Decomposing Reasoning Efficiency in Large Language Models

Daniel Kaiser[*†] **Ali Ramezani-Kebrya**[*‡] **Arnoldo Frigessi**[*‡] **Benjamin Ricaud**[*†]

[†]daniel.kaiser@uit.no

Abstract

While large language models (LLMs) are typically evaluated on accuracy metrics alone, real-world deployment requires careful control of computational efficiency. Building on the CogniLoad benchmark [Kaiser et al., 2025], we introduce a unified efficiency metric (i.e., correct answers per 1'000 output tokens) enabling direct cross-model comparison. We further provide an interpretable factorization into context robustness, logic robustness, and token appetite. Evaluating 15 state-of-the-art reasoning LLMs on CogniLoad, we find that some models fail due to logic errors, others consume excessive tokens, and a few hit context limits. Tokens are an imperfect but practical proxy for computation load, permitting consistent comparisons across closed and open models. By decomposing overall efficiency into actionable components, our framework identifies concrete targets for improving LLM reasoning efficiency.

1 Introduction

Performance on a task is not the only factor influencing the deployment of LLMs. Two models with comparable accuracy may differ substantially in terms of token usage (*i.e.,* cost and latency). This disparity is particularly evident on reasoning tasks where models may explore redundant solution paths, produce verbose chain-of-thought, or derail into reasoning loops.

In this paper, we propose a unified metric and a diagnostic factorization, benchmark reasoning LLMs by measuring their token usage, and link it to different properties of the reasoning task and reasoning outcomes. We leverage the reasoning benchmark CogniLoad [Kaiser et al., 2025] grounded in Cognitive Load Theory [Sweller, 1988], which generates tasks with controlled cognitive load across three dimensions: intrinsic difficulty d, task length N, and distractor density ρ. This allows us to define an actionable measure of efficiency and uncover where reasoning models struggle. We focus on CogniLoad benchmark because it specifically evaluates the pure deductive reasoning ability of LLMs within their context without requirement of any domain knowledge (*e.g.,* math, code, etc).

Token Efficiency. We define efficiency as the ratio of correct answers on CogniLoad relative to the average number of thousand output tokens (T). This is a global measure, *i.e.,* it is computed over the complete CogniLoad parameter surface. We denote the token efficiency by E_0:

$$E_0 \;=\; \frac{1000 \cdot P(\text{success})}{\mathbb{E}[T]} \qquad \text{(TokenEff)}$$

where $P(\text{success})$ is the probability of giving the correct answer (solving the puzzle), estimated over all given puzzles. The token appetite $\mathbb{E}[T]$ in the denominator is the average number of output

[*]Integreat - Norwegian Centre for Knowledge-Driven Machine Learning

[†]UiT - The Arctic University of Norway

[‡]University of Oslo

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Efficient Reasoning.

tokens generated (including thinking trace + response) for each puzzle. This measure allows us to calculate a single score for each LLM and enables us to produce a token-efficiency leaderboard (see Table 1).

Tokens as a Practical Proxy for Computation. We define efficiency in terms of generated output tokens to achieve some $P(\text{success})$ and through it compare open and proprietary models without having to know model details: model size, FLOPs, or energy usage. Because per-token cost is usually the billed unit by model providers and they vary by model architecture, we treat tokens as a measurable, model-agnostic proxy for computation. For open models, compute-normalized variants (e.g., size$\times$tokens) can complement E_0 while for proprietary models, this information is typically unavailable.

In our second experiment, we explore how E_0 depends on CogniLoad's three cognitive load dimensions as E_d, E_N and E_ρ. Following this experiment we regress the output tokens of LLMs across the CogniLoad parameter surface and investigate how these cognitive dimensions influence the token appetite of the models.

In our final experiment, we go deeper into the analysis of reasoning failures and the generation of tokens. For this, we decompose $P(\text{success})$ further into interpretable robustness factors: $P(\text{success}) = r_{\text{ctx}} \cdot r_{\text{logic}}$, where $r_{\text{ctx}} = 1 - P(\text{context failure})$ and $r_{\text{logic}} = 1 - P(\text{logic failure} \mid \text{no context failure})$. Combined with token appetite $\mathbb{E}[T]$, this decomposition separates three distinct weaknesses in terms of efficiency: (i) hitting the context cap, (ii) making logical errors, and (iii) spending too many tokens when correct. More precisely, for each evaluation, we classify the outcome as success (correct solution for the puzzle), context failure (exhausting the 32'768 combined token limit), or logic failure (incorrect answer without hitting the limit).

2 Related Work

Reasoning Efficiency and Overthinking in LLMs: LLMs often produce unnecessarily long chains of thought on easy problems, inflating token use and sometimes lowering accuracy. Han et al. [2025] show that step-by-step solutions on simple math can introduce unnecessary steps and hurt accuracy, while Chiang and yi Lee [2024] find redundant calculations on trivial GSM8K-Zero items, with elaborate traces that occasionally derail correct answers. Recent surveys [Sui et al., 2025] synthesize these patterns and outline causes and mitigations for overthinking and underthinking.

Balancing Over- and Underthinking: OptimalThinkingBench [Aggarwal et al., 2025] formalizes both failure modes with an overthinking-adjusted accuracy and shows that no current model achieves the optimal balance: reasoning models over-explain trivial queries, while faster non-reasoning models underthink hard ones. This framing motivates adaptive policies that scale the amount of reasoning to instance difficulty rather than using a fixed chain-of-thought budget.

Token-budget-aware Reasoning Approaches: Recent work reduces superfluous reasoning via budgeting, early stopping, and training. TALE [Han et al., 2025] prompts with token budgets that scale with difficulty, cutting tokens by about 67% with minimal accuracy loss. Dynasor-CoT [Fu et al., 2025a] terminates reasoning when confidence is high, saving up to 29% tokens at similar accuracy. Length-controllable fine-tuning (CoT-Valve [Ma et al., 2025]) and length-harmonizing pruning (O1-Pruner [Luo et al., 2025]) shorten traces without degrading correctness. CoThink [Fan et al., 2025] uses a concise outline from an instruct model to guide a reasoner, reducing tokens by $\tilde{2}2\%$ with negligible loss. Complementary dynamic controls further improve efficiency with Deep Think with Confidence [Fu et al., 2025b] filtering redundant paths using internal confidence signals, and TRAAC [Singh et al., 2025] pruning low-utility steps online using attention and difficulty estimation—both cutting tokens substantially while maintaining accuracy. Offline CoT compression via step entropy [Li et al., 2025b] likewise removes redundant steps with small accuracy impact. Together, these strategies trim reasoning while aiming to preserve reasoning quality.

Efficiency-aware Evaluation Metrics: Standard evaluations often ignore computational cost, favoring methods that simply spend more tokens. Budget-aware evaluation [Wang et al., 2024] charges for compute and frequently eliminates purported gains. Metrics such as Accuracy per Computation Unit (ACU) [Ma et al., 2025] and reasoning-efficiency ratios [Fan et al., 2025] quantify accuracy–cost trade-offs. Beyond scalar ratios, efficiency-frontier analyses define best-known accu-

Model	E_0 in %	$P(S)$ in %	$\mathbb{E}[T]$
o3-2025-04-16	15.21	89.27	5'869
gpt-5-2025-08-07	13.38	92.29	6'897
DeepSeek-R1-Distill-Llama-70B	12.30	56.46	4'592
o4-mini-2025-04-16	12.19	75.97	6'232
Qwen3-32B	11.47	50.26	4'383
gpt-5-mini-2025-08-07	8.87	78.79	8'878
Qwen3-30B-A3B	8.74	44.64	5'105
DeepSeek-R1-Distill-Qwen-32B	7.76	49.59	6'393
gpt-5-nano-2025-08-07	5.86	45.64	7'792
gemini-2.5-pro	5.20	75.00	14'416
gemini-2.5-flash	3.98	57.00	14'332
Phi-4-reasoning	3.38	40.62	12'013
Phi-4-reasoning-plus	3.08	46.27	15'046
Phi-4-mini-reasoning	2.79	22.34	8'005
DeepSeek-R1-Distill-Qwen-1.5B	0.97	14.21	14'690

Table 1: Efficiency Leaderboard by $E_0 = 1000\,P(\text{success})/\mathbb{E}[T]$. The token appetite $\mathbb{E}[T]$ is the average number of token generated by each model to solve the puzzles. $P(S)$ is short for $P(\text{success})$, the overall accuracy score for each model.

racy–length curves and measure a Reasoning Efficiency Gap (REG) [Gao et al., 2025], showing most systems lie well off the frontier with their method REO-RL narrowing but not closing this gap.

Benchmarks and Diagnostic Frameworks: CogniLoad [Kaiser et al., 2025] manipulates length, difficulty, and distractor density to probe failure modes with length strongly degrading accuracy for several models, indicating weak context robustness. Think-Bench [Li et al., 2025a] evaluates "thinking efficiency" with human-annotated key steps and metrics such as total tokens, tokens to first correct step, reflection tokens, and step precision/recall, finding widespread overthinking on simple problems and that larger models, while more accurate, often spend more tokens per unit of reasoning quality. Complementary to these, OptimalThinkingBench [Aggarwal et al., 2025] explicitly balances over- and underthinking via thresholded "thinking tokens", emphasizing efficiency-sensitive correctness.

A Unified and Decomposable Efficiency Metric: We introduce a single, model-agnostic score: correct answers per 1'000 output tokens—computed from black-box outputs. We further decompose efficiency into three interpretable factors: context robustness, logic robustness, and token appetite (tokens used per solution). Unlike prior work that reports multiple metrics without a synthesis, our factorization yields an actionable index that separates contextual, logical, and verbosity-driven reasoning inefficiencies, extending budget-aware evaluation [Wang et al., 2024] and complementing efficiency-frontier views [Gao et al., 2025], especially for proprietary models that hide their chain-of-thought and details about the underlying model.

3 Results

We evaluate 15 LLMs on CogniLoad's factorial design with three dimensions of complexity: intrinsic difficulty $d \in \{1, 3, 5, 7, 10\}$, task length $N \in \{20, 50, 100, 250\}$, and distractor density $\rho \in \{5, 10, 25, 50, 75, 90, 95\}\%$. This yields 140 parameter configurations for which we generated 10 random puzzles each, totaling 1'400 tasks per model. All LLMs are evaluated on the same CogniLoad puzzles.

Each CogniLoad attempt is labeled as one of: success (exact match), context failure (hitting the 32'768 combined token limit), or logic failure (incorrect answer without hitting the limit). We count output tokens T as all model-generated tokens (reasoning trace + final answer). Models use default decoding settings and for the GPT-5 series models we set the reasoning effort to "medium".

3.1 Overall Efficiency and Leaderboard

Table 1 ranks models by E_0 and reveals that efficiency leadership differs significantly compared to accuracy leadership.

Efficiency vs Accuracy: o3-2025-04-16 is most efficient, $E_0 = 15.21\%$, solving correctly 89.27% of puzzles with 5'869 output tokens on average. gpt-5-2025-08-07 attains the highest accuracy (92.29%) but lower efficiency $E_0 = 13.38\%$ with 6'897 tokens (16% more tokens than o3).

Token Appetite Varies Widely: Gemini-2.5-pro achieves 75% success but uses 14'416 tokens on average while by contrast, o4-mini-2025-04-16 obtains a comparable performance of 75.97% with just 6'232 tokens (56% fewer).

3.2 Efficiency across Cognitive Load Dimensions

To better understand how task properties influence the efficiency of LLMs, we use the three CogniLoad dimensions and vary one while averaging accuracy over the remaining two. The three parameter-dependent efficiencies are plotted in Fig. 1.

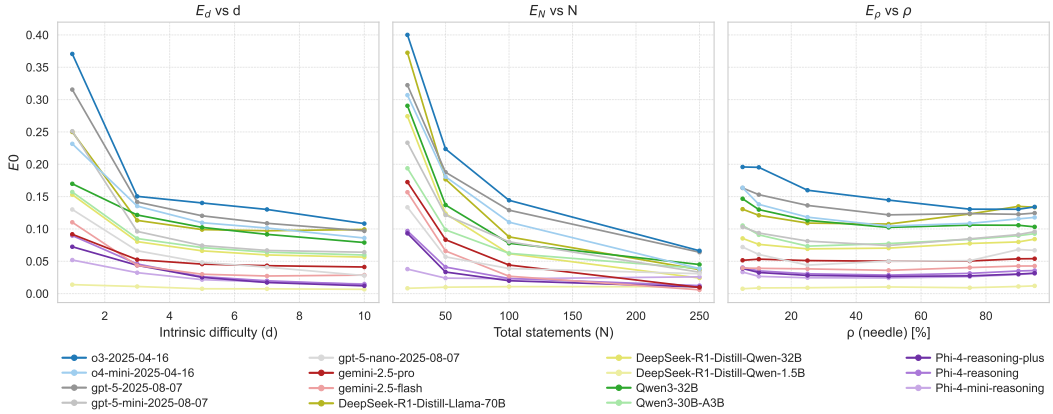

Figure 1: Token efficiency of recent LLMs with respect to the CogniLoad dimensions. E_d (left) and E_N (middle) show that the efficiency is impacted by the difficulty d and length N of the puzzles. All models exhibit diminishing curves of efficiency. Surprisingly, the ratio of distractors in the task has little influence on the efficiency (E_ρ on the right).

The left panel (vs. d) shows a sharp drop from $d = 1$ to $d = 3$, followed by a much slower decline through $d = 10$. Strong models retain a consistent margin after the early drop, and the ordering among top models is largely preserved across difficulty levels. This pattern suggests that most of the difficulty-related efficiency loss is incurred by the first increase in difficulty, with diminishing additional loss of efficiency thereafter.

The center panel (vs. N) reveals the primary bottleneck. Increasing task length from $N = 20$ to $N = 50$ eliminates a large share of efficiency E_N for every model, with a further steady decay out to $N = 250$. By $N = 250$, efficiency is typically 70–90% lower than at $N = 20$, and some lower-capacity models approach zero. Despite this overall contraction, the relative ranking among stronger models remains stable, indicating that sustained, stateful reasoning over long sequences is the central driver of efficiency loss.

The right panel (vs. ρ) varies the fraction of distractors in the puzzle text. Most models exhibit a shallow U-shape: E_ρ is higher when ρ is very low or very high and tends to be lowest at intermediate values (roughly $\rho \approx 40$–70%). The amplitude of this effect is modest compared to the impact of N, and several models deviate slightly from the U-shape (e.g., nearly flat or gently increasing with ρ).

3.3 What Drives Token Appetite?

To quantify how task properties inflate output token usage, we regress per-instance output tokens T on successful cases only via an OLS regression of the form:

$$T = \beta_0 + \beta_d \left(\frac{d-1}{10}\right) + \beta_N \left(\frac{N-20}{250}\right) + \beta_\rho \left(\frac{\text{needles}}{N} - 0.05\right) + \varepsilon,$$

where $d \in \{1, 3, 5, 7, 10\}$ is intrinsic difficulty, $N \in \{20, 50, 100, 250\}$ is task length, and $\text{needles}/N \in [0.05, 0.95]$ is the fraction of PoI-relevant statements (higher ρ means fewer dis-

Model	β_0	β_d	β_N	β_ρ
DeepSeek-R1-Distill-Llama-70B	2'487***	1'593***	6'225***	-545***
DeepSeek-R1-Distill-Qwen-1.5B	6'732***	-2'027***	-2'409***	1'758***
DeepSeek-R1-Distill-Qwen-32B	2'620***	1'745***	9'410***	-501***
Phi-4-mini-reasoning	7'581***	1'072***	-382*	358*
Phi-4-reasoning	3'949***	7'926***	11'925***	784***
Phi-4-reasoning-plus	5'913***	8'452***	14'003***	647***
Qwen3-30B-A3B	3'885***	387***	4'174***	-593***
Qwen3-32B	3'370***	2	4'233***	-595***
gemini-2.5-flash	2'351***	12'395***	20'132***	314
gemini-2.5-pro	3'766***	9'901***	21'386***	444
gpt-5-2025-08-07	2'293***	5'113***	8'649***	-1'301***
gpt-5-mini-2025-08-07	2'283***	8'224***	10'279***	-585
gpt-5-nano-2025-08-07	4'230***	5'151***	5'172***	874*
o3-2025-04-16	2'275***	4'425***	7'842***	-2'147***
o4-mini-2025-04-16	2'862***	4'449***	7'560***	-1'782***

Table 2: Stars denote significance (* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$).
Predictors: β_0 (intercept), β_d (difficulty-norm), β_N (total-statements-norm), β_ρ (rho-norm).

tractors). The predictors for d and N are affine-normalized to anchor at the grid minimum; for ρ we subtract 0.05 so that $\rho = 0$ corresponds to the minimum fraction in CogniLoad.

Length Dominates Token Appetite. The length coefficient β_N is the largest driver across models (Table 2). It is strongly positive for all high-capacity systems, confirming that longer sequences substantially increase token appetite: Gemini-2.5-pro and -flash exhibit the largest slopes ($\beta_N \approx 21'386$ and $20'132$), the Phi family shows large positive slopes ($\approx 11'925$ and $14'003$), GPT-5 variants and o3/o4 are sizable but smaller ($8'649$, $10'279$, $7'842$, $7'560$), and mid-tier Qwen models are more modest ($\approx 4'174$–$4'233$). The two small models show negative β_N (DeepSeek-R1-Distill-Qwen-1.5B: $-2'409$; Phi-4-mini-reasoning: -382), indicating that among their successful cases, solutions tend to be guesses, consistent with short-circuiting under load rather than the results of genuine correct reasoning.

Intrinsic Difficulty Increases Verbosity in Specific Families. The difficulty coefficient β_d is positive and significant for most models, with very large effects in the Phi ($\approx 7'926$, $8'452$) and Gemini families ($12'395$, $9'901$), and moderate effects for GPT-5 and o3/o4 ($5'113$, $8'224$; $4'425$, $4'449$). In contrast, some smaller models show weak or even negative dependence (Qwen3-32B: 2, not significant; DeepSeek-R1-Distill-Qwen-1.5B: $-2'027$), suggesting that, conditional on success, they do not expand their traces much with intrinsic complexity and may instead rely on guessed answers on easier puzzle instances.

Distractor Effects Are Modest and Family-specific. The distractor ratio coefficient β_ρ (higher means more distractors) is smaller in magnitude than β_N and varies in sign across families. It is negative and often significant for GPT-5 and o3/o4 ($-1'301$, $-2'147$, $-1'782$) and for some Qwen variants (≈ -595), indicating that when a larger share of the puzzle are needles, successful solutions include longer reasoning traces. In contrast, the Phi family and DeepSeek-R1-Distill-Qwen-1.5B show positive and significant β_ρ (e.g., 784, 647, $1'758$), suggesting they shorten traces distractors become more frequent. For Gemini-2.5 models, β_ρ is positive but not significant, reinforcing that distractor density has comparatively minimal influence on token appetite for these systems. Overall, the direction and size of β_ρ are secondary to β_N and appear model-family dependent.

Baselines Vary Substantially. Intercepts β_0 (the expected output tokens at d=1, N=20, ρ=0.05) highlight baseline concision differences: o3 has the smallest baseline (343^*), GPT-5 is low ($1'122$), while Phi-4-mini-reasoning and DeepSeek-R1-Distill-Qwen-1.5B are much higher ($7'903$, $8'315$). These baseline gaps help explain the token-appetite component in the efficiency decomposition.

In sum, token appetite scales primarily with task length N, with intrinsic difficulty d contributing notably in specific families, and distractor density playing a smaller, model-dependent role.

3.4 Efficiency and Failure Modes

We propose an intuitive distinction between the three different outcomes of the LLMs answers: success, logic failure and failure due to reached context limit, and decompose Eq. (TokenEff) into different parts by taking its logarithm. Since $P(\text{success}) = r_{\text{ctx}} \cdot r_{\text{logic}}$, the quantity $\log(E_0)$ is a sum of terms, each one related to one of the key parameter in our efficiency analysis. We obtain the log-space identity $\log E_0 = \log r_{\text{ctx}} + \log r_{\text{logic}} - \log \mathbb{E}[T] + \text{const}$. To facilitate comparison between different LLMs, we compute the difference of $\log(E_0)$ with the most efficient model (i.e. o3-2025-04-16) used as the reference model :

$$\Delta \log E_0 \;=\; \Delta \log r_{\text{ctx}} \;+\; \Delta \log r_{\text{logic}} \;-\; \Delta \log \mathbb{E}[T]. \qquad \text{(FailDecomp)}$$

This leads to an additive contribution analysis where positive values indicate improvements and negative values indicate degradations of relative efficiency relative to the reference. We compute 95% confidence intervals via bootstrap resampling of puzzle IDs over 400 iterations.

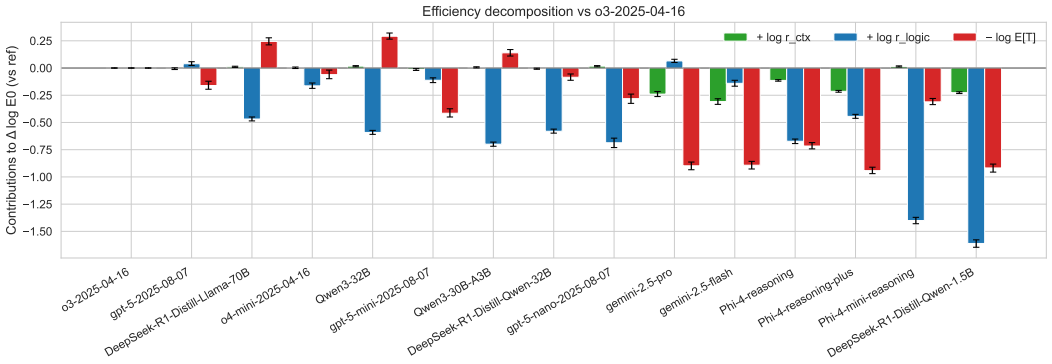

Figure 2: Decomposition of $\Delta \log E_0$ relative to o3-2025-04-16 into context robustness $(+\log r_{\text{ctx}})$, green, logic robustness $(+\log r_{\text{logic}})$, blue, and token appetite $(-\log \mathbb{E}[T])$, red. The dominant pattern across almost all models is that the largest negative contribution comes from logic robustness.

The results of this decomposition are shown in Fig. 2. The difference in efficiency with respect to the reference is driven primarily by a higher probability of logic failures, not by hitting the context cap and only by small differences in reasoning verbosity.

The second pattern concerns token appetite and is LLM-family dependent. Some models are comparatively concise and gain efficiency through a positive $\Delta \log \mathbb{E}[T]$. DeepSeek-R1-Distill-Llama-70B and several Qwen variants illustrate this: their red bars are positive, indicating fewer output tokens than the reference, yet this advantage is outweighed by a sizable negative $\Delta \log r_{\text{logic}}$. Other models pay a substantial verbosity tax. The Gemini models exhibits large token penalties in the chart while the Gemini-2.5-pro version compensates for it through positive $\log r_{\text{logic}}$). The Phi family (Phi-4-reasoning, Phi-4-reasoning-plus, Phi-4-mini-reasoning) combines very large logic deficits with additional token penalties, resulting in the biggest overall shortfalls in E_0.

A third observation is that context robustness variations are minor. Green bars cluster near zero for most models, with only modest positive values for a few (e.g., gemini-2.5-pro). This indicates that, in CogniLoad, the 32k token cap is rarely the binding constraint. When context failures occur, they likely reflect derailed or looping traces rather than genuine limits within the available context limit.

Two contrasts are especially informative. First, gpt-5-2025-08-07 is nearly at parity with the reference on both robustness terms and loses only slightly due to a higher token appetite. Second, models such as Qwen3-32B and Qwen3-30B-A3B are relatively frugal in tokens but still trail the reference mainly due to logic robustness, suggesting that making the reasoning steps more reliable would deliver larger efficiency gains than more concise reasoning.

Overall, the decomposition shows that gains of a few tenths in $\Delta \log \mathbb{E}[T]$ are rarely sufficient to offset drops of 0.5–1.5 in $\Delta \log r_{\text{logic}}$. Improving reasoning fidelity via better verifiers, self-correction, or training signals that penalize invalid deductions is a potential remedy for improving E_0.

4 Limitations

Tokens as Computation Proxy. Token counts are a practical but imperfect proxy for computation: per-token latency, FLOPs, and energy vary with model size, architecture, and inference stack. We therefore emphasize E_0 as a model-agnostic metric that allows comparison across closed and open models. Where feasible (i.e., open models), compute-normalized variants can complement E_0.

Deductive and Synthetic Reasoning Scope. CogniLoad targets sequential deductive reasoning without external tools or prior knowledge. This isolates reasoning fidelity from retrieval and world knowledge, but it does not cover other reasoning paradigms (e.g., inductive, analogical) or multimodal settings. Thus, while token efficiency on CogniLoad may differ from reasoning efficiency in specific domains (eg. math, coding, or open-domain QA), it reveals fundamental sensitivities of pure reasoning to length, intrinsic difficulty, and distractors.

Single Decoding Setup. We evaluate each model with default decoding (and a standardized "medium" reasoning effort where available). Different temperatures, stop criteria, or thought constraints can alter token appetite and probability of success. We partially mitigate this by aggregating over 1 400 puzzles per model.

No External Verifiers or Tools. As we focus on the efficiency of pure reasoning ability, our test protocol forbids tool use and external verifiers as models must solve the puzzle only with their generated reasoning tokens. This isolates intrinsic reasoning efficiency but does not evaluate tool-augmented systems.

5 Discussion

The decomposition turns the vague goal of "token efficiency" into concrete, model-specific targets. In addition, our analysis of how E_0 depends on task properties (i.e. the three dimensions of cognitive load) and the model behavior (i.e. logic failure, context failure, and verbosity) open several future directions for improvements of reasoning in LLMs. When $\Delta \log r_{\text{logic}}$ dominates, improvements should prioritize reasoning quality through better verifiers or self-correction mechanisms and training data that emphasizes formal consistency. When $-\Delta \log \mathbb{E}[T]$ dominates, one should focus on concise plans, early stopping, and mechanisms that avoid redundant reasoning steps. When $\Delta \log r_{\text{ctx}}$ is substantial, mitigation should first target derailing traces and self-loops before increasing context size.

6 Acknowledgements and Support

This work was supported by the Research Council of Norway through its Centres of Excellence scheme, Integreat - Norwegian Centre for Knowledge-Driven Machine Learning, project number 332645.

Ali Ramezani-Kebrya was supported by the Research Council of Norway through FRIPRO Grant under project number 356103 and its Centres of Excellence scheme, Integreat - Norwegian Centre for knowledge-driven machine learning under project number 332645.

We acknowledge NRIS Norway for awarding this project access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium through Sigma2, project number nn12027k.

References

P. Aggarwal, S. Kim, J. Lanchantin, S. Welleck, J. Weston, I. Kulikov, and S. Saha. Optimalthinkingbench: Evaluating over and underthinking in llms, 2025. URL https://arxiv.org/abs/2508.13141.

C.-H. Chiang and H. yi Lee. Over-reasoning and redundant calculation of large language models, 2024. URL https://arxiv.org/abs/2401.11467.

S. Fan, B. Qin, P. Han, S. Shang, Y. Wang, and A. Sun. The price of a second thought: On the evaluation of reasoning efficiency in large language models, 2025. URL https://arxiv.org/abs/2505.22017.

Y. Fu, J. Chen, Y. Zhuang, Z. Fu, I. Stoica, and H. Zhang. Reasoning without self-doubt: More efficient chain-of-thought through certainty probing. In *ICLR 2025 Workshop on Foundation Models in the Wild*, 2025a.

Y. Fu, X. Wang, Y. Tian, and J. Zhao. Deep think with confidence, 2025b. URL https://arxiv.org/abs/2508.15260.

J. Gao, S. Yan, Q. Tan, L. Yang, S. Xu, W. Fu, Z. Mei, K. Lyu, and Y. Wu. How far are we from optimal reasoning efficiency?, 2025. URL https://arxiv.org/abs/2506.07104.

T. Han, Z. Wang, C. Fang, S. Zhao, S. Ma, and Z. Chen. Token-budget-aware LLM reasoning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

D. Kaiser, A. Frigessi, A. Ramezani-Kebrya, and B. Ricaud. CogniLoad: A synthetic natural language reasoning benchmark with tunable length, intrinsic difficulty, and distractor density, 2025. URL https://arxiv.org/abs/2509.18458.

Z. Li, Y. Chang, and Y. Wu. Think-bench: Evaluating thinking efficiency and chain-of-thought quality of large reasoning models, 2025a. URL https://arxiv.org/abs/2505.22113.

Z. Li, J. Zhong, Z. Zheng, X. Wen, Z. Xu, Y. Cheng, F. Zhang, and Q. Xu. Compressing chain-of-thought in llms via step entropy, 2025b. URL https://arxiv.org/abs/2508.03346.

H. Luo, L. Shen, H. He, Y. Wang, S. Liu, W. Li, N. Tan, X. Cao, and D. Tao. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning, 2025. URL https://arxiv.org/abs/2501.12570.

X. Ma, G. Wan, R. Yu, G. Fang, and X. Wang. Cot-valve: Length-compressible chain-of-thought tuning, 2025. URL https://arxiv.org/abs/2502.09601.

J. Singh, J. C.-Y. Chen, A. Prasad, E. Stengel-Eskin, A. Nambi, and M. Bansal. Think right: Learning to mitigate under-over thinking via adaptive, attentive compression, 2025. URL https://arxiv.org/abs/2510.01581.

Y. Sui, Y.-N. Chuang, G. Wang, J. Zhang, T. Zhang, J. Yuan, H. Liu, A. Wen, S. Zhong, N. Zou, H. Chen, and X. Hu. Stop overthinking: A survey on efficient reasoning for large language models, 2025. URL https://arxiv.org/abs/2503.16419.

J. Sweller. Cognitive load during problem solving: Effects on learning. *Cognitive science*, 12(2): 257–285, 1988.

J. Wang, S. Jain, D. Zhang, B. Ray, V. Kumar, and B. Athiwaratkun. Reasoning in token economies: Budget-aware evaluation of llm reasoning strategies, 2024. URL https://arxiv.org/abs/2406.06461.