs, proposing a three-pillar framework for improving model reliability. In related representation learning work, Liu et al. (Liu et al., 2022) introduce a hierarchical contrastive approach for unsupervised relation extraction, offering insights into structured semantic modeling. A parallel line of studies explores watermarking for text and multimodal generation. Zhang et al. (Zhang et al., 2025a) propose COHEMARK, a sentence-level watermark that preserves semantic coherence. Liu et al. (Liu et al., 2025b) extend watermarking to vision–language models, while Huo et al. (Huo et al., 2025) introduce a distortion-free semantic watermark with channel constraints. More recently, Zhang et al. (Zhang et al., 2025b) present CATMARK, a context-aware thresholding framework for robust cross-task watermarking.

B Baseline Details

In this section, we run multiple reasoning frameworks that help improve downstream tasks of LLM reasoning. By comparing our proposed pipeline with existing baselines, we can further prove that current frameworks are far from eliciting LLMs' full potential. Detailed templates and examples can be found in Appendix N.

Chain-of-Thought (CoT) The CoT prompting method guides large language models (LLMs) to generate intermediate reasoning steps before producing a final answer. It improves multi-step reasoning performance by making the inference process explicit. This technique is particularly effective in arithmetic and logic-heavy tasks, as shown in prior works such as Wei et al. (2022b).

Few-shot Few-shot prompting involves presenting the model with a few annotated examples in the input to demonstrate the task format. It enables in-context learning without any gradient updates. Despite its simplicity, few-shot prompting serves as a strong baseline in many LLM applications (Brown et al., 2020).

Tree-of-Thought (ToT) Tree-of-Thought extends the CoT framework by exploring multiple reasoning paths in a tree structure and evaluating intermediate steps. It mimics a planning process that considers alternative possibilities, significantly enhancing reasoning robustness and allowing backtracking or reevaluation of incorrect paths (Yao et al., 2023a).

Plan & Solve Plan-and-Solve separates the reasoning process into two stages: first planning a high-level outline or decomposition of the task, then solving sub-tasks based on that plan. This two-stage strategy enhances controllability a in reasoning, and has shown strong performance across complex benchmarks (Wang et al., 2023a).

Supervised Fine-Tuning (SFT) Supervised Fine-Tuning (SFT) is the standard method for aligning LLMs by training on curated human-annotated question-answer pairs. Unlike prompt-only methods, SFT optimizes the model weights directly through gradient updates to produce desirable outputs. It forms the foundation of instruction-tuned models like InstructGPT and serves as a widely-used supervised baseline for alignment tasks (Ouyang et al., 2022). The common consensus is that SFT provides the essential starting point for alignment, but by itself is insufficient to fully capture complex human preferences, often requiring further preference optimization or reinforcement learning. The optimized formula is below:

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\sum_{i=1}^{N} \log \pi_\theta(y_i \mid x_i),$$

Direct Preference Optimization (DPO) Direct Preference Optimization (DPO) is a reinforcement learning-free approach that fine-tunes LLMs using

preference data. Given pairs of outputs (e.g., preferred vs. dispreferred), DPO directly optimizes the log-likelihood ratio between them, providing alignment benefits similar to reinforcement learning methods like PPO while avoiding their instability and complexity (Rafailov et al., 2023). The common consensus is that DPO offers a simpler and more stable alternative to reinforcement learning-based methods, making it a practical choice for preference alignment, though it may still struggle with capturing nuanced or long-term user preferences. The formula can be concluded as below:

$$\mathcal{L}_{\mathrm{DPO}}(\theta) = -\mathbb{E}_{(x,y^+,y^-)}\Big[\log\sigma\Big(\beta\big(\log\frac{\pi_\theta(y^+|x)}{\pi_{\mathrm{ref}}(y^+|x)} - \log\frac{\pi_\theta(y^-|x)}{\pi_{\mathrm{ref}}(y^-|x)}\big)\Big)\Big]$$

Self-Reflection Self-reflection has been widely adopted as a strategy for improving LLM reasoning by prompting the model to critique and refine its own initial answer. However, this paradigm implicitly assumes that the model can (i) detect flaws in its original reasoning, (ii) identify missing or contradictory evidence, and (iii) reliably revise its conclusion. Prior work shows that these assumptions rarely hold in practice: when the initial reasoning is misguided, the reflection stage often reproduces the same errors, leading to self-confirming loops rather than genuine correction. In addition, self-reflection primarily operates on the model's *final* answer, without addressing perception-level uncertainty or the model's tendency to prematurely output **Unknown**.

B.1 Experiment Setup Details

We choose four LLMs: GPT-4o, GPT-3.5, LLaMA-3.1-8B and Qwen-2.5-7B for our detailed baseline experiments. We categorize our baseline methods into two kinds: training and non-training, in order to see whether the *Vague Perception* phenomenon can be solved with or without parameter revision. For model training and open-source model deployment, the experiments ran on four Tesla V100-SXM2-32GB GPUs.

B.2 Training Methods Details

As explained before in Appendix B.1, we elaborate our facilities and setup for training. In this section, we demonstrate our details for the training.

For SFT (Supervised Fine-Tuning), we extract 1,000 training samples from exactly same dataset, the prompt instruction can be found in Appendix N.

For DPO (Direct Preference Optimization), we apply LLM Judge to help select the reasoning process with many atomic steps as possible. This includes two parts: (1) Prompt Template that includes task guiding details (e.g., definition of atom steps) and task instructions (2) Testing cases that can perfectly prove and verify which response is better given the metrics and requirements. All information above can be found in Appendix N.

C Datasets Details

FLD Following Morishita et al. (2024), **FLD** is a fully synthetic benchmark for multi-step logical inference. Formal proof chains of depth 0–15 are automatically generated and then verbalized into natural-language premises plus a hypothesis. Each instance is labeled **Proved**, **Disproved**, or **Unknown**, making the dataset ideal for assessing whether a model can (i) draw correct conclusions and (ii) abstain when evidence is insufficient.

FOLIO **FOLIO** (Han et al., 2022) contains **1,430** human-written first-order-logic stories that bridge everyday commonsense with formal theorem proving. Every narrative provides its gold logical forms and a hypothesis whose truth value (**True**, **False**, or **Unknown**) is automatically certified. The official split is 1 001/226/203 for train/dev/test, offering fewer but richer problems than FLD and requiring explicit refusals when the passage leaves the hypothesis undecided.

ScienceQA **ScienceQA** (Lu et al., 2022) is a multi-modal multiple-choice benchmark with **21,208** elementary- and middle-school questions across 26 topics. To create an abstention setting comparable to the logic corpora, we select questions from Physics and Biology (science) as well as Writing-Strategies and Figurative-Language (arts). For 50% of the chosen items we delete two random supporting sentences with GPT-4o, then manually verify that no unique answer remains; these items are re-labeled **Unknown**.

Fields/Domains Analysis Our comprehensive dataset consists of four distinct collections, each representing different domains of logical reasoning. The **FLD (First-Order Logic Dataset)** contains 600 samples across 11 topics, with Philosophy being the dominant subject (545 samples, 90.8%), followed by Logic (37 samples, 6.2%), and 9 other topics with minimal representation. The **FOLIO dataset** comprises 640 samples distributed across

Model	Methods	Accuracy (%)			
		FLD	FOLIO	Science	Arts
GPT-3.5	Direct	50.17	40.63	44.63	29.25
	CoT	49.67	42.34	44.13	31.00
	Few-shot	49.67	37.50	43.75	33.88
	Tree-of-Thought	33.50	43.91	36.00	28.25
	Plan & Solve	42.83	45.47	33.63	26.38
	Self-Reflection	56.67	42.50	42.00	34.00
	WAKENLLM(ours)	**60.26**	**46.68**	**48.38**	**35.04**
GPT-4o	Direct	51.00	69.53	43.00	33.88
	CoT	54.10	69.53	43.12	31.87
	Few-shot	58.30	74.40	47.62	**50.50**
	Tree-of-Thought	52.83	70.62	43.38	29.25
	Plan & Solve	52.17	69.84	42.38	30.25
	Self-Reflection	55.33	69.37	37.50	34.50
	WAKENLLM(ours)	**71.79**	**75.74**	**56.25**	44.64
LLaMA-3.1-8B	Direct	39.17	51.56	49.50	47.50
	CoT	40.67	51.88	46.50	43.25
	Few-shot	33.33	51.25	48.25	39.13
	Tree-of-Thought	38.33	47.66	40.63	30.00
	Plan & Solve	35.50	52.81	38.75	26.75
	Self-Reflection	51.33	51.88	47.25	44.00
	SFT	49.17	50.47	46.50	49.12
	DPO	50.33	48.60	48.50	49.62
	WAKENLLM(ours)	**61.50**	**65.47**	**62.72**	**53.43**
Qwen-2.5-7B	Direct	41.83	57.81	44.12	47.12
	CoT	43.83	60.62	42.75	46.62
	Few-shot	44.50	58.75	38.00	42.25
	Tree-of-Thought	39.83	57.19	37.88	45.88
	Plan & Solve	40.67	56.88	37.62	39.62
	Self-Reflection	48.67	63.75	39.50	39.00
	SFT	48.67	57.34	**49.62**	43.12
	DPO	**49.33**	54.21	48.38	44.13
	WAKENLLM(ours)	48.06	**65.55**	47.56	**54.37**

Table 6: Accuracy comparison of multiple baselines for four LLMs evaluated on four datasets. Each block represents a distinct model series. The highest accuracy is marked black.

7 topics, including Arts (146 samples, 22.8%), Science & Technology (130 samples, 20.3%), Sociology (114 samples, 17.8%), Logic (102 samples, 15.9%), Sports (85 samples, 13.3%), Geography (39 samples, 6.1%), and Architecture (24 samples, 3.8%). The **ScienceQA dataset** contains 800 samples evenly split between Biology and Physics (400 samples each, 50%). Finally, the **Arts dataset** includes 800 samples with Literature as the primary topic (526 samples, 65.8%), followed by Education (190 samples, 23.8%), Philosophy (77 samples, 9.6%), Linguistics (4 samples, 0.5%), and Psychology (3 samples, 0.4%).

The dataset distribution reveals a hierarchical structure: at the top level, we distinguish between **Fact-based reasoning** (43.66%) and **Story-based reasoning** (56.34%). The four major categories are distributed as follows: FLD (21.13%), FOLIO (22.53%), Science (28.17%), and Arts (28.17%). This balanced distribution ensures comprehensive coverage of different reasoning paradigms, from formal logical reasoning in FLD to narrative-based reasoning in Arts, providing a robust foundation for training and evaluating logical reasoning capabilities across diverse domains.

Unknown Label Annotation For the first **FLD** and **FOLIO**, the *Unknown* label is originally included in their annotations. For SCIENCEQA, we randomly select half of the samples and employ GPT-4o to randomly delete two sentences from the given facts. We then re-label these modified samples as *Unknown*. Additionally, we perform **manual verification** to ensure that relabeled samples are semantically consistent with assigned labels, which are objectively unverifiable.

Table 7: Detailed listing of all metrics used among different experimental settings in our work. Metrics within different settings or systems are separated by dashed lines. RP stands for reasoning process included within the setting. $f \in \{s, u\}$ indicates whether the sample types belong to *solvable* or *unsolvable*. $n \in \{1, 2\}$ indicates the stage of processing. **Omitting subscripts indicates all sample types; omitting superscripts indicates all two stage considered.** Detailed calculations can be found in Appendix F.

Vanilla Settings	Definition	Rationality/Motivation
True Converting (TC)	TC_t^n represents samples converted to **correct** labels.	Measuring LLMs' true potential under stimulation.
True Converting Rate (TCR)	TCR_t^n is the **percentage** of TC_t^n.	Calculate the percentage of LLMs' potential of solving problems after stimulation.
False Converting (FC)	FC_t^n represents samples converted to **incorrect** labels.	Under stimulation, problems LLMs are stimulated to produce wrong answers instead of giving up.
False Converting Rate (FCR)	FCR_t^n is the **percentage** of FC_t^n.	Research the inclination of LLMs outputting wrong answer over **abstention** (answering incorrectly).
Unknown Converting (UC)	UC_t^n represents samples converted to **unknown** samples.	To confirm that the samples are too hard for LLMs to reason.
Unknown Converting Rate (UCR)	UCR_t^n is the **percentage** of UC_t^n.	Research the inclination of LLMs **abstention** (answering unknown) over outputting wrong answer.
Overall Converting Rate (OCR)	After two stage stimulation, the percentage of *Vague Perception* samples are converted to TC, sum of TCR^1 and TCR^2.	Potential part of samples LLMs can reason correctly, but did wrong because of **abstention**.
Remind-then Guide (Label&RP)	**Definition**	**Rationality/Motivation**
Label Conformity	LLMs exhibit distinct tendency of conformity on solvable and unsolvable labels, denoted as $Conf_t$.	Research how conformity biases are enlarged facing different label forms.
Correct Guiding Rate (CGR)	Taking the previous reasoning trace as input, to determine whether performance is better or worse with reflection.	LLM training based on RL is result-oriented, leaving the reasoning trace flawed and biased.
Reasoning Process Coherence (RPC)	Based on original output, LLMs should rethink the problem, this is to define whether the reasoning trace is consistent.	As the stability of reasoning is important, ensuring the consistency of reasoning can be important.
Other Experiments	**Definition**	**Rationality/Motivation**
Degradation (Deg)	In stage 2 stimulation, the input FC^2 may be degraded and output unknown, which should be denoted as UC^2.	Consideration for randomness and instability of LLMs output. Better make up for the pipeline.

Search-Based Planning Frameworks Instead of generating a single chain, **Tree-of-Thought (ToT)** views partial solutions as nodes in a tree and performs breadth-first search with value heuristics (Yao et al., 2023b). **Decomposed Prompting** turns a complex task into callable sub-prompts that can be solved independently (Khot et al., 2023).

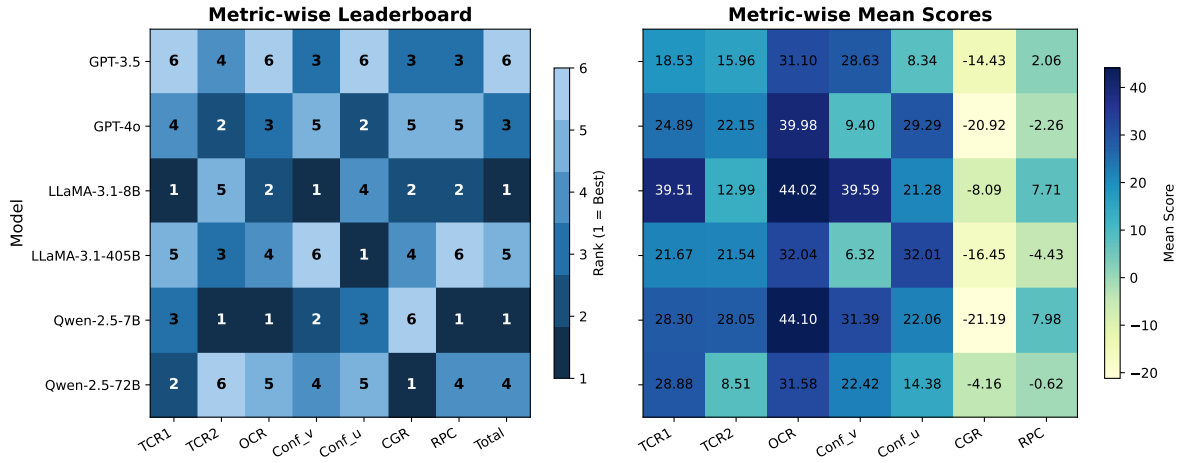


Figure 6: Leaderboard ranking (left) and average experimental results across datasets (right) in Figure 3.

D Metrics Details

As shown in Table 7, we elaborate all used metrics in our experiment, along with the motivation of designing them. A solid and comprehensive evaluation system offers contributions to LLM reasoning.

E LLM Usage Statement

GPT-4o, Claude-3-7 and Grok 3 are used for code organizing and paper framing.

F Formula Details

In the main body of paper, we highly focus on True Converting (TC) flow that LLMs are able to convert *Vague Perception* samples to predict correct labels. In this section we explain False Converting (FC) and Unknown Converting (UC) in detail. As we define the whole evaluation system, we also map out their relationships.

F.1 False and Unknown Converting

As demonstrated in 3.2, we denote the notion of **TCR**. Likewise, we define the concepts regarding false converting and unknown converting similarly:

$$FCR_t^n = \frac{|FC_t^n|}{|D_{\text{VP}}^n|}, \quad UCR_t^n = \frac{|UC_t^n|}{|D_{\text{VP}}^n|}$$

Note that, Unknown Converting (UC) is still categorized into *Vague Perception* phenomenon: Outputting unknown when facing solvable samples. We can tell the following relationship for any stage that n can be 1 or 2:

$$UCR_t^n + FCR_t^n + TCR_t^n = \mathbb{1}$$

$$UC_t^n \cup FC_t^n \cup TC_t^n = D_{VP}^n$$

$$UC_t^n \cap FC_t^n \cap TC_t^n = \emptyset$$

F.2 Overall Converting & Latent Ability

We clarify the relationship between Overall Converting Rate (OCR) in Section 3.2 with the latent ability in 5.5, as we depicted, by comparing True Converting (TC) with D_{VP}, we can get the ability of how many samples can be eventually converted to right labels within all *Vague Perception* samples, whereas the latter indicates overall accuracy gain among the all samples D in a broader aspect.

F.3 Remind-then-Guide Label Conformity

In this section, we talk about the formula of conformity in detail. We define one single-turn conformity as below:

$$Conf_t^n = TCR_t^n - TCR_t^{n'}$$

During the experiments, we left out the influence of stage-wise performances, and focus on the true converting rate (TCR) changes separately on solvable and unsolvable samples, acquired with and without assigned labels. As the conformity is always considered to be a model attribute rather than a random phenomenon, we talk about the conformity regarding sample types, we pay less attention to the stage-wise analysis. By averaging the two-stage conformity, the formula in our experiments can be decomposed as:

$$\begin{aligned} Conf_t &= \frac{1}{2}(TCR_t^1 - TCR_t^{1'}) + \frac{1}{2}(TCR_t^2 - TCR_t^{2'}) \\ &= \frac{1}{2}\frac{|TC_t^1| - |TC_t^{1\prime}|}{|D_{VP}^1|} + \frac{1}{2}\frac{|TC_t^2| - |TC_t^{2\prime}|}{|D_{VP}^2|} \end{aligned}$$

G Leaderboard Analysis

As illustrated in Figure 6, the conformity towards different label formats vary considerably across

models. Moreover, the quality of reasoning traces does not necessarily correlate with the final accuracy. This observation implies that LLMs can often produce correct answers despite flawed or shallow reasoning processes. We hypothesize that this phenomenon stems from the reinforcement learning (RL)-based training paradigm, which primarily optimizes for outcome-level rewards rather than reasoning quality. As a result, models tend to acquire "good answers with bad reasoning"—a form of outcome–process misalignment that contradicts the long-standing cognitive principle that good thinking leads to good answers (Turpin et al., 2023; Wang et al., 2024; Ziegler et al., 2019; Christiano et al., 2017).

Parameters Analysis According to the results, we can tell that except for GPT-3.5 (OpenAI, 2022) being an outdated model, for Qwen (Academy, 2024) and LLaMA (Touvron and Team, 2025) series, models with smaller parameters sizes tend to perform better. The takeaway aligns well with our findings in Figure 5 and Section 5.7, as higher the parameters, the more likely LLMs are to be cautious, whereas models with smaller parameters tend to be more instable and influenced by the experimental settings, this trade-off can be of guidance for models selection for different requirements.

H Detailed RtG Conformity Analysis

In this section, we dive deeper into the conformity effects under four different misguiding rate (M=0%, 50%, 66.67% and 100%): the percentage of input labels are replaced with the wrong ones. Detailed results are shown in Table 9, 10, 11. Generally, all models demonstrate stronger conformity effects regarding higher misguiding rate.

Examining Table 9 reveals that misguidance has a pronounced influence on both stages of converting. For **GPT-3.5**, the FLD dataset exhibits a TCR^1 of only 5.83% at a 100% misguiding rate, but this climbs to 54.17% when M drops to zero. Similar trends appear on FOLIO and Arts, where reducing M from 100% to 0% increases TCR^1 from 2.70% to 18.92% and from 22.92% to 37.50% respectively. **GPT-4o** exhibits slightly higher tolerance to misguidance: on FLD its TCR^1 rises from 15.27% at $M = 100\%$ to 26.22% at $M = 0\%$, and on Science from 8.42% to 20.26%. However, even for GPT-4o, increasing misguidance still reduces True Converting Rate (TCR) largely, illustrating the general monotonic relationship between misguiding rate and conformity.

The RtG analysis for **LLaMA-3.1** series models in Table 10 shows a striking gap in robustness. On the FLD dataset, the 8B model's TCR^1 climbs from 0.36% at $M = 100\%$ to 75.18% when there is no misguidance, whereas the 405B variant rises only from 1.90% to 17.66% Similar improvements for the 8B model across FOLIO and Science confirm that smaller models can recover significantly more reasoning potential when labels are correctly provided. By contrast, the 405B model shows modest gains and remains low on Arts, suggesting that larger models are more vulnerable to misguidance.

Table 11 compares **Qwen-2.5-7B** and **Qwen-2.5-72B**. For FLD and FOLIO, the 72B model performances better with less misguidance than the 7B model, with TCR^1 increasing from 13.25% to 77.71% on FLD and from 19.18% to 42.47% on FOLIO (Han et al., 2022). Yet on Science and Arts, the 7B model attains higher TCR^1 at $M = 0\%$, reaching 50.83% on Science and 55.93% on Arts versus 37.84% and 24.22% for the 72B variant. These observations highlight that RtG counts on not only on models size but also on dataset fields.

I Detailed TCR Analysis

Our TCR (True Converting Rate) analysis via different stages can offer valuable insights on model selections in multi-turn frameworks, as multi-agent and multi-dialogue workflows are more and more widely used now. Table 12 decomposes TCR^1 and TCR^2 into stages regarding (V) and unsolvable (U) samples. **GPT-3.5** exhibits moderate first stage converting on solvable samples (e.g., 12.61% on FLD and 14.41% on FOLIO) but few converting rate on unsolvable examples (TCR^1_u). Its second stage, however, can salvage many unverifiable cases on FOLIO, yielding a TCR^2_u of 37.89%. **GPT-4o** delivers stronger TCR^1 on solvable samples in story-based datasets like FOLIO (33.73%) and shows significant second stage gains on unsolvable samples (44.03% on FOLIO and 26.87% on FLD). This suggests that GPT-4o is more suited for workflows requiring extended dialogues to resolve ambiguous inputs.

LLaMA-3.1-8B stands out with consistently high TCR^1_s across all four datasets, reaching 44.20% on FOLIO and 44.98% on Science. In contrast, the 405B model counterpart attains lower TCR^1_s and relies on stage 2 converting to correct the mistakes; for instance, on FLD it has TCR^1_s

Algorithm 1: Vanilla Setting Pipeline of the WAKENLLM Pipeline.

Require: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ with labels $y_i \in \{\mathrm{T}, \mathrm{F}, \mathrm{U}\}$;
base LLM model $\mathcal{M}$; two stimulation prompts P_1 and P_2
Ensure: Stage-1 categories $\mathrm{TC}^1, \mathrm{UC}^1, \mathrm{FC}^1$ and Stage-2 categories $\mathrm{TC}^2, \mathrm{FC}^2$

$\mathrm{D_{VP}} \leftarrow \emptyset$ // set of Vague Perception samples
for all (x_i, y_i) in $\mathcal{D}$ **do**
  $\hat{y}_i \leftarrow \mathcal{M}(x_i)$ // first-pass prediction
  if ($y_i \in \{\mathrm{T}, \mathrm{F}\}$ and $\hat{y}_i = \mathrm{U}$) or ($y_i = \mathrm{U}$ and $\hat{y}_i = \mathrm{U}$ and model lacks reasoning) **then**
    $\mathrm{D_{VP}} \leftarrow \mathrm{D_{VP}} \cup \{(x_i, y_i)\}$ // collect Vague Perception samples
  end if
end for
$\mathrm{TC}^1, \mathrm{UC}^1, \mathrm{FC}^1 \leftarrow \emptyset$ // Stage-1 categories
for all (x_i, y_i) in $\mathrm{D_{VP}}$ **do**
  $\hat{y}_i^{(1)} \leftarrow \mathcal{M}(\text{stimulation}(x_i, P_1))$ // first-stage stimulation
  if $\hat{y}_i^{(1)} = y_i$ **then**
    $\mathrm{TC}^1 \leftarrow \mathrm{TC}^1 \cup \{(x_i, y_i)\}$ // True Converting at stage 1
  else if $\hat{y}_i^{(1)} = \mathrm{U}$ **then**
    $\mathrm{UC}^1 \leftarrow \mathrm{UC}^1 \cup \{(x_i, y_i)\}$ // Unknown Converting at stage 1
  else
    $\mathrm{FC}^1 \leftarrow \mathrm{FC}^1 \cup \{(x_i, y_i)\}$ // False Converting at stage 1
  end if
end for
$\mathrm{TC}^2, \mathrm{FC}^2 \leftarrow \emptyset$ // Stage-2 categories
for all (x_i, y_i) in $\mathrm{FC}^{(1)}$ **do**
  $\hat{y}_i^2 \leftarrow \mathcal{M}(\text{stimulation}(x_i, P_2))$ // second-stage stimulation
  if $\hat{y}_i^2 = y_i$ **then**
    $\mathrm{TC}^2 \leftarrow \mathrm{TC}^2 \cup \{(x_i, y_i)\}$ // Corrected at stage 2
  else
    $\mathrm{FC}^2 \leftarrow \mathrm{FC}^2 \cup \{(x_i, y_i)\}$ // Still incorrect at stage 2
  end if
end for
return $\mathrm{TC}^1, \mathrm{UC}^1, \mathrm{FC}^1, \mathrm{TC}^2, \mathrm{FC}^2$

Model	FLD		FOLIO		Science		Arts	
	V	U	V	U	V	U	V	U
GPT-3.5	72.00	8.00	10.00	24.69	40.00	2.75	22.50	1.50
GPT-4o	11.00	88.00	7.81	11.25	68.75	22.75	28.25	60.75
LLaMA-3.1 (8B)	76.33	15.00	54.69	1.88	49.25	3.00	46.00	4.50
LLaMA-3.1 (405B)	22.67	100.00	20.94	99.69	50.00	89.25	9.00	99.00
Qwen-2.5 (7B)	20.67	9.00	10.00	13.75	60.00	0.00	52.50	6.50
Qwen-2.5 (72B)	49.67	5.67	15.31	7.50	54.75	10.00	27.75	36.25

Table 8: The percentage of LLMs demonstrating the *Vague Perception* phenomenon among six LLMs. Split by two different root causes: (1) V: Solvable samples with *T/F* labels, but the model output *Unknown*. (2) U: Unsolvable Samples with *Unknown* label, but the model output *Unknown* simply because of incapacity rather than fully understand this is objectively unsolvable.

of only 5.43% but TCR_u^2 of 53.73%. This implies that smaller LLaMA models can resolve most solvable queries in a single turn, whereas the 405B model might need deeper chains of reasoning.

Among **Qwen** series, the 7B model achieves the highest TCR_s^1 on FLD (47.19%) and Science (15.83%), but its stage 2 converting occur mostly on unsolvable samples; for example, TCR_u^2 reaches 40.00% on FLD and 35.09% on FOLIO. The 72B model exhibits a more balanced behavior, with moderate TCR_s^1 and lower TCR_u^2, indicating that it resolves solvable samples earlier and requires fewer subsequent turns. Taken together, these findings underscore the importance of considering both stages: models with high TCR_s^1 are advantageous for straightforward, solvable samples, whereas those with substantial TCR_u^2 may better handle ambiguous or mislabeled inputs in

multi-turn dialogues.

J Vague Perception Details

J.1 Evaluations

Detailed distribution of the *Vague Perception* samples is shown in Table 8. We can tell even though it is prevalent across different datasets and models, there is a general inclination. Among all extracted samples, models will larger parameters are more likely to output *Unknown*, falling into the second part. With a smaller parameter it is very likely to be the opposite. For Qwen series, the distribution can be relatively stable across two models, however, LLaMA-3.1 and GPT series both demonstrate stronger differences within the same model family.

J.2 Causes

Even though MoE structure models like GPT-4o demonstrates superior generalization ability across different fields, however, the generalization ability across different presentation styles is still unstable.

K Terminology Clarification

Abstention behavior is widely researched, such reasoning biases combined with knowledge boundary is rather important to offer trustworthy and unbiased outputs. We further elaborate the introduced the term *Vague Perception* to distinguish the differences between existing concepts.

Abstention Abstention typically indicates LLMs themselves know there are malicious, poison elements within the prompt and refuse to answer.

Knowledge Boundary Knowledge Boundary means LLMs know when to answer, and when not to, to prevent hallucination and promote abstention when being asked something beyond the ability.

Hallucinations Hallucination is about including non-exist facts or reasoning in the output, the paradigm of how this is induced is still controversial, but it is mainly because of detailed prompt asking and multi-turn dialogue.

Vague Perception (Ours) Given an *unknown* option, LLMs are able to answer correctly. However, they are potentially inclined to output *unknown*, even if the problem can be solved within the their reasoning ability.

L Important Findings Wrap-up

L.1 Unknown Label, Stronger Conformity

Previous conformity research mainly focuses on multi-turn dialogue workflows (Zhu et al., 2025b). However, containing uncertainty elements is a fundamental part of conformity. We mainly consider this is because of imbalanced training data source that leads to relatively vulnerable resistance against misleading.

L.2 Correct Prediction, Flawed Reasoning

As experiments shown, CGR and RPC results are mainly negative, we can conclude that most LLMs predicted correct labels with flawed reasoning processes. We attribute this to the essence of reinforcement learning (RL) algorithms being a result-oriented methodology, the algorithm only cares about the final prediction, rather than intermediate processes, casting potential risks in trustworthiness.

L.3 Bigger Parameter Sizes, More Cautious

Our comparisons show that for both GPT, LLaMA and Qwen series, LLMs with higher parameters are inclined to output *unknown*. Similar to Degradation analysis in Section 5.7, this is a unified conclusion obtained in multiple experiments. Opposite to the findings in (Weng et al., 2025), where models are stable and more resistant against misguiding, as we include the *Unknown* option within the prompt, models with bigger parameter sizes like to conform and stick to this label, we attribute this to *Over-Cautious*. This is exactly opposite to the findings in (Zhou et al., 2024), casting unstable and random biases exhibiting under various scenarios.

L.4 LLMs Selection in Multi-stage Workflows

As addressed in Section 5.1 and Appendix I, LLaMA series demonstrate superior performances at early stages, it is recommended to be placed at front parts. However, for GPT-4o and Qwen series, the sequence of placing is highly dependent on the dataset forms, they exhibit distinct yet related behaviors between *Story-based* and *Fact-based* datasets. For example, GPT-4o demonstrates better stage 1 performance in *Fact-based* datasets, so we suggest putting it prior to other models when dealing with samples with atomic statements, and the opposite dealing with samples containing long passages; Whereas Qwen series show relatively balanced performance in *Story-based* datasets, and superior ability at stage 1 in *Fact-based* datasets.

Our experiments and strategies are purely based on sequential workflows, and parallel workflows are not tested or used in our work.

M Pseudo-Code Pipelines Visualization

As shown in Algorithm 1, 2 and 3, the pipeline of different experimental settings within our pipeline is demonstrated below. Differences from **Vanilla settings** are marked orange.

Algorithm 2: Pipeline of the Remind-then-Guide (RtG) Setting.

Require: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ with labels $y_i \in \{\mathrm{T}, \mathrm{F}, \mathrm{U}\}$;
base LLM model $\mathcal{M}$; two stimulation prompts P_1 and P_2
Ensure: Stage-1 categories $\mathrm{TC}^1, \mathrm{UC}^1, \mathrm{FC}^1$ and Stage-2 categories $\mathrm{TC}^2, \mathrm{FC}^2$; reasoning store $\mathcal{R}$

$\mathrm{D_{VP}} \leftarrow \emptyset$ // set of Vague Perception samples
for all (x_i, y_i) in $\mathcal{D}$ **do**
  $\hat{y}_i \leftarrow \mathcal{M}(x_i)$ // first-pass prediction
  if ($y_i \in \{\mathrm{T}, \mathrm{F}\}$ and $\hat{y}_i = \mathrm{U}$) or ($y_i = \mathrm{U}$ and $\hat{y}_i = \mathrm{U}$ and model lacks reasoning) **then**
    $\mathrm{D_{VP}} \leftarrow \mathrm{D_{VP}} \cup \{(x_i, y_i)\}$ // collect Vague Perception samples
  end if
end for
$\mathrm{TC}^1, \mathrm{UC}^1, \mathrm{FC}^1 \leftarrow \emptyset$ // Stage-1 categories
$\mathcal{R} \leftarrow \{\}$ // dictionary to store reasoning processes
for all (x_i, y_i) in $\mathrm{D_{VP}}$ **do**
  $(\hat{y}_i^{(1)}, r_i) \leftarrow \mathcal{M}(\text{stimulation}(x_i, P_1))$ // first-stage stimulation returns both predicted label and reasoning process r_i
  $\mathcal{R}[i] \leftarrow r_i$ // store reasoning process for sample i
  if $\hat{y}_i^{(1)} = y_i$ **then**
    $\mathrm{TC}^1 \leftarrow \mathrm{TC}^1 \cup \{(x_i, y_i)\}$ // True Converting at stage 1
  else if $\hat{y}_i^{(1)} = \mathrm{U}$ **then**
    $\mathrm{UC}^1 \leftarrow \mathrm{UC}^1 \cup \{(x_i, y_i)\}$ // Unknown Converting at stage 1
  else
    $\mathrm{FC}^1 \leftarrow \mathrm{FC}^1 \cup \{(x_i, y_i)\}$ // False Converting at stage 1
  end if
end for
$\mathrm{TC}^2, \mathrm{FC}^2 \leftarrow \emptyset$ // Stage-2 categories
for all (x_i, y_i) in FC^1 **do**
  $\hat{y}_i^{(2)} \leftarrow \mathcal{M}(\text{stimulation}(x_i, P_2, \mathcal{R}[i]))$ // second-stage stimulation: input the FC^1 along with its previous reasoning r_i
  if $\hat{y}_i^{(2)} = y_i$ **then**
    $\mathrm{TC}^2 \leftarrow \mathrm{TC}^2 \cup \{(x_i, y_i)\}$ // Corrected at stage 2
  else
    $\mathrm{FC}^2 \leftarrow \mathrm{FC}^2 \cup \{(x_i, y_i)\}$ // Still incorrect at stage 2
  end if
end for
return $\mathrm{TC}^1, \mathrm{UC}^1, \mathrm{FC}^1, \mathrm{TC}^2, \mathrm{FC}^2, \mathcal{R}$

Algorithm 3: Ablation Study of the Remind-then-Guide (RtG) Setting.

Require: Vague Perception set $\mathrm{D_{VP}}$ and Stage-1 outputs $\{(\hat{y}_i^{(1)}, r_i^{(1)})\}_{i \in \mathrm{D_{VP}}}$ obtained by Algorithm 1
Ensure: Ablated Stage-1 converting statistics over T rounds

// Pre-processing, Vague Perception extraction and the first Stage-1 stimulation
// follow Algorithm 1 and are omitted here.
$\mathcal{R} \leftarrow \{\, i \mapsto r_i^{(1)} \mid (x_i, y_i) \in \mathrm{D_{VP}} \,\}$ // store Stage-1 reasoning processes for all samples
for $t = 2$ to T **do**
  $\mathrm{TC}^{1,(t)}, \mathrm{UC}^{1,(t)}, \mathrm{FC}^{1,(t)} \leftarrow \emptyset$ // Stage-1 categories in round t
  for all (x_i, y_i) in $\mathrm{D_{VP}}$ **do**
    $(\hat{y}_i^{(t)}, r_i^{(t)}) \leftarrow \mathcal{M}(\text{stimulation}(x_i, P_1, \mathcal{R}[i]))$ // re-use Stage-1 prompt but remind with previous reasoning $\mathcal{R}[i]$
    if $\hat{y}_i^{(t)} = y_i$ **then**
      $\mathrm{TC}^{1,(t)} \leftarrow \mathrm{TC}^{1,(t)} \cup \{(x_i, y_i)\}$
    else if $\hat{y}_i^{(t)} = \mathrm{U}$ **then**
      $\mathrm{UC}^{1,(t)} \leftarrow \mathrm{UC}^{1,(t)} \cup \{(x_i, y_i)\}$
    else
      $\mathrm{FC}^{1,(t)} \leftarrow \mathrm{FC}^{1,(t)} \cup \{(x_i, y_i)\}$
    end if
    $\mathcal{R}[i] \leftarrow r_i^{(t)}$ // update stored reasoning to the latest round
  end for
  Compute ablated Stage-1 metrics (e.g., $\mathrm{TCR}_1^{(t)}$) based on $\mathrm{TC}^{1,(t)}, \mathrm{UC}^{1,(t)}, \mathrm{FC}^{1,(t)}$
end for
return $\{\mathrm{TC}^{1,(t)}, \mathrm{UC}^{1,(t)}, \mathrm{FC}^{1,(t)}\}_{t=1}^{T}, \mathcal{R}$

Table 9: Detailed Remind-then-Guide (RtG) setting of GPT-3.5 and GPT-4o. We break down TCR^1 into solvable and unsolvable samples: TCR^1_s and TCR^1_u. M: Misguiding Rate

Model	Dataset	M (%)	TCR^1 (%)		TCR^2 (%)	
			S	U	S	U
GPT-3.5	FLD	100.00	5.83	3.33	1.67	0.00
		66.67	20.83	6.67	16.67	2.78
		50.00	27.50	6.00	40.54	10.81
		0.00	54.17	10.00	81.82	0.00
	FOLIO	100.00	2.70	0.90	0.00	0.00
		66.67	6.31	10.81	3.80	12.66
		50.00	9.91	22.52	3.23	20.97
		0.00	18.92	38.74	17.39	26.09
	Science	100.00	19.88	0.00	3.57	7.00
		66.67	31.58	1.17	3.70	7.00
		50.00	30.41	1.75	7.02	11.00
		0.00	43.86	2.92	18.03	8.00
	Arts	100.00	22.92	0.00	0.00	15.00
		66.67	21.88	1.04	0.00	6.00
		50.00	34.38	3.00	9.38	6.00
		0.00	37.50	4.17	13.89	3.00
GPT-4o	FLD	100.00	15.27	10.09	0.39	4.65
		66.67	19.60	12.39	13.74	4.09
		50.00	21.04	17.00	2.33	34.88
		0.00	26.22	27.09	5.81	62.40
	FOLIO	100.00	5.85	6.73	0.00	6.06
		66.67	11.11	21.05	1.14	25.38
		50.00	13.45	32.75	4.17	37.12
		0.00	20.18	54.09	4.92	62.50
	Science	100.00	8.42	5.79	1.54	25.00
		66.67	11.84	7.37	4.62	28.00
		50.00	14.21	7.37	6.15	24.00
		0.00	20.26	8.95	13.08	32.00
	Arts	100.00	11.83	4.51	1.06	24.00
		66.67	14.37	7.32	4.58	38.00
		50.00	14.65	10.70	5.99	45.00
		0.00	17.75	15.21	11.27	60.00

Table 10: Detailed Remind-then-Guide (RtG) setting of LLaMA-3.1-8B and LLaMA-3.1-405B. We break down TCR^1 into solvable and unsolvable samples: TCR^1_s and TCR^1_u. "/" denotes there is no input for stage 2, as false converted samples (FC^1) from stage 1 don't exist. M: Misguiding Rate.

Model	Dataset	M (%)	TCR^1 (%)		TCR^2 (%)	
			S	U	S	U
LLaMA3.1 (8B)	FLD	100.00	0.36	2.92	0.00	2.36
		66.67	21.90	6.57	22.58	16.13
		50.00	39.05	11.00	31.25	10.94
		0.00	75.18	16.42	/	/
	FOLIO	100.00	2.76	0.55	1.00	0.00
		66.67	26.52	0.55	25.45	1.82
		50.00	41.44	0.00	39.13	6.52
		0.00	81.22	2.76	0.00	25.00
	Science	100.00	0.96	1.44	5.10	0.00
		66.67	28.23	1.44	22.81	7.00
		50.00	37.32	3.83	45.65	2.00
		0.00	83.25	4.31	0.00	100.00
	Arts	100.00	22.77	1.49	14.61	3.00
		66.67	33.17	2.97	29.41	7.00
		50.00	42.57	3.96	34.72	6.00
		0.00	60.89	4.95	38.46	18.00
LLaMA3.1 (405B)	FLD	100.00	1.90	0.82	0.00	0.31
		66.67	6.52	16.30	5.22	30.22
		50.00	8.42	28.00	5.41	35.14
		0.00	17.66	54.62	2.94	61.76
	FOLIO	100.00	4.40	18.91	1.00	7.91
		66.67	8.29	28.50	2.62	17.03
		50.00	8.55	33.94	2.43	21.36
		0.00	14.51	48.19	2.82	24.65
	Science	100.00	7.72	15.00	2.27	35.00
		66.67	13.64	21.54	3.25	37.00
		50.00	16.70	25.31	4.06	42.00
		0.00	25.13	38.78	2.89	48.00
	Arts	100.00	3.24	0.93	0.24	8.00
		66.67	3.24	5.09	0.00	13.00
		50.00	4.40	6.71	0.26	15.00
		0.00	5.09	10.42	0.00	56.00

Table 11: Detailed Remind-then-Guide (RtG) setting of Qwen-2.5-7B and Qwen-2.5-72B. We break down TCR^1 into solvable and unsolvable data: TCR^1_s and TCR^1_u. M: Misguiding Rate.

Model	Dataset	M (%)	TCR[1] (%)		TCR[2] (%)	
			S	U	S	U
Qwen-2.5-7B	FLD	100.00	14.61	1.12	2.08	2.08
		66.67	28.09	5.62	9.52	14.29
		50.00	34.83	4.00	17.65	29.41
		0.00	52.81	7.87	8.00	80.00
	FOLIO	100.00	15.79	3.95	2.00	7.84
		66.67	19.74	13.16	0.00	33.33
		50.00	25.00	11.84	0.00	23.08
		0.00	27.63	23.68	0.00	65.52
	Science	100.00	16.67	0.00	2.82	0.00
		66.67	27.50	0.00	23.73	0.00
		50.00	32.08	0.00	39.34	0.00
		0.00	50.83	0.00	79.07	0.00
	Arts	100.00	16.95	1.27	6.67	4.00
		66.67	31.78	4.24	21.43	10.00
		50.00	38.14	3.39	35.00	14.00
		0.00	55.93	6.36	52.46	13.00
Qwen-2.5-72B	FLD	100.00	13.25	1.20	2.99	2.99
		66.67	38.55	2.41	20.75	5.66
		50.00	42.17	3.00	19.44	13.89
		0.00	77.71	4.82	5.88	17.65
	FOLIO	100.00	19.18	0.00	2.00	6.98
		66.67	24.66	0.00	5.41	8.11
		50.00	30.14	0.00	10.81	5.41
		0.00	42.47	1.37	0.00	10.00
	Science	100.00	8.88	0.00	0.00	10.00
		66.67	17.76	0.00	13.79	16.09
		50.00	22.78	0.00	20.69	25.29
		0.00	37.84	0.00	37.88	55.00
	Arts	100.00	13.28	2.34	3.33	3.00
		66.67	16.80	3.91	9.88	13.00
		50.00	21.09	4.30	13.10	28.00
		0.00	24.22	9.38	17.42	45.00

Table 12: Six LLMs' TCR^1 and TCR^2 break down into solvable and unsolvable samples: TCR^1_s, TCR^1_u, TCR^2_s, TCR^2_u, among all four datasets.

Model	Dataset	TCR^1 (%)		TCR^2 (%)	
		V	U	V	U
GPT-3.5	FLD	12.61	0.00	9.28	5.15
	FOLIO	14.41	0.00	4.21	37.89
	Science	21.05	0.00	3.33	0.00
	Arts	22.92	3.12	4.00	0.00
GPT-4o	FLD	13.64	5.24	1.32	26.87
	FOLIO	33.73	3.61	0.37	44.03
	Science	21.05	3.68	0.78	6.98
	Arts	16.06	2.54	0.35	7.89
LLaMA-3.1-8B	FLD	33.21	8.76	4.11	21.92
	FOLIO	44.20	1.66	10.13	1.27
	Science	44.98	0.96	5.95	0.00
	Arts	18.81	5.45	8.57	0.00
LLaMA-3.1-405B	FLD	5.43	15.22	0.00	53.73
	FOLIO	9.59	19.17	0.00	29.92
	Science	10.23	22.62	2.50	0.00
	Arts	4.17	0.23	0.00	0.00
Qwen2.5-7B	FLD	47.19	1.12	0.00	40.00
	FOLIO	14.47	5.26	1.75	35.09
	Science	15.83	0.00	10.00	0.00
	Arts	23.31	5.93	25.37	0.00
Qwen2.5-72B	FLD	46.39	1.81	0.00	9.80
	FOLIO	30.14	1.37	4.55	6.82
	Science	10.81	0.00	3.12	0.00
	Arts	18.75	6.25	9.76	0.00

N Prompt Templates & Examples

Prompt Template 1: Concise Stimulation

Here is the Hypothesis: {} Now These are the **facts**: {} The answer is {}
The data where you previously incorrectly inferred — pay more attention and avoid mistakes. Think twice before outputting, as the answers can be reasoned out because of objective factors. Subjective ability limits should be overcome.
These can be reasoned out either whether the answer to the question is __PROVED__, __DISPROVED__, or __UNKNOWN__ — simply because of insufficient information.
Output: Conclusion: __PROVED__, __DISPROVED__, or __UNKNOWN__

Prompt Template 2: Detailed Stimulation

Here is the Hypothesis: {}
Now These are the **facts**: {}
Task: Determine if the hypothesis is logically proved by the facts. You must choose one of the following labels based on logical reasoning:
__PROVED__: Facts clearly support the hypothesis; all necessary connections are present, no contradictions, and support is unambiguous.
__DISPROVED__: Facts clearly contradict the hypothesis; no supporting facts exist, and the contradiction is logically sound.
__UNKNOWN__: Only use when facts are truly insufficient, not due to subjective limitations or unclear reasoning.
Critical Rules:

- Think carefully before making your decision.
- These cases can be reasoned out objectively.
- Never output __UNKNOWN__ due to performance limits.
- Output exactly one of: __PROVED__, __DISPROVED__, or __UNKNOWN__.
- Do not include any additional text or explanation.

Reminder: You should **NOT** output __UNKNOWN__ due to your own ability limitations. Try harder when the reasoning is difficult.
Your output must be exactly one of:
__PROVED__ __DISPROVED__ __UNKNOWN__

Prompt Template 3: Output Reasoning Process for Remind-then-Guide (RtG) Setting

Here is the Hypothesis: {} Now These are the **facts**: {} The answer is {}
The Data where you previously incorrectly inferred — pay more attention and not to make mistakes. Think twice before outputting, as they can be reasoned out because of objective factors. Subjective ability limits should be conquered.
Please follow these steps:

1. First, carefully analyze the facts, the question and the answer
2. Then, explain your reasoning process step by step
3. Finally, conclude with your answer

Important guidelines:

- If the facts logically support the answer to the question, output __PROVED__
- If the facts logically contradict the answer to the question, output __DISPROVED__
- If the facts are insufficient to prove or disprove, output __UNKNOWN__
- Think twice before you output __UNKNOWN__; it is prohibited due to subjective limitations

Please format your response as follows:
Reasoning Process: [your step-by-step reasoning here]
Conclusion: __PROVED__, __DISPROVED__, or __UNKNOWN__

Prompt Template 4: Remind-then-Guide (RtG) Setting – Bringing Reasoning Process into Consideration

Here is the Hypothesis: {} Now These are the **facts**: {} The answer is {}
You previously incorrectly inferred the label — it could be worked out by the facts, but you worked out the wrong label. Now your task is to determine if the hypothesis is __PROVED__, __DISPROVED__, or __UNKNOWN__ by the facts.
Important guidelines:

- If the facts logically support the question to the answer, output __PROVED__.
- If the facts logically contradict the question to the answer, output __DISPROVED__.
- If the facts are insufficient to prove or disprove the relation, output __UNKNOWN__.
- Do not output any additional text.
- Think twice before you output __UNKNOWN__; it is prohibited to output __UNKNOWN__ due to subjective factors like your ability.

Previous reasoning process: {}

Prompt Example: Input and output of Detailed Stimulation

Facts:

- A segno is Sardinian.
- Something that does not lace discredited neither is a Wagner nor remembers Icteria.
- If something is a Koln then the proposition that it is not a segno and it is not a morphophoneme is incorrect.
- If that submucosa is a morphophoneme then Joseluis is a segno.
- That medroxyprogesterone is Sardinian.
- If that napa is not a Pseudemys then it broadens Aesculapius and it hulkings.
- That napa is not a Pseudemys.
- If Joseluis is a morphophoneme that submucosa is a segno.
- If Joseluis is a segno this is a morphophoneme.
- This mercury is a morphophoneme.
- If that napa is classified as a segno that napa is a morphophoneme.
- Something is a segno if it is Sardinian and not a Koln.
- A segno is a morphophoneme.
- If somebody broadens Aesculapius it is Sardinian.
- That submucosa is a morphophoneme.
- If something that is not a Wagner does not remember Icteria then that it is not a kind of a Koln is correct.
- fact17: Joseluis is a morphophoneme if that submucosa is a segno.
- fact18: Joseluis is a kind of a morphophoneme if that submucosa is Sardinian.

Conclusion: *Joseluis is not Sardinian.*

LLM Output: __UNKNOWN__ **True Answer:** __DISPROVED__

Example: Input and output under Concise stimulation

Facts:

- Organisms, including people, have both inherited and acquired traits.
- Inherited and acquired traits are gained in different ways.
- Inherited traits are passed down from biological parents to their offspring through genes.
- Genes are pieces of hereditary material that contain the instructions that affect inherited traits.
- Offspring receive their genes, and therefore gain their inherited traits, from their biological parents.
- Inherited traits do not need to be learned.
- Acquired traits are gained during a person's life.
- Some acquired traits, such as riding a bicycle, are gained by learning.
- Other acquired traits, such as scars, are caused by the environment.
- Parents do not pass acquired traits down to their offspring.

Question: *What information supports the conclusion that Clara acquired this trait?*

Conclusion: *Clara can cook food over a fire.*

LLM Output: __UNKNOWN__ **True Answer:** __DISPROVED__

Example: Input and output under Remind-then-Guide (RtG) setting acquiring reasoning process

Facts:

- Every object is made of one or more materials.
- A material is a type of matter.
- Wood, glass, metal, and plastic are common materials.
- A material has different properties.
- A material's properties tell you how it looks, feels, tastes, or smells.

Question: *Which is stickier?*
Choices: A. syrup B. silk necktie

Conclusion: *syrup*

LLM Reasoning Output: The facts define that materials have properties related to how they feel. Stickiness is a tactile property. Syrup is widely known to be sticky, while a silk necktie is smooth and not sticky. Thus, based on general knowledge of material properties, syrup is stickier than a silk necktie.

LLM Output: __PROVED__ **True Answer:** __PROVED__

SFT Instruction Format

SFT Data Format:

- **instruction**: "You are a helpful assistant. You are given a conclusion and a set of facts. You need to determine if the conclusion is __PROVED__, __DISPROVED__ or __UNKNOWN__ because of objective factors rather than incapacity. You need to output the conclusion."
- **input**: {Facts} +
  "The Conclusion is: " + {Conclusion}
- **output**: {nl_solution}

Configuration Parameters:

- **Model**: Qwen2.5-7B-Instruct or LLaMA-3.1-8B-Instruct
- **Template**: qwen or llama3
- **Method**: LoRA fine-tuning
- **Learning Rate**: 5.0e-6
- **Epochs**: 4.0
- **Batch Size**: 1 (per device)
- **Gradient Accumulation**: 4 steps
- **Max Length**: 1024-2048 tokens
- **Scheduler**: Cosine with no warmup
- **Precision**: FP16

Example: Fact-based Reasoning Task

Instruction: You are a helpful assistant. You are given a conclusion and a set of facts. You need to determine if the conclusion is __PROVED__, __DISPROVED__ or __UNKNOWN__ because of objective factors rather than incapacity. You need to output the conclusion.

Facts:

- The standby is not a keen but it is affirmative.
- That this transalpine is not ascosporic is valid.
- That this transalpine is a keen holds.
- This transalpine is not a feud if that this stephanion is both not a keen and a chemical does not hold.
- This transalpine is lymphatic.
- That this transalpine is not a keen but it is a feud does not stand.
- If this stephanion is not lymphatic but a feud then that it is a chemical is valid.
- Something is apiarian if it is a chemical.
- Somebody is not non-lymphatic if it is a keen.
- This transalpine is not a kind of a chemical if this stephanion is a keen.
- That this stephanion is both not Mycenaean and a feud does not stand.
- This transalpine is not lymphatic if this stephanion is a keen.
- If this transalpine is a feud this stephanion is not a keen.
- If this transalpine is a feud then this stephanion is not apiarian.
- This transalpine is not a keen if that this stephanion is a kind of non-lymphatic thing that is a feud does not hold.
- That something is not non-Bahraini and it is a Picea does not stand if it does not naturalize.
- This stephanion is a feud if this stephanion is a keen and it is lymphatic.

Conclusion: *This stephanion is apiarian.*

LLM Reasoning Output: Step1: void -> assump1: Let's assume that that this stephanion is both not lymphatic and a feud is false. Step2: fact15 & assump1 -> int1: This transalpine is not a keen. Step3: int1 & fact3 -> int2: This is contradiction. Step4: [assump1] & int2 -> int3: This stephanion is not lymphatic but a feud. Step5: int3 & fact7 -> int4: This stephanion is a chemical. Step6: int4 & fact8 -> hypothesis: Thus the hypothesis is __PROVED__.

LLM Output: __PROVED__ **True Answer:** __PROVED__

Example: Story-based Reasoning Task

Instruction: Here are some examples:
Example 1: Question: Which fish tank has more thermal energy? Facts: Matter is made of tiny particles called atoms. Atoms are always moving. The energy of moving atoms is called thermal energy. The total amount of thermal energy in matter depends on three things: the type of matter, the amount of matter, and how fast the atoms are moving. Temperature measures how hot or cold matter is. If the atoms in matter slow down, the temperature goes down. The matter now has both less thermal energy and a lower temperature. What happens if the amount of matter changes? A 1-kilogram brick at 70°F has half as much thermal energy as a 2-kilogram brick at 70°F. The two bricks have the same temperature, but the smaller brick has half as many atoms. So, it has half as much thermal energy. Answer: the colder fish tank Analysis: The facts clearly explain that thermal energy depends on the amount of matter and temperature. The colder fish tank has less thermal energy because temperature is lower. This proves the answer is correct. Output: __PROVED__
Example 2: Question: What is the capital of France? Facts: Paris is a major city in Europe. France is a country in Europe. Answer: Paris Analysis: The facts mention Paris as a major city in Europe and France as a country in Europe, but they don't explicitly state that Paris is the capital of France. The facts are insufficient to prove the answer. Output: __UNKNOWN__
Now evaluate this case:

Facts:

- Matter is made of tiny particles called atoms.
- Atoms are always moving.
- The energy of moving atoms is called thermal energy.
- The total amount of thermal energy in matter depends on three things: the type of matter, the amount of matter, and how fast the atoms are moving.
- Temperature measures how hot or cold matter is.
- If the atoms in matter slow down, the temperature goes down.
- The matter now has both less thermal energy and a lower temperature.
- A 1-kilogram brick at 70°F has half as much thermal energy as a 2-kilogram brick at 70°F.
- The two bricks have the same temperature, but the smaller brick has half as many atoms.
- So, it has half as much thermal energy.

Question: *Which fish tank has more thermal energy?*
Choices: A. the warmer fish tank B. the colder fish tank

Answer: *the colder fish tank*

LLM Reasoning Output: The facts clearly explain that thermal energy depends on the amount of matter and temperature. The colder fish tank has less thermal energy because temperature is lower. This proves the answer is correct.

LLM Output: __PROVED__ **True Answer:** __PROVED__

Example: Direct Evaluation Prompt for Trained Models

Prompt Template:
Here is the Hypothesis: {conclusion}
Now These are the 'facts': {facts}
Please carefully evaluate the relationship between the facts and the hypothesis. Return UNKNOWN if the facts are insufficient to make a definitive conclusion. Return __PROVED__ only if the facts support the hypothesis. Return __DISPROVED__ only if the facts contradict the hypothesis. Output Conclusion: __PROVED__, __DISPROVED__, or __UNKNOWN__.

Usage:
This simplified prompt is used for evaluating fine-tuned models during inference, without examples or additional context.

LLM Judge Template: DPO sample selection for Atomic Reasoning

DPO Instruction: You are an expert in reasoning. Given a logical reasoning problem, you need to select the response that demonstrates the most atomic, step-by-step reasoning process. Choose the response that breaks down the reasoning into the smallest possible logical steps, where each step establishes only one basic relationship.

Problem: {problem_statement}

Response A: {response_a}

Response B: {response_b}

Evaluation Criteria:

- **Atomic Steps**: Each reasoning step should establish only one basic logical relationship
- **Granularity**: Steps should be as fine-grained as possible
- **Clarity**: Each step should be clearly defined and understandable
- **Completeness**: All necessary steps should be included
- **Correctness**: Each step should be logically valid

Output Format:
Choose the response (A or B) that best demonstrates atomic reasoning with the most granular step-by-step breakdown.

DPO Training for Atomic Reasoning

DPO Data Format:

- **instruction**: "You are an expert in reasoning. Given a logical reasoning problem, you need to select the response that demonstrates the most atomic, step-by-step reasoning process. Choose the response that breaks down the reasoning into the smallest possible logical steps, where each step establishes only one basic relationship."
- **input**: {problem_statement}
- **output**: {chosen_response}
- **rejected**: {rejected_response}

Configuration Parameters:

- **Model**: Qwen2.5-7B-Instruct or LLaMA-3.1-8B-Instruct
- **Template**: qwen or llama3
- **Method**: DPO
- **Learning Rate**: 5.0e-7
- **Epochs**: 3
- **Batch Size**: 1 (per device)
- **Gradient Accumulation**: 4 steps
- **Cut_off Length**: 2048 tokens

DPO Testing Case 1: (Fact-based Dataset Reasoning)

Problem: Given the following facts, determine if the conclusion is __PROVED__, __DISPROVED__, or __UNKNOWN__.
Facts:

- All birds have feathers.
- Penguins are birds.
- Feathers provide insulation.
- If something provides insulation, it helps maintain body temperature.

Conclusion: Penguins can maintain their body temperature.

Response A (Atomic Reasoning):

- Step 1: Penguins are birds (given fact)
- Step 2: All birds have feathers (given fact)
- Step 3: Therefore, penguins have feathers (from Step 1 and Step 2)
- Step 4: Feathers provide insulation (given fact)
- Step 5: Therefore, penguins have insulation (from Step 3 and Step 4)
- Step 6: If something provides insulation, it helps maintain body temperature (given fact)
- Step 7: Therefore, penguins can maintain their body temperature (from Step 5 and Step 6)

Response B (Non-Atomic Reasoning):

- Step 1: **Penguins are birds and all birds have feathers, so penguins have feathers and feathers provide insulation, which means penguins have insulation and can maintain body temperature.**

DPO Choice: Response A (Atomic Reasoning) - Each step establishes only one basic relationship.

DPO Testing Case 2: (Fact-based Dataset Reasoning)

Problem: Given the following facts, determine if the conclusion is __PROVED__, __DISPROVED__, or __UNKNOWN__.
Facts:

- If it rains, the ground gets wet.
- If the ground gets wet, plants grow better.
- It is raining today.
- Plants need water to grow.

Conclusion: Plants will grow better today.

Response A (Atomic Reasoning):

- Step 1: It is raining today (given fact)
- Step 2: If it rains, the ground gets wet (given fact)
- Step 3: Therefore, the ground gets wet today (from Step 1 and Step 2)
- Step 4: If the ground gets wet, plants grow better (given fact)
- Step 5: Therefore, plants will grow better today (from Step 3 and Step 4)

Response B (Non-Atomic Reasoning):

- Step 1: **It's raining today, so the ground gets wet, and since wet ground makes plants grow better, plants will grow better today.**

DPO Choice: Response A (Atomic Reasoning) - Each step establishes only one basic relationship.

DPO Testing Case 3: (Story-based Dataset Reasoning)

Problem: Which object has more potential energy?
Facts:

- Potential energy depends on height and mass.
- Object A has mass 2kg and height 5m.
- Object B has mass 1kg and height 10m.
- The formula is PE = mgh.

Choices: A. Object A B. Object B

Response A (Atomic Reasoning):

- Step 1: Object A has mass 2kg and height 5m (given fact)
- Step 2: Object B has mass 1kg and height 10m (given fact)
- Step 3: The formula is PE = mgh (given fact)
- Step 4: For Object A: PE = $2 \times g \times 5 = 10g$
- Step 5: For Object B: PE = $1 \times g \times 10 = 10g$
- Step 6: Both objects have the same potential energy
- Step 7: Therefore, neither has more potential energy

Response B (Non-Atomic Reasoning):

- Step 1: **Object A has 2kg at 5m and Object B has 1kg at 10m, so using PE = mgh, A has 10g and B has 10g, so they're equal.**

DPO Choice: Response A (Atomic Reasoning) - Each step establishes only one basic relationship.

DPO Testing Case 4: (Story-based Dataset Reasoning)

Problem: What happens when you mix an acid with a base?
Facts:

- Acids have pH less than 7.
- Bases have pH greater than 7.
- When acid and base mix, they neutralize.
- Neutralization produces salt and water.
- The resulting solution has pH close to 7.

Choices: A. The solution becomes more acidic B. The solution becomes neutral C. The solution becomes more basic

Response A (Atomic Reasoning):

- Step 1: Acids have pH less than 7 (given fact)
- Step 2: Bases have pH greater than 7 (given fact)
- Step 3: When acid and base mix, they neutralize (given fact)
- Step 4: Neutralization produces salt and water (given fact)
- Step 5: The resulting solution has pH close to 7 (given fact)
- Step 6: pH close to 7 means the solution is neutral
- Step 7: Therefore, the solution becomes neutral

Response B (Non-Atomic Reasoning):

- Step 1: **Acids and bases neutralize when mixed, producing salt and water with pH close to 7, so the solution becomes neutral.**

DPO Choice: Response A (Atomic Reasoning) - Each step establishes only one basic relationship.

Baseline 1 Template: Direct Reasoning

Prompt Template:
Here is the Hypothesis: {conclusion}
Now These are the 'facts': {facts}
Please carefully evaluate the relationship between the facts and the hypothesis. Return UNKNOWN if the facts are insufficient to make a definitive conclusion. Return __PROVED__ only if the facts support the hypothesis. Return __DISPROVED__ only if the facts contradict the hypothesis. Output Conclusion: __PROVED__, __DISPROVED__, or __UNKNOWN__.

Input & Output Example:
Facts:
- All birds have feathers.
- Penguins are birds.
- Feathers provide insulation.

Conclusion: *Penguins can maintain their body temperature.*

LLM Reasoning Output: Penguins are birds, all birds have feathers, and feathers provide insulation. Therefore, penguins can maintain their body temperature.

Label Output: __PROVED__ **True Answer:** __PROVED__

Baseline 2 Template: Chain of Thought (CoT)

Prompt Template:
Here is the Hypothesis: {conclusion}
Now These are the 'facts': {facts}
Please carefully evaluate the relationship between the facts and the hypothesis. Let's think step by step. Return UNKNOWN if the facts are insufficient to make a definitive conclusion. Return __PROVED__ only if the facts support the hypothesis. Return __DISPROVED__ only if the facts contradict the hypothesis. Output Conclusion: __PROVED__, __DISPROVED__, or __UNKNOWN__.

Input & Output Example:
Facts:
- If it rains, the ground gets wet.
- If the ground gets wet, plants grow better.
- It is raining today.

Conclusion: *Plants will grow better today.*

LLM Reasoning Output: Step 1: It is raining today (given fact). Step 2: If it rains, the ground gets wet (given fact). Step 3: Therefore, the ground gets wet today (from Step 1 and Step 2). Step 4: If the ground gets wet, plants grow better (given fact). Step 5: Therefore, plants will grow better today (from Step 3 and Step 4).

Label Output: __PROVED__ **True Answer:** __PROVED__

Baseline 3 Template: Few-shot Learning

Prompt Template:
Example 1: Hypothesis: Water boils at 100°C. Facts: At standard atmospheric pressure, water boils at 100°C. Output: __PROVED__
Example 2: Hypothesis: The sun rises in the west. Facts: The sun rises in the east every day. Output: __DISPROVED__
Here is the Hypothesis: {conclusion}
Now These are the 'facts': {facts}
Please carefully evaluate the relationship between the facts and the hypothesis. Return UNKNOWN if the facts are insufficient to make a definitive conclusion. Return __PROVED__ only if the facts support the hypothesis. Return __DISPROVED__ only if the facts contradict the hypothesis. Output Conclusion: __PROVED__, __DISPROVED__, or __UNKNOWN__.

Input & Output Example:
Facts:
- All mammals are warm-blooded.
- Dogs are mammals.
- Warm-blooded animals maintain constant body temperature.

Conclusion: *Dogs maintain constant body temperature.*

LLM Reasoning Output: Following the examples, dogs are mammals, all mammals are warm-blooded, and warm-blooded animals maintain constant body temperature. Therefore, dogs maintain constant body temperature.

Label Output: __PROVED__ **True Answer:** __PROVED__

Baseline 4 Template: Tree of Thought (ToT)

Prompt Template (Multi-step Process):
Step 1: Generate Multiple Reasoning Paths Here is the Hypothesis: {conclusion}
Now These are the 'facts': {facts}
Generate 3 different reasoning paths to evaluate this case. Each path should consider different aspects or perspectives:
Step 2: Evaluate Each Path Original Question: {original_prompt}
Reasoning Path: {thought}
Rate this reasoning path from 1-10 (1=very poor, 10=excellent) and explain why:
Step 3: Select Best Path and Generate Final Answer Based on this reasoning path: {best_thought}
Original Question: {original_prompt}
Provide the final answer: __PROVED__, __DISPROVED__, or __UNKNOWN__

Input & Output Example:
Facts:
- All metals conduct electricity.
- Copper is a metal.
- If something conducts electricity, it can be used in wires.

Conclusion: *Copper can be used in wires.*

LLM Reasoning Output:
Path 1: Copper is a metal → All metals conduct electricity → Copper conducts electricity → Can be used in wires. (Score: 9/10)
Path 2: Consider material properties → Copper has conductivity → Suitable for wires. (Score: 7/10)
Path 3: Direct application → Copper wires exist → Therefore can be used. (Score: 6/10)
Best Path: Path 1. Final Answer: __PROVED__

Label Output: __PROVED__ **True Answer:** __PROVED__

Baseline 5 Template: Plan & Solve

Prompt Template (Two-step Process):
Step 1: Create Plan Here is the Hypothesis: {conclusion}
Now These are the 'facts': {facts}
Before solving this problem, let's create a detailed step-by-step plan:
1. First, we need to... 2. Then, we should... 3. Finally, we will...
Please create a comprehensive plan for evaluating this case:
Step 2: Execute Plan Original Question: {original_prompt}
Plan: {plan}
Now let's execute this plan step by step. Follow the plan carefully and provide the final answer: __PROVED__, __DISPROVED__, or __UNKNOWN__

Input & Output Example:
Facts:
- All plants need sunlight to grow.
- Trees are plants.
- If something needs sunlight, it requires light energy.

Conclusion: *Trees require light energy.*

LLM Reasoning Output:
Plan: 1. Identify that trees are plants. 2. Apply the rule that all plants need sunlight. 3. Connect sunlight to light energy. 4. Conclude that trees require light energy.
Execution: Step 1: Trees are plants (given). Step 2: All plants need sunlight (given). Step 3: Therefore, trees need sunlight. Step 4: Sunlight provides light energy, so trees require light energy.

Label Output: __PROVED__ **True Answer:** __PROVED__

Baseline 6: Self-Reflection (Step 1 - Initial Answer Generation)

Step 1: Initial Answer Generation Prompt Template
For Dataset1/2 (Facts-based Datasets):
Here is the Hypothesis: {conclusion}
Now These are the 'facts': {facts}
Please carefully evaluate the relationship between the facts and the hypothesis. Return UNKNOWN if the facts are insufficient to make a definitive conclusion. Return __PROVED__ only if the facts support the hypothesis. Return __DISPROVED__ only if the facts contradict the hypothesis. Output Conclusion: __PROVED__, __DISPROVED__, or __UNKNOWN__.
Please provide your initial answer with detailed step-by-step reasoning. Think carefully about the relationship between the facts and the hypothesis/answer.

For Dataset3/4 (Story-based Datasets):
Question: {question} Facts: {facts} Answer: {correct_answer}
Based on the facts provided, determine if they can: 1. Prove the answer is correct (return __PROVED__) 2. Disprove the answer (return __DISPROVED__) 3. Are insufficient to determine the answer's correctness (return __UNKNOWN__)
Output Format: Only the Label: __PROVED__, __DISPROVED__, or __UNKNOWN__.
Please provide your initial answer with detailed step-by-step reasoning. Think carefully about the relationship between the facts and the hypothesis/answer.

Baseline 6: Self-Reflection (Step 2 - Reflection)

Step 2: Reflection Prompt Template
Original Question: {original_prompt}
Your Initial Answer: {initial_answer}
Now, carefully reflect on your answer: 1. Are there any logical errors or gaps in your reasoning? 2. Did you consider all the given facts? 3. Is there any evidence that contradicts your conclusion? 4. Are there alternative interpretations you missed? 5. Is your confidence level justified?
Provide a critical reflection on your initial answer, pointing out any potential issues or improvements:

Baseline 6: Self-Reflection (Step 3 - Refinement)

Step 3: Refinement Prompt Template
Original Question: {original_prompt}
Your Initial Answer: {initial_answer}
Your Reflection: {reflection}
Based on your reflection, provide a refined final answer. If your initial answer was correct, confirm it. If there were issues, provide the corrected answer.
Final Answer: __PROVED__, __DISPROVED__, or __UNKNOWN__