ReEfBench: Quantifying the Reasoning Efficiency of LLMs

Zhizhang Fu *
Westlake University
fuzhizhang@westlake.edu.cn

Yuancheng Gu *
Imperial College London
yuancheng.gu22@alumni.imperial.ac.uk

Chenkai Hu
New York University
ckh326@nyu.edu

Hanmeng Liu
Hainan University
liuhanmeng@hainanu.edu.cn

Yue Zhang †
Westlake University
zhangyue@westlake.edu.cn

Abstract

Test-time scaling has enabled Large Language Models (LLMs) to tackle complex reasoning, yet the limitations of current Chain-of-Thought (CoT) evaluation obscure whether performance gains stem from genuine reasoning or mere verbosity. To address this, (1) we propose a novel neuro-symbolic framework for the non-intrusive, comprehensive process-centric evaluation of reasoning grounded in First-Order Logic. (2) Through this lens, we identify four distinct behavioral prototypes and diagnose the failure modes. (3) We examine the impact of inference mode, training strategy, and model scale. Our analysis reveals that extended token generation is not a prerequisite for deep reasoning. Furthermore, we reveal critical constraints: mixing long and short CoT data in training risks premature saturation and collapse, while distillation into smaller models captures behavioral length but fails to replicate logical efficacy due to intrinsic capacity limits.

1 Introduction

Test-time scaling (Zhang et al., 2025) has empowered LLMs to tackle complex problems by allocating more compute to reasoning steps (OpenAI, 2024). However, this paradigm reveals a perplexing inefficiency: models often exhibit "overthinking", generating protracted thinking chains even for trivial tasks like calculating $2 + 3$ (Chen et al., 2025b). Given this observation, however, further **quantifying** the discrepancy systematically remains challenging due to the lack of process-centric evaluation, which is necessary for distinguishing genuine cognitive depth from mere computational inflation.

To this end, a simple intuition is to measure efficiency using $W = P \cdot t$, where t represents the computational consumption (e.g., token or step count) and W denotes the abstract "Logical Depth" achieved. Consequently, P represents the **reasoning efficiency**—the logical gain per unit of computation. Efficient models maximize P, achieving the necessary logical depth W with minimal cost t, whereas "overthinking" models increase t without a proportional gain in W.

While it is relatively easier to quantify the Cost (t) (e.g., token or step count), accurately quantifying Logical Depth (W) requires a specialized evaluation framework with three key attributes: (1) **A Formal Basis** (e.g., First-Order Logic), to ensure the reasoning path has an objective, calculable logical depth; (2) **Decoupling of Logic and Knowledge**, to evaluate pure reasoning capabilities without interference from knowledge; and (3) **Controllability and Scalability**, to generate problems of precise and varying levels of logical depth. To this end, existing datasets such as ProntoQA (Saparov and He, 2023) quantify reasoning over FOL, but suffer from limitations in scalability and, crucially, lack an evaluation mechanism to calculate logical depth and extensive process behaviors.

To address this, we construct a comprehensive neuro-symbolic evaluation framework, Reasoning Efficiency Bench (ReEfBench), as illustrated in Figure 1. Starting with the generation of test instances (Phase A) and the acquisition of model responses (Phase B), our pipeline utilizes an LLM parser to decompose the reasoning text into logical nodes (Phase C). We then apply deterministic rules to identify node-level attributes such as logical depth (Phase D), which are finally aggregated to compute comprehensive *Behavioral Indicators* (Phase E). This approach combines the generalization of neural models with the rigor of symbolic logic, enabling a non-intrusive and quantifiable evaluation of the reasoning process.

We use this framework to conduct a comprehensive evaluation of 25 open-source and closed-source models, quantifying their reasoning depth

*These authors contributed equally to this work
†Corresponding author

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 20325–20346

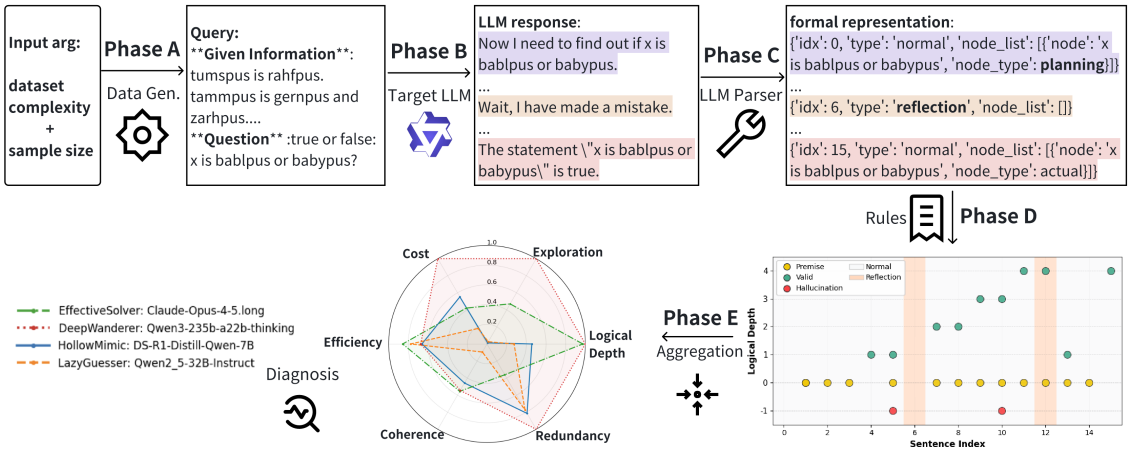


Figure 1: Overview of our framework. We generate scalable, controllable first-order logic (FOL) data that enables precise verification of logical depth (Phase A). Target LLM generates a response for each query (Phase B). The pipeline parses a target LLM's response into formal representations (Phase C), computes its logical depth and correctness via rule-based verifiers (Phase D), evaluates the normalized output along six dimensions—Logical Depth, Cost, Exploration, Efficiency, Coherence, and Redundancy—and finally classifies it into one of four behavioral prototypes: EffectiveSolver, DeepWanderer, HollowMimic, and LazyGuesser (Phase E).

and corresponding cost. Given the quantitative results, we identify four behavioral prototypes: the **Deep Wanderer** (high token consumption, exhaustive exploration; e.g., Qwen3-235B-thinking), the **Effective Solver** (high efficiency, precise reasoning; e.g., Claude-Opus-4.5), the **Hollow Mimic** (diluted expansion, verbose but shallow; e.g., Deepseek-R1-Qwen-7B), and the **Lazy Guesser** (saturation/collapse, minimal effort; e.g., Qwen2.5-32B-Instruct). This taxonomy reveals two successful pathways—adaptive strategies that optimize either efficiency or exploration—and two failure modes characterized by unproductive verbosity or cognitive saturation or collapse.

Further, we analyze the impact of inference mode, training strategy, and model scale on these behaviors. Our results reveal that while Long CoT (OpenAI, 2024; DeepSeek-AI et al., 2025) generally yields higher Logical Depth (W) than Short CoT (Wei et al., 2022) under similar settings, Short CoT proves capable of reaching substantial depths, often rivaling Long CoT. Specifically, we find that: (1) when trained with reasoning tasks, Short CoT can approach the depth of Long CoT, particularly when enhanced with reflection mechanisms; (2) Distilling Long CoT capabilities into small models often leads to "behavioral mimicry", extending t without increasing W due to intrinsic capacity constraints; and (3) mixing long and short CoT data risks disrupting model strategies, often causing premature saturation and collapse.

Our contributions are threefold: 1. We propose the first neuro-symbolic evaluation framework in FOL for deterministic, non-intrusive reasoning quantification. 2. By quantifying efficiency, we identify four **behavioral prototypes** and diagnose critical failure modes. 3. We challenge the assumption that deep reasoning requires extensive token consumption (Chen et al., 2025a).

We release ReEfBench (including data and methods) at https://github.com/Harryking1999/LoG.

2 Related Work

Long CoT and Evaluation Challenges. Recent reasoning LLMs leverage test-time scaling to tackle complex problems by allocating increased computation to reasoning steps (OpenAI, 2024). While Long CoT effectively improves reasoning capabilities (OpenAI, 2024; DeepSeek-AI et al., 2025), extended chains often manifest as "overthinking"—redundant verification or verbosity with minimal accuracy gains (Chen et al., 2025b; Peng et al., 2025). This motivates a comprehensive evaluation of the CoT process to provide the insights necessary for diagnosing these inefficiencies.

Current evaluation methods are insufficient for diagnosing these process-level discrepancies. Human annotation is prohibitively expensive, while "LLM-as-a-Judge" (Fu et al., 2024) approaches suffer from biases and noisy judgments (Lee et al., 2025). Similarly, early automated metrics largely rely on pure rule-based statistics or uninterpretable scores (Golovneva et al., 2023; Prasad et al., 2023), failing to capture the hallmarks of System 2 rea-

Framework	Dataset Property			Evaluation Property			
	Scalable	FOL	Logic-Only	LogDepth	BehProc	Interp	NonIntr
ReEfBench (Ours)	✓	✓	✓	✓	✓	✓	✓
FOLIO (Han et al., 2024)	✗	✓	✗	–	–	–	–
ProntoQA (Saparov and He, 2023)	∼	✓	✓	✗	✗	✓	✗
ZebraLogic (Lin et al., 2025)	✗	✗	✓	✓	✗	✓	✗
LogiNumSynth (Liu et al., 2025)	✓	✓	✓	✓	✗	✓	✗
Sys2Bench (Parashar et al., 2025)	–	–	–	✗	✓	✓	✓
CognitiveBehaviors (Gandhi et al., 2025)	–	–	–	✗	✓	✓	✗
Roscoe (Golovneva et al., 2023)	–	–	–	✓	∼	∼	✓
ReCEval (Prasad et al., 2023)	–	–	–	✓	✗	✗	✓

Table 1: Comparison of existing frameworks/datasets against our method across dataset properties (Scalable, FOL: First-Order Logic, Logic-Only) and evaluation properties (LogDepth: Logical Depth, BehProc: Behavioral Process, Interp: Interpretable, NonIntr: Non-intrusive). The symbol ∼ indicates partial attainment. Furthermore, "✗" denotes that a feature is relevant but missing, while "-" denotes inapplicability (i.e., evaluation frameworks lack dataset properties, and dataset contributions lack evaluation properties)

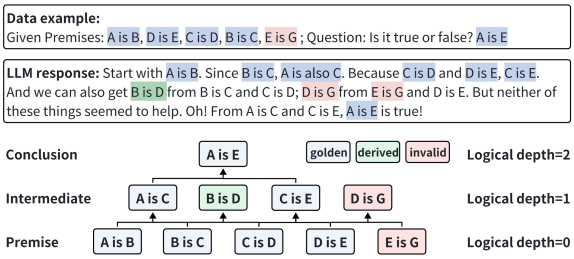


Figure 2: Example of our dataset: Premise, Intermediate and Conclusion. Dataset Complexity = max(Logical depth) = 2.

soning like exploration and reflection (Chen et al., 2025a). In logical reasoning, metrics often lack granularity or flexibility. As the most prevailing process metric, "Validity" (Saparov and He, 2023; Zhou et al., 2025; Wei et al., 2025) serves merely as a binary strict accuracy metric. While some step-level evaluations exist, they are typically intrusive (Saparov and He, 2023; Lin et al., 2025; Liu et al., 2025), enforcing rigid output formats that limit model flexibility. Furthermore, recent studies on exploration and reflection lack a holistic framework to assess logical depth and cognitive behaviors simultaneously (Parashar et al., 2025; Nie et al., 2024; Heyman and Zylberberg, 2025; Xie et al., 2025). In contrast, by combining the flexibility of neural parsing with the rigor of symbolic verification, our approach mitigates the brittleness of rule-based metrics and the stochasticity of LLM judges, enabling non-intrusive, comprehensive assessment of both logical depth and cognitive behaviors. The systematical comparison is in Table 1.

Neural Symbolic Approaches. Neuro-symbolic AI converges connectionist pattern recognition with symbolic rigor. In the LLM era, this paradigm typically enhances reasoning capabilities by parsing natural language into formal logic (e.g., SQL, FOL) for rigorous execution by external solvers (Olausson et al., 2023; Pan et al., 2023; Wang et al., 2025). In this work, we repurpose this architecture for evaluation. Instead of aiding the model in solving tasks, we employ small LLMs solely to parse semantic variability into structured forms, delegating metric calculation to a symbolic module.

3 Method

As illustrated in Figure 1, ReEfBench consists of five main stages: scalable dataset construction (Phase A; Section 3.1), target LLM generation (Phase B; Section 3.2), response decomposition/parsing (Phase C; Section 3.2), structured processing of the decomposed reasoning nodes (Phase D; Section 3.3) and metrics of reasoning process (Phase E; Section 3.4).

3.1 Reasoning Data Construction

Dataset Formulation. As illustrated in Figure 2, each instance in our dataset consists of a set of *premises*, a target *conclusion*, and a step-by-step *golden solution*. The target model is presented with the premises and required to verify the truthfulness of the conclusion. Structurally, the basic **logical nodes** follow the form "x is A", representing "all instances of x are members of A". To support complex reasoning, we employ *Modus Ponens* (e.g., inferring "Rex is a vertebrate" from "Rex is a dog" and "All dogs are vertebrates") as the atomic inference rule, extended with conjunction ($\wedge$) and disjunction ($\vee$), yielding the four deduction rules

summarized in Table 10 (Appendix A.2).

Scalable Data Construction. As outlined in Phase A of Figure 1, our data construction process is parameterized by the dataset *complexity level* C and sample size N. Referring to Figure 2, we define the *complexity level* C as the *Logical Depth* of the conclusion node, corresponding to the maximum number of inference layers in the golden solution. To construct these samples, we employ a stochastic backward-chaining process: we initialize a random conclusion node and recursively expand the graph backward by sampling valid deduction rules to generate the necessary valid premises until the reasoning chain reaches the target logical depth C (detailed in Algorithm 1, Appendix A.5). Upon completion, the leaf nodes serve as the *premises*, the root node acts as the *conclusion*, and the remaining nodes constitute the *intermediate steps* of the golden solution in Figure 2. Finally, to increase task difficulty, we augment each instance with C extra invalid premises (distractors).

3.2 LLM response and LLM-based extraction

The reasoning data generated in Section 3.1 (Phase A in Figure 1) is passed to target LLMs to obtain responses (Phase B). The details of the prompt and hyper-parameters are shown in Appendix B.3. The responses will then be processed through the pipeline below (Phase C).

Decomposition and Parsing. Given a response, we first decompose it into individual sentences based on punctuation marks (such as periods, question marks, etc.) to significantly reduce the difficulty of processing for LLMs. Subsequently, we use an LLM-based parser to identify the **type of each sentence**, classifying them as either *normal* or *reflection* (Xie et al., 2025), and to extract the logical nodes within the sentences, determining the **type of logical nodes** as either *actual* or *planning*, representing logical nodes that denote actual events and planning-related logical nodes, respectively. As for the example in Figure 1, "The statement x is bablpus or babypus is true" is classified as an *actual* node, denoted "x is bablpus or babypus". Similarly, representative examples of *planning* and *reflection* are also explicitly illustrated in Phase C of Figure 1. The prompts and code for the LLM parser can be found in our open-source repository.

Parser Validation. To ensure high reliability, we construct a validation set of 1,000 samples from 5 models manually labeled by three independent human annotators. The annotators demonstrate exceptional consistency with a 95.1% agreement rate. On this high-quality dataset, the parser based on Qwen2.5-32B-Instruct (Qwen et al., 2025) achieves an F1 of 94.3%, confirming its effectiveness in replicating human classification behavior. We provide a detailed discussion on parsing stability in Appendix B.4, along with concrete examples.

3.3 Rule-based Node Processing

As in Figure 1 Phase C, we apply a rule-based method to classify the correctness of each *actual* node produced in Section 3.2 (using Algorithm 2 in Appendix A.5). Concurrently, a separate procedure calculates the *logical depth* for arbitrary actual nodes with backward chaining. For nodes explicitly present in the golden solution, we directly retrieve their depth (Figure 2). For correct nodes absent from the golden solution, we calculate their depth by identifying the deepest nodes required to derive them (detailed in Algorithm 4, Appendix A.5). For instance, following Figure 2, although "B is D" is not in the golden solution, it is assigned a *logical depth* of 1 based on its logical antecedents.

For *planning* nodes and *reflection* sentences, we assess their functional utility by examining whether subsequent nodes (within a fixed contextual window) exhibit measurable changes in logical depth or breadth. For instance, we validate a planning node if it is followed by a corresponding actual inference, whereas a reflection sentence is considered effective only if it yields a novel or deeper logical node within a local window (e.g., within the next 5 sentences). To confirm our evaluation is robust to this specific window size, we provide a comprehensive ablation study in Appendix B.4.

3.4 Evaluation Metrics

We aggregate the node classifications and attributes derived in Section 3.3, alongside basic statistics (e.g., token counts), into six interpretable diagnostic scores (visualized via the radar chart in Figure 1). Specifically: (1) *Logical Depth* (S_{ld}) reflects reasoning capability via the achieved valid logical depth; (2) *Cost* (S_{cost}) captures computational consumption, combining total token count and frequency of reflection/planning steps; (3) *Exploration* (S_{exp}) counts unique, correct logical nodes explored; (4) *Efficiency* (S_{eff}) integrates token efficiency (tokens per depth increment) and effective span (normalized position where new node

generation stops); (5) *Coherence* (S_{coh}) assesses whether meta-cognitive steps (planning/reflection) lead to actual logical progress; (6) *Redundancy* (S_{red}) quantifies repetition at sentence and node levels. Given that these dimensions often involve disparate units (e.g., S_{cost} combines raw token counts and step frequencies), we require a unified scale for aggregation. Therefore, we perform max-normalization to map all raw sub-metrics into the $[0, 1]$ range. The final score for each dimension is computed as the average of its normalized components. Complementarily, we also utilize the interpretable raw values for fine-grained analysis. For further details on calculating these six metrics and the raw metrics, please refer to Appendix B.2.

4 Experiments

We conduct extensive experiments over ReEf-Bench, with representative commercial and open-source models to evaluate their reasoning processes. We show main results in Section 4.2, identify four distinct behavioral prototypes in Section 4.3, and subsequently investigate the dynamic trajectories of model behaviors across varying complexity levels in Section 4.4.

4.1 Experimental Settings

We utilize the scalable dataset construction method from Section 3.1 to generate evaluation sets spanning Complexity Levels 3 to 11, with 100 samples per level. Detailed statistics of these generated datasets are provided in Appendix B.5. Our evaluation suite encompasses 25 diverse LLMs, including proprietary SOTA models (e.g., Claude-4.5 series (Anthropic, 2025a,b)) and open-weights models (e.g., Qwen series (Qwen et al., 2025; Yang et al., 2025; Team, 2025) and DeepSeek-R1 series (DeepSeek-AI et al., 2025)), covering both standard instruction-tuned and reasoning-enhanced paradigms. For nomenclature, we append .long or .short suffixes to models with distinct thinking or non-thinking modes (e.g., Qwen3-32B.long denotes the thinking mode), whereas single-mode models retain their original abbreviations. Detailed model settings can be found in Table 11 in Appendix B.1.

4.2 Main Results

To capture model behaviors under maximum stress, we focus exclusively on the most challenging subset (Complexity Level 11). In Table 2, we report the statistics of the six metrics defined in Section 3.4 across 14 representative models, where we observe substantial behavioral divergence. Frontier reasoning models nearly reach the maximal logical depth of 11, yet their behaviors diverge markedly. Qwen3-235B-thinking consumes 16.6k tokens, reflecting exhaustive Exploration (1.0), high Redundancy (0.77), and low Efficiency (0.47), whereas Claude-Opus-4.5.long uses much fewer tokens (3.5k) with high Efficiency (0.60) but lower Redundancy (0.28) and exploration (0.47). At smaller-scale ($\leq$32B), models with short CoT (e.g., Qwen2.5-32B-Instruct) generally incur low cost (<2k tokens) but rarely exceed depth 5, while long-CoT models (e.g., QwQ-32B) spend more tokens (3k–6k) yet struggle to match the depth of frontier models, peaking around depth 7.1. Specifications for all 25 evaluated models are provided in Table 17 in Appendix B.8.

4.3 The Reasoning Landscape

To systematize the patterns in Table 2, we map the models into a normalized reasoning space defined by Logical Depth (S_{ld}, y-axis) and Cost (S_{cost}, x-axis). We apply K-means clustering with $k = 4$ specifically to capture the **behavioral prototypes** inherent to the plane's four quadrants, yielding four distinct reasoning archetypes visualized in Figure 3. We then employ the remaining four diagnostic metrics (Efficiency, Exploration, Coherence, Redundancy) to interpret their internal mechanisms. To accurately characterize these prototypes, we compute the average score of each metric weighted by the confidence score S_c (Table 2).

Deep Wanderer This cluster forms a distinct behavioral regime characterized by simultaneous high cost ($S_{cost} \approx 0.88$) and deep reasoning ($S_{ld} \approx 1.0$). In our current evaluation, it is uniquely represented by Qwen3-235B-thinking. While models like QwQ-32B and Qwen3-235B.long approach this cluster in Figure 3, they lack the extreme depth or cost. Diagnostically, this archetype trades efficiency for coverage: it combines low Efficiency (0.47) with high Redundancy (0.77) and Exploration (1.00). This suggests the model reaches deep reasoning states by rigorously expanding the search space and tolerating redundant steps.

Effective Solver. Models in this category, such as Claude-Opus-4.5.long and Qwen3-235B-Instruct, maintain high task performance ($S_{ld} \approx 0.86$) but with significantly reduced cost ($S_{cost} \approx 0.37$). Un-

#	Model	Classification		Core Metrics		Diagnostic Metrics				Raw Stats.	
		Category	S_c	S_{ld}	S_{cost}	S_{exp}	S_{eff}	S_{coh}	S_{red}	**Depth**	**Token(k)**
1	Qwen3-235B-thinking	DeepWanderer	1.00	1.00	0.88	1.00	0.47	0.41	0.77	10.54	16.8
2	Qwen3-235B-Instruct	EffectiveSolver	0.82	0.95	0.37	0.83	0.59	0.28	0.62	9.96	3.4
3	DeepSeek-R1		0.80	0.86	0.41	0.34	0.59	0.58	0.48	9.04	3.7
4	Claude-Opus-4.5.long		0.80	0.97	0.37	0.47	0.60	0.42	0.28	10.27	3.5
5	Qwen3-235B.long		0.67	0.81	0.46	0.29	0.57	0.31	0.55	8.54	4.1
6	Claude-Opus-4.5.short		0.33	0.74	0.24	0.62	0.70	0.47	0.37	7.82	1.4
7	DS-R1-Qwen-7B	HollowMimic	1.00	0.46	0.49	0.01	0.47	0.35	0.62	4.80	6.0
8	Qwen3-14B.long		0.39	0.47	0.39	0.09	0.54	0.34	0.57	4.90	3.4
9	QwQ-32B		0.34	0.68	0.61	0.14	0.48	0.32	0.62	7.12	5.7
10	Qwen3-32B.long		0.29	0.58	0.35	0.24	0.56	0.33	0.52	6.14	2.7
11	Qwen2.5-32B-Inst	LazyGuesser	1.00	0.28	0.16	0.03	0.55	0.07	0.59	2.90	0.7
12	Qwen3-4B.short		1.00	0.45	0.17	0.02	0.67	0.11	0.54	4.70	0.7
13	Qwen3-235B.short		0.74	0.54	0.20	0.12	0.70	0.43	0.53	5.70	1.0
14	Qwen3-4B.long		0.62	0.42	0.29	0.03	0.55	0.39	0.48	4.38	1.7
Category Avg (weighted)		DeepWanderer		**1.00**	**0.88**	**1.00**	0.47	0.41	**0.77**	**10.54**	**16.8**
		EffectiveSolver		0.86	0.37	0.42	0.60	0.40	0.46	9.10	3.4
		HollowMimic		0.52	0.45	0.09	0.50	**0.42**	0.60	5.43	4.7
		LazyGuesser		0.45	0.21	0.09	**0.62**	0.25	0.51	4.78	1.1

Table 2: Model classification results based on K-means clustering, showing category assignments, confidence scores (S_c), core metrics (S_{ld}, S_{cost}), and diagnostic metrics (S_{exp}, S_{eff}, S_{coh}, S_{red}) for representative models. The rightmost "Raw Stats." columns are included for reference, where Depth represents the Average Logical Depth (max 11 in this dataset) and Token.(k) denotes the Token Count (in thousands). Category averages are weighted by confidence scores (S_c) and calculated based on the full set of 25 models (Table 17).

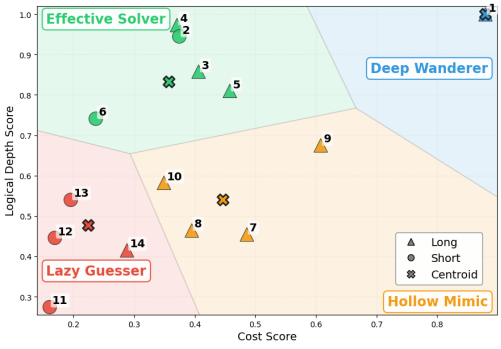


Figure 3: Models plotted by Logical Depth (S_{ld}) vs. Cost (S_{cost}); triangles = Long CoT, circles = Short CoT, stars = centroids. Model IDs listed in Table 2.

like Deep Wanderers, they favor direct derivation, exhibiting the lowest Redundancy (0.46) and limited search breadth (0.42). Surprisingly, this category includes many "thinking" models (notably Claude-Opus-4.5.long), challenging the assumption that reasoning enhancement necessitates verbose exploration (Chen et al., 2025a).

Hollow Mimic. Prevalent in smaller reasoning models (such as QwQ-32B), this category is defined by a severe mismatch between effort and outcome: they invest significant cost (0.45, 2nd highest) for mediocre logical depth (0.52). While their low Efficiency (0.50) and high Redundancy (0.60) mirror the Deep Wanderer, their Exploration is critically low ($\approx$ 0.09), meaning they generate text without expanding the logical search radius. Crucially, these models maintain high Process Coherence (0.42) comparable to successful solvers. This reveals a "performative reasoning" failure: explicit planning and reflection behaviors are correctly triggered and structured, but fail to translate into genuine logical progress.

Lazy Guesser. This cluster includes standard instruction models (Qwen2.5-32B-Instruct) and small reasoning models (Qwen3-4B.long) that fail at high complexities. They exhibit the lowest cost (0.21). Although their Efficiency score appears high (0.61), this is an artifact of brevity rather than competence. Counter-intuitively, their Redundancy is notable (0.51), indicating that "lazy" failure is rarely a concise refusal; instead, these models often get stuck in restatements, unable to propel the reasoning forward.

Model	Mode	Method	Data	Depth Δ
QwQ-32B	Long	SFT+RL	R+G	+97.1%
Qwen2.5-32B-In	Short	SFT	G	-
Claude-Opus-4.5	Long	-	-	+8.8%
	Short	-	-	-
Qwen3-235B-Th	Long	-	R+G	+1.6%
Qwen3-235B-In	Short	-	R+G	-
Qwen3-32B	Long	SFT+RL	R+G	+6.1%
	Short	SFT+RL	R+G	-
Qwen3-14B	Long	SFT	R+G	+6.5%
	Short	SFT	R+G	-

Table 3: Training Paradigm Analysis. We compare Long vs. Short modes at Complexity Level 11. **Th/In**: Thinking/Instruct. **Data**: R (Reasoning), G (General). Δ shows relative gain. Note that QwQ-32B and Qwen2.5-32B-In. represent early model iterations, while the others are current. For raw statistics and additional models, please refer to Table 16 in Appendix B.8.

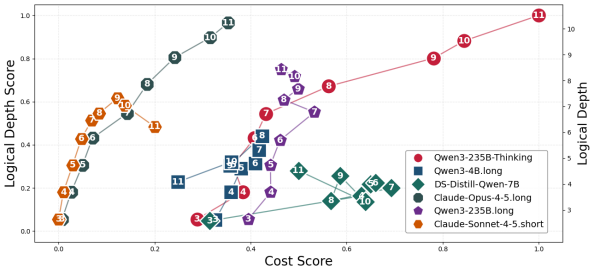


Figure 4: Performance trajectories across varying complexity levels (3–11). The visualization illustrates three distinct behavioral patterns defined in Section 4.3: (1) Adaptive Scaling (e.g., Qwen3-235B-thinking, Claude-Opus-4.5.long); (2) Diluted Expansion (e.g., Claude-Sonnet-4.5.short); and (3) Saturation (e.g., DS-Distill-Qwen-7B) & Collapse (e.g., Qwen3-4B.long). Complete trajectories for all evaluated models are presented in Figure 7 in Appendix B.8.

4.4 Dynamic Trajectories: Scaling with Complexity

We analyze the performance trajectories of models across varying complexity levels ($C = 3$ to 11). Figure 4 illustrates three representative archetypes identified from these trajectories: *Adaptive Scaling* (successful reasoning), and two failure modes—*Saturation/Collapse* (Shojaee et al., 2025) (insufficient effort) and *Diluted Expansion* (unproductive verbosity), as introduced below.

Adaptive Scaling. In an ideal scenario, both cost and logical depth increase as complexity grows. Models exhibiting this pattern correctly identify increased difficulty and generate proportional tokens to resolve it. For instance, Claude-Opus-4.5.long demonstrates high-efficiency solving, while Qwen3-235B-thinking achieves similar success through exhaustive exploration.

Saturation and Collapse. As complexity rises, some models (e.g., DS-Distill-Qwen-7B) fail to further scale their token generation, causing logical depth to plateau (*Saturation*). More critically, other models (e.g., Qwen3-4B.long) experience *Collapse* (Shojaee et al., 2025), where both token count and logical depth significantly decrease compared to lower complexity levels, indicating a breakdown in reasoning maintenance.

Diluted Expansion. Other models become verbose without being smarter (e.g., Claude-Sonnet-4.5.short). This pattern involves increased token count in response to difficulty without corresponding gains in logical depth. The implication is clear: for models lacking intrinsic reasoning abilities, scaling output length is necessary but insufficient.

5 Analysis

Following the identification of behavioral prototypes in Section 4, we examine the impact of inference mode (Long vs. Short CoT), training strategy, and model scale. We apply the $W = Pt$ intuition proposed in the Introduction to interpret these effects, formulating Logical Depth as **Work** (W) and Token Count as **Time** (t) to derive **Efficiency** ($P = W/t$).

5.1 Interpreting Behavioral Trajectories

Under this lens, the behavioral prototypes map to distinct dynamic trajectories governing the trade-off between Efficiency (P) and Cost (t). **Adaptive Scaling** represents the successful equilibrium where models effectively scale effort (t) to match rising difficulty (required W). Crucially, while successful strategies increase t in response to difficulty, they do so with distinct gradients. The *High-t Strategy* ("Deep Wanderer") rapidly expands the tokens to accumulate the necessary W, often relying on exhaustive search. Conversely, the *High-P Strategy* ("Effective Solver") also scales t, but at a significantly slower rate.

In contrast, failure modes represent breakdowns in this scaling mechanism. **Diluted Expansion** ("Hollow Mimic") mimics the rapid t-expansion of the High-t strategy ($\Delta t \gg 0$) but fails to reach proportional Logical Depth. Alternatively, **Saturation & Collapse** ("Lazy Guesser") occurs when the

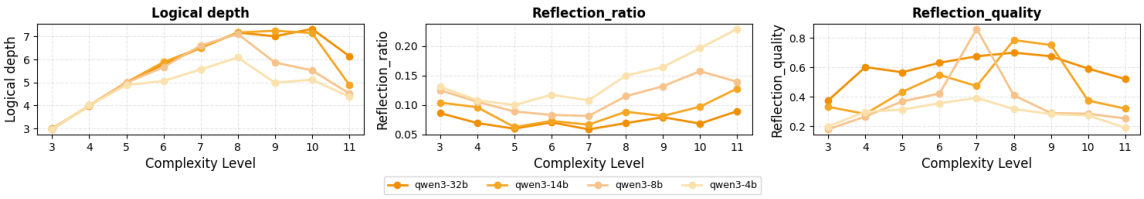


Figure 5: Qwen3-32B vs 14B vs 8B vs 4B. Three subplots compare key reasoning behaviors (logical depth, reflection ratio and quality) across model sizes as problem complexity increases. Smaller distilled models exhibit more sophisticated behaviors, but fail to emulate behavioral efficiency and cannot translate these sophisticated behaviors into deeper reasoning.

Model	Logical Depth	Reflection%
Claude-Opus-4.5.long	10.3	0.7%
Qwen3-235B-Instruct	10.0	1.8%
Claude-Opus-4.5.short	7.8	0.8%
Qwen3-235b.short	5.7	0.1%
Qwen3-32b.short	5.7	0.0%
Qwen2.5-32B-Instruct	2.9	0.0%

Table 4: Impact of Reflection on Logical Depth. Comparing Effective Solvers (top) against Lazy Guessers (bottom) reveals that successful models maintain reflection steps even in short reasoning process.

model fails to scale t. Facing increased complexity, the model either retreats to stagnant t or decreasing t, causing W to plateau or drop despite the need for greater depth.

5.2 Long CoT vs. Short CoT

According to Table 3, while early High-t models (Long CoT (Chen et al., 2025a)) hold a significant advantage in logical depth—e.g., QwQ-32B exceeds Qwen2.5-32B-Instruct by over 97%—current Short CoT (Wei et al., 2022) models are rapidly bridging this divide. The logical depth gap has narrowed to under 10% in modern iterations, with Qwen3-235B-Instruct trailing its thinking counterpart by a negligible 1.61%. Table 3 reveals that this convergence is methodology-agnostic yet correlated with the inclusion of reasoning data. This aligns with Yu et al. (2025), who observes that shortening long chains while preserving structure retains reasoning capability. Further, Table 4 indicates that high-performing short models consistently incorporate *reflection*, underscoring the critical role of "Short CoT + Reflection". Consequently, as short CoT achieve comparable depth at significantly reduced costs, deployment strategies are shifting from functional partitioning (Long for reasoning, Short for general) to efficiency trade-offs.

5.3 Mixed Training

We analyze Qwen3 models under pure "thinking" (Long CoT) vs. mixed (Long + Short) training. As visualized in Figure 4, while the pure Qwen3-235B-thinking maintains robust **Adaptive Scaling** by increasing t with complexity, the mixed Qwen3-235B.long suffers premature **Saturation**, failing to scale token generation significantly earlier. This structural degradation persists across other scales (pure QwQ-32B vs. mixed Qwen3-32B.long, pure Qwen3-30B-thinking vs. mixed Qwen3-30B.long), as detailed in Figure 9 in Appendix B.8.

We suggest that incorporating short-CoT data (High-P) risks negatively interfering with the High-t mechanism, creating a reluctance to scale computation when necessary. This interference offers a potential explanation for why Qwen3-2507 (Qwen Team, 2025) has abandoned mixed training strategies in favor of specialized tuning.

5.4 Distillation

We examine the limits of distilling Long CoT (High-t) strategies from larger teachers into smaller students using the Qwen3 lineage (14B, 8B, 4B) supervised by a 32B teacher. As visualized in Figure 5, smaller models (4B, 8B) generate more reflection steps than the teacher at lower difficulties.

However, this appearance is deceptive. While the *quantity* (frequency) of reflection mimics or exceeds the teacher, the semantic *quality* degrades strictly with model size ($32B > 14B > 8B > 4B$), failing to translate into Logical Depth (W). We observe an identical dissociation between frequency and quality in planning behaviors, as detailed in Figure 10 in Appendix B.8. In contrast, Qwen3-14B marks a clear divergence: it successfully maintains both reflection quality and logical depth aligned with the 32B teacher. This distinct separation indicates that without sufficient parametric capacity (with 14B emerging as the critical

threshold), forcing a Long CoT strategy results in "Diluted Expansion"—the model mimics the form of reasoning to minimize distillation loss without grasping the intrinsic logic.

6 Conclusion

We introduce a neuro-symbolic framework that grounds natural language reasoning in FOL to deterministically quantify reasoning efficiency of LLMs. Through this lens, we identify four reasoning prototypes and diagnose their behavioral characteristics. Our analysis reveals that Long CoT is not necessary for deep reasoning. Furthermore, mixing long and short CoT in training risks strategy interference, while distillation often yields behavioral mimicry. Our dataset, ReEfBench, and the evaluation method can be used for efficiency evaluation of the reasoning process.

Limitations

We acknowledge three limitations. First, restricting evaluation to First-Order Logic (FOL) prioritizes verification rigor over generality; consequently, our findings may not strictly apply to domains like open-ended QA. Second, our non-intrusive design limits our analysis to verifying the logic of the naturally generated text, rather than forcing the externalization of implicit reasoning steps. Finally, regarding the four reasoning archetypes: while these patterns are real, the boundaries are not absolute. A model's specific category is relative and can be fuzzy; however, by analyzing a diverse set of models, we ensure that the overall classification and statistical trends remain objective.

Acknowledgments

This publication has been supported by the National Natural Science Foundation of China (NSFC) Key Project under Grant Number 62336006.

References

Anthropic. 2025a. Introducing Claude Opus 4.5. Accessed: 2026-01-03.

Anthropic. 2025b. Introducing Claude Sonnet 4.5. Accessed: 2026-01-03.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025a. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *Preprint*, arXiv:2503.09567.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025b. Do NOT think that much for 2+3=? on the overthinking of long reasoning models. In *Forty-second International Conference on Machine Learning*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Jinlan Fu, See Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. Gptscore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576.

Kanishk Gandhi, Ayush K Chakravarthy, Anikait Singh, Nathan Lile, and Noah Goodman. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective STars. In *Second Conference on Language Modeling*.

Olga Golovneva, Moya Peng Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. ROSCOE: A suite of metrics for scoring step-by-step reasoning. In *The Eleventh 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, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, and 16 others. 2024. FOLIO: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22017–22031, Miami, Florida, USA. Association for Computational Linguistics.

Alex Heyman and Joel Zylberberg. 2025. Evaluating the systematic reasoning abilities of large language models through graph coloring. *arXiv preprint arXiv:2502.07087*.

Chungpa Lee, Thomas Zeng, Jongwon Jeong, Jy-yong Sohn, and Kangwook Lee. 2025. How to correctly report llm-as-a-judge evaluations. *arXiv preprint arXiv:2511.21140*.

Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. 2025. Zebralogic: On the scaling limits of llms for logical reasoning. *arXiv preprint arXiv:2502.01100*.

Yiwei Liu, Yucheng Li, Xiao Li, and Gong Cheng. 2025. Loginumsynth: Synthesizing joint logical-numerical reasoning problems for language models. *arXiv preprint arXiv:2510.11031*.

Allen Nie, Yi Su, Bo Chang, Jonathan N Lee, Ed H Chi, Quoc V Le, and Minmin Chen. 2024. Evolve: Evaluating and optimizing llms for exploration. *arXiv preprint arXiv:2410.06238*.

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

OpenAI. 2024. Learning to reason with LLMs. Technical report.

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

Shubham Parashar, Blake Olson, Sambhav Khurana, Eric Li, Hongyi Ling, James Caverlee, and Shuiwang Ji. 2025. Inference-time computations for llm reasoning and planning: A benchmark and insights. *arXiv preprint arXiv:2502.12521*.

Keqin Peng, Liang Ding, Yuanxin Ouyang, Meng Fang, and Dacheng Tao. 2025. Revisiting overthinking in long chain-of-thought from the perspective of self-doubt. *Preprint*, arXiv:2505.23480.

Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. 2023. ReCEval: Evaluating reasoning chains via correctness and informativeness. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10066–10086, Singapore. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Qwen Team. 2025. Qwen3-235b-a22b-thinking-2507. `https://huggingface.co/Qwen/Qwen3-235B-A22B-Thinking-2507`.

Abulhair Saparov and He He. 2023. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *The Eleventh International Conference on Learning Representations*.

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

Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. 2025. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. *arXiv preprint arXiv:2506.06941*.

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

Xiangyu Wang, Haocheng Yang, Fengxiang Cheng, and Fenrong Liu. 2025. Adaptive selection of symbolic languages for improving llm logical reasoning. *Preprint*, arXiv:2510.10703.

Anjiang Wei, Yuheng Wu, Yingjia Wan, Tarun Suresh, Huanmi Tan, Zhanke Zhou, Sanmi Koyejo, Ke Wang, and Alex Aiken. 2025. SATBench: Benchmarking LLMs' logical reasoning via automated puzzle generation from SAT formulas. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33820–33837, Suzhou, China. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. 2025. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *Preprint*, arXiv:2502.14768.

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

Bin Yu, Hang Yuan, Haotian Li, Xueyin Xu, Yuliang Wei, Bailing Wang, Weizhen Qi, and Kai Chen. 2025. Long-short chain-of-thought mixture supervised fine-tuning eliciting efficient reasoning in large language models. *Preprint*, arXiv:2505.03469.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, and 1 others. 2025. A survey on test-time scaling in large language models: What, how, where, and how well? *arXiv preprint arXiv:2503.24235*.

Yujun Zhou, Jiayi Ye, Zipeng Ling, Yufei Han, Yue Huang, Haomin Zhuang, Zhenwen Liang, Kehan Guo, Taicheng Guo, Xiangqi Wang, and 1 others. 2025. Dissecting logical reasoning in llms: A fine-grained evaluation and supervision study. *arXiv preprint arXiv:2506.04810*.

Category	Sub-type	Formal Definition	Description
Planning	-	-	Meta-cognitive statements outlining strategy (e.g., "Let us assume...") without asserting a logical conclusion.
Actual	**Premise**	$s \in \mathcal{P}$	Fundamental facts explicitly provided in the problem description.
	Derived	$\mathcal{P} \vdash s$	Logically valid deductions implied by $\mathcal{P}$
	Hallucination	$\mathcal{P} \nvdash s$	Statements that cannot be logically inferred from $\mathcal{P}$

Table 5: Taxonomy of reasoning node types. Nodes are first classified into *Planning* or *Actual* steps by the parser. Actual steps are further categorized based on their logical validity and relationship to the canonical ground truth graph $\mathcal{G}^\star$ and premise set $\mathcal{P}$.

Level	Inst.	Distr.	Avg Prem.	Avg Steps
3	100	3	6.3	11.2
4	100	4	10.8	20.4
5	100	5	18.0	35.6
6	100	6	26.6	55.1
7	100	7	41.7	89.8
8	100	8	70.5	158.0
9	100	9	115.8	266.2
10	100	10	186.4	435.2
11	100	11	309.8	732.1

Table 6: Dataset statistics for the main experiments. Inst., Distr., and Prem. denote instances, distractors, and average premises, respectively. Both the number of premises and the required proof steps scale exponentially with the complexity level.

Level	Ann. 1	Ann. 2	Ann. 3	Average
Level 3	100%	100%	100%	**100%**
Level 5	40%	25%	30%	**32%**

Table 7: Human baseline performance under a strict 20-minute time limit. Accuracy drops significantly at Level 5 due to cognitive overload caused by the expanding premise count.

Archetype	LD.	Engage.	Behavioral Trait
Effective Solver	High	Low	*Efficient & Correct*
Deep Wanderer	High	High	*Exhaustive & Correct*
Lazy Guesser	Low	Low	*Direct Failure*
Hollow Mimic	Low	High	*Inefficient Failure*

Table 8: Definition of Behavioral Archetypes based on the Quadrants of the Logical Depth-Cost Plane.

A Methodology Supplement

A.1 Dataset Details

We calculate the accuracy of all the models evaluated with data from Complexity Level 3 to Level 11 (Table 9). Despite the fundamental logic relying on simple Modus Ponens combined with and/or operators, the most advanced model (Claude-Opus-4.5.long) exhibits a model accuracy still declines rapidly (0.62) under high complexity. For clarity, models bearing the suffixes .long or .short (e.g., Claude-Sonnet-4.5.long, Qwen3-32B.short) denote dual-mode variants of the same model. Models without such suffixes are single-mode versions. Regarding model versions and release dates: the Qwen3-235B-thinking, Qwen3-235B-instruct, Qwen3-30B-thinking, and Qwen3-30B-instruct models are specialized single-mode variants derived from their respective base models (Qwen3-235B and Qwen3-30B) and correspond to the 2507 training snapshot. The two Claude-Opus-4.5 variants were released on November 1 (1101), while the Claude-Sonnet-4.5 variants date from September 29 (0929). DeepSeek-R1 was released on May 28 (0528).

A.2 Table for deduction rules

Table 10 shows the deduction rules we apply to construct our datasets.

A.3 Main Concepts for Methodology

Logical Graph. A *Logical Graph* is a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \tau)$, where:

- $\mathcal{V}$ is a set of *statements*, each has form $X \vdash Y$;
- $\mathcal{E}$ encodes inference dependencies (i.e., an edge (u, v) exists if v is derived using u);
- τ assigns each node a type label from PREMISE, DERIVED, PLANNING, or HALLUCINATION.

A *logical complete tree* is a tree in which all leaf-to-root paths have equal length. The *canonical LoG*, denoted $\mathcal{G}^\star = (\mathcal{V}^\star, \mathcal{E}^\star, \tau^\star)$, is the unique logical complete tree that (i) contains all and only sound inferences from $\mathcal{P}$, and (ii) excludes all HALLUCINATION nodes.

Model	Level_3	Level_4	Level_5	Level_6	Level_7	Level_8	Level_9	Level_10	Level_11
Claude-Sonnet-4.5.long	100%	100%	100%	100%	99%	99%	98%	92%	62%
Claude-Sonnet-4.5.short	100%	99%	99%	94%	85%	56%	66%	45%	33%
Claude-Opus-4.5.long	100%	100%	100%	100%	97%	99%	98%	87%	63%
Claude-Opus-4.5.short	100%	100%	100%	100%	95%	74%	56%	49%	32%
Deepseek-R1	100%	100%	100%	99%	97%	79%	70%	25%	17%
Qwen3-235B.long	98%	96%	97%	92%	90%	73%	39%	25%	14%
Qwen3-235B.short	97%	95%	92%	62%	74%	47%	23%	15%	7%
Qwen3-30B.long	100%	91%	90%	54%	28%	9%	2%	3%	0%
Qwen3-30B.short	98%	83%	69%	41%	24%	9%	6%	3%	5%
Qwen3-30B-thinking	78%	72%	92%	79%	48%	33%	9%	3%	0%
Qwen3-30B-instruct	92%	91%	86%	69%	59%	43%	17%	14%	7%
Qwen3-235B-thinking	94%	99%	100%	100%	94%	91%	82%	63%	51%
Qwen3-235B-instruct	98%	96%	98%	98%	99%	95%	82%	49%	30%
Qwen2.5-32B-instruct	93%	54%	40%	25%	15%	11%	8%	6%	5%
QwQ-32B	99%	95%	91%	84%	57%	26%	10%	6%	1%
DS-R1-Qwen-32B	99%	92%	93%	74%	35%	16%	10%	10%	9%
DS-R1-Qwen-7B	56%	29%	32%	9%	7%	3%	1%	1%	3%
Qwen3-32B.short	93%	93%	94%	62%	68%	40%	16%	13%	11%
Qwen3-32B.long	98%	99%	100%	79%	74%	50%	20%	11%	4%
Qwen3-14B.long	97%	94%	95%	80%	69%	42%	27%	4%	3%
Qwen3-14B.short	90%	84%	75%	43%	49%	32%	19%	15%	10%
Qwen3-8B.long	95%	93%	89%	76%	52%	33%	8%	3%	0%
Qwen3-8B.short	81%	72%	68%	60%	35%	19%	11%	6%	0
Qwen3-4B.long	94%	92%	77%	28%	22%	9%	0%	2%	0%
Qwen3-4B.short	90%	70%	54%	37%	35%	16%	13%	6%	2%

Table 9: Model accuracy across difficulty levels (Level 3 to Level 11), showing performance degradation as complexity increases, with notable variations among models and configurations (e.g., .long and .short).

Deduction rule	Formal definition	Natural language example
Modus Ponens	$\frac{f(a) \quad \forall x\,(f(x) \vdash g(x))}{g(a)}$	Alex is a cat. All cats are carnivores. Alex is a carnivore.
Conjunction Introduction	$\frac{A \quad B}{A \wedge B}$	Alex is a cat. Alex is orange. Alex is a cat and orange.
Conjunction Elimination	$\frac{A \wedge B}{A}$	Alex is a cat and orange. Alex is a cat.
Disjunction Introduction	$\frac{A}{A \vee B}$	Alex is a cat. Alex is a cat or orange.

Table 10: Four deduction rules used in our dataset, derived from Modus Ponens with logical conjunction ($\wedge$) and disjunction ($\vee$). Adapted from Saparov et al. (2023).

Logical Depth (D). We define logical depth at the statement level. Let $\mathcal{S}_0 = \mathcal{P}$. The depth of a statement s is the smallest k such that s can be derived in k inference steps:

$$\text{depth}(s) = \begin{cases} 0 & \text{if } s \in \mathcal{P}, \\ k & \text{if } s \in \mathcal{S}_k \text{ and } s \notin \bigcup_{i=0}^{k-1} \mathcal{S}_i, \\ -1 & \text{if } \mathcal{P} \nvdash s \text{ (i.e., hallucination).} \end{cases}$$

For non-canonical graphs, we compute depth recursively. Define $\mathcal{C}_0 = \mathcal{P}$, and for $k \geq 0$:

$$\mathcal{C}_{k+1} = \left\{ Z \;\middle|\; \begin{array}{l} \exists m \geq 1,\ \exists Y_1, \ldots, Y_m \text{ s.t.} \\ (Y_1 \wedge \cdots \wedge Y_m) \vdash Z \text{ is valid}, \\ Y_1, \ldots, Y_m \in \bigcup_{t=0}^{k} \mathcal{C}_t, \\ \max_{1 \leq \ell \leq m} \min\{t \mid Y_\ell \in \mathcal{C}_t\} = k \end{array} \right\}.$$

Then $\text{depth}(s) = k$ iff $s \in \mathcal{C}_k \setminus \bigcup_{t<k} \mathcal{C}_t$. Importantly, depth is *unique* for any sound statement, regardless of the derivation path.

Moreover, the number of premises required to derive a conclusion at depth d scales as $n_{\text{premise}} \sim m^d$,

Algorithm 1 Logical Reasoning Graph Generation

Require: Maximum reasoning depth H;
Element vocabulary $\mathcal{E}$;
Deduction rules $\{\text{MP}, \text{CE}, \text{CI}, \text{DI}\}$;
Hard mode flag HARD
Ensure: A logical reasoning graph G
Initialize unused elements $\mathcal{E}' \leftarrow \mathcal{E}$
Sample an initial conclusion c_0 using $1 \sim 3$ elements from $\mathcal{E}'$
Initialize graph G with root node c_0
for $d = 1$ to H **do**
  for each leaf conclusion c at depth d **do**
    Select a valid deduction rule r
    Generate premises $\mathcal{P} \leftarrow r(c)$
    Add $(\mathcal{P} \Rightarrow c)$ to G
  end for
end for
if HARD **then**
  Add irrelevant premises as distractors at maximum depth
end if
return G

where m is a constant branching factor determined by the dataset.

Logical Breadth (B and $B^\star$). We define *logical breadth* as the size of the reachable logical closure:

$$B = |\{s \mid \mathcal{P} \vdash s\}|,$$

i.e., the total number of distinct statements derivable from $\mathcal{P}$. The *minimal necessary breadth* $B^\star$ is the size of the smallest subset of this closure that suffices to derive the target conclusion, which captures the essential reasoning scope required for a given task.

Reasoning Cost (t). We proxy computational cost by the number of tokens in the model's CoT response, denoted t. While imperfect, this provides a practical, observable measure of reasoning effort.

A.4 Definition of Node Types

Table 5 shows the definition of all node types.

A.5 Algorithm Details

`data_generation` This method automatically generates multi-hop logical reasoning graphs using a fixed set of deductive rules (Modus Ponens, Conjunction Elimination, Conjunction Introduction, Disjunction Introduction). Starting from a randomly constructed conclusion, the algorithm recursively expands the graph backward by selecting admissible inference rules under structural constraints to avoid trivial or cyclic reasoning. Each graph is generated via breadth-first expansion up to a predefined depth, and an optional hard mode augments the premises with logically irrelevant distractors. The resulting graphs are finally converted into question–answer pairs for evaluating logical reasoning capabilities.

Algorithm 2 Backward Chaining Proof Search

function ISPROVABLE(τ, Π, V, d, t_0)
  Input: target τ, premises Π, visited V, depth d, start time t_0
  Output: $(provable, trace)$
  if timeout or $d > d_{max}$ or $\tau \in V$ **then return** $(\bot, \emptyset)$
  end if
  $V \leftarrow V \cup \{\tau\}$
  if $\exists \pi \in \Pi : \tau \equiv \pi$ **then return** $(\top, \{\pi\})$
  end if
  $\mathcal{P} \leftarrow$ FINDPATHS(τ, Π) ▷ Get inference paths
  Sort $\mathcal{P}$ by $(|\mathcal{I}_p|, \text{priority}(r_p))$ ▷ $\mathcal{I}_p$: intermediates, r_p: rule
  for each path $p \in \mathcal{P}$ **do**
    $\Pi_p \leftarrow \emptyset$, $provable \leftarrow \top$
    for each intermediate $\iota \in \mathcal{I}_p$ **do**
      $(v, \Pi_\iota) \leftarrow$ ISPROVABLE$(\iota, \Pi, V', d+1, t_0)$ ▷ $V' = V$
      if $\neg v$ **then** $provable \leftarrow \bot$; **break**
      end if
      $\Pi_p \leftarrow \Pi_p \cup \Pi_\iota$
    end for
    if $provable$ **then**
      $V \leftarrow V \setminus \{\tau\}$
      return $(\top, \Pi_p)$
    end if
  end for
  $V \leftarrow V \setminus \{\tau\}$
  return $(\bot, \emptyset)$
end function

`is_provable` This function implements a backward chaining algorithm that: 1. Checks if a target conclusion can be derived from given premises 2. Uses memoization (visited set) to prevent circular reasoning 3. Enforces timeout and depth limits to prevent infinite loops 4. First checks if the target is directly in premises (base case) 5. Then explores multiple reasoning paths using inference rules (MP, CE, CI, etc.) 6. Recursively proves intermediate steps needed for each path 7. Returns the first successful proof path found, with optional trace information.

`get_equivalent_depth` This function computes an "equivalent depth" for a logical node by: 1. Checking if it already exists in the LoG tree (returns depth-1) 2. Attempting to prove it from premises and tracking which premises are used 3.

Algorithm 3 Backward Chaining Proof Search

```
function IsProvable(τ, Π, V, d)
    if d > d_max or τ ∈ V then return ⊥
    end if
    if ∃π ∈ Π : τ ≡ π then return ⊤
    end if
    V ← V ∪ {τ}
    for path (r, I) ∈ FindPaths(τ, Π) do
        if ∀ι ∈ I : IsProvable(ι, Π, V, d+1) then
            return ⊤
        end if
    end for
    return ⊥
end function
```

Algorithm 4 - Get Equivalent Depth

```
function GetEquivalentDepth(node, logTree)
    n ← FindLogNodeByOutput(node.original)
    if n ≠ null then return n.depth − 1
    end if
    (ok, tr) ← IsProvable(node, {s ∈ stmtList | s.type = "premise"})
    if ¬ ok then return −1
    end if
    used ← {p.original | p ∈ tr.usedPremises}
    minD ← logScale; exact ← false
    for all n in logTree do
        if n.depth < minD ∧ (n.req ⊆ used ∨ Pred(n.output) = node.output) then
            minD ← n.depth; exact ← (n.req = used)
        end if
    end for
    return if minD = logScale then 0 elif exact then minD−1 else Max(minD−2, 0)
end function
```

Finding the shallowest matching node in the LoG tree via two strategies: Premise coverage: nodes whose required premises are covered by the proof; Output predicate matching: nodes with identical output predicates 4. Returning adjusted depth based on match quality (exact match: depth-1, partial match: depth-2).

planning and reflection's context window The effective context window for planning nodes comprises the 5 trailing sentences relative to the planning node position, inclusive of the current sentence under consideration.

Similarly, the impact window for reflection sentences extends over a span of 5 sentences when computing reflection-associated metrics, such as interval gain.

B Experiment Supplement

B.1 Details for Models

Table 11 shows thinking Mode, Model Source, training method and training data for each models.

B.2 Evaluation metrics

In this subsection, we enumerate all metrics implemented in our codebase. The metrics are organized into a two-tier structure: (1) a high-level taxonomy grouping metrics into *Core* and *Diagnostic* categories (Table 12), which guide process-level evaluation; and (2) a comprehensive list of base and derived metric implementations at sentence and node levels (Table 13).

Node-Level Metrics. For node definition, refer to Table 5. We evaluate actual nodes by their count (premise/derived/hallucination), correctness, and depth. Planning nodes are assessed by count, correctness, and effectiveness (whether they generate new actual nodes within a given window). Derived metrics include exploration precision, reasoning accuracy, premise and depth coverage, node duplication ratio, and incorrect ratio.

Sentence-Level Metrics. We track basic verbosity (sentence count, token count, node count) and distinguish reflection sentences by their count and ratio. Efficiency metrics include first correct step, step efficiency (depth advancement per expenditure), node efficiency (correct nodes per sentence), and reflection efficiency. We measure reasoning spans (effective, forward, and overall) based on the relative position of the last novel step. For reflection sentences specifically, we track whether their context windows produce new nodes, deeper nodes, or hallucinations. Finally, we compute sentence duplication ratio to identify repeated reasoning patterns.

B.3 Generation Hyperparameters and Prompts

The hyperparameters employed for API invocation comprised: temperature = 0, maximum token allocation of 24,000 for reasoning-enabled models and 8,000 for non-reasoning models, with all other API parameters maintained at their default values.

The prompt we used for each LoG question is as follows:

Model	Thinking Mode	Source	Training Method	Data
Qwen Series				
QwQ-32B	Long	Open	SFT + RL	reason + general
Qwen3-235B	Long / Short	Open	SFT + RL	reason + general
Qwen3-235B-Instruct	Short	Open	-	reason + general
Qwen3-235B-thinking	Long	Open	-	reason + general
Qwen3-30B	Long / Short	Open	SFT	reason + general
Qwen3-30B-Instruct	Short	Open	-	reason + general
Qwen3-30B-thinking	Long	Open	-	reason + general
Qwen3-4B/8B/14B	Long / Short	Open	SFT	reason + general
Qwen3-32B	Long / Short	Open	SFT + RL	reason + general
Qwen2.5-32B-Instruct	Short	Open	SFT	general
Claude Series				
Claude-Opus-4.5	Long/Short	Closed	-	-
Claude-Sonnet-4.5	Long/Short	Closed	-	-
DeepSeek Series				
DeepSeek-R1(671B)	Long	Open	SFT + RL	reason + general
DS-R1-Qwen-7B	Long	Open	SFT	reason + general
DS-R1-Qwen-32B	Long	Open	SFT	reason + general

Table 11: Model training methods classification, detailing Thinking Mode, Source (Open/Closed), Training Method (SFT, RL), and Data type for models across Qwen, Claude, and DeepSeek series.

"Please answer the question based on the given information:
Given Information: {tmp_information}
Note: In this context, 'A is B' has the same meaning as 'a rabbit is a mammal' — it means A belongs to category B, not that A equals B.
Question: {tmp_question}
Please reason step by step, show your reasoning process and put your final answer in \boxed{}."

B.4 Robustness of the Evaluation Pipeline

Parser Stability Across Complexities. A potential concern regarding LLM-based evaluation is whether the parser's accuracy degrades as the target model's reasoning chain becomes longer or more complex. In our framework, the parser's accuracy is invariant to the global reasoning complexity because our pipeline first segments the entire reasoning response into individual sentences. The parser then processes these atomic units sequentially. Therefore, regardless of whether the reasoning depth is 3 or 11, the linguistic complexity of the input unit (the single sentence) remains relatively constant.

Table 15 provides typical examples randomly sampled from standard CoT (*Qwen2.5-32B-Instruct*) and Long-CoT (*DeepSeek-R1*) responses. Even though Long-CoT models produce linguistically richer and more convoluted sentences, the parser reliably breaks them down into atomic logical units without loss of fidelity. Note that while most actual logic nodes follow the standard predicate format ("A is B"), planning nodes sometimes consist solely of the target entity.

Impact of Context Window Size In our framework, the efficacy of meta-cognitive nodes (e.g., planning and reflection) is evaluated based on whether they tangibly aid subsequent reasoning steps within a local window (denoted as W). Theoretically, a smaller window risks under-recalling valid downstream impacts, while a larger window risks over-recalling coincidental progress.

To determine if our evaluation is overly sensitive to this hyperparameter, we conduct an ablation study using window sizes $W \in \{3, 5, 7\}$ across all 25 evaluated models, focusing on the diagnostic metrics directly affected by this window length (Reflection Efficiency and Valid Planning). As shown in Table 14, while absolute metric scores naturally exhibit slight variations, the relative rankings of the models remain remarkably stable. The Spearman's rank correlation (ρ) between our default setting ($W = 5$) and alternative settings is exceptionally high (averaging > 0.94), and the pairwise inversion rates are extremely low (mostly under 8.5%). We thus conclude that our evaluation framework is highly robust, and the comparative insights drawn in our paper are independent of the choice of the local window.

B.5 Dataset Statistics and Human Baseline

Dataset Statistics It should be noted that our data construction framework (Phase A) serves as a scal-

Category	Metric	Code Metric	Meaning
Core	*Logical Depth* (S_{ld})	Max Depth	Average maximum correct logical depth achieved
	Cost (S_{cost})	Reflection Count Planning Count Verbosity	# of reflection steps during reasoning # of planning nodes during reasoning square root of # of tokens generated during reasoning
Diagnostic	*Exploration*	Explored Node Count	# of actual nodes provable from given premises
	Efficiency	Step Efficiency Effective Span	Effective logical depth advancement per token Relative position of last novel and correct step
	Coherence	Reflection Efficiency Valid Planning	Depth gain induced by reflection in a given window Proportion of *valid* planning steps executed
	Redundancy	Sentence Duplication Node Duplication Ratio	Proportion of duplicated sentences during reasoning Repeated visits to identical logical nodes

Table 12: Taxonomy of reasoning behavior metrics. Core metrics define the primary evaluation axes (Logical Depth, Cost), while diagnostic metrics capture process-level characteristics (Exploration, Efficiency, Coherence, Redundancy) underlying reasoning performance. Within each group, all metrics are first normalized to ensure comparability across metrics, and are then aggregated—by weighted averaging—into a single group-level score. For verbosity, we apply a square root transformation prior to normalization to mitigate the high variance in raw token counts.

able generator, allowing for customizable dataset sizes and complexities. For our main experiments, we generated 100 instances per complexity level ($C = 3$ to $C = 11$). To dynamically scale the difficulty of the reasoning context, we established a rule where complexity Level C incorporates exactly C logically irrelevant distractor sentences.

Table 6 provides the comprehensive statistics for the generated dataset used in our evaluation. As complexity increases, both the average number of necessary premises and the total proof steps scale exponentially. For instance, while Level 3 problems require an average of 6.3 premises, Level 11 problems demand managing nearly 310 premises and executing over 700 atomic proof steps, creating an extreme stress test for LLM context management.

Human Baseline To benchmark LLM performance against human cognition, we conduct a small-scale human evaluation. We evaluate three independent human annotators on a subset of Level 3 and Level 5 tasks (20 samples each) under a strict 20-minute time limit.

As shown in Table 7, human accuracy is perfect (100%) at Level 3 but drops precipitously to an average of 32% at Level 5. This drastic degradation is expected: as shown in Table 6, Level 5 problems contain an average of 18 premises. The complexity escalates even further at Level 6 (≈ 27 premises) and Level 7 (≈ 42 premises), which far exceeds human cognitive capacity and working memory limits for manual verification.

B.6 K-means Classification Result

To move beyond isolated metric analysis, we employ a semi-supervised clustering framework to identify generalized behavioral archetypes:
(1) **Unsupervised Clustering & Semantic Mapping:** We first apply K-means clustering ($k = 4$) on the normalized 2D feature space defined by Logical Depth and Cost. To interpret the clusters, we define four "Ideal Archetypes" corresponding to the quadrants of the plane (Table 8) and utilize the Hungarian Algorithm to map empirical centroids to these semantic labels.
(2) **Boundary-Relative Confidence:** To rigorously quantify how "typical" a model is of its category, we propose a boundary-aware confidence score (S_c). Unlike simple centroid distance, this metric considers the geometric decision boundaries (Voronoi partitions). For a model point P assigned to cluster centroid C_{own}, the confidence is defined:

$$S_c(P) = \min_{C_{enemy}} \left(\min \left(1.0, \frac{d(P, B)}{d(C_{own}, B)} \right) \right) \tag{1}$$

where $d(\cdot, B)$ denotes the perpendicular distance to the boundary. Intuitively, $S_c = 1.0$ indicates the model resides deeper in its region than the centroid itself (hyper-typical), while $S_c \approx 0$ implies the model lies on the decision boundary.

B.7 Model Setting Details

Table 11 shows the Training paradigms and datasets of all models evaluated. Table 12 illustrates the taxonomy of metrics.

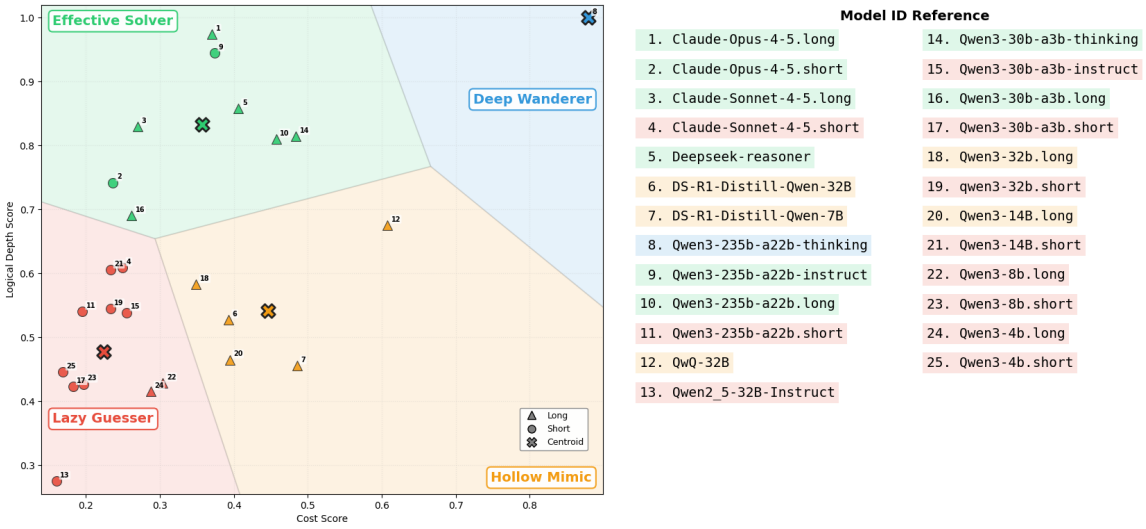


Figure 6: K-means Classification

B.8 Experiment Results Details

Table 16 summarizes the logical depth and the number of tokens used by different models. Table 17 is the full version of Table 2. Figure 7 illustrates the variations in Logical Depth and Cost across all models under different levels of complexity. Figure 8 shows the comparison of short models with and without reflection.

C Analysis Supplement

C.1 analysis plots

We present additional empirical insights through two key comparative visualizations. Figure 9 illustrates the performance divergence between separately trained and mixed training configurations across 32B, 30B, and 235B reasoning models, highlighting the detrimental impact of mixed training on sustaining deep logical Cost. Complementarily, Figure 10 examines the behavioral fidelity of distilled smaller models (4B–14B) relative to the Qwen3-32B teacher, revealing that while smaller models can mimic sophisticated reasoning patterns, only the 14B variant successfully aligns both behaviorally and capability-wise with the teacher, underscoring the intrinsic limitations of token-efficient reasoning in under-capacitated models.

Category	Level	Type	Metric	Meaning
Base	Node	Actual node	Count	Number of premise/derived/ hallucination nodes
			Correctness	Whether actual node is correct with given premise
			Depth	Log node depth or calculated equivalent depth
		Planning node	Count	Number of planning nodes & unique
			Correctness	Statement can be proved with given premises
			Effectiveness/valid	New actual node achieved from planning in given window
	Sentence	Verbosity	Sentence count	Number of sentences during reasoning
			Token count	Number of tokens during reasoning
			Node count	Number of nodes during reasoning
		Reflection	Sentence count	Number of reflection sentences
		Reflection effect	Has new node	Count if reflection window creates new node
			Has deeper node	Count if reflection window creates deeper node
			Has new hallucination	Count if reflection window creates new hallucination node
Derived	Node	All node	Node Duplication Ratio	Repeated visits to identical logical nodes
		Actual node	Exploration precision	Proportion of explored nodes on minimum log graph
			Reasoning accuracy	Proportion of correct nodes on minimum log graph
			Premise coverage	Proportion of premise used
			Depth coverage	Proportion of depth covered during reasoning
			Depth	Average maximum correct logical depth achieved
			Incorrect ratio	Proportion of hallucination actual nodes
			Interval depth	Length of complete reasoning subtree interval
	Sentence	Efficiency	First correct step	First sentence containing correct nodes
			Step Efficiency	Effective logical depth advancement per expenditure
			Node Efficiency	Average correct nodes per sentence
			Reflection Efficiency	Depth gain induced by reflection in given window
		Spans	Effective Span	Relative position of last novel and correct step
			Forward reasoning span	Relative position of last novel and deepest step
			Reasoning span	Relative position of last novel step
		Duplication	Sentence duplication ratio	Proportion of duplicated sentences during reasoning

Table 13: Full reasoning evaluation metrics, categorized into Base (node/sentence-level counts and correctness) and Derived (aggregated performance indicators like exploration, efficiency, span, and duplication ratios) for comprehensive behavioral analysis.

Metric	Setting Pair	Avg Spearman ρ	Min ρ	Avg Inversion Rate	Max Inversion Rate
Reflection Efficiency	$W = 3$ vs $W = 5$	0.9475	0.8964	6.53%	11.41%
	$W = 5$ vs $W = 7$	0.9628	0.9155	4.91%	9.59%
Valid Planning	$W = 3$ vs $W = 5$	0.9458	0.8700	8.24%	14.67%
	$W = 5$ vs $W = 7$	0.9751	0.9546	4.86%	6.67%

Table 14: Ablation study on context window size (W). High Spearman correlation and low inversion rates demonstrate that model rankings are highly stable across different window settings.

Model	Segmented Sentence (Input)	Parsed Node Type	Parsed Logic Nodes
Qwen2.5-32B-Instruct	- Given: dohcpus is liydpus.	Actual	dohcpus is liydpus
	- yuhvpus is vontpus.	Actual	yuhvpus is vontpus
	Identify yunqpus: "yunqpus is tawjpus.	Planning Actual	yunqpus yunqpus is tawjpus
	- Given: corbpus is qozwpus.	Actual	corbpus is qozwpus
	Determine what each of these categories (neywpus, xuxspus, bacdpus, xalwpus) can be: - neywpus is bactpus.	Planning Planning Planning Planning Actual	neywpus xuxspus bacdpus xalwpus neywpus is bactpus
DeepSeek-R1	First, I need to determine if x is both babgpus and babkpus.	Planning	x is babgpus and babkpus
	- jeqhpus is babkpus and muzvpus and buzmpus and zoyqpus.	Actual	jeqhpus is babkpus and muzvpus and buzmpus and zoyqpus
	So, I need to see if x (woprpus) is eventually a member of babkpus or if there's a path to babkpus.	Planning	x is babkpus
	yatypus is vemlpus, so yatypus belongs to vemlpus.	Actual Actual	yatypus is vemlpus yatypus is vemlpus
	In the given, it says "vemlpus is liqbpus" and "vemlpus is hujgpus", so vemlpus belongs to both liqbpus and hujgpus.	Actual Actual Actual	vemlpus is liqbpus vemlpus is hujgpus vemlpus is liqbpus and hujgpus

Table 15: Examples of sentence-level parsing across different models. Each logic node extracted by the parser corresponds to a specific node type, ensuring a transparent mapping between the model's natural language response and the formal representation.

Model	Type	Avg Depth	Avg Depth Delta	Avg Token	Avg Token Delta
QwQ-32B	long	5.96	97.13%	4336.67	879.18%
DS-R1-Qwen-32B	long	5.06	67.45%	4283.78	867.24%
Qwen2.5-32B-Instruct	short	3.02	-	442.89	-
Claude-Opus-4.5	long	6.88	8.76%	1398.00	108.24%
	short	6.32	-	671.33	-
Claude-Sonnet-4.5	long	6.68	14.20%	931.67	41.66%
	short	5.85	-	657.67	-
Qwen3-235B-thinking	long	6.88	1.61%	6850.44	295.70%
Qwen3-235B-Instruct	short	6.77	-	1731.22	-
Qwen3-235B	long	6.33	17.31%	4127.33	429.30%
	short	5.39	-	779.78	-
Qwen3-32B	long	5.76	6.05%	2161.22	152.58%
	short	5.44	-	855.67	-
Qwen3-30B-thinking	long	6.28	12.63%	4582.00	242.34%
Qwen3-30b-Instruct	short	5.58	-	1338.44	-
Qwen3-30B	long	6.08	36.97%	2416.22	217.65%
	short	4.44	-	760.67	-
Qwen3-14B	long	5.64	6.45%	3099.78	224.58%
	short	5.30	-	955.00	-
Qwen3-8B	long	5.24	14.95%	2996.67	312.76%
	short	4.56	-	726.00	-
Qwen3-4B	long	4.78	6.91%	2714.56	289.52%
	short	4.47	-	696.89	-

Table 16: Model Depth and Token Analysis, comparing average logical depth, depth delta (relative increase from short to long mode), average token count, and token delta across models and thinking modes.

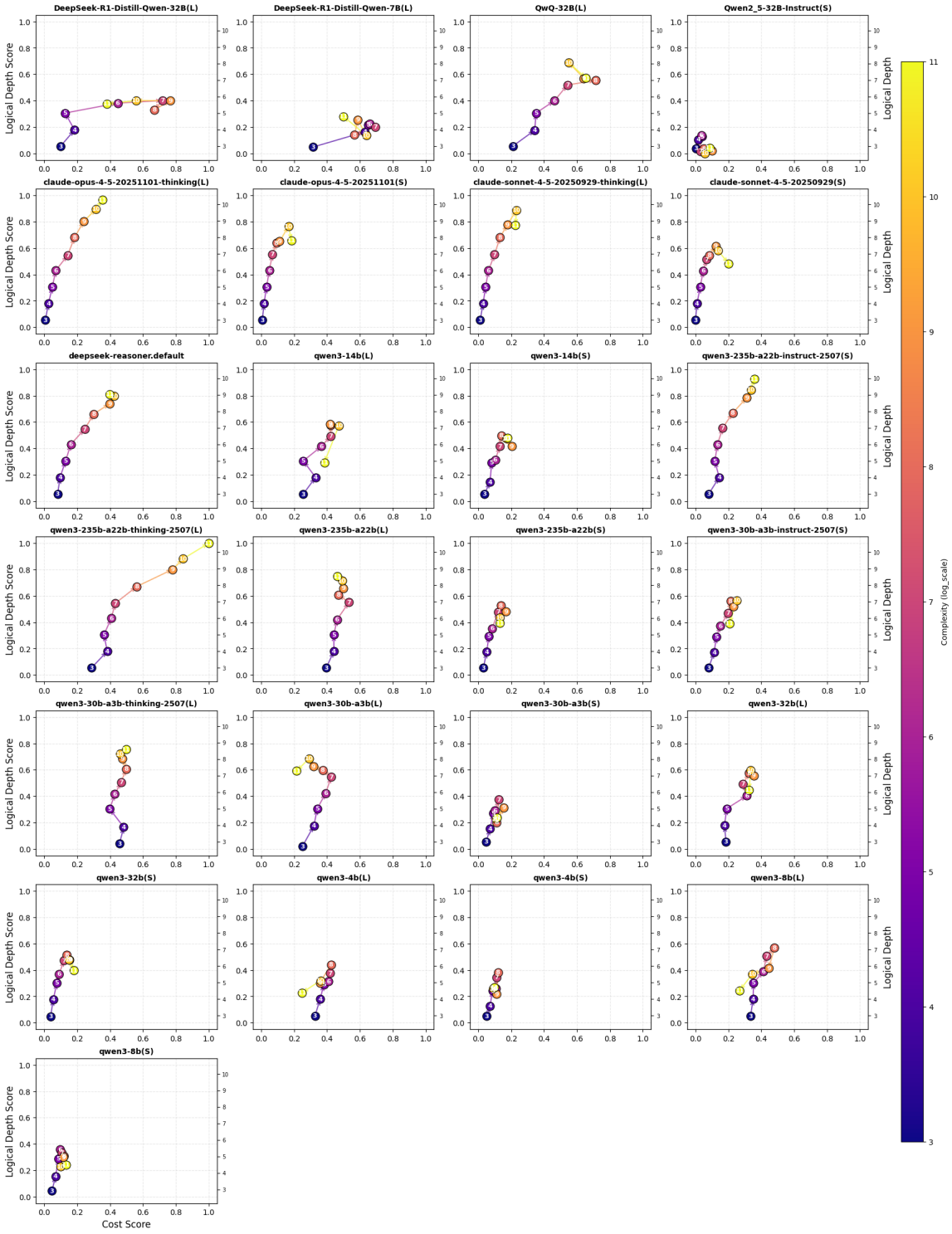


Figure 7: Full set of reasoning trajectories for all 25 evaluated models, plotting Logical Depth Score against Cost Score across increasing complexity (color-coded by depth from log C=3 to 11). Each subplot visualizes a model's adaptation pattern under varying problem difficulty.

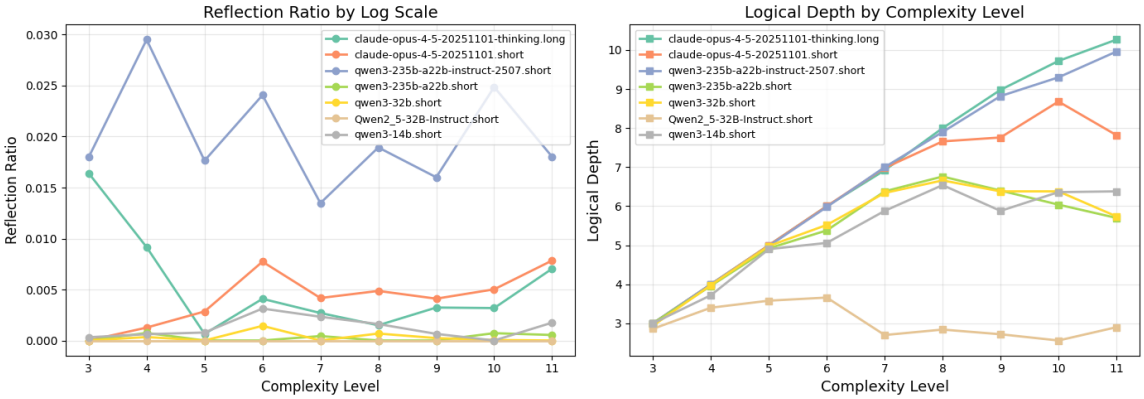
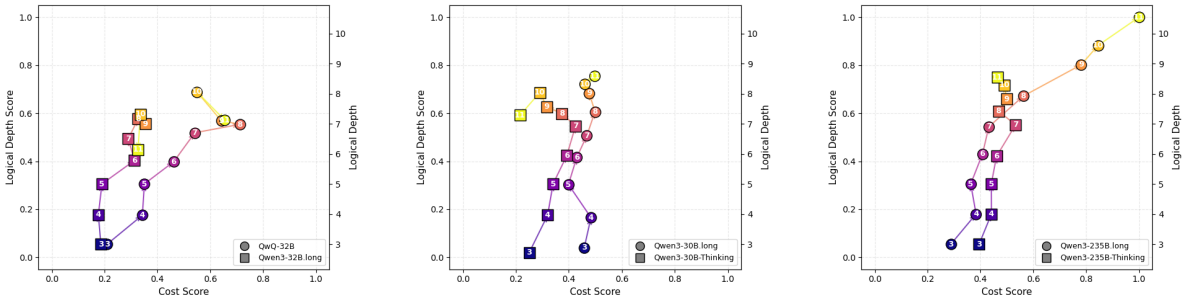


Figure 8: Reflection Ratio and Max Depth by Log Scale. The three models (Qwen3-235B-Instruct, Claude-Opus-4.5.long, Claude-Opus-4.5.short) that maintain an advantage in logical depth in the right figure also demonstrate superior reflection ratios in the left figure. The Qwen3-235B.short model, which shows negligible reflection behavior, significantly lags behind the Qwen3-235B-A22B-Instruct model in logical depth, despite having comparable parameter scales.

#	Model	Classification		Core Metrics		Diagnostic Metrics				Raw Stats.	
		Category	S_c	S_{ld}	S_{cost}	S_{exp}	S_{eff}	S_{coh}	S_{red}	**Depth**	**Tok.(k)**
1	Qwen3-235B-thinking	DeepWanderer	1.00	1.00	0.88	1.00	0.47	0.41	0.77	10.54	16.8
2	Claude-Sonnet-4.5.long	EffectiveSolver	0.82	0.83	0.27	0.38	0.67	0.26	0.38	8.74	1.9
3	Qwen3-235B-Instruct		0.82	0.95	0.37	0.83	0.59	0.28	0.62	9.96	3.4
4	DeepSeek-R1		0.80	0.86	0.41	0.34	0.59	0.58	0.48	9.04	3.7
5	Claude-Opus-4.5.long		0.80	0.97	0.37	0.47	0.60	0.42	0.28	10.27	3.5
6	Qwen3-235B.long		0.67	0.81	0.46	0.29	0.57	0.31	0.55	8.54	4.1
7	Qwen3-30B-thinking		0.58	0.81	0.48	0.06	0.52	0.50	0.53	8.58	5.3
8	Claude-Opus-4.5.short		0.33	0.74	0.24	0.62	0.70	0.47	0.37	7.82	1.4
9	Qwen3-30B.long		0.12	0.69	0.26	0.04	0.64	0.59	0.39	7.28	1.5
10	DS-R1-Qwen-7B	HollowMimic	1.00	0.49	0.46	0.01	0.47	0.35	0.62	4.80	6.0
11	DS-R1-Qwen-32B		0.53	0.53	0.39	0.12	0.50	0.76	0.59	5.56	3.4
12	Qwen3-14B.long		0.39	0.47	0.39	0.09	0.54	0.34	0.57	4.90	3.4
13	QwQ-32B		0.34	0.68	0.61	0.14	0.48	0.32	0.62	7.12	5.7
14	Qwen3-32B.long		0.29	0.58	0.35	0.24	0.56	0.33	0.52	6.14	2.7
15	Qwen2.5-32B-Inst	LazyGuesser	1.00	0.28	0.16	0.03	0.55	0.07	0.59	2.90	0.7
16	Qwen3-30B.short		1.00	0.42	0.18	0.06	0.63	0.07	0.51	4.46	0.8
17	Qwen3-8B.short		1.00	0.43	0.20	0.05	0.63	0.17	0.49	4.50	1.0
18	Qwen3-4B.short		1.00	0.45	0.17	0.02	0.67	0.11	0.54	4.70	0.7
19	Qwen3-235B.short		0.74	0.54	0.20	0.12	0.70	0.43	0.53	5.70	1.0
20	Qwen3-32B.short		0.65	0.55	0.23	0.18	0.64	0.41	0.53	5.74	1.4
21	Qwen3-4B.long		0.62	0.42	0.29	0.03	0.55	0.39	0.48	4.38	1.7
22	Qwen3-30B-Instruct		0.59	0.54	0.26	0.27	0.58	0.47	0.45	5.68	1.6
23	Qwen3-8B.long		0.45	0.43	0.30	0.07	0.55	0.40	0.44	4.52	1.9
24	Qwen3-14B.short		0.35	0.23	0.61	0.08	0.68	0.23	0.55	6.38	1.4
25	Claude-Sonnet-4.5.short		0.30	0.61	0.25	0.34	0.61	0.42	0.40	6.42	1.6
Category Avg (weighted)		DeepWanderer		**1.00**	**0.88**	**1.00**	0.47	0.41	**0.77**	**10.54**	**16.8**
		EffectiveSolver		0.86	0.37	0.42	0.60	0.40	0.46	9.10	3.4
		HollowMimic		0.52	0.45	0.09	0.50	**0.42**	0.60	5.43	4.7
		LazyGuesser		0.45	0.21	0.09	**0.62**	0.25	0.51	4.78	1.1

Table 17: Full model classification results. The **Raw Stats.** columns (rightmost) display absolute Average Logical Depth and Token Count (in thousands) for reference. Note that the category averages are weighted by confidence scores (S_c).



(a) 32B Seperate vs Mixed (b) 30B Seperate vs Mixed (c) 235B Seperate vs Mixed

Figure 9: Comparative Analysis of Reasoning Models. Performance trajectories of (a) 32B, (b) 30B, and (c) 235B model variants under separate vs. mixed configurations, plotting Logical Depth against Cost across increasing complexity. It can be observed that, in these three settings, the independently trained thinking models (QwQ-32B, Qwen3-30B-thinking, Qwen3-235B-thinking) experience saturation or collapse later than their counterparts trained with long/short mixed methods. This reflects the disruptive effect of mixed training on high-consumption strategies.

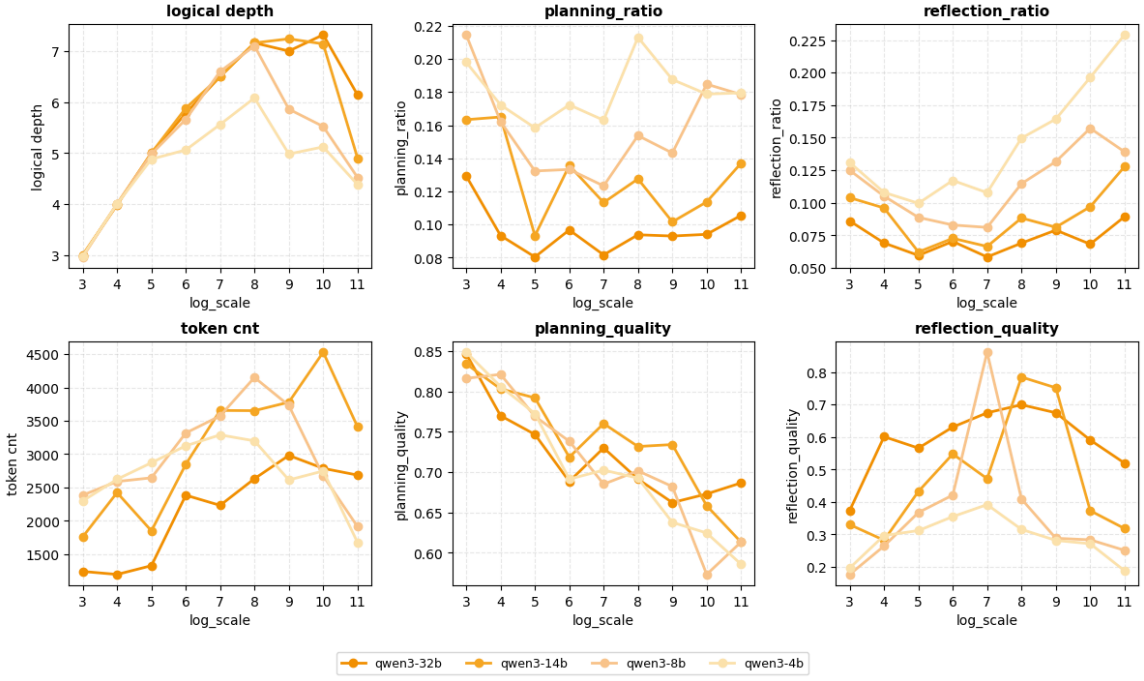


Figure 10: Qwen3-32B vs 14B vs 8B vs 4B. Six subplots compare key reasoning behaviors (max depth, planning/reflection ratio and quality, token count) across model sizes as problem complexity (log_scale) increases. Smaller distilled models exhibit more sophisticated behaviors, even surpassing the teacher model; however, they fail to emulate behavioral efficiency and cannot translate these sophisticated behaviors into deeper reasoning. Only the 14B model shows a high degree of alignment in both behavior and capabilities with the 32B teacher model. This indicates that the effectiveness of the model's reasoning in expanding tokens is intrinsically constrained by its capabilities.