A Study of Large Language Models' Reasoning Capabilities in Complex Scenarios: Investigating Logic Performance under Quantization, Finite KV-Cache, and Time Constraints

Anonymous authors
Paper under double-blind review

Abstract

As the parameter scale of Large Language Models (LLMs) continues to expand, performance predictions based on scaling laws face severe challenges under multiple physical deployment constraints. This study uses a classical propositional logic formal deduction task as the evaluation benchmark—requiring models to strictly apply 12 standard inference rules to progressively deduce from given premises to a target conclusion and output a deterministic binary judgment (TRUE/FALSE)—to systematically evaluate 8 architectural variants of the Qwen 3.5 series (covering BF16 and Q4_K_M quantized versions from 0.8B to 9B scales). We analyze their empirical performance under conditions of "parameter quantization," "finite KV-cache," and "time constraints (token output truncation)." Our research focuses on systematically evaluating and identifying the optimal model scale for practical deployment under restricted resources. We find that due to the strict hard truncation of the output window, traditional evaluation metrics decouple: Formal Evaluation Scores (FES) more accurately reflect the model's actual reasoning depth, while mean accuracy primarily demands a rapid response. By comparing a micro full-precision model and a medium-scale quantized model with nearly identical VRAM footprints, we reveal a critical deployment trade-off. Ultimately, this study concludes that in resource-constrained edge environments, model selection must be dictated by the specific task focus: if deep reasoning and complex reflection are prioritized, deploying a larger quantized model is advisable; conversely, if rapid response and truncation avoidance are critical, a smaller full-precision (e.g., BF16) model is the optimal choice.

1 Introduction

Enhancing the complex logical reasoning capabilities of Large Language Models (LLMs) is a core pathway to achieving Artificial General Intelligence (AGI). Modern open-weight architectures, notably the Qwen 3.5 series (Qwen Team, 2026), span seamlessly from 0.8B edge models to massive cloud engines. Because they unify intuitive projection and reflection modes with strict continuity, they provide an ideal testbed for examining the scaling laws of Test-Time Compute (Snell et al., 2024) and step-by-step reasoning dynamics.

However, scaling Test-Time Compute contradicts practical deployment physics. "Verbosity bias" drives large, aligned models to indiscriminately over-generate reasoning tokens. We demonstrate via Qwen 3.5 that under real-world physical constraints—specifically parameter quantization, limited KV-cache, and strict output token truncation—the apparent advantages of larger models in extended chain-of-thought reasoning are substantially reduced.

To measure these degradation mechanisms, this study selected classical propositional logic deduction as the core task. The experiment employs a two-stage prompting architecture: the first stage requires the model to perform step-by-step formal deduction using 12 standard inference rules (Modus Ponens, Modus Tollens, Hypothetical Syllogism, etc.); the second stage requires a deterministic binary judgment (TRUE/FALSE). This design offers three

methodological advantages: (1) unique answers and verifiable paths eliminate scoring ambiguity; (2) binary output naturally adapts to truncation analysis—the model either outputs the judgment or loses it entirely; (3) controllable deduction steps (S2–S10) enable systematic isolation of constraint impacts.

Based on Qwen 3.5, we constructed a test matrix covering 0.8B to 9B parameters (BF16 and Q4_K_M versions), conducting over 200 groups of intervention tests across reasoning stages (S2–S10) and context capacities (L0–L4). This paper systematically evaluates the performance trade-offs under physical constraints to distinguish the most viable model scales for practical deployment, and derives a "Mixture Decoding Truncation Survival Model" that mathematically unifies instruction compliance, Markovian reasoning fidelity, and truncation survival probability to self-consistently explain the observed performance divergences.

2 Research methodology and experimental design

2.1 Evaluation model selection and matrix configuration

This experiment selected the Qwen 3.5 family as the foundational architecture and systematically built a test matrix containing 8 independent variants across two orthogonal dimensions: Model Size and Precision/Quantization. All models were sourced from Unsloth (Unsloth AI, 2026) on Hugging Face. The configurations are shown in Table 1.

ID	Model Architecture	Numerical Precision	GGUF File Identifier	Est. VRAM Usage
C1	Qwen3.5-0.8B	BF16 (Control)	`Qwen3.5-0.8B-BF16.gguf`	∼1.5 GB
E1	Qwen3.5-0.8B	Q4_K_M (Exp)	`Qwen3.5-0.8B-Q4_K_M.gguf`	∼0.5 GB
C2	Qwen3.5-2B	BF16 (Control)	`Qwen3.5-2B-BF16.gguf`	∼4.0 GB
E2	Qwen3.5-2B	Q4_K_M (Exp)	`Qwen3.5-2B-Q4_K_M.gguf`	∼1.2 GB
C3	Qwen3.5-4B	BF16 (Control)	`Qwen3.5-4B-BF16.gguf`	∼7.5 GB
E3	Qwen3.5-4B	Q4_K_M (Exp)	`Qwen3.5-4B-Q4_K_M.gguf`	∼2.5 GB
C4	Qwen3.5-9B	BF16 (Control)	`Qwen3.5-9B-BF16.gguf`	∼17.0 GB
E4	Qwen3.5-9B	Q4_K_M (Exp)	`Qwen3.5-9B-Q4_K_M.gguf`	∼5.5 GB

Table 1: Experimental model matrix configuration and physical constraint parameters

2.2 Multi-dimensional environmental constraint interventions

The experiment established three intervention dimensions covering 200 data intersection points:

- **Logic Depth Decay**: Ascending steps from S2 to S10, testing resistance to error accumulation in lengthy deduction chains.
- **Context Pressure**: Pressure levels from L0 (8K tokens) to L4 (512 tokens), testing stability under sharply increased information density and constrained generation budgets.
- **Multiple Constraint Overlay**: Cross-product of S-series and L-series, observing non-linear evolution paths across difficulty levels.

2.3 Core evaluation metrics

The experiment adopted a dual-track evaluation system:FES is assigned by Gemini 3.1 Pro under a fixed rubric; Accuracy is computed directly from the model judgment against ground truth.

- **Mean Accuracy (%)**: Matching of the final logical result in the 512-token output window.

- **Formal Evaluation Score (FES)**: A rubric-based score assigned to the Stage 1 reasoning output by Gemini 3.1 Pro, measuring format adherence, step-reference consistency, designated rule-label usage, and structural completeness. FES is intended to capture formal reasoning compliance rather than final-answer correctness.

3 Quantitative experimental results

3.1 Evolution of logical depth and scale evaluation

As shown in Table 2, the expected positive correlation between model scale and accuracy does not hold under the evaluated constraint conditions; smaller models consistently achieve higher mean accuracy.

ID	Model Spec	S2	S3	S5	S7	S10	Mean
C1	Qwen3.5-0.8B-BF16	80.0%	76.0%	64.0%	68.0%	50.0%	**67.6%**
E1	Qwen3.5-0.8B-Q4	60.0%	58.0%	68.0%	56.0%	50.0%	58.4%
C2	Qwen3.5-2B-BF16	56.0%	34.0%	68.0%	50.0%	58.0%	53.2%
E2	Qwen3.5-2B-Q4	12.0%	32.0%	32.0%	32.0%	54.0%	32.4%
C3	Qwen3.5-4B-BF16	24.0%	0.0%	0.0%	0.0%	0.0%	4.8%
E3	Qwen3.5-4B-Q4	2.0%	0.0%	0.0%	0.0%	0.0%	0.4%
C4	Qwen3.5-9B-BF16	18.0%	0.0%	0.0%	0.0%	0.0%	3.6%
E4	Qwen3.5-9B-Q4	6.0%	8.0%	0.0%	0.0%	0.0%	2.8%

Table 2: Accuracy across logic depth (model $\times$ steps)

Among all evaluated models, C1 achieved the highest mean accuracy at 67.6% while still maintaining 50.0% accuracy at S10. Conversely, **large models (4B, 9B) exhibited sharp accuracy degradation**, with accuracy remaining at 0.0% across all conditions beyond S3.

3.2 Resistance and collapse under context pressure

In context pressure tests from L0 (8K) to L4 (512) (Table 3), different scales displayed vastly different behaviors.

ID	Model Spec	L0 (8K)	L1 (4K)	L2 (2K)	L3 (1K)	L4 (512)	Mean
C1	Qwen3.5-0.8B-BF16	68.0%	70.0%	70.0%	64.0%	66.0%	**67.6%**
E1	Qwen3.5-0.8B-Q4	54.0%	58.0%	48.0%	60.0%	72.0%	58.4%
C2	Qwen3.5-2B-BF16	54.0%	54.0%	50.0%	50.0%	58.0%	53.2%
E2	Qwen3.5-2B-Q4	38.0%	38.0%	42.0%	28.0%	16.0%	32.4%
C3	Qwen3.5-4B-BF16	6.0%	6.0%	6.0%	6.0%	0.0%	4.8%
E3	Qwen3.5-4B-Q4	0.0%	0.0%	0.0%	2.0%	0.0%	0.4%
C4	Qwen3.5-9B-BF16	4.0%	4.0%	4.0%	4.0%	2.0%	3.6%
E4	Qwen3.5-9B-Q4	4.0%	4.0%	4.0%	2.0%	0.0%	2.8%

Table 3: Accuracy across context pressure levels (model $\times$ pressure)

C1 (0.8B) proved absolute stability: accuracy fluctuated only within 64.0%–70.0% regardless of context compression. **E2 (2B-Q4) exposed extreme sensitivity**: accuracy plummeted from 38.0% (L0) to 16.0% (L4). Notably, **E1 (0.8B-Q4) exhibited a "Pressure-Positive" anomaly**—accuracy *increased* from 54.0% to 72.0% as context shrank, suggesting that compression forced the model to focus on core information while filtering noise. The cross-product overlay effects of logic depth and context pressure are detailed in Appendix A.

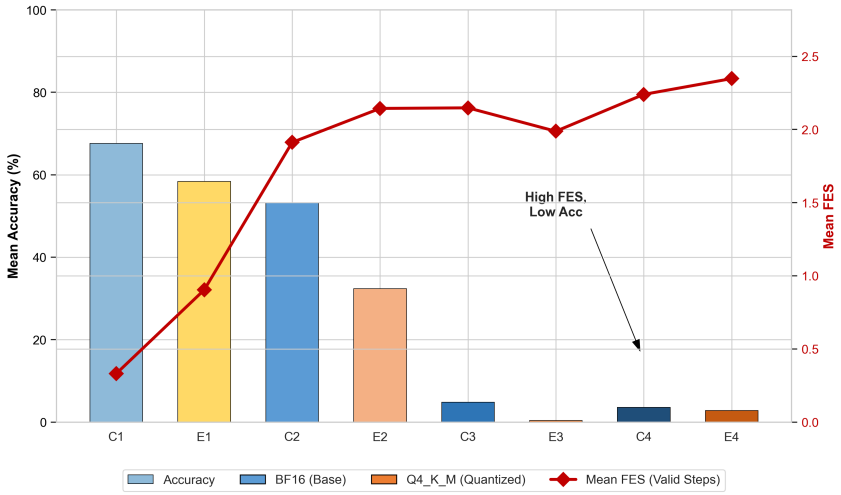

Figure 1: Global performance overview (accuracy vs FES)

4 Discussion

This section combines empirical data and visualizations to explore the divergences in reasoning behavior across model scales.

4.1 Global performance decoupling

Figure 1 reveals a striking cross-divergence: **large models retain complex reasoning capabilities but entirely lack the ability to complete judgments within the constrained output window**. As model scale increases from C1 to C4, FES surges from 0.33 to 2.24, indicating that 9B models scrupulously follow CoT frameworks for deep, step-by-step reflection. However, accuracy tends toward zero because strongly RLHF-aligned models tend to generate extended reasoning sequences spanning thousands of tokens, causing the final judgment token to fall outside the 512-token output budget. This creates a systematic evaluation artifact: the model completes the reasoning process but fails to emit the final judgment within the allotted token budget. The divergence between FES and accuracy thus reflects a fundamental mismatch between the model's cognitive architecture and the physical constraints of the deployment environment.

4.2 Logic depth decay and precision robustness

Figure 2 confirms that **accuracy monotonically decreases with reasoning steps**, consistent with error accumulation theory (Wei et al., 2022; Dziri et al., 2023). Even for the robust C1 model (0.8B, BF16), accuracy monotonically decreases from 80.0% at S2 to 50.0% at S10. This aligns with the theoretical prediction that in multi-step deduction chains, any logical error in an intermediate step propagates through subsequent steps, progressively increasing the probability of an incorrect final answer.

Concurrently, comparing BF16 (solid lines) with Q4 (dashed lines) of the same color reveals that **full-precision models consistently outperform quantized counterparts**. The persistent elevation of C1 and C2 curves above E1 and E2 strongly demonstrates that the continuous manifold of high-precision floating-point parameters holds irreplaceable critical value in maintaining logical state transmission fidelity across multi-step deductions.

4.3 Context pressure and quantization interaction

Figure 3 reveals that **larger models incur more pronounced capability depreciation under extreme context pressures (L3, L4)**. The dashed trajectories in the radar chart (e.g., E2)

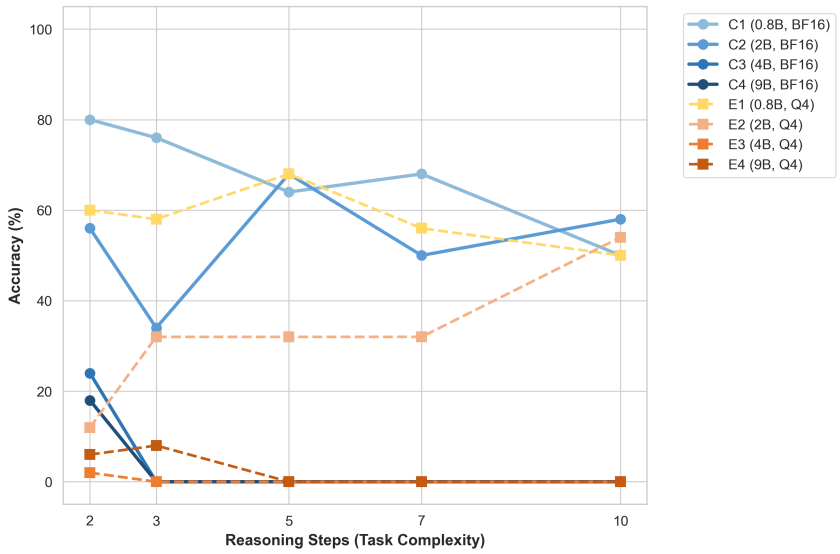

Figure 2: Logic decay curves over reasoning steps

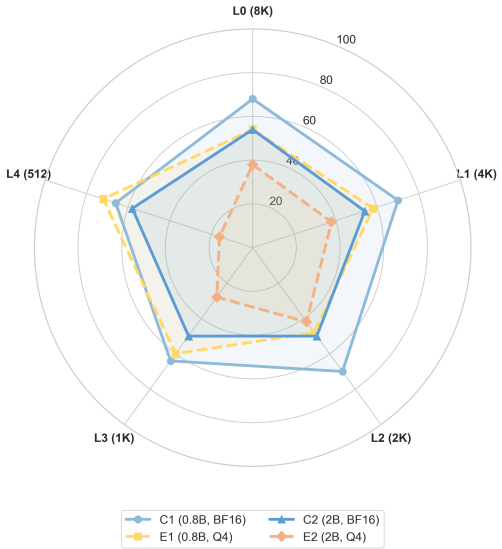

Figure 3: Context pressure robustness

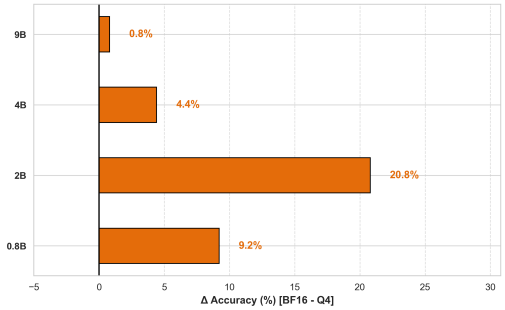

Figure 4: Quantization degradation by size

experience severe centripetal collapse toward the L4 dimension (512 tokens). For medium-to-large scale quantized models that are highly dependent on vast Attention Heads to maintain long-range context correlations, once the input window is subject to extreme compression and information density sharply increases alongside noise, their ability to maintain long-range contextual associations degrades significantly under extreme context compression.

Figure 4 further illustrates the quantization degradation gradient across model sizes. The weight precision loss brought by Q4_K_M quantization compounds with context pressure to create a dual destructive effect. This is particularly severe for 2B-scale models, where the quantized variant (E2) loses over 20 percentage points compared to its BF16 counterpart (C2), whereas the gap narrows at extreme scales where both variants already approach zero accuracy (see Appendix B for detailed truncation analysis).

4.4 FES analysis and small model judgment mechanisms

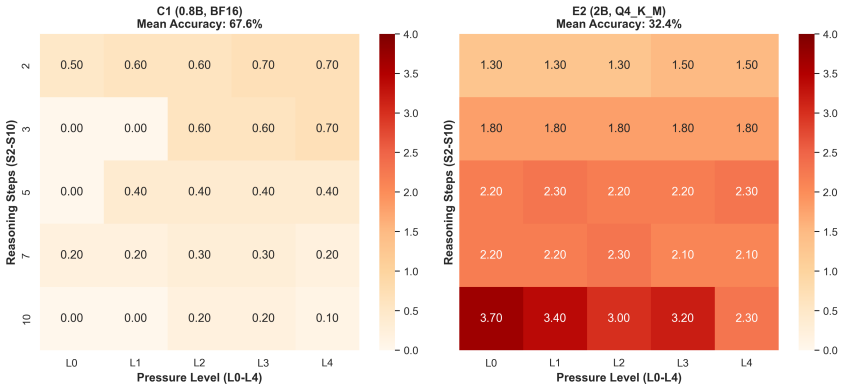

Figure 5: FES heatmap (C1 vs E2)

Due to the existence of the strict 512-token output truncation window, traditional evaluation metrics decouple in this task. The Formal Evaluation Score (FES) more accurately reflects the model's actual reasoning depth and internal reflection steps, whereas final Accuracy primarily demands a rapid response capability to physically output the judgment before truncation occurs.

The FES heatmap in Figure 5 and Table 4 provide the most direct empirical evidence for this mechanism by comparing C1 (0.8B, BF16) and E2 (2B, Q4_K_M) at the deepest reasoning level S10. Notably, both models consume a nearly identical VRAM budget ($\sim$1.5 GB vs $\sim$1.2 GB), making them direct competitors for resource-constrained edge deployment.

Model ID	L0 (8K)	L1 (4K)	L2 (2K)	L3 (1K)	L4 (512)
C1 (Qwen3.5-0.8B, BF16)	0.00	0.00	0.20	0.20	0.10
E2 (Qwen3.5-2B, Q4_K_M)	3.70	3.40	3.00	3.20	2.30

Table 4: C1 (0.8B) vs E2 (2B): FES score comparison at S10

The heatmap reveals a stark contrast: E2 (2B) stably garners high FES scores of 2.30–3.70 across all pressure levels at S10, indicating it performs significantly more reasoning steps and attempts to maintain a deep reflection structure. However, this extended reasoning process makes it highly susceptible to the output length limit, severely dragging down its mean accuracy (32.4%).

Conversely, C1 (0.8B) maintains near-zero FES across all conditions. It generates concise outputs that consistently fall within the token budget, thereby avoiding truncation and achieving a substantially higher mean accuracy (67.6%). This comparison yields a crucial conclusion: while deploying a larger quantized model (E2) under the same VRAM constraints brings deeper internal reasoning, one must heavily consider the output length limit scenario. If generation budgets are tight, prioritizing fast-responding micro-models is more practically viable.

4.5 VRAM dividend and effective reasoning rate

Figure 6 reveals the profound dialectic between model quantization and VRAM utilization. **Under constricted KV-caches, quantized models are more VRAM-conservative, yielding genuinely higher effective reasoning rates.** The earlier assertion that BF16 models have superior pinpoint accuracy hinges on the reality that pre-training distributions experience less fidelity loss compared to quantized variants.

However, in practical industrial or edge-side contexts with capped KV-cache capacity, physical VRAM operates as an insurmountable bedrock bottleneck. Even though high-

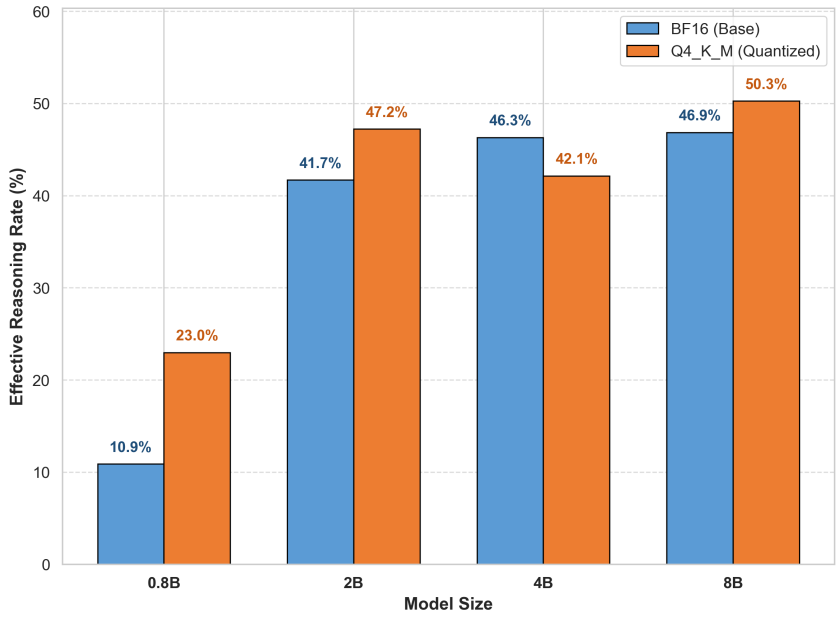

Figure 6: Effective reasoning rate (FES/step) by model size

precision models are theoretically superior, quantization algorithms such as Q4_K_M provide more than twice the context storage capacity under equivalent VRAM constraints. If the physical output ceiling imposing catastrophic truncations were structurally relaxed, this additional context capacity would allow quantized models to sustain longer generation sequences under the same VRAM budget. In such scenarios, quantized models could leverage lengthier generation permits, likely enabling them to surpass full-precision models forced to terminate reasoning prematurely due to VRAM exhaustion—potentially reversing the accuracy gap between quantized and full-precision models.

5 Theoretical Modeling: Mixture Decoding Truncation Survival Model

The observed system accuracy A_{obs} is not a monotone function of parameter count. Instead, it emerges from the interaction of three hidden factors under our benchmark: (1) instruction-following compliance, (2) Markovian reasoning fidelity along the deduction chain, and (3) survival probability under a hard output-length budget. We formalize these factors as expectations over two decoding trajectories: a Deep Reasoning Path that attempts full chain-of-thought deduction, and a Shortcut Guessing Path that relies on heuristic pattern matching without generating a complete proof.

5.1 Deep Reasoning Path Probability Modeling

Markovian logic fidelity. In the deep reasoning trajectory, each inference step must correctly transmit the internal logical state from one line of the proof to the next. Let the per-step error probability at numerical precision bit-width b be $\varepsilon(b) \in (0, 1)$. For a deduction chain of depth S, the probability that all steps remain logically correct is then

$$P_{\text{logic}}(S, b) = [1 - \varepsilon(b)]^S. \tag{1}$$

When $\varepsilon(b)$ is small, the binomial term admits a standard exponential approximation,

$$P_{\text{logic}}(S, b) \approx \exp\big(-S\,\varepsilon(b)\big), \tag{2}$$

which makes the multi-step degradation explicitly linear in both depth S and the effective error rate.

To capture the impact of quantization, we decompose the error rate into a full-precision baseline and an additive penalty:

$$\varepsilon(b) = \varepsilon_0 + \delta \cdot 2^{-2b}, \quad \varepsilon_0 > 0,\ \delta > 0. \tag{3}$$

Here ε_0 denotes the intrinsic per-step error at full precision (e.g., BF16), while $\delta 2^{-2b}$ models additional noise introduced by low-bit quantization. Substituting into the exponential approximation yields

$$P_{\text{logic}}(S, b) \approx \exp(-S\varepsilon_0) \cdot \exp\big(-S\delta 2^{-2b}\big), \tag{4}$$

naturally factorizing logic fidelity into a full-precision term and a quantization penalty term. This aligns with the empirical finding that deeper chains and more aggressive quantization jointly accelerate accuracy degradation.

Log-normal truncation survival. Let L denote the length of the model's produced chain of thought under the deep reasoning path. Empirically, L exhibits a heavy-tailed distribution that grows with model size and alignment strength. We model L with a log-normal distribution

$$L \sim \text{LogNormal}\big(\mu_L(M), \sigma_L^2(M)\big), \tag{5}$$

where the parameters $\mu_L(M)$ and $\sigma_L(M)$ increase with model size M and stronger RLHF alignment.

Given a hard output budget W (e.g., 512 tokens), the probability that the reasoning chain finishes before truncation is the CDF of the log-normal:

$$P_{\text{survive}}(W; M) = F_L\big(W; \mu_L(M), \sigma_L(M)\big). \tag{6}$$

As M and the alignment strength grow, $\mu_L(M)$ shifts rightward, making $\ln W - \mu_L(M)$ increasingly negative and driving $P_{\text{survive}}(W; M)$ toward zero. This captures the "alignment penalty" whereby highly aligned large models tend to generate proofs far longer than the available budget.

Deep-path success probability. Combining Markovian fidelity and truncation survival, the success probability of the deep reasoning path for a model with size M, precision b, deduction depth S, and output window W is

$$P_{\text{deep}}(S, W, b, M) = P_{\text{logic}}(S, b) \cdot P_{\text{survive}}(W; M). \tag{7}$$

This term represents the probability that the model both maintains logically correct intermediate states and manages to output the final judgment before being truncated.

5.2 Shortcut Guessing Path and Mixture Formulation

When a model's capacity is insufficient to reliably follow the structured proof instructions—typically for very small parameter scales such as 0.8B—it often fails to engage in full chain-of-thought reasoning. In this shortcut regime, the model bypasses most formal steps (yielding near-zero FES) and instead emits a fast decision based on shallow patterns in the premises.

We characterize this behavior with two ingredients:

- A compliance factor $\alpha(M, S) \in [0, 1]$, denoting the probability that the model attempts the deep reasoning protocol at depth S. This factor is empirically correlated with the Formal Evaluation Score.
- A shortcut accuracy $P_{\text{guess}}(M, S) \in [0.5, 1)$, representing the success rate of the heuristic shortcut path when the model does not follow the full protocol. Unlike a purely random binary guess (0.5), $P_{\text{guess}}(M, S)$ is allowed to exceed 0.5 because small models can still exploit local feature fitting (e.g., recognizing connectives or frequent patterns) without producing a valid proof trace.

The overall observed accuracy is then modeled as a mixture of the two trajectories:

$$A_{\text{obs}}(S, W, b, M) = \alpha(M, S)\, P_{\text{deep}}(S, W, b, M) + [1 - \alpha(M, S)]\, P_{\text{guess}}(M, S). \tag{8}$$

This unified expression explains both extremes:

Large, highly aligned models: here $\alpha(M, S) \approx 1$ and $P_{\text{guess}}(M, S)$ contributes negligibly. However, the expected chain length far exceeds W, so $P_{\text{survive}}(W; M) \approx 0$. Consequently, $P_{\text{deep}} \approx 0$ and the first term vanishes, leading to $A_{\text{obs}} \approx 0$ despite strong internal reasoning.

Small, micro-scale models: these models often exhibit $\alpha(M, S) \ll 1$ and very low FES, but retain a shortcut accuracy $P_{\text{guess}}(M, S)$ that can empirically reach the 0.6–0.7 range. In this case A_{obs} is dominated by the second term, matching the observed $\sim 67.6\%$ accuracy of 0.8B models without requiring a contradictory assumption of "pure random guessing".

By fitting $\alpha(M, S)$ and $P_{\text{guess}}(M, S)$ from data while keeping P_{deep} grounded in the Markov-plus-truncation structure above, the Mixture Decoding Truncation Survival Model offers a self-consistent mathematical explanation for the divergent behaviors of large and small models under the same physical constraints.

6 Conclusion and future directions

Based on the reconstructed rigorous mathematical logic, our experimental conclusions must be updated to highlight the interplay of alignment penalties, quantization degradation, and compute allocation:

Conclusion 1: Micro-models are optimally suited for strictly constrained edge environments due to their evasion of the "Alignment Tax." The experimental results demonstrate that strongly aligned large models tend to generate extended reflection sequences whose lengths substantially exceed the output budgets available on edge devices ($W = 512$). In contrast, small models bypass physical truncation constraints by directly outputting binary judgments. Therefore, for strict real-world engineering constraints where output budget is fixed, prioritizing micro-models yields a significantly higher valid completion rate.

Conclusion 2: The core cost of quantization is the erosion of Markov state fidelity, not surface-level accuracy cliffs. After excluding truncation artifacts, directly comparing the 2B and 4B models across BF16 and Q4 precisions unveils the true destructive mechanism of quantization. Low-bit quantization does not instantaneously render a model "incompetent"; rather, it introduces noise into hidden-layer feature transmission, reducing the per-step deduction fidelity λ. For tasks requiring multi-step deduction ($S \geq 5$) under high context pressure ($L \geq 2$), such errors accumulate exponentially according to λ^S, leading to significant degradation in multi-step reasoning accuracy.

Conclusion 3: The optimization trajectory for Test-Time Compute in edge deployment demands a paradigm shift. Facing stringent physical memory and output time constraints, reducing model parameter scale alone is insufficient to address the fundamental constraints. To genuinely realize complex logical reasoning on the edge, the future engineering focus must pivot to "output budget control" for large language models. Specifically, we must develop algorithms capable of suppressing the verbosity bias of large models via Instruction Tuning within a given maximum token budget W, or employing specific decoding interventions (e.g., adaptive depth restrictions) to directly elevate their survival probability $P_{survive}$ within the constrained window. Combining strong logical fidelity with token-efficient generation is necessary to achieve effective performance in edge deployment scenarios.

Ethics Statement

Our research aligns with the COLM 2026 Code of Ethics by transparently disclosing the fundamental limitations and potential hazards of deploying Large Language Models (LLMs) in resource-constrained environments.

The performance disparities revealed in our study demonstrate a critical vulnerability: while deep RLHF alignment enhances moral and safety compliance in large models, it simultaneously induces an uncontrollable "Verbosity Bias." Under strict physical constraints such as finite KV-cache and hard output token truncation, this bias leads to output truncation, preventing the model from emitting final judgments within the available token budget.

We emphasize a profound safety and engineering concern: deploying large-scale models in edge computing scenarios without accounting for physical output constraints poses significant risks. In such environments, the model's failure to emit a final judgment due to output truncation may result in undetected incomplete responses, which poses risks in safety-critical applications. We urge the community to thoroughly evaluate physical operational limits before deployment to prevent incomplete or erroneous model outputs from causing adverse outcomes in safety-critical applications.

References

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jian, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D Hwang, et al. Faith and fate: Limits of transformers on compositionality. *arXiv preprint arXiv:2305.10601*, 2023.

Qwen Team. Qwen 3.5: The new qwen series models. GitHub repository https://github.com/QwenLM/Qwen, 2026.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Unsloth AI. Qwen 3.5 gguf model collection. Hugging Face https://huggingface.co/collections/unsloth/qwen35, 2026.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

Appendix

Contents

1 Introduction 1
2 Research methodology and experimental design 2
2.1 Evaluation Model Selection and Matrix Configuration 2
2.2 Multi-dimensional Environmental Constraint Interventions 2
2.3 Core Evaluation Metrics 2
3 Quantitative experimental results 3
3.1 Evolution of Logical Depth and Scale Evaluation 3
3.2 Resistance and Collapse under Context Pressure 3
4 Discussion 4
4.1 Global Performance Decoupling 4
4.2 Logic Depth Decay and Precision Robustness 4
4.3 Context Pressure and Quantization Interaction 4
4.4 FES Analysis and Small Model Judgment Mechanisms 6
4.5 VRAM Dividend and Effective Reasoning Rate 6
5 Theoretical modeling: Mixture Decoding Truncation Survival Model 7
5.1 Deep Reasoning Path Probability Modeling 7
5.2 Shortcut Guessing Path and Mixture Formulation 8
6 Conclusion and future directions 9

A Overlay effects of logic depth and context pressure 13
B Truncation mask and performance evaluation 13
C FES detailed comparison across all 8 models 13
D Extended derivations for the Mixture Model 14
D.1 Markovian Fidelity and Quantization Degradation 14
D.2 Log-Normal Truncation Survival Analysis 14
D.3 Shortcut Guessing Baseline Assessment 14
E Runtime Configuration and Two-Stage Prompt Templates 15
E.1 Inference framework 15
E.2 Two-stage prompting design 15
E.3 Context pressure and decoding settings 15
E.4 Stage 1 prompt for Qwen models 15
E.5 Stage 2 prompt for Qwen models 16
E.6 Output compliance and final accuracy 17

E.7 FES evaluation 17
E.8 Reproducibility note 17

F LLM Usage Disclosure 18

In this appendix, we provide supplementary experimental data, extended theoretical derivations, additional visualizations, and implementation details that support the findings presented in the main text. Appendix A presents the cross-product overlay effects of logic depth and context pressure. Appendix B analyzes how output truncation masks the true reasoning potential of large models. Appendix C provides the detailed FES heatmap comparing all 8 models. Appendix D groups the extended theoretical derivations, where Appendix D.1 formalizes the quantization-enhanced error accumulation in Markovian fidelity, Appendix D.2 presents the Log-Normal truncation survival analysis, and Appendix D.3 assesses the shortcut guessing baseline. Appendix E documents the runtime pipeline and two-stage prompt templates for reproducibility.

A Overlay effects of logic depth and context pressure

Table 5 shows the comprehensive accuracy when considering the cross-product of logic steps (S-series) and context pressure (L-series), averaged across all models.

Logic Steps	L0 (8K)	L1 (4K)	L2 (2K)	L3 (1K)	L4 (512)	Mean
S2 (2 steps)	33.8%	35.0%	32.5%	31.2%	28.7%	**32.2%**
S3 (3 steps)	27.5%	30.0%	28.8%	21.2%	22.5%	26.0%
S5 (5 steps)	28.7%	28.7%	30.0%	28.7%	28.7%	29.0%
S7 (7 steps)	25.0%	26.2%	23.8%	27.5%	26.2%	25.8%
S10 (10 steps)	27.5%	26.2%	25.0%	26.2%	27.5%	26.5%

Table 5: Comprehensive accuracy: difficulty depth $\times$ context pressure

At the simplest S2 level, the highest average accuracy is 32.2%. As steps increase, accuracy rapidly converges to the 25%–29% range, demonstrating the performance ceiling under complex cross-constraints.

B Truncation mask and performance evaluation

From the accuracy gradient in Figure 4 and Figure 2, within the 0.8B–2B interval, judgment accuracy visibly decays with increasing parameters (e.g., 67.6% in C1 to 53.2% in C2, with quantized models losing more). However, this trend encounters systematic data distortion when expanding to 4B and 9B. Large models failing to output correct judgments under constrained scenarios severely distort the statistical distributions. A 0% accuracy rooted in 512-token truncation statistically masks the true logical reasoning potential of 4B/9B models. Thus, relying solely on superficial accuracy to map the Scaling Laws of language models in constraint-driven tasks is fundamentally infeasible.

C FES detailed comparison across all 8 models

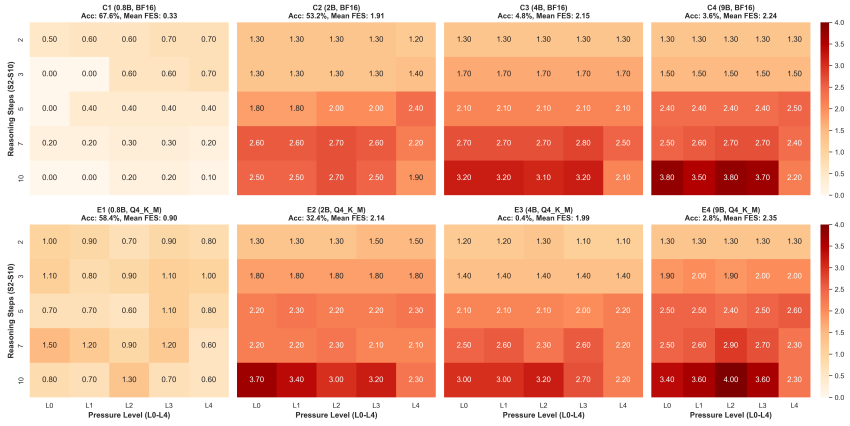

Figure 7: FES (valid inference steps) heatmap across all 8 models

To further quantify the global formal performance contrast, Figure 7 presents the detailed FES data across all 8 models. Across all Step$\times$Level combinations, the heatmap reveals a clear gradient: larger models (C3, E3, C4, E4) consistently produce higher FES scores, while small models (C1, E1) remain anchored near zero across all conditions, further confirming the "fast thinking" mechanism described in Section 4.

D Extended derivations for the Mixture Model

D.1 Markovian fidelity and quantization degradation

The deep reasoning path assumes that each application of a logical inference rule faithfully transmits the internal state of the proof. Let the per-step error probability at precision bit-width b be $\varepsilon(b)$, modeled as

$$\varepsilon(b) = \varepsilon_0 + \delta \cdot 2^{-2b}, \quad \varepsilon_0 > 0,\ \delta > 0. \tag{9}$$

Here ε_0 is the baseline error under full-precision BF16, while $\delta 2^{-2b}$ captures the extra noise introduced by low-bit quantization; this term grows as the bit-width decreases.

For a deduction chain of depth S, the probability of preserving a correct logical state throughout the entire chain is

$$P_{\text{logic}}(S, b) = [1 - \varepsilon(b)]^S \approx \exp\big(-S\varepsilon(b)\big) = e^{-\varepsilon_0 S} \cdot e^{-S\delta 2^{-2b}}. \tag{10}$$

The first factor corresponds to the full-precision Markovian fidelity, while the second factor is an exponential penalty driven jointly by depth S and quantization level b. This decomposition theoretically accounts for the rapid accuracy collapse of 2B/4B Q4 models at higher reasoning depths ($S \geq 5$) observed in the main experiments.

D.2 Log-Normal truncation survival analysis

In strongly aligned models, the length L of the generated chain of thought is well approximated by a log-normal distribution,

$$L \sim \text{LogNormal}\big(\mu_L(M), \sigma_L^2(M)\big), \tag{11}$$

where both $\mu_L(M)$ and $\sigma_L(M)$ increase with model size and alignment strength.

Given a hard output budget W, the probability that the final judgment token is emitted before truncation is the CDF of the log-normal:

$$P_{\text{survive}}(W; M) = \frac{1}{2} + \frac{1}{2}\,\text{erf}\left(\frac{\ln W - \mu_L(M)}{\sigma_L(M)\sqrt{2}}\right). \tag{12}$$

As M scales from 0.8B to 9B, empirical measurements show $\mathbb{E}[L]$ rapidly surpassing 512 tokens, causing $\ln W - \mu_L(M)$ to become highly negative. In this limit, the error function approaches -1, and $P_{\text{survive}}(W; M)$ tends to 0, formalizing the "alignment penalty artifact" where large models systematically lose their final answers due to truncation.

D.3 Shortcut guessing baseline assessment

For extremely small models (e.g., 0.8B) faced with complex formal deduction tasks, instruction compliance often collapses, and the model rarely follows the requested proof-style protocol. In this regime, the compliance factor satisfies $\alpha(M, S) \to 0$, and the mixture model reduces to

$$\lim_{\alpha \to 0} A_{\text{obs}}(S, W, b, M) = P_{\text{guess}}(M, S). \tag{13}$$

Because the task is a binary TRUE/FALSE judgment, a purely random guess would yield $P_{\text{guess}} = 0.5$. However, empirical observations show that 0.8B models consistently achieve 60–70% accuracy even with near-zero FES, indicating that $P_{\text{guess}}(M, S)$ is boosted by local feature fitting, shallow pattern recognition, and dataset biases.

This analysis clarifies that the apparent "advantage" of micro-models in constrained settings stems from their reliance on robust shortcut heuristics that bypass both deep reasoning and truncation. Their behavior is thus captured by an elevated but still heuristic $P_{\text{guess}}(M, S)$, rather than by ideal Markovian fidelity or survival under long proof traces.

E Runtime Configuration and Two-Stage Prompt Templates

This appendix documents the implementation-level settings used in the main experiment, including the inference framework, decoding configuration, and the two-stage prompt templates. The purpose of this appendix is to improve reproducibility by making the runtime pipeline explicit.

For readability, we provide an English rendering of the prompts below.

E.1 Inference framework

All experiments were executed locally under the `llama.cpp` framework through the `llama-server` HTTP backend. Each model was loaded as a GGUF checkpoint, and inference was issued through a unified completion-based interface. This setup allowed us to test the same Qwen model family under different context-window constraints while keeping the serving framework fixed.

The evaluated model family consisted of eight Qwen3.5 variants spanning four parameter scales and two precision settings: 0.8B, 2B, 4B, and 9B, each tested in BF16 and Q4_K_M form. These eight variants formed the complete model axis of the experimental matrix.

E.2 Two-stage prompting design

The experiment used a two-stage prompting pipeline. In Stage 1, the model was asked to generate a structured formal deduction trace from the premises to the target conclusion. In Stage 2, the model was given the original problem together with the Stage 1 reasoning trace and was required to output only a final binary judgment.

This design separated formal reasoning generation from final answer emission. It also made it possible to analyze cases in which a model produced an extended reasoning trace but failed to emit the final judgment under a constrained output budget.

E.3 Context pressure and decoding settings

We implemented five context-pressure levels by varying the context window allocated to the inference server: L0 = 8192, L1 = 4096, L2 = 2048, L3 = 1024, and L4 = 512. These settings were used to simulate progressively tighter memory and generation constraints.

Decoding was deterministic throughout the main experiment, with temperature fixed at 0.0. Stage 1 reasoning generation used pressure-dependent token budgets, whereas Stage 2 used a fixed output budget for the final judgment stage.

E.4 Stage 1 prompt for Qwen models

The Stage 1 prompt was designed to force the model into a proof-style formal deduction mode. It explicitly restricted the model to a predefined set of inference rules and required every derived line to follow a numbered proof format.

Stage 1 system prompt

```
You are a strict formal-logic verifier. Your task is to derive the target
conclusion from the given premises step by step.

You may use only the following 12 inference rules, each referenced by a
method label:

- method1: Modus Ponens
- method2: Modus Tollens
- method3: Hypothetical Syllogism
- method4: Disjunctive Syllogism
```

```
- method5: Addition
- method6: Simplification
- method7: Conjunction
- method8: Constructive Dilemma
- method9: Destructive Dilemma
- method10: Double Negation
- method11: De Morgan's Laws
- method12: Contraposition

Formatting rules are extremely important.

First, list the premises in order. If there are 3 premises, then steps 1,
2, and 3 must be the premises themselves.

After that, every new deduction step must follow this exact format:

{step number}. {derived logical formula} ({referenced step numbers}, methodX)

Example:

1. p
2. p -> q
3. r
4. q (1, 2, method1)
5. q | r (4, method5)

In this stage, output only the numbered deduction process with explicit
references and method labels. Do not output any final judgment, reflection,
explanation, or extra commentary.
```

Stage 1 user prompt template

```
Please try to derive the target conclusion from the given premises step by
step.

Premises:
1. {premise_1}
2. {premise_2}
...
n. {premise_n}

Target conclusion: {conclusion}
Required deduction depth: {steps_count}

Please strictly follow the required proof format and provide the derivation
step by step.
```

To reduce unwanted preambles and encourage immediate structured generation, the assistant response was prefixed so that generation began directly from:

```
1.
```

E.5 Stage 2 prompt for Qwen models

The Stage 2 prompt was designed to suppress any additional reasoning and force the model to emit only the final binary decision. This stage evaluated whether the model could successfully deliver the judgment after the reasoning stage.

Stage 2 system prompt

```
You are a strict formal-logic verifier.

You will be given a problem statement and a reasoning trace. At this stage,
you must not perform any additional reasoning, explanation, or analysis.

You must output one and only one of the following two lines:

JUDGMENT: TRUE
JUDGMENT: FALSE

Your response must end immediately after the judgment line.
```

Stage 2 user prompt template

```
Problem:
Premises:
1. {premise_1}
2. {premise_2}
...
n. {premise_n}

Conclusion: {conclusion}
Required deduction depth: {steps_count}

Reasoning trace (possibly truncated):
{reasoning_tail}

Now output only the final judgment.
```

Only the tail portion of the Stage 1 reasoning trace was passed into Stage 2. This design preserved the final portion of the generated deduction while keeping the second-stage prompt within the active context budget.

E.6 Output compliance and final accuracy

Final accuracy was computed directly from the Stage 2 output by comparing the model's emitted binary judgment against the ground-truth validity label of the corresponding proposition instance. No external language model was used to determine the final accuracy label.

Any Stage 2 response that failed to produce exactly one compliant judgment line was treated as non-compliant for strict final-answer evaluation. In particular, if the model did not end with either "JUDGMENT: TRUE" or "JUDGMENT: FALSE", the output was counted as incorrect under the accuracy metric.

E.7 FES evaluation

The Formal Evaluation Score (FES) was assigned to the Stage 1 reasoning output and was intended to measure formal compliance rather than final-answer correctness. Under a fixed rubric, the evaluator examined whether the reasoning trace preserved explicit step numbering, valid step references, required method labels, and overall proof-like structural completeness.

E.8 Reproducibility note

The main purpose of this appendix is to expose implementation details that materially affect the reported results, especially the serving framework, context-window constraints, deterministic decoding, and the strict two-stage prompt design. Since the observed behavior

is sensitive to formatting and output-budget control, these implementation details should be treated as part of the experimental specification rather than incidental engineering choices.

F LLM Usage Disclosure

In accordance with the COLM 2026 policy regarding the use of Large Language Models (LLMs), we explicitly disclose the use of AI assistance in our evaluation pipeline.

Specifically, Gemini 3.1 Pro was used as an external evaluator for assigning the Formal Evaluation Score (FES) to the Stage 1 reasoning outputs generated by the tested models.

The FES evaluation was designed to assess the degree to which a model's Stage 1 output followed the required formal deduction protocol. Under a fixed evaluation rubric, Gemini 3.1 Pro examined whether the response adhered to the required numbered proof format, maintained valid references to earlier steps, invoked the designated inference-rule labels (e.g., method1–method12), and preserved the overall structural completeness of the deduction trace. The resulting FES was used as a measure of formal reasoning compliance rather than final-answer correctness.

We emphasize that Gemini 3.1 Pro was not used to determine the final accuracy reported in the main experiment. Accuracy was computed directly by comparing the final binary judgment produced by the evaluated model against the ground-truth validity label of each proposition instance.

Gemini 3.1 Pro only served as an external evaluator for FES scoring and did not generate the reasoning traces or the final judgments reported as model outputs. To reduce evaluator variance, we used a fixed prompt template and a consistent scoring rubric across all experimental conditions.