The Illusion of Procedural Reasoning: Measuring Long-Horizon FSM Execution in LLMs

Mahdi Samiei[1], **Mahdi Mansouri**[1], **Mahdieh Soleymani Baghshah**[1]

[1]Department of Computer Engineering, Sharif university of Technology
mm.samiei@sharif.edu, mhd.mansouri01@sharif.edu, soleymani@sharif.edu

Abstract

Large language models (LLMs) have achieved remarkable results on tasks framed as reasoning problems, yet their true ability to perform procedural reasoning, executing multi-step, rule-based computations remains unclear. Unlike algorithmic systems, which can deterministically execute long-horizon symbolic procedures, LLMs often degrade under extended reasoning chains, but there is no controlled, interpretable benchmark to isolate and measure this collapse. We introduce Finite-State Machine (FSM) Execution as a minimal, fully interpretable framework for evaluating the procedural reasoning capacity of LLMs. In our setup, the model is given an explicit FSM definition and must execute it step-by-step given input actions, maintaining state consistency over multiple turns. This task requires no world knowledge, only faithful application of deterministic transition rules, making it a direct probe of the model's internal procedural fidelity. We measure both Turn Accuracy (local correctness) and Task Accuracy (global long-horizon correctness) to disentangle immediate computation from cumulative state maintenance. Empirical results reveal systematic degradation as task horizon or branching complexity increases. Models perform significantly worse when rule retrieval involves high branching factors (many actions per state) than when memory span is long (many states, few actions). Larger models show improved local accuracy but remain brittle under multi-step reasoning unless explicitly prompted to externalize intermediate steps. These findings expose a consistent illusion of procedural reasoning: LLMs can mimic algorithmic behavior for short traces but fail to sustain coherent execution as procedural depth grows. FSM-based evaluation offers a transparent, complexity-controlled probe for diagnosing this failure mode and guiding the design of inductive biases, memory mechanisms, and reasoning scaffolds that enable genuine long-horizon procedural competence. By grounding "reasoning" in measurable execution fidelity rather than surface correctness, this work helps establish a rigorous experimental foundation for understanding and improving the algorithmic reliability of LLMs.

Introduction

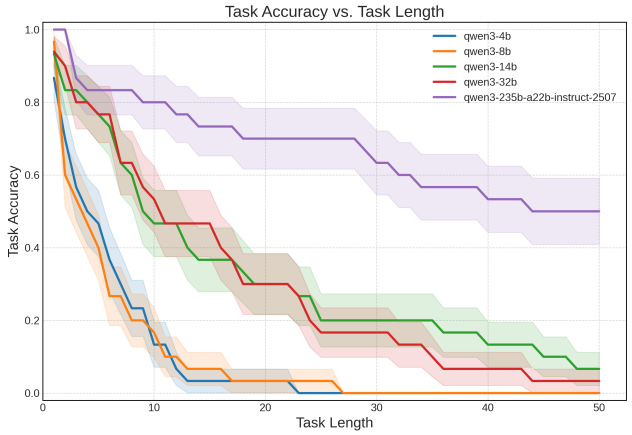
LLMs have demonstrated striking performance on a wide range of benchmarks that are often described as requiring reasoning. Yet a central question remains unsettled: do LLMs possess genuine reasoning competence, or are they primarily leveraging statistical pattern matching over surface forms and short-horizon reasoning problems? Recent studies have cast doubt on the former view by showing that LLM performance frequently collapses as samples complexity increases, measured by longer compositions, deeper dependency chains, or more distractors and that apparent successes are often concentrated on lower complexity instances that may be susceptible to template memorization or training data contamination (Shojaee et al. 2025; Mirzadeh et al. 2025; Sun et al. 2025; Sinha et al. 2025; Paqaleh et al. 2025; Zhou et al. 2025) . These observations have sharpened the debate over whether current LLMs truly reason or merely extrapolate familiar patterns.

A growing line of work proposes decomposing reasoning into two phases: planning and execution (Shojaee et al. 2025; Sinha et al. 2025) . Planning concerns deriving a solution strategy or algorithm for a problem; execution concerns faithfully carrying out that strategy step by step. Empirically, even when the algorithmic plan is supplied to the model, performance can still degrade precipitously as instance complexity rises, implicating failures in the execution phase rather than in plan discovery per sample.

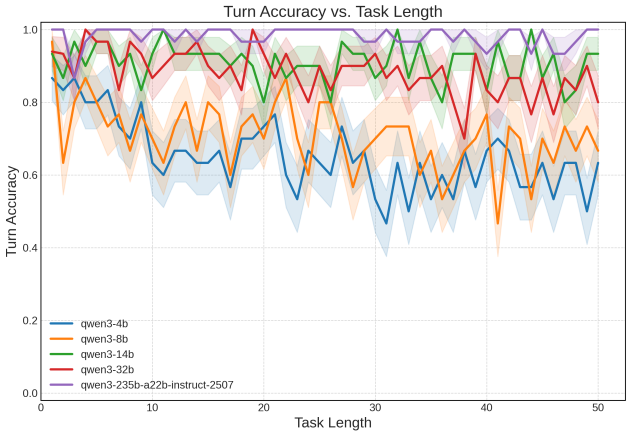
Sinha et al. (2025) provide complementary evidence from purely executional tasks in which the "plan" reduces to simple retrieval and aggregation—for example, retrieving numeric values by given word keys and summing them, yet models struggle to generalize beyond short-horizon templates. This work also highlight a self-conditioning effect whereby incorrect intermediate outputs feed back into the context, increasing the likelihood of subsequent errors and inducing error cascades over multi-step interactions.

Despite substantial progress, existing benchmarks rarely disentangle planning from execution under precisely controlled complexity, nor do they provide a minimal environment in which the correctness of each intermediate step is unambiguous. Building on these perspectives, we propose a complementary line of inquiry: FSMs as a controllable probe of reasoning and execution complexity in LLMs. FSMs represent explicit, interpretable procedural structures with precisely quantifiable complexity—defined by the number of states, transitions, and branching dependencies. When an LLM is tasked with executing an FSM—simulating the state updates for a given input se-

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.



(a) **Task Accuracy vs. Model Scale.** While scaling improves cumulative execution fidelity, even the largest model (Qwen3-235B) reaches only about 50% overall task accuracy, indicating persistent long-horizon degradation.



(b) **Turn Accuracy vs. Model Scale.** Larger Qwen models achieve higher per-turn correctness when executing FSM transitions. Accuracy increases steadily with parameter count, suggesting stronger local rule adherence in larger models.

Figure 1: Task accuracy and Turn accuracy comparison for different models

quence from an initial state until a goal condition is met—it must perform a sequence of discrete, deterministic reasoning steps. Unlike symbolic puzzles or text-based math problems, FSMs eliminate ambiguity in what constitutes a correct step, thereby enabling precise measurement of systematic reasoning failures as structural complexity and horizon length increase.

Our experiments show that even bare-bones FSM execution is challenging for current LLMs. Accuracy declines predictably with the number of actions and input length, and we observe clear self-conditioning dynamics: an early state-tracking mistake propagates through subsequent steps and amplifies downstream error rates. Moreover, enabling "thinking" prompting does not resolve the collapse at higher complexities, suggesting that the bottleneck lies in reliable stepwise execution rather than plan articulation. FSM-based evaluation thus offers a minimal, interpretable environment to quantify this failure mode and to ground future research on inductive biases, modular controllers, training curricula, and external memory mechanisms that may enable true generalization to increasing procedural complexity.

Formulation

To completely evaluate an LLM's ability to maintain long-term, procedural state, we employ the FSM as a ground-truth framework. An FSM is a computational model that is always in one of a finite set of states. Its 'next' state is determined only by its 'current' state and a given 'action,' according to a predefined set of transition rules. This makes it an ideal tool for this experiment: it is computationally simple for a classical machine but challenging for an LLM, as it requires perfect adherence to arbitrary rules and flawless state memory over long interactions, rather than relying on semantic or world knowledge.

Formally, we define our FSM as a 4-tuple $M = (Q, \Sigma, \delta, q_0)$, where:

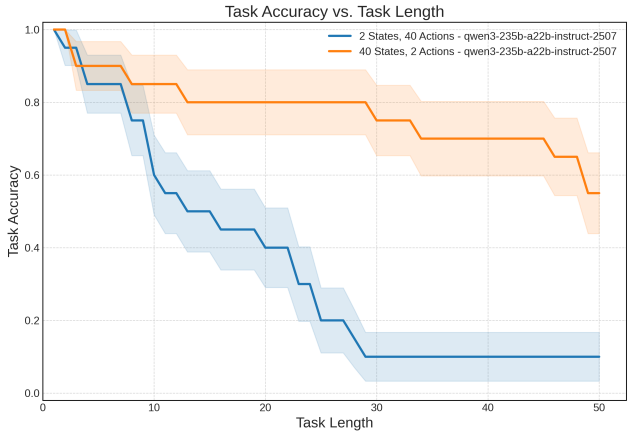
- Q is a finite set of states.
- Σ is a finite set of actions.
- $\delta : Q \times \Sigma \to Q$ is the transition function that maps a (state, action) pair to a new state.
- $q_0 \in Q$ is the designated initial state.

To ensure the task tests rule-following rather than semantic inference, we populate the states (Q) and actions (Σ) from disjoint sets of simple, single-token English words (e.g., states as nouns like 'cat', 'desk'; actions as adjectives like 'red', 'fast'). This prevents the model from "guessing" transitions based on word association.

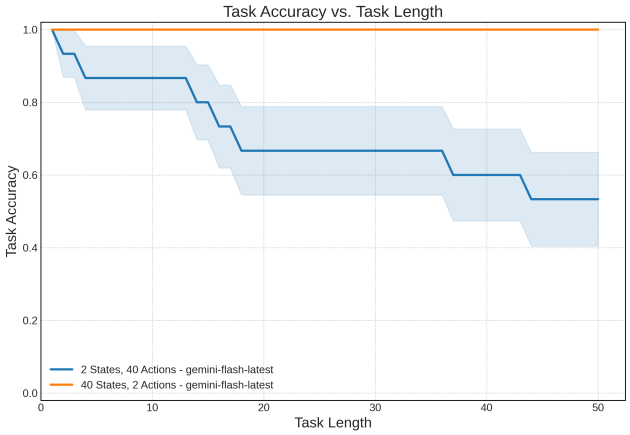
The complete FSM definition—including all states, actions, the initial state q_0, and a full list of transitions (e.g., `From cat, on action red, go to desk.`)—is provided to the LLM in a system prompt. This prompt sets the LLM's role as an FSM executor and strictly defines the output format as `<state>FinalState</state>`. At each conversational turn, the model receives a sequence of actions and outputs the resulting state, which becomes the starting state for the next turn.

To evaluate performance, we define two metrics:

1. **Turn Accuracy:** This metric evaluates the model's computational correctness on a turn-by-turn basis. It checks if the model's reported state at turn t is the correct result after applying the action sequence given in the prompt from turn $t-1$. Turn Accuracy answers the question: "Did the model perform this single calculation correctly, started from the previous state even if it was wrong?"
2. **Task Accuracy:** This is a much stricter metric measuring long-term fidelity. It checks if the model's reported state at turn t is the true state, assuming a perfect path from the initial state q_0. Task accuracy is 1 only if every preceding turn (including the current one) was also correct.



(a) Task accuracy comparison for a 2-state/40-action vs 40-state/2-action setup. The setup with 2 states and 40 actions are much harder to solve for an LLM while using 40 states and 2 actions will lead to much better accuracy for long-horizon tasks.



(b) Task accuracy comparison for a 2-state/40-action vs 40-state/2-action setup. Gemini-2.5-flash can reach only about 50% accuracy at the end of a task with 50 steps with only 2 states and 40 actions meanwhile this model can reach 100% accuracy for a 40-state/2-action setup.

Figure 2: Task accuracy comparison for a Wide & Shallow setup vs Deep & Narrow setup.

Task Accuracy answers the question: "Has the model remained on the correct path since the very beginning?"

The distinction is critical: Turn accuracy measures the model's immediate processing ability, while task accuracy measures its long-term state-holding capacity. A model can have a high turn accuracy (it calculates correctly from its own, flawed state) but a low task accuracy (it was knocked off the correct path many turns ago and never recovered).

Experiments

We applied our FSM formulation to evaluate several LLMs. Our experiments were designed to investigate three primary questions: (1) the effect of model scale on state-tracking fidelity, (2) the impact of FSM structure (state-space vs. action-space complexity), and (3) the model's ability to handle multi-step sequential instructions within a single turn.

Scaling Effects on Rule Adherence

We first investigated the effect of model scale on a baseline FSM task (4 states, 5 actions). As shown in Figure 2a, we tested various Qwen models of different parameter sizes. The results indicate a strong positive correlation between model scale and both **Turn Accuracy** and **Task Accuracy**. This suggests that larger models possess superior capabilities for strict rule-adherence and long-term state maintenance, with Qwen3-235b substantially outperforming all other models in this test. Even though this outperforming model could get about 50% accuracy in the tasks.

A key finding, visible in Figure 2a, is the effect of "negative self-conditioning," particularly in smaller models. These models exhibit a clear decaying pattern in Turn Accuracy over time.

This is a significant observation. While Turn Accuracy is measured relative to the model's own previous state—and should, in theory, remain independent of past task failures—the data suggests it is not. This pattern, which echoes the findings of Sinha et al. (2025), implies that as a model makes mistakes, its ability to perform the next computational step correctly (even from its own flawed state) is also degraded.

Therefore, for these models, Turn Accuracy is not truly independent of the conversational history. It appears that as the context window becomes "polluted" with the model's own errors, its immediate processing ability is compromised, creating a cycle of compounding failure.

State-Space vs. Action-Space: A Counter-Intuitive Finding

A surprising and central finding of our work is illustrated in Figure 2. We tested two high-performing models (Qwen3-235b and Gemini-2.5-flash) on two FSM configurations with an identical number of total transitions (80):

- **Config 1 (Wide & Shallow):** 2 states, 40 actions.
- **Config 2 (Deep & Narrow):** 40 states, 2 actions.

Intuition might suggest that Config 1 (fewer states) would be easier, as even a random guess of the final state would have a 50% probability of being correct.

Our findings demonstrate the opposite: both models performed significantly better on the FSM with 40 states and 2 actions.

We hypothesize this is not a failure of *state memory* but a failure of *rule retrieval*. In the 2-state/40-action FSM, each state has a very high *branching factor*. When processing a turn, the model must locate the single correct transition rule from a list of 40 very similar rules (e.g., `From state_A, on action_1...`, `From state_A, on action_2...`). This "needle in a haystack" problem appears to confuse the model's attention

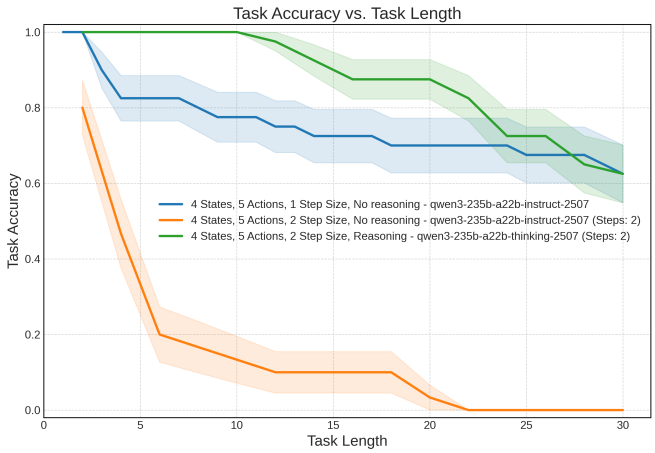


Figure 3: Increasing step size to 2 will result a huge performance degradation on a 4-state/5-action setup. It indicated that steps should also be atomized to reach high task accuracy. Using reasoning in this setup will lead to much higher performance in the cost of reasoning tokens generated by the model.

mechanism, leading to frequent errors. Conversely, the 40-state/2-action FSM has a low branching factor (2 rules per state), making rule retrieval trivial, even if the total *state space* is large.

This leads to a practical design principle: when building LLM-based systems (like multi-agent frameworks or Chain-of-Thought prompts), it is preferable to design workflows with many simple decision points (many states, few actions) rather than few complex decision points (few states, many actions).

Instruction Complexity and Mitigation via Reasoning

Finally, we investigated the impact of *instruction complexity* per turn, as shown in Figure 3. In all previous experiments, we used a step size of one (one action per turn). In this experiment, we also tested a step size of two (two sequential actions in a single prompt).

As the plot illustrates, performance degraded completely when models were asked to process two sequential actions in one turn. This highlights a critical limitation: LLMs struggle to perform multiple, sequential computational steps ”in-head” within a single generation. It is far more effective to break the process into distinct user prompts.

Crucially, we found this degradation can be significantly mitigated by enabling model reasoning. When the model was prompted to ”think” through the two steps, it effectively used its context window as a scratchpad to record the *intermediate state* after the first action, before proceeding to the second. This externalization of the intermediate step allowed it to compute the final state correctly. The results were much better, though this came at the cost of additional reasoning tokens.

Conclusion

This work introduced FSMs as a minimal yet powerful framework for probing the procedural reasoning capabilities of large language models. By decoupling symbolic execution from semantic interpretation, FSM-based evaluation allows precise measurement of a model’s ability to maintain and manipulate internal state over long horizons. Across a range of models and configurations, we observed a consistent pattern: LLMs can follow short procedural traces accurately but fail predictably as the complexity or temporal depth of the task increases. The degradation follows a self-conditioning dynamic, where early local errors propagate through subsequent steps, leading to global collapse in long-term state fidelity.

Our experiments reveal several key insights. First, model scale improves short-horizon execution but does not fundamentally eliminate long-horizon instability, indicating that scaling alone may not yield genuine procedural competence. Second, the sharp performance asymmetry between wide and deep FSMs highlights that failures often stem from rule retrieval under high branching factors rather than from limited memory over large state spaces. Finally, the catastrophic drop in accuracy for multi-step (two-action) instructions demonstrates the fragility of internal, unexternalized reasoning, an effect that can be partially mitigated when the model is encouraged to externalize intermediate steps through explicit reasoning or scratchpad generation.

Together, these findings suggest that current LLMs exhibit an illusion of procedural reasoning: they can imitate the appearance of algorithmic control in short contexts but lack stable internal mechanisms for multi-step execution. FSM-based evaluation offers a simple, interpretable testbed for isolating this failure mode and guiding the development of architectures and prompting strategies that incorporate compositional inductive bias, explicit memory, or modular execution scaffolds. Future research should explore integrating symbolic controllers, structured reasoning traces, or recurrent supervision to endow models with the ability to sustain coherent, rule-based computation across extended horizons, an essential step toward genuine systematic reasoning.

References

Mirzadeh, S. I.; Alizadeh, K.; Shahrokhi, H.; Tuzel, O.; Bengio, S.; and Farajtabar, M. 2025. GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models. In *The Thirteenth International Conference on Learning Representations*.

Paqaleh, M. M. S.; Marioriyad, A.; Tahmasebi-Zadeh, A.; Fereydooni, M.; Ghaznavai, M.; and Baghshah, M. S. 2025. Bridging Reasoning to Learning: Unmasking Illusions using Complexity Out of Distribution Generalization. arXiv:2510.06274.

Shojaee, P.; Mirzadeh, I.; Alizadeh, K.; Horton, M.; Bengio, S.; and Farajtabar, M. 2025. The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity. arXiv:2506.06941.

Sinha, A.; Arun, A.; Goel, S.; Staab, S.; and Geiping, J. 2025. The Illusion of Diminishing Returns: Measuring Long Horizon Execution in LLMs. arXiv:2509.09677.

Sun, Y.; Hu, S.; Zhou, G.; Zheng, K.; Hajishirzi, H.; Dziri, N.; and Song, D. 2025. OMEGA: Can LLMs Reason Outside the Box in Math? Evaluating Exploratory, Compositional, and Transformative Generalization. arXiv:2506.18880.

Zhou, Y.; Liu, H.; Chen, Z.; Tian, Y.; and Chen, B. 2025. GSM-Infinite: How Do Your LLMs Behave over Infinitely Increasing Context Length and Reasoning Complexity? arXiv:2502.05252.