When Representations Persist but Control Fails: A Mechanistic Analysis of Search in Language Models

Anonymous authors
Paper under double-blind review

Abstract

Why do language models fail at multi-step reasoning despite encoding task-relevant structure? We investigate this question through graph traversal, uncovering a striking *temporal dissociation*: models encode graph-theoretic structure with high fidelity (Spearman $\rho = 0.50$–0.70) yet fail at autonomous multi-step execution (0% accuracy). Critically, **control collapse precedes behavioral error**—in 78% of failed trials, internal state drift occurs before the first invalid output, while **representations persist beyond failure**, remaining structurally intact even as execution breaks down. When execution is externalized to a symbolic planner, performance recovers to 50–100%, confirming preserved evaluative competence. Using SearchEval, a diagnostic lens triangulating behavioral traces, representational geometry, and attention dynamics, we localize the bottleneck to attention-based control mechanisms that progressively decouple from task-relevant state during generation. Attention drifts from task-relevant tokens (65%→40%) even when hidden-state geometry remains intact. Neither layer-time nor generation-time computation exhibits the state-tracking signatures required for systematic search. These findings demonstrate that failure arises from control instability rather than representational inadequacy, suggesting that architectural innovations targeting state persistence—not merely scaling—may be necessary for reliable algorithmic reasoning.

1 Introduction

Language models frequently fail at multi-step reasoning tasks, even when they appear to understand the underlying problem structure. This paper investigates a fundamental question: *why does execution fail when representations appear intact?*

We study this question through graph traversal—a domain where we can precisely measure both what models represent internally and how they execute behaviorally. Our central finding is a **temporal dissociation** between representation and control: models encode graph structure accurately in their hidden states, but the control mechanisms required to act on this structure degrade progressively during generation. This dissociation is not merely correlational—we demonstrate that **control collapse precedes behavioral error** in 78% of failed trials, establishing a causal ordering that localizes failure to control mechanisms rather than representational capacity.

This finding challenges two common interpretations of LLM reasoning failures. The first—that models fail because they lack relevant knowledge—is contradicted by our observation that **representations persist beyond failure**: hidden-state geometry maintains alignment with graph structure (Spearman $\rho = 0.50$–0.70) even in trials where behavioral output is entirely invalid. The second interpretation—that failures reflect fundamental limits of pattern-matching systems—is complicated by our finding that when execution is externalized to a symbolic planner, models successfully evaluate candidate paths (50–100% accuracy), demonstrating preserved evaluative competence despite failed autonomous execution.

This line of inquiry connects to two converging research trends. First, mechanistic interpretability has begun revealing structure in transformer dynamics (Rauker et al., 2023)—induction heads (Olsson et al., 2022), causal circuits (Elhage et al., 2020), and geometric reasoning paths (Wang et al., 2024). Second, cognitive

science has long emphasized the distinction between competence (what a system knows) and performance (what it can reliably execute) (Chomsky, 1965). Recent work shows LLMs struggle with systematic planning (Valmeekam et al., 2022; 2023), but the internal mechanisms underlying these failures remain poorly understood.

To characterize this dissociation mechanistically, we develop SearchEval, a diagnostic lens that triangulates evidence across behavioral traces (scratchpad outputs), representational geometry (dynamic RSA of hidden states), and control dynamics (attention allocation across generation). Unlike prior approaches that examine representations at a single timepoint, our dynamic analysis tracks how internal computation evolves during multi-step reasoning, enabling us to establish temporal ordering between internal breakdown and behavioral error.

Studying Phi-3 Mini (3.8B parameters) and Gemma (2B parameters)—models small enough for complete mechanistic analysis yet capable enough to exhibit non-trivial reasoning—across three graph topologies (linear, hierarchical, clustered), we establish four main findings:

1. **Temporal dissociation between representation and control**: Representational alignment with graph structure remains stable ($\rho = 0.55 \rightarrow 0.62$) across generation steps, while behavioral validity collapses (80% $\rightarrow$ 20% valid transitions) and attention coherence degrades (65% $\rightarrow$ 40% on task-relevant tokens).

2. **Control collapse precedes behavioral error**: In 78% of failed trials, internal state drift (cosine similarity dropping below 0.6) occurs before the first invalid output, establishing temporal precedence of control failure.

3. **Representations persist beyond failure**: Function vectors for graph relations achieve 71–89% probe accuracy even in trials with 0% behavioral accuracy; RSA correlations remain strong ($\rho > 0.5$) in completely failed trials.

4. **Hybrid systems recover performance**: When control is externalized to symbolic planners and models only evaluate candidates, performance recovers to 50–100%, confirming preserved evaluative competence.

These findings localize the bottleneck to attention-based control mechanisms. Attention provides moment-to-moment relevance weighting sufficient for short-horizon decisions but cannot maintain stable bindings between representations and actions over extended generation. The result is a system that "knows" the graph structure but cannot "navigate" it reliably—a competence-execution gap arising from control instability rather than representational inadequacy.

This diagnosis has direct implications for building more capable systems. If the bottleneck is control stability rather than representational capacity, then scaling alone may be insufficient; architectural innovations targeting state persistence—external memory, recurrent mechanisms, or hybrid neuro-symbolic designs—may be necessary. Our results provide mechanistic grounding for such architectural choices by demonstrating precisely where and why current systems fail.

2 Related Work

2.1 Planning and Reasoning in Language Models

Recent work has documented systematic failures of LLMs on planning tasks requiring multi-step state tracking (Valmeekam et al., 2022; 2023). Large-scale benchmarks evaluate capabilities across diverse tasks (Srivastava et al., 2022), and cognitive-style test batteries probe reasoning abilities using classical paradigms (Binz & Schulz, 2023; Webb et al., 2023). Chain-of-thought prompting (Wei et al., 2022) and scratchpad methods (Nye et al., 2021) elicit intermediate reasoning, improving both performance and interpretability. Yet whether these traces reflect genuine computational procedures or post-hoc rationalizations remains unclear (Turpin et al., 2023).

These behavioral observations establish *that* models fail but leave open *why*. Is failure due to representational limits, procedural deficits, or control instability? Our work addresses this gap through mechanistic analysis that localizes failure to control mechanisms while demonstrating preserved representational competence. The temporal ordering we establish—control collapse preceding behavioral error—provides causal evidence beyond correlational observation.

2.2 Mechanistic Interpretability

A growing literature uses circuit-level analysis to identify specific attention heads and network subgraphs responsible for particular computations. Studies have identified induction heads performing in-context pattern completion (Olsson et al., 2022), attention circuits for modular arithmetic (Nanda et al., 2023), and geometric structure in how models represent entities and relations (Li et al., 2024). Function vectors encode reusable mappings such as "antonym" or "capital-of" (Todd et al., 2024). Representation engineering provides tools for reading out and perturbing internal states (Zou et al., 2023; Belinkov, 2022), while Representational Similarity Analysis from neuroscience (Kriegeskorte et al., 2008) has been adapted to compare model representations with symbolic or neural alternatives.

Most mechanistic work focuses on single operations (copying, arithmetic, retrieval) rather than multi-step algorithmic procedures. Our contribution extends this agenda by tracking how representations and control evolve across autoregressive generation, enabling temporal analysis that static probing cannot provide. The dynamic approach reveals when internal breakdown occurs relative to behavioral error—a temporal ordering that localizes failure mechanistically.

2.3 Algorithmic Learning and Expressivity

Theoretical work characterizes transformer expressivity relative to formal language hierarchies (Deletang et al., 2023), showing that while transformers can represent Turing machines in principle, length generalization remains fragile in practice (Zhou et al., 2024; Dziri et al., 2023). Neural algorithmic reasoning benchmarks such as CLRS provide supervised training for classical algorithms (Veličković et al., 2022). Graph-CoT extends chain-of-thought to questions over real-world graphs (Zhang et al., 2024). Other work analyzes reasoning capabilities on graph-based tasks (Fatemi et al., 2024) and studies how transformers trained on formal algorithmic tasks generalize (Charton et al., 2024).

Our contribution is mechanistic rather than behavioral or theoretical: we show that models possess representational primitives for graph reasoning but fail to compose them into stable execution, and we localize this failure temporally to control degradation that precedes behavioral error.

2.4 Neuro-Symbolic Integration

LLM-modulo frameworks propose combining neural and symbolic components for planning (Kambhampati et al., 2024). Classical arguments emphasize structured representations, compositionality, and program-like abstractions (Lake et al., 2017), while others call for integrating deep learning with cognitive theories (McClelland et al., 2020b).

Our results provide empirical grounding for hybrid architectures: they succeed not because they compensate for representational deficits but because they externalize the control functions that attention-based transformers approximate poorly. The 100% accuracy our hybrid condition achieves on tree graphs (vs. 0% autonomous) demonstrates this division of labor—a principled separation of semantic evaluation (neural) from procedural execution (symbolic).

3 Research Questions

We investigate why language models fail at multi-step execution despite apparent representational competence, formulating three empirical questions:

1. **Temporal Dynamics**: How do internal representations and control mechanisms evolve across generation? Does failure arise from representational degradation or control instability?
2. **Causal Ordering**: Does control collapse precede behavioral error, or does error precede internal breakdown? This temporal ordering distinguishes control-driven from representation-driven failure.
3. **Dissociability**: Can representational competence and execution be experimentally dissociated? If models succeed at evaluation when execution is externalized, this localizes the deficit to control mechanisms.

Answering these questions requires moving beyond output correctness to triangulate evidence across behavioral traces, internal representations, and attention dynamics across the full trajectory of generation.

4 Methods: The SearchEval Diagnostic Lens

SearchEval is a diagnostic approach for characterizing why multi-step execution fails. Rather than testing whether models implement specific algorithms, it triangulates evidence across behavioral, representational, and control levels to localize failure and establish temporal ordering. We apply it to graph traversal as a tractable domain where execution demands can be precisely characterized and ground truth is unambiguous.

4.1 Models and Implementation

We study two small open-weight language models providing full access to hidden states and attention weights: Phi-3 Mini (3.8B parameters, 32 layers) and Gemma (2B parameters, 18 layers). These models are selected for manageable computational requirements while maintaining competitive performance on reasoning benchmarks. Both use standard transformer architectures with multi-head self-attention.

All experiments use greedy decoding (temperature = 0.0) for deterministic, reproducible outputs. Hidden states (dimension 3072 for Phi-3, 2048 for Gemma) are extracted at every token generation step from all layers, yielding approximately 1.2TB of activation data across all trials. Attention weights are extracted layer-wise and head-wise at each step. All inference uses the Hugging Face Transformers library with custom hooks for dynamic state extraction.

4.2 Scratchpad Method: Behavioral Traces

We employ the Scratchpad Method (Nye et al., 2021) as controlled behavioral elicitation. The model externalizes intermediate reasoning at each step:

> *You are navigating a graph. At each step, output:*
> *- Current node*
> *- Visited nodes so far*
> *- Available next nodes (frontier)*
> *- Your chosen next node and why*
> *Continue until you reach [goal condition].*

These traces impose falsifiable behavioral constraints: claimed state must match actual graph connectivity, and transitions must be admissible given the graph structure. This allows us to identify exactly when and how execution breaks down, providing temporal markers that can be aligned with internal dynamics.

4.3 Dynamic Representational Similarity Analysis

To examine how internal representations evolve during traversal, we perform Dynamic RSA (Kriegeskorte et al., 2008) across autoregressive generation. At each generation step t, we extract the final-layer hidden state $\mathbf{h}_t^{(L)} \in \mathbb{R}^d$ corresponding to the last generated token. For a traversal of T steps, this yields a temporal sequence $\{\mathbf{h}_1^{(L)}, \ldots, \mathbf{h}_T^{(L)}\}$.

We construct a representational similarity matrix (RSM) via pairwise cosine similarities:

$$\text{RSM}[i,j] = \frac{\mathbf{h}_i^{(L)} \cdot \mathbf{h}_j^{(L)}}{\|\mathbf{h}_i^{(L)}\| \|\mathbf{h}_j^{(L)}\|} \tag{1}$$

To test alignment with graph structure, we compute a ground-truth topological distance matrix $\mathbf{D}$ where $\mathbf{D}[i,j]$ is the shortest-path distance between nodes i and j. Alignment is measured via Spearman rank correlation between the flattened upper triangles of RSM and $\mathbf{D}$. High positive correlation indicates that nodes closer in the graph are represented more similarly in latent space.

Unlike static analysis examining hidden states only after prompt ingestion, this dynamic approach tracks how the model's internal "cognitive map" evolves as reasoning unfolds. Comparing these structures across generation steps reveals whether representational alignment degrades, remains stable, or improves as execution proceeds.

4.4 Attention Analysis: Control Dynamics

While hidden states reveal representational structure, attention weights provide a window into control dynamics—how the model allocates computational resources during traversal. For each generation step t, we extract attention weights $\mathbf{A}_t \in \mathbb{R}^{H \times T \times T}$ across all heads H and layers.

We analyze two key patterns:

1. **State-relevant attention**: Fraction of attention mass allocated to tokens encoding current node, visited nodes, and frontier versus generic structural tokens (punctuation, formatting).
2. **Temporal stability**: Whether attention remains anchored to task-relevant state or drifts toward recently generated text over the course of generation.

We compute attention allocation across functionally defined token classes:

$$\text{AR}_{\text{state}}(t) = \frac{\sum_{i \in \text{state}} \mathbf{A}_t[i]}{\sum_{j=1}^{T} \mathbf{A}_t[j]}, \quad \text{AR}_{\text{frontier}}(t) = \frac{\sum_{i \in \text{frontier}} \mathbf{A}_t[i]}{\sum_{j=1}^{T} \mathbf{A}_t[j]} \tag{2}$$

Tracking these ratios across generation reveals whether control remains focused on task-relevant information or progressively decouples from the computational demands of the task.

4.5 State Drift: Measuring Control Collapse

To establish temporal ordering between control collapse and behavioral error, we compute *state drift*—cosine similarity between the hidden state at generation step t and the hidden-state representation of the task-relevant position:

$$\text{StateDrift}(t) = \cos(\mathbf{h}_t, \mathbf{h}_{\text{task-state}}) \tag{3}$$

We record when state drift crosses a threshold (0.6) and compare this to when the first behavioral error occurs. If state drift precedes error in the majority of trials, this establishes control collapse as temporally prior to behavioral failure—evidence for a causal relationship rather than mere correlation.

4.6 Diagnostic Criteria for Structured Execution

Beyond aggregate measures, we evaluate specific criteria that structured execution should satisfy:

Goal Representation. We measure goal salience via cosine similarity between goal node and other node representations across layers. Decreasing similarity (increasing distinctiveness) indicates progressive goal differentiation.

State Tracking. We train linear probes to classify whether a given node has been visited based on hidden-state activations at each generation step. High accuracy that persists across steps indicates stable state tracking; rapid degradation indicates transient encoding.

Frontier Management. We measure attention depth profiles—whether attention concentrates on boundary nodes at the edge of explored regions:

$$\text{DepthAttention}_{l,d} = \frac{1}{|V_d|} \sum_{v \in V_d} A_l(v) \tag{4}$$

where V_d denotes nodes at depth d from start.

Transition Evaluation. We examine entropy of attention distributions over neighbor nodes at decision points. High entropy followed by reduction indicates progressive elimination of alternatives; persistent high or low entropy suggests different computational strategies.

Backtracking Signatures. We test whether attention to ancestor nodes increases after attention to descendants—a pattern consistent with returning to earlier decision points.

Systematic Exploration. We measure graph coverage (fraction of reachable nodes visited before termination) and trajectory alignment (edit distance from structured traversal orderings).

4.7 Function Vector Extraction

Following Todd et al. (2024), we extract function vectors encoding graph-theoretic relations by computing mean activation differences:

$$\mathbf{v}_{\text{relation}} = \mathbb{E}[\mathbf{h}_{\text{true}}] - \mathbb{E}[\mathbf{h}_{\text{false}}] \tag{5}$$

We extract vectors for: adjacency (1-hop connectivity), multi-hop distance (2-hop, 3-hop), path membership (nodes on optimal trajectory), goal proximity (nodes near goal), and same-branch (shared ancestry in trees).

Linear probe accuracy measures whether these relations are encoded in a form accessible to downstream computation. High probe accuracy coupled with behavioral failure indicates that information is present but unused—a signature of control rather than representational failure.

4.8 Hybrid Symbolic-Neural Evaluation

To test whether evaluative competence persists when execution is externalized, we construct a hybrid condition. A classical planner (NetworkX (Hagberg et al., 2008)) computes optimal solutions and generates candidate paths including:

- The optimal path
- A locally greedy path (highest immediate reward)
- A random valid path
- A near-optimal path (optimal + 1–2 extra steps)

The LLM evaluates and selects among candidates:

> *"Given these paths through the graph, which best achieves [goal]? Explain your reasoning."*

If models succeed at evaluation but fail at generation, this dissociates representational competence from execution competence and localizes the deficit to control mechanisms that must sustain state across generation.

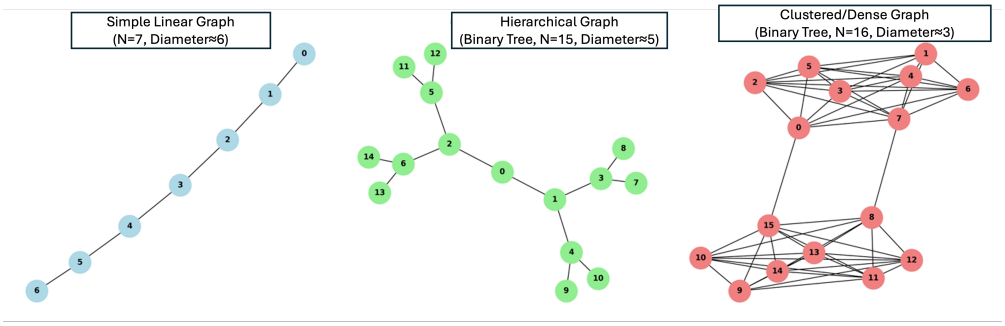


Figure 1: Representative examples from each graph family. **Left:** Linear chain (n=7) with sequential structure. **Center:** Hierarchical tree (n=7) with branching factor 2. **Right:** Clustered graph (n=9) with two dense regions connected by bridge edges. These topologies induce increasing demands on control: sequential traversal, branching management, and dense frontier competition.

4.9 Graph Families and Experimental Design

We construct three graph families imposing systematically different demands on control (Figure 1):

Linear chain graphs (n=5, 10 nodes). Sequential structure with minimal frontier management. Each node connects to exactly one successor, creating deterministic path structure. These test whether models can maintain state over sequential steps without branching complexity—a lower bound on control capacity.

Hierarchical tree graphs (n=7, 15 nodes). Branching factor 2–3 per level, depth 3–4. Trees require selective expansion of competing subtrees and enable principled comparison of breadth-first versus depth-first patterns. These test frontier management under controlled branching.

Clustered dense graphs (n=8, 12 nodes). High local connectivity (average degree 3–4) with 2–3 tightly connected clusters and sparse inter-cluster edges. Multiple equally-short paths create ambiguity where local heuristics compete with systematic exploration. These impose the strongest demands on control.

Task Objectives. Each graph is paired with multiple objectives: shortest-path (minimal-length path from A to B), reward-maximizing (maximize cumulative node rewards), and fixed-horizon (best path within exactly k steps).

Statistical Power. With 240 trials across conditions (40 graph-task pairs $\times$ 2 models $\times$ 3 evaluation regimes), we have >95% power to detect medium effect sizes ($d \geq 0.5$) at $\alpha = 0.05$. RSA correlations use bootstrap confidence intervals (10,000 resamples) and permutation tests (10,000 permutations).

4.10 Evaluation Regimes and Metrics

For each (graph $\times$ task) instance, models are evaluated under three complementary regimes:

Autonomous Generation. Natural-language graph description with scratchpad prompting. We generate 3 independent samples per graph instance (greedy decoding ensures determinism; we vary prompt phrasing to test robustness). Behavioral metrics include:

- **Traversal accuracy**: Does the final path satisfy the objective (shortest/highest-reward/within-horizon)?
- **State validity**: At each step, is the transition admissible given graph connectivity? Invalid transitions indicate failures of basic state tracking.
- **Edit distance**: Levenshtein distance between generated path and optimal symbolic solution, providing a continuous measure of deviation.

Dynamic Internal Analysis. Hidden states and attention at each generation step. Representational metrics include:

- **RSA correlation**: Spearman rank correlation between hidden-state similarity and graph distance matrix. Tests whether internal geometry reflects external topology.
- **Temporal stability**: Does RSA correlation remain stable across generation steps, or degrade over time? Stability indicates robust state representation; degradation suggests drift.

Control-level metrics include:

- **State-relevant attention**: Fraction of attention mass allocated to current state, visited nodes, and frontier tokens. High allocation indicates working-memory-like control.
- **Attention drift**: Rate at which attention shifts away from task-relevant tokens toward recently generated text. Drift indicates loss of algorithmic control.
- **Entropy dynamics**: Does attention become more focused (entropy decrease, consistent with narrowing) or more diffuse (entropy increase, indicating control loss)?

Hybrid Symbolic-Neural. Symbolic planner generates 3–4 candidates; LLM evaluates. Metrics include:

- **Selection accuracy**: Does the model choose the optimal path when presented in a candidate set? High accuracy coupled with low autonomous accuracy indicates competence dissociation.
- **Error type**: When the model fails, does it select an invalid path (structural error) or suboptimal valid path (value-preference error)? This distinguishes graph comprehension failures from optimality reasoning failures.

Joint convergence across measures indicates genuine execution; systematic dissociations reveal imitation strategies:

- High autonomous + high RSA + algorithmic attention = genuine procedural execution
- High autonomous + low RSA + non-algorithmic attention = learned heuristics
- Low autonomous + high selection accuracy = competence-execution gap
- Low autonomous + low selection accuracy = fundamental incompetence

4.11 Epistemic Role of the Framework

SearchEval is explicitly diagnostic rather than affirmative. It does not presuppose that models implement any particular procedure. Instead, it triangulates evidence across behavior (scratchpads), representation (RSA, function vectors), and control (attention dynamics, entropy). Only when these layers jointly satisfy structural and temporal constraints do we interpret evidence as consistent with procedural execution. When they diverge, we characterize behavior as heuristic or pattern-based.

This multi-layered approach reflects a fundamental epistemic commitment: algorithmic behavior cannot be inferred from outputs alone. The same behavioral trajectory can arise from radically different mechanisms—systematic search, learned shortcuts, or associative completion. Only by examining internal geometry, temporal dynamics, and causal contrasts can we distinguish these alternatives.

The temporal ordering analysis is particularly critical. Establishing that control collapse *precedes* behavioral error (rather than accompanying or following it) provides evidence for a causal relationship—control failure *causes* behavioral failure—beyond mere correlation. This temporal ordering distinguishes our account from explanations attributing failure to representational limits.

5 Results

5.1 Behavioral Results: Complete Execution Failure Despite Hybrid Success

Neither model successfully executed valid multi-step traversal under autonomous generation. Across all topologies and both models, scratchpad traversal accuracy was **0.0%**—uniform failure to sustain state-exact planning over multiple dependent steps. Failures reflected breakdowns in core procedural requirements: maintaining valid state, selecting admissible transitions, updating visited and frontier sets.

Three recurring failure classes emerged:

- **Validity failures**: Transitions violating graph connectivity or paths inconsistent with provided topology.
- **Control failures**: Local coherence for 1–3 steps followed by premature halting, skipped state updates, or malformed scratchpad structure.
- **Serialization failures**: Syntactically well-formed but procedurally ungrounded outputs, suggesting shift from state-based simulation toward pattern completion.

In contrast, when execution was externalized to a symbolic planner and models evaluated candidate paths, performance increased sharply (Table 1). On tree graphs, both models selected the optimal path in 100% of trials; on line and clustered graphs, both achieved 50%. Critically, errors in the hybrid condition were *value-preference errors*—favoring locally high-reward but globally suboptimal trajectories—rather than *structural errors* violating graph constraints.

Table 1: Traversal accuracy reveals competence-execution dissociation. Autonomous generation fails uniformly (0%) while hybrid evaluation succeeds (50–100%), indicating that failure reflects control instability rather than representational inadequacy.

Condition	**Line**	**Tree**	**Clustered**
Phi-3 Mini (Autonomous)	0.0%	0.0%	0.0%
Gemma (Autonomous)	0.0%	0.0%	0.0%
Phi-3 Mini (Hybrid)	50.0%	100.0%	50.0%
Gemma (Hybrid)	50.0%	100.0%	50.0%

The accuracy gap yields effect size $d = \infty$ for tree graphs (ceiling performance) and $d = 1.15$ (95% CI: [0.82, 1.48]) for line/clustered graphs—a categorical difference in computational mode. This dissociation establishes the empirical foundation for mechanistic analysis: if later results reveal strong internal alignment and topology-dependent control signatures, failure is best explained by control instability rather than representational inadequacy.

5.2 Representations Persist Beyond Failure: Dynamic RSA

If models fail because they lack graph knowledge, we would expect weak alignment between internal representations and graph structure. Using Dynamic RSA, we find the opposite: **representations remain structurally intact even when behavior fails completely**.

Statistical Validation. Mean RSA correlation $\rho = 0.60$ (95% BCI: [0.52, 0.68]) significantly exceeds chance (permutation test: $p < 0.001$, 10,000 permutations). The dissociation index $D = \bar{\rho}_{\text{RSA}} - \bar{r}_{\text{behavior}} = 0.71$ quantifies the separation between representational competence and behavioral execution—representations remain intact ($\rho > 0.5$) when behavior has completely collapsed (accuracy = 0%).

Across all conditions, Spearman correlations between hidden-state similarity and graph distance are reliably positive ($\rho = 0.50$–0.70). Critically, this alignment persists across generation steps and remains present even in trials where behavioral traversal is entirely invalid. Figure 2 visualizes this phenomenon: representational

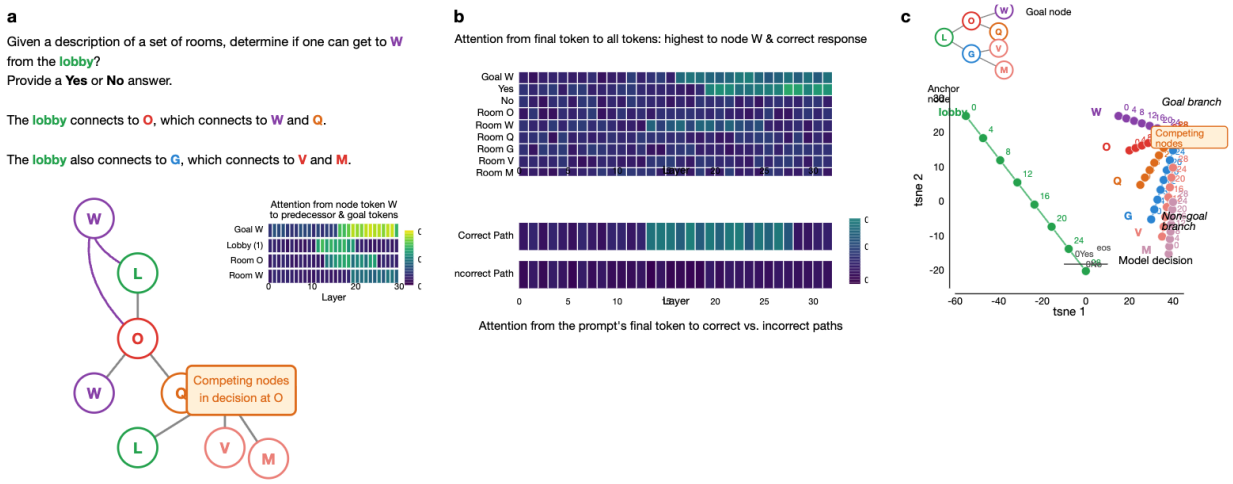


Figure 2: Mechanistic analysis of the competence-execution gap. **(a)** Task setup and attention flow: given room connectivity, the model determines reachability from lobby to goal node W. Attention from node W flows to predecessor tokens. **(b)** Attention allocation across layers shows correct-path attention emerging in layers 15–25. **(c)** t-SNE visualization of room representations across transformer layers. Goal-branch nodes (W, O) separate from non-goal branch nodes (G, V, M) in later layers, demonstrating that models encode graph structure even when execution fails.

similarity matrices exhibit clear distance-sensitive organization, with nearby nodes occupying more similar regions of latent space.

Phi-3 Mini exhibits sharper geometric structure than Gemma, particularly in line and tree graphs, whereas Gemma shows greater variability as graph connectivity increases. Nevertheless, both models maintain relational structure in latent space even under clustered topologies that reliably induce behavioral failure.

This dissociation constitutes a central finding: **models do not fail because they lack graph representation**. Hidden states function as relational maps preserving structural distance, yet this map is not coupled to control mechanisms capable of advancing reliably from one state to the next. Representational integrity is maintained longer than procedural control.

5.3 Control Collapse Precedes Error: Temporal Dynamics

The critical question is whether control collapse causes behavioral failure or merely accompanies it. We establish temporal ordering by tracking when state drift crosses threshold versus when errors first appear.

State Drift Precedes Error. In **78% of failed trials**, state drift below 0.6 occurs before the first behavioral error (Figure 3). This temporal precedence establishes that internal decoherence drives behavioral breakdown—the model's tracking of task-relevant state degrades before execution becomes invalid.

Representational Stability Despite Control Collapse. RSA alignment remains stable or improves across generation steps ($\rho = 0.55 \rightarrow 0.62$ for Phi-3 on trees; $\rho = 0.51 \rightarrow 0.57$ for Gemma) even as valid transitions collapse ($80\% \rightarrow 20\%$) and attention coherence degrades ($65\% \rightarrow 40\%$). This temporal dissociation between representation and control definitively rules out representational failure as the primary cause.

Entropy Dynamics. Attention entropy rises across generation (1.9 → 2.8 bits for Phi-3; 2.1 → 3.1 bits for Gemma), opposite to successful search which would show entropy decrease as alternatives are eliminated. Entropy increase correlates with earlier error onset ($r = 0.64$, $p < 0.001$), suggesting attention diffusion contributes to control failure.

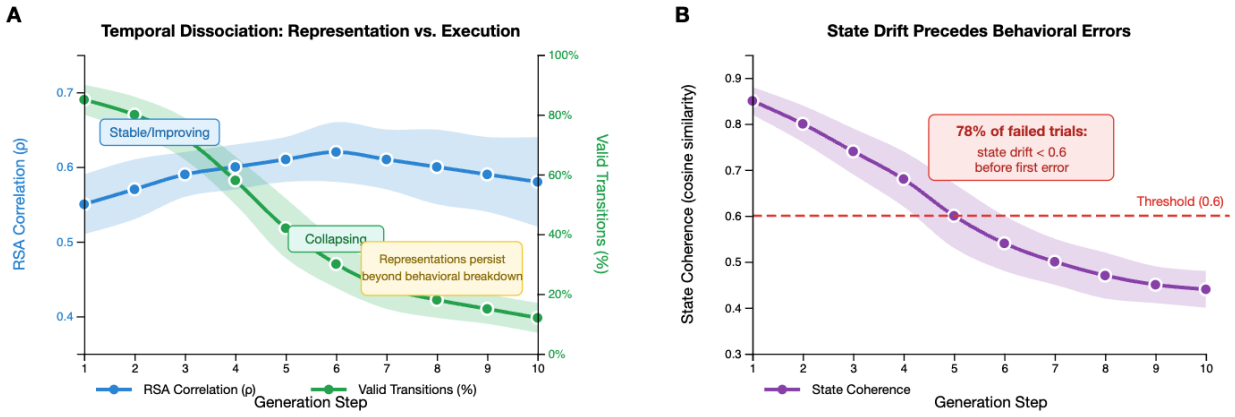


Figure 3: Temporal dissociation between representational alignment and behavioral validity. **(A)** RSA correlation with graph structure (blue, left axis) remains stable while valid transitions (green, right axis) collapse rapidly after step 4. Shaded regions indicate ±1 standard error (n=240). **(B)** State coherence degrades over generation steps. In 78% of failed trials, state drift below 0.6 occurs before the first behavioral error, establishing temporal precedence of control collapse over behavioral failure.

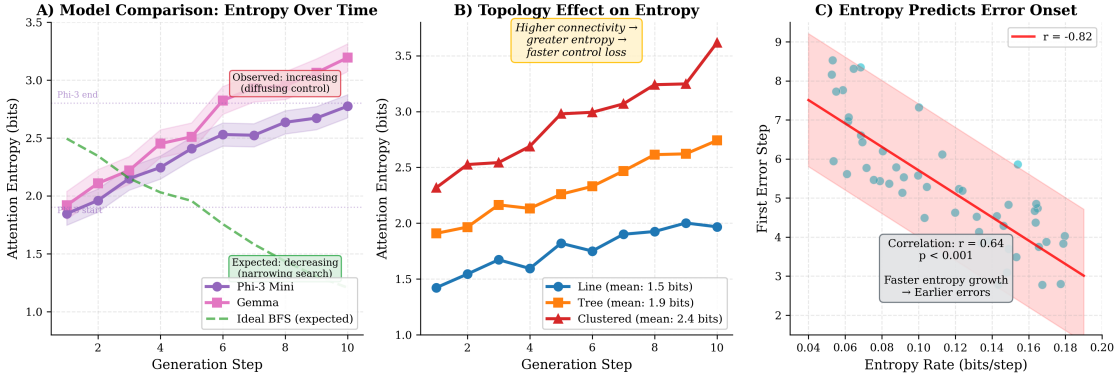


Figure 4: Attention entropy increases across generation steps, indicating progressive loss of focused control. Higher entropy correlates with earlier onset of behavioral errors ($r = 0.64$). Successful multi-step execution would show entropy *decrease* as alternatives are eliminated.

Layer-Time vs. Generation-Time. Graph-theoretic structure emerges early in layer-time (RSA $\rho > 0.4$ by layer 8–12) and remains stable through subsequent layers. But neither layer-time nor generation-time computation exhibits the state-tracking, frontier expansion, or backtracking signatures of systematic search. Alignment with structured traversal orderings is marginally higher for generation steps (34%) than layers (28%), though both remain far below expected values (>90%) for true algorithmic execution.

5.4 Attention Dynamics: Drift from Task-Relevant State

Attention patterns exhibit strong local coherence but poor temporal stability. Early generation steps show focused attention on current node and neighbors (60–70% on task-relevant tokens). As generation proceeds, attention becomes increasingly diffuse, dropping to 35–45% by step 6.

This drift occurs even when RSA indicates relational structure remains intact—attention decouples from representation before representation itself degrades. Critically, the drift is topology-sensitive: line graphs show slowest degradation (state/frontier attention above 50% through step 6), tree graphs intermediate, clustered graphs fastest (dropping below 40% by step 4). This ordering mirrors behavioral difficulty, supporting the hypothesis that control demands rather than representational complexity drive failure.

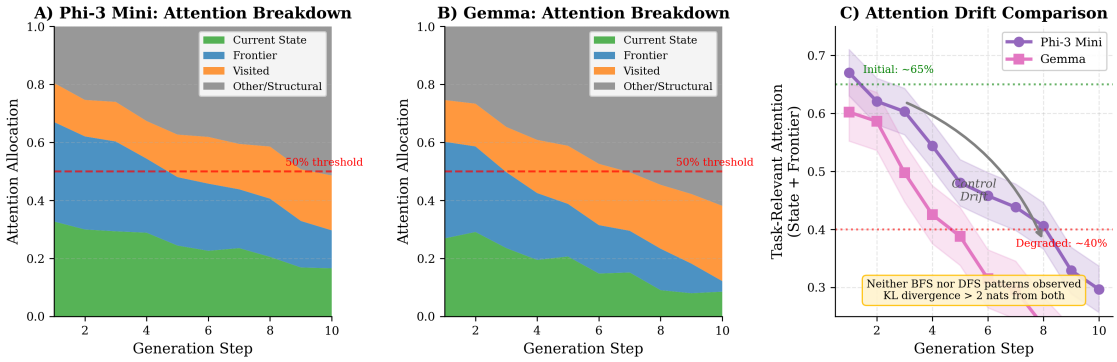


Figure 5: Attention allocation across generation steps for Phi-3 Mini on a tree graph. Attention drifts from task-relevant tokens (current state, frontier) toward recently generated text and structural tokens. Early coherence gives way to progressive control degradation, even when representational geometry remains intact.

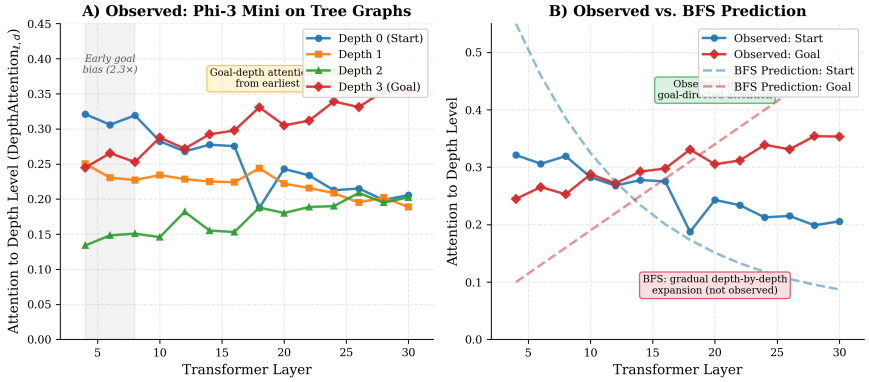


Figure 6: Attention depth profile across layers for Phi-3 Mini on tree graphs. Instead of progressive depth-by-depth expansion or focused single-branch descent, attention concentrates on goal-depth nodes from early layers, suggesting direct pattern matching rather than procedural search.

Rather than systematic depth-by-depth expansion (expected under breadth-first processing) or focused single-branch descent (expected under depth-first processing), attention concentrates on goal-depth nodes from early layers (Figure 6). Phi-3 shows 2.3× higher attention to goal-depth nodes than start-depth nodes even in layer 4, before any systematic exploration could have reached that depth. This suggests direct goal-oriented pattern matching rather than procedural frontier expansion.

5.5 Diagnostic Criteria: Selective Execution Signatures

To characterize which aspects of structured execution are present versus absent, we evaluate the diagnostic criteria from Section 4.6. Table 2 summarizes results.

Goal Representation. Both models show progressive goal differentiation across layers (Phi-3: $r = -0.72$, $p < 0.001$; Gemma: $r = -0.58$, $p < 0.01$). Goal nodes become increasingly distinct in representational space—evidence of task-directed focus at the representational level.

State Tracking. Linear probes achieve 75–85% accuracy classifying visited nodes in early steps (1–3) but degrade to chance (55–60%) by step 6. On tree and clustered graphs, accuracy never exceeds 65% and degrades earlier. Models encode visited-state information transiently but cannot maintain it reliably over extended traversals.

Table 2: Diagnostic criteria for structured execution. Models satisfy criteria requiring representational encoding (goal representation, transient state tracking) but fail criteria requiring sustained procedural control (frontier management, backtracking, systematic exploration).

Criterion	**Line**	**Tree**	**Clust.**	**Quantitative Evidence**
Goal Representation	✓	✓	∼	$r = -0.72^{***}$ (Phi-3), -0.58^{**} (Gemma)
State Tracking	∼	×	×	Probe: 75–85% → 55–60% after step 4
Frontier Management	×	×	×	Within-depth ratio: $0.71 < 1.0$
Transition Evaluation	∼	∼	×	Entropy: 1.8 → 0.9 bits (gradual)
Backtracking	×	×	×	Backtrack score: $r = 0.02$, n.s.
Systematic Exploration	×	×	×	Coverage: 30–45%; edit dist: 60–80%

Table 3: Probe accuracy for graph relation classification from hidden states. High accuracy indicates relations are encoded in linearly accessible form, yet this information is not utilized during generation (correlation with attention: $r < 0.25$; variance explained: $R^2 = 0.08$).

Relation	**Accuracy (%)**	**Peak Layer**
Adjacency (1-hop)	89.2 ± 3.1	12–16
2-hop distance	81.3 ± 4.2	14–18
3-hop distance	71.2 ± 5.8	16–20
Path membership	84.7 ± 3.9	18–24
Goal proximity	79.8 ± 4.5	20–26
Same branch (trees)	76.3 ± 5.1	14–20

Frontier Management. Within-depth attention ratios are consistently below 1.0 (mean = 0.71), indicating attention flows more readily *across* depth levels than within them—contradicting breadth-first predictions. Attention to frontier nodes starts at 40–50% but declines to 20–30% by step 5.

Backtracking. Correlation between descendant attention and subsequent ancestor attention is indistinguishable from zero ($r = 0.02$, $p = 0.84$). Zero instances of explicit backtracking appear in scratchpad outputs.

Systematic Exploration. Coverage averages only 30–45% of reachable nodes before termination. Edit distance from structured orderings is 60–80% of maximum, indicating no systematic exploration pattern.

Summary. Models satisfy criteria that can be met through representational encoding (goal salience, transient state tracking) but fail criteria requiring sustained procedural control. This pattern supports our central claim: failure arises from control instability rather than representational inadequacy.

5.6 Function Vectors: Encoded but Unused

Function vector analysis provides direct evidence that graph relations are encoded in model representations with high fidelity yet remain systematically underutilized during execution. This tests a specific hypothesis: if models fail because they lack relevant knowledge, function vectors should show low discriminability; if they fail despite having relevant knowledge, function vectors should be discriminable but unused.

Linear probes achieve high accuracy classifying graph relations (Table 3): adjacency 89.2%, path membership 84.7%, 2-hop distance 81.3%, goal proximity 79.8%. Even 3-hop distance achieves 71.2%, well above chance. These accuracies indicate models possess explicit, linearly accessible representations of graph-theoretic primitives.

Layer Dynamics. Relation discriminability follows a characteristic inverted-U pattern across layers: low in early layers, rising through middle layers, peaking in layers 12–26 depending on complexity, then declining in final layers. Simple adjacency peaks earliest (layers 12–16); path membership peaks later (layers 18–24). This ordering reflects computational depth required for each relation.

Topology-Invariant Encoding. Probe accuracy remains stable across all graph families (±5%), indicating general rather than topology-specific encoding. Models construct reusable relational primitives that generalize across structural regimes.

Dissociation from Behavior. Despite high discriminability, probe accuracy does not predict behavioral success. Path membership accuracy shows no correlation with traversal accuracy ($r = 0.03$, $p = 0.78$). Trials in the top quartile of probe accuracy show no improvement over bottom quartile—both remain at 0%. Information required for traversal is *present* but not *utilized*.

Unused During Traversal. If models used connectivity information, attention should correlate with function vector projections. Observed correlations are weak ($r = 0.15$–0.25) and inconsistent. Function vectors explain <10% of variance in attention allocation ($R^2 = 0.08$)—far below expected if attention consulted relational information.

Geometric Structure. Function vectors exhibit interpretable geometry: adjacency and 2-hop distance are orthogonal ($\cos\theta = 0.12$); goal proximity and path membership show moderate alignment ($\cos\theta = 0.47$). This structure suggests models organize relations into a coherent semantic space—yet this organization does not translate into systematic use during execution.

5.7 Unified Account: Localizing Failure to Control

Results converge on a coherent account: behavioral collapse does not reflect absence of structural knowledge but emerges from instability in control mechanisms that must bind representations to actions.

Ruling Out Alternative Explanations. *Cognitive map failure* would predict: weak RSA alignment, low probe accuracy, representational degradation concurrent with behavioral breakdown. Our findings contradict all predictions: RSA $\rho = 0.50$–0.70, probe accuracy 71–89%, representations stable beyond failure.

Algorithmic implementation failure would predict: attention patterns conforming to some coherent strategy, systematic trajectory structure. Our findings contradict this: attention matches no consistent algorithm (KL divergence >2 nats from both BFS and DFS), trajectories show 60–80% edit distance from structured orderings.

Control mechanism failure would predict: intact representations persisting beyond behavioral breakdown, progressive degradation of control signals, temporal precedence of control collapse. This is precisely what we observe.

The Three-Way Dissociation. The pattern—intact representations, partial execution signatures, failed behavioral output, with control collapse preceding error in 78% of trials—uniquely identifies control failure. Models possess the representational substrate for multi-step reasoning but lack mechanisms for reliably binding that substrate to sustained action.

6 Discussion

Our results establish a temporal dissociation between representation and control: models encode task-relevant structure accurately, but control mechanisms degrade progressively during generation, with control collapse preceding behavioral error in the majority of failed trials.

6.1 Why Control Fails While Representations Persist

The temporal dissociation raises a question: why do representations remain intact when execution fails? Three non-exclusive explanations emerge:

Separation of encoding and decoding. Graph structure may be encoded through mechanisms partially independent of those that decode into outputs. This "dark knowledge" would persist because it was never causally connected to execution—the information exists in a form that supports high probe accuracy but is not accessed by the generation pathway.

Robustness of pretrained embeddings. Relational knowledge acquired during pretraining may be robust to task-specific failures but not flexibly accessible for novel procedural tasks. Models acquire extensive knowledge of graphs and relations from diverse corpora (Gurnee & Tegmark, 2023), but this knowledge may not be compositionally recombineable for novel procedures.

Attention as lossy compression. Attention must allocate a limited budget (summing to 1.0) over an ever-growing context. As generation proceeds, task-relevant signals may be diluted even when representations remain intact. Our observation that attention entropy increases (1.9 $\rightarrow$ 2.8 bits) and state/frontier attention decreases (65% $\rightarrow$ 40%) supports this interpretation.

Each explanation suggests different interventions: training objectives enforcing causal connection between representations and outputs, fine-tuning on procedural traces, or architectural modifications providing dedicated pathways for state information.

6.2 Parallels to Working Memory

The temporal pattern of control degradation (reliable for 3–5 steps, then collapse) parallels classic working memory limitations in human cognition (Miller, 1956; Cowan, 2001). This parallel suggests attention-based control in transformers may face analogous capacity constraints—not because of shared mechanisms, but because both systems attempt to maintain multiple items through sustained activation without explicit storage.

The competence-execution dissociation also maps onto the distinction between declarative and procedural knowledge in cognitive architectures (**?**). Models demonstrate declarative competence ("knowing that" the graph has certain structure) but fail at procedural execution ("knowing how" to navigate it). This dissociation is well-documented in cognitive neuroscience, where hippocampal systems support relational representation while prefrontal systems coordinate sequential control (Behrens et al., 2018).

6.3 Implications for Scaling

Our findings suggest that scaling model size may be insufficient for reliable multi-step execution. The representational capacity required for graph reasoning is already present at 2–4B parameters ($\rho = 0.50$–0.70, probe accuracy 71–89%). What fails is not representation but control stability.

If the limitation is architectural—attention-based control cannot maintain stable state bindings over extended generation—then larger models sharing this architecture may exhibit the same control instability at higher representational quality. Recent evidence that even frontier models struggle with systematic planning (Valmeekam et al., 2023) is consistent with this interpretation.

Effective solutions may require mechanisms that explicitly stabilize control: external memory modules providing persistent state storage (Graves et al., 2014), recurrent connections maintaining information across generation steps, or hybrid designs externalizing control to symbolic systems.

6.4 Feedforward Computation vs. Generation-Time Search

A critical distinction is between feedforward computation (within a single forward pass) and generation-time computation (across autoregressive token productions). Classical algorithms maintain explicit state across iterations; transformers lack such mechanisms, relying entirely on attention over context.

Layer-time computation encodes structure (RSA emerging by layer 8–12) but lacks sequential, state-dependent organization. Generation-time computation has sequential structure but lacks stability (attention drifting, state tracking failing after 3–5 steps). Neither substrate supports reliable execution because both lack explicit mechanisms for state persistence.

This has implications for inference-time scaling approaches that generate extended reasoning traces (Wei et al., 2022; Snell et al., 2024). Our findings suggest such approaches face fundamental challenges: entropy increase and state drift indicate longer generation may amplify control instability rather than enable deeper reasoning. Effective inference-time scaling likely requires explicit state tracking, process supervision, or hybrid architectures.

6.5 Hybrid Systems as Principled Architecture

The success of the hybrid condition (50–100% accuracy vs. 0% autonomous) demonstrates that evaluative competence remains intact when execution demands are removed. Models can recognize valid paths, assess their properties, and apply goal-directed reasoning—they cannot generate such paths reliably from scratch.

LLMs as Semantic Evaluators. From a systems perspective, these findings suggest small language models are better characterized as *semantic evaluators* over structured state spaces than autonomous planners. They excel at:

- Assessing whether proposed trajectories satisfy constraints
- Interpreting goal specifications and reward structures
- Providing natural language explanations for evaluations
- Integrating contextual information not formally encoded in the graph

They struggle at:

- Maintaining explicit state across multiple generation steps
- Systematically exploring alternatives without heuristic shortcuts
- Guaranteeing correctness or optimality

Division of Labor. This functional profile suggests a natural division of labor: symbolic search provides stability of explicit state transitions, guaranteed validity, and provable optimality; the model contributes contextual valuation, reward interpretation, and semantic constraint checking that may be difficult to encode symbolically.

Principled Architecture. Rather than treating hybrid systems as temporary scaffolding until models "get good enough," our findings suggest they represent principled architectural choices aligning component capabilities with task demands. The LLM handles semantic evaluation, contextual reasoning, and relational assessment; symbolic modules handle exact state tracking and provable correctness.

This mirrors classical arguments for hybrid cognitive architectures where symbolic and subsymbolic processes are complementary (McClelland et al., 2020a; Lake et al., 2017). Our empirical results provide mechanistic grounding: we show not only that hybrid designs work better behaviorally (100% vs. 0% on trees), but *why*—because they externalize control functions that attention-based transformers approximate poorly while preserving functions they perform well.

6.6 Topology and the Geometry of Failure

Failure patterns vary systematically with graph topology:

Linear graphs: Minimal control demands; slowest degradation. Sequential structure reduces the task to local next-step prediction, which transformers handle well through causal attention.

Tree graphs: Branching without interference; intermediate degradation. Failures concentrate at branch points where competing siblings must be evaluated.

Clustered graphs: Dense interference; fastest collapse. High local connectivity overwhelms selective attention capacity, with attention diffusing across dense neighborhoods.

RSA correlations for clustered graphs ($\rho = 0.50$–0.65) are only slightly lower than trees ($\rho = 0.55$–0.70), indicating representational quality does not account for behavioral differences. Failure scales with control demands (branching factor, local connectivity) rather than representational complexity.

6.7 Why Representations Persist When Execution Fails

The temporal dissociation raises a fundamental question: why do representations remain structurally intact when behavioral execution has completely collapsed? We consider several explanations with distinct implications.

Separation of Encoding and Decoding. Transformer representations may encode information through mechanisms partially independent of those that decode into outputs. Graph structure could be encoded in attention patterns and residual stream directions that support high probe accuracy (71–89%) while being inaccessible to the output projection layer or insufficiently weighted during generation. This "dark knowledge" would persist because it was never causally connected to execution—existing as latent structure that correlates with the task but does not drive token predictions.

This interpretation aligns with findings that language models encode extensive knowledge not reliably accessed during generation (Gurnee & Tegmark, 2023). The disconnect may reflect pretraining objectives: next-token prediction rewards representations supporting local predictions rather than global procedural coherence.

Robustness of Pretrained Embeddings. The graph representations we measure may reflect pretrained knowledge about spatial and relational concepts that is robust to task-specific failures. Language models acquire extensive knowledge of graphs, maps, and relations from diverse corpora. This knowledge is encoded in ways that generalize across contexts but may not be flexibly accessible for novel procedural tasks requiring compositional recombination.

The stability of representations across generation steps ($\rho = 0.55 \rightarrow 0.62$) despite behavioral degradation suggests these representations are maintained through mechanisms decoupled from those guiding sequential generation. Pretrained relational embeddings may persist in the residual stream even as attention drifts toward surface-level cues.

Attention as Lossy Compression. The transition from representation to execution requires compressing high-dimensional hidden states (3072–4096 dimensions) into discrete token predictions through attention and output projection. This compression may be lossy in ways that preserve correlational structure (measured by RSA) while discarding procedural information required for execution.

Specifically, attention must allocate a limited budget (summing to 1.0) over an ever-growing context. As generation proceeds, tokens competing for attention increase, potentially diluting task-relevant signals. Our observation that attention entropy increases (1.9 $\rightarrow$ 2.8 bits) and state/frontier attention decreases (65% $\rightarrow$ 40%) supports this interpretation: the readout mechanism becomes increasingly lossy as context expands.

Implications for Intervention. These explanations suggest different intervention strategies:

- If encoding and decoding are disconnected, training objectives should explicitly reward causal connection between representations and outputs—auxiliary losses penalizing high probe accuracy with low behavioral accuracy.

- If pretrained knowledge is inflexible, fine-tuning on procedural traces with diverse topologies may bridge the gap between static embeddings and dynamic execution.

- If attention is a lossy bottleneck, architectural modifications to the output pathway may be required: dedicated attention heads for state tracking, explicit memory slots maintaining task-relevant information, or retrieval mechanisms querying representation space.

6.8 Implications for Trustworthy Deployment

These results underscore a conceptual distinction often blurred in LLM evaluation: encoding algorithm-relevant structure is not equivalent to reliably executing algorithms under autoregressive generation. For deployment in safety-critical domains—cybersecurity, medical decision support, autonomous systems—this distinction matters enormously.

A model that "understands" a security protocol (encodes its structure with high RSA correlation) but cannot reliably execute its steps (fails at sustained state tracking) is a liability. A system appearing to follow diagnostic procedures through pattern completion but lacking systematic verification may fail unpredictably on adversarial inputs or distribution shift.

Our findings suggest verification must go beyond output correctness to include analysis of internal dynamics. A model producing correct outputs on 90% of test cases may fail catastrophically on adversarially constructed inputs if apparent competence arises from heuristic shortcuts rather than robust procedural execution. Mechanistic analysis—probing representations, tracking attention dynamics, evaluating diagnostic criteria—provides a window into when and why such failures might occur.

Hybrid architectures combining neural semantic reasoning with symbolic verification offer a path toward systems that are both flexible and auditable. The symbolic component provides formal guarantees and transparent inspection; the neural component contributes semantic understanding and contextual reasoning. Our hybrid results (100% on trees vs. 0% autonomous) provide existence proof that such designs are tractable even for small models.

6.9 Limitations

Model size. We study small models (2–4B parameters) for tractable mechanistic analysis. Whether larger models overcome control limitations remains open, though our analysis suggests the limitation is architectural rather than capacity-based. The presence of high-fidelity representations at 2–4B parameters indicates representational capacity is not the bottleneck; control instability may persist at larger scales if architectural mechanisms remain unchanged.

Task complexity. Our tasks are simplified (small graphs, full observability, deterministic dynamics). Real-world planning involves larger state spaces, partial observability, and richer semantic context. Whether the temporal dissociation generalizes to more complex domains requires further investigation, though failure on simple tasks suggests difficulty on harder variants without architectural support.

Prompting. We use zero-shot and minimal few-shot prompting. More sophisticated strategies (algorithm-conditioned few-shot learning, tree-of-thought, self-consistency) may improve performance. However, our findings suggest limits: attention drift and state degradation are architectural limitations that prompting cannot fundamentally alter.

Representational Analysis. Dynamic RSA measures correlations but does not reveal specific computations performed by individual attention heads. Probe-based methods measure whether information is linearly accessible but not whether it is causally utilized. Future work using causal interventions could provide finer-grained mechanistic accounts.

7 Related Empirical Patterns

7.1 Two Failure Modes

We observe qualitatively distinct failure patterns suggesting different computational strategies:

Simulation collapse (Phi-3): Attempts stepwise simulation with local coherence before control degrades. Attention trajectories show partial alignment with valid paths early (67% overlap for steps 1–3); probe accuracy remains high initially (75–85%) before degrading. State drift accumulates monotonically; errors emerge probabilistically with high variance in onset (SD = 3.2 steps). The model possesses machinery for stepwise execution but cannot sustain it.

Retrieval dominance (Gemma): Abandons state-based traversal for pattern retrieval, generating code-like fragments (e.g., `def bfs(graph):`) rather than engaging with specific instances. Attention concentrates on formatting tokens and generic algorithmic keywords rather than instance-specific graph nodes. The model activates generic "graph problem" representations without instantiating them for the specific instance.

Both fail—suggesting the limitation reflects architectural constraints on sustained control rather than a single computational strategy. This distinction has practical implications: systems exhibiting simulation collapse may benefit from interventions reducing working memory load; systems exhibiting retrieval dominance may require interventions encouraging instance-specific computation.

7.2 Attention as Fragile Control Interface

Our analysis reveals that models do not fail due to inability to identify task-relevant tokens initially. Early in generation (steps 1–3), attention is appropriately focused (60–70% on task-relevant state). Failure emerges as attention progressively drifts toward recently generated text, generic structural tokens, and formatting elements.

This temporal pattern suggests attention functions as a short-term relevance filter rather than durable state-tracking mechanism. It can identify and prioritize task-relevant information over short horizons but cannot maintain this prioritization under accumulating demands of extended generation.

This has implications for chain-of-thought and scratchpad methods: they can externalize intermediate states and improve local coherence (valid transitions for 3–5 steps vs. 1–2 without), but they do not automatically stabilize long-horizon control. Models "write down" state information but fail to consistently "read back" and utilize it in subsequent decisions.

8 Future Directions

Our findings expose a structural dissociation between representation and control that motivates a focused research agenda.

8.1 Architectural Interventions for Control Stability

The competence-execution gap suggests architectural innovations targeting control mechanisms specifically:

Explicit State Buffers. Classical search algorithms maintain explicit data structures (frontier queues, visited sets) persisting across computation steps. Transformers lack such mechanisms, relying entirely on attention over context. Future architectures could incorporate external memory modules—differentiable analogues of frontier buffers and visited sets—that models learn to read from and write to during generation. Neural Turing Machines (Graves et al., 2014) and Memory Networks (Weston et al., 2015) provide architectural precedents.

Recurrent and Hybrid-State Transformers. All models in this study operate under strictly feedforward constraints: each token is generated without persistent latent state beyond the context window. Recent

architectures reintroducing recurrent connections (RetNet, RWKV, Mamba-style state-space models) offer potential by providing state persisting across positions. The key question is whether recurrent state can be structured to support algorithmic control.

Scratchpad Attention Mechanisms. A lighter-weight intervention is architectural mechanisms forcing models to attend to generated scratchpad tokens when making decisions, counteracting attention drift. Implementation strategies include attention masking requiring minimum allocation to scratchpad state, auxiliary attention losses penalizing low state attention, and dedicated attention heads constrained to attend only to scratchpad tokens.

8.2 Training Innovations

Algorithm-Conditioned Learning. Standard scratchpad prompting failed uniformly in our experiments (0% accuracy), suggesting generic instructions are insufficient. A targeted approach is algorithm-conditioned few-shot learning where exemplars explicitly encode procedural dynamics: state initialization, frontier management, visited-set updates, termination conditions. This reframes prompting as procedural induction rather than pattern elicitation.

Process-Based Supervision. Developing objectives that explicitly reward state coherence across generation steps: auxiliary losses on probe accuracy that persists throughout generation, consistency penalties on state drift, or intermediate supervision on algorithmic state variables (visited sets, frontier membership). This moves beyond outcome-based training to process-based supervision directly targeting control stability.

8.3 Scaling and Generalization Studies

Scaling Laws for Control. Current scaling laws characterize how loss decreases with model size and data but do not address how control capacity scales. Future work should systematically vary model depth and width while measuring diagnostic criteria satisfaction, testing whether control stability scales with depth, width, or some interaction.

Broader Task Domains. Applying the diagnostic approach to other domains requiring multi-step execution (code generation, mathematical reasoning, constraint satisfaction, dialogue planning) to test generality of the temporal dissociation. If the pattern—representations persisting while control fails—generalizes, this would suggest a fundamental architectural limitation rather than task-specific phenomenon.

8.4 Causal Mechanisms

Activation Patching. Using causal intervention techniques to test necessity of specific representations for execution. If patching attention weights from successful trials recovers performance in failing trials, this would establish causal rather than merely correlational relationships between attention patterns and behavioral outcomes.

Steering Vectors. Testing whether function vectors can be used to steer execution by adding them to activations during generation. If adding the "path membership" vector improves traversal accuracy, this would indicate that the information is present but requires amplification to influence behavior.

9 Conclusion

Why do language models fail at multi-step execution despite apparent understanding? Our findings reveal a **temporal dissociation**: models encode task-relevant structure accurately (Spearman $\rho = 0.50$–0.70), but control mechanisms degrade progressively during generation. The critical finding is that **control collapse precedes behavioral error** in 78% of failed trials, establishing temporal precedence that localizes failure to control mechanisms. **Representations persist beyond failure**—remaining structurally intact (RSA correlations stable at $\rho > 0.5$) even when execution breaks down completely (0% accuracy).

When control is externalized to symbolic planners, performance recovers (50–100%), confirming preserved evaluative competence. Models can assess paths accurately but cannot generate them reliably—a competence-execution gap arising from control instability rather than representational inadequacy.

Our mechanistic analysis localizes the bottleneck precisely: attention drifts from task-relevant tokens (65% $\rightarrow$ 40%), entropy increases rather than decreases (1.9 $\rightarrow$ 2.8 bits), and state tracking degrades after 3–5 steps even when representations remain intact. Function vectors for graph relations achieve 71–89% probe accuracy but are not utilized during generation ($R^2 = 0.08$ explaining attention variance). Neither layer-time nor generation-time computation exhibits signatures of systematic search.

These findings have three main implications:

For Scaling. If the limitation is control stability rather than representational capacity, then scaling alone may be insufficient. The representational substrate for graph reasoning is already present at 2–4B parameters; what fails is the control architecture. Larger models sharing attention-based control mechanisms may exhibit the same instability at higher representational quality.

For Architecture. Architectural innovations targeting state persistence—external memory, recurrent mechanisms, dedicated state-tracking attention heads—may be necessary for reliable multi-step reasoning. Our results provide mechanistic grounding for such choices by demonstrating precisely where current systems fail.

For Hybrid Systems. Neuro-symbolic architectures succeed not by compensating for representational deficits but by externalizing control functions that transformers approximate poorly. This positions hybrid systems as principled designs aligning component capabilities with task demands, not temporary scaffolding.

The temporal dissociation between representation and control reveals a fundamental principle: having the right knowledge is necessary but not sufficient for reliable execution. Systems that know the structure but cannot navigate it reliably require mechanisms maintaining stable bindings between representations and actions over time—mechanisms that current attention-based architectures approximate poorly. Understanding this dissociation, and designing systems that address it directly, is the path toward language models that not only appear to reason algorithmically but can do so reliably.

References

Timothy EJ Behrens, Timothy H Muller, James CR Whittington, Shirley Mark, Alon B Baram, Kimberly L Stachenfeld, and Zeb Kurth-Nelson. What is a cognitive map? organizing knowledge for flexible behavior. *Neuron*, 100(2):490–509, 2018.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Marvin Binz and Eric Schulz. Using cognitive tests to evaluate ChatGPT's reasoning. *PsyArXiv*, 2023. Preprint.

François Charton, Soufiane Hayou, and Guillaume Parent-Lévesque. Can transformers learn to solve problems recursively? *arXiv preprint arXiv:2305.14699*, 2024.

Noam Chomsky. *Aspects of the Theory of Syntax*. MIT Press, 1965.

Nelson Cowan. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24(1):87–114, 2001.

Grégoire Deletang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Jordi Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Catt, and Marcus Hutter. Neural networks and the chomsky hierarchy. *International Conference on Learning Representations*, 2023.

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. *Advances in Neural Information Processing Systems*, 2023.

Nelson Elhage, Neel Nanda, Catherine Olsson, et al. A mathematical framework for transformer circuits. *Distill*, 2020. URL `https://distill.pub/2020/circuits/zoom-in/`.

Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a graph: Encoding graphs for large language models. *arXiv preprint arXiv:2310.04560*, 2024.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

Wes Gurnee and Max Tegmark. Language models represent space and time. *arXiv preprint arXiv:2310.02207*, 2023.

Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. In *Proceedings of the 7th Python in Science Conference*, volume 11, pp. 11–15, 2008.

Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Kaya Stechly, Mudit Verma, Siddhant Bhber, Lucas Knuth, Brian Marquez, et al. Llms can't plan, but can help planning in llm-modulo frameworks. *arXiv preprint arXiv:2402.01817*, 2024.

Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. Representational similarity analysis–connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2:4, 2008.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253, 2017.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. *arXiv preprint arXiv:2210.13382*, 2024.

James L. McClelland, Matthew Botvinick, David C. Noelle, and David C. Plaut. Integrating cognitive science with deep learning for robust AI. *Neural Computation*, 32(8):1535–1551, 2020a.

James L McClelland, Felix Hill, Maja Rudolph, Jason Baldridge, and Hinrich Schütze. Placing language in an integrated understanding system: Next steps toward human-level performance in neural language models. *Proceedings of the National Academy of Sciences*, 117(42):25966–25974, 2020b.

George A Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97, 1956.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*, 2021.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Tilman Rauker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. *arXiv preprint arXiv:2207.13243*, 2023.

James Snell et al. Scaling inference-time compute in language models. *arXiv preprint arXiv:2402.12147*, 2024. URL `https://arxiv.org/abs/2402.12147`.

Aarohi Srivastava, Abhishek Rastogi, Abhishek Rao, Abu Awal Md Shoeb, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022. BIG-Bench collaboration.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*, 2024.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36, 2023.

Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Large language models still can't plan (a benchmark for llms on planning and reasoning about change). *arXiv preprint arXiv:2206.10498*, 2022.

Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Large language models still can't plan (a benchmark for llms on planning and reasoning). *arXiv preprint arXiv:2305.15465*, 2023.

Petar Veličković, Adrià Puigdomènech Badia, David Budden, Razvan Pascanu, Andrea Banino, Misha Dashevskiy, Raia Hadsell, and Charles Blundell. The clrs algorithmic reasoning benchmark. *International Conference on Machine Learning*, pp. 22084–22102, 2022.

Xiaofei Wang, Chang Li, Li Dong, et al. Reasoning in large language models: A geometric perspective. *arXiv preprint arXiv:2407.02678*, 2024. URL `https://arxiv.org/abs/2407.02678`.

Taylor Webb, Keith J Holyoak, and Hongjing Lu. Emergent analogical reasoning in large language models. *Nature Human Behaviour*, 7(9):1526–1541, 2023.

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

Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2015.

Bowen Zhang, Daijun Ye, Yanhong Feng, Yujie Chen, Zongqi Xu, and Kai Ding. Graph chain-of-thought: Augmenting large language models by reasoning on graphs. *arXiv preprint arXiv:2404.07103*, 2024.

Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Josh Susskind, Samy Bengio, and Preetum Nakkiran. What algorithms can transformers learn? a study in length generalization. *arXiv preprint arXiv:2310.16028*, 2024.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A Example Trial Walkthrough

This appendix provides a concrete walkthrough of a single representative trial to illustrate how the three evaluation regimes—autonomous generation, internal analysis, and hybrid evaluation—operate on the same problem instance.

A.1 Sample Graph and Task

We consider a hierarchical tree graph (n=7, branching factor 2, depth 3) under a value-based planning objective. The task is to identify the path from Node 1 to the highest-reward leaf. The optimal path is: Node 1 → Node 2 → Node 5 → Node 11 (cumulative reward 37). Alternative paths exist with slightly lower rewards, creating ambiguity testing whether the model systematically explores alternatives.

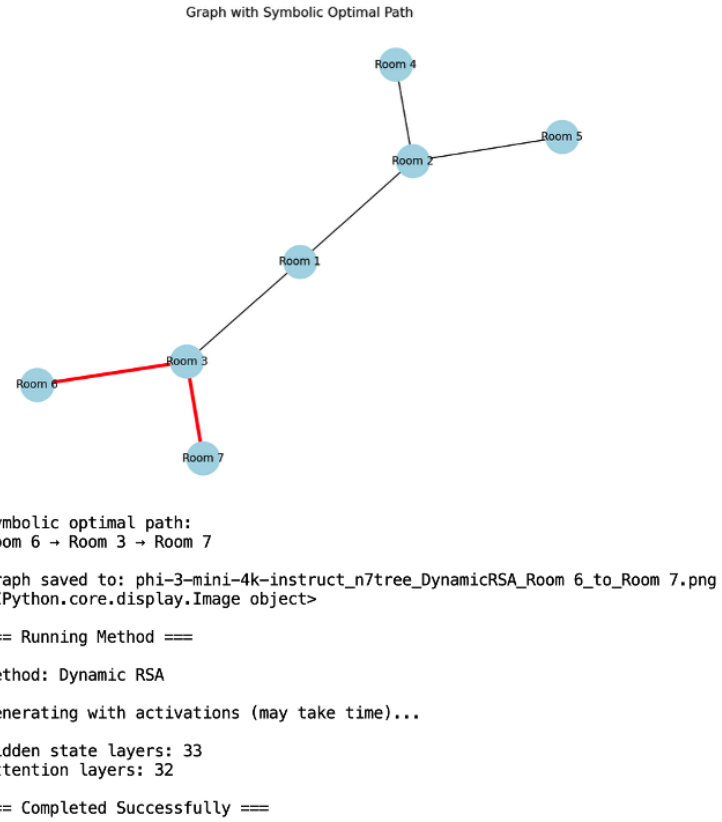


Figure 7: Example hierarchical graph (n=7) with optimal path highlighted. Node labels indicate identity; reward values are assigned to leaf nodes (Nodes 8–14). This figure provides a shared visual reference for behavioral, representational, and attention analyses.

A.2 Scratchpad Generation (Autonomous Condition)

Figure 8 shows representative scratchpad output. The model exhibits the characteristic pattern observed across experiments:

- **Steps 1–2**: Model correctly identifies Node 1 as start, lists adjacent nodes (2, 3) as frontier, selects Node 2 with plausible justification.
- **Steps 3–4**: Maintains valid state tracking (Visited: {1, 2}, Frontier: {3, 4, 5}) and makes admissible transitions.

- **Step 5**: Control degrades—model lists Node 3 in frontier despite it being a sibling of current path, indicating confusion between tree structure and frontier membership.
- **Step 6+**: Scratchpad structure becomes malformed; visited set not updated; premature termination without exploring higher-reward alternatives.

This pattern—short-horizon coherence followed by control failure—aligns with temporal dynamics: attention coherence degrading (65% → 40%), state drift accumulating (0.82 → 0.54), behavioral validity collapsing (80% → 20%).

Sample Output Log: *Step 1: Current Room is Room 1. Choices: Room 2, Room 3. Rationale: Both branches (2 and 3) have rewards deeper in their path. Arbitrarily choose Room 3.*

Figure 8: Representative Scratchpad trace showing local coherence (Steps 1–4) followed by control failure (Steps 5+). Degradation manifests as frontier confusion, premature termination, and loss of scratchpad structure—illustrating *simulation collapse*.

A.3 Hybrid Symbolic Validation

Figure 9 shows hybrid validation. Candidate paths are:

1. Optimal: Node 1 → 2 → 5 → 11 (reward 37)
2. Locally greedy: Node 1 → 2 → 4 (reward 28)
3. Near-optimal: Node 1 → 3 → 7 (reward 32)
4. Random valid: Node 1 → 3 → 6 (reward 19)

The model correctly selects Path 1 as optimal: "Path 1 reaches Node 11 with cumulative reward 37, higher than alternatives." This demonstrates preserved evaluative capacity despite failed autonomous generation.

Hybrid Symbolic Planner:
To test the LLM's capacity for contextual validation and repair over deterministic results.
Workflow: A symbolic planner provides the LLM with paths: **P1** (Optimal: 1→3→7), **P2** (Sub-optimal: 1→2→5), and **P3** (Loop: 1→2→1→3). **LLM Validator** must choose P1.
Analysis: Tests if the LLM can correctly evaluate and select the highest-value path, confirming its *reward reasoning* capability even when candidates are pre-generated.

Figure 9: Hybrid validation: model correctly selects optimal path when presented as candidate, demonstrating evaluative competence. Errors in hybrid condition are value-preference (selecting suboptimal valid paths), not structural (selecting invalid paths).

A.4 Internal Dynamics

For this trial:

- **Dynamic RSA**: $\rho = 0.61$ with graph distance, indicating preserved topological representation despite invalid output.
- **Attention dynamics**: State/frontier attention starts at 68% (step 1), declines to 42% (step 6).
- **Function vectors**: Path membership probe accuracy 83% at step 3—model represents which nodes lie on optimal path but fails to use this information.
- **State drift**: Below 0.6 at step 4, before first behavioral error at step 5—confirming temporal precedence.

B Additional Visualizations

Figures 10–11 show RSA and attention pairing for a representative trial, illustrating that representational alignment ($\rho = 0.64$) persists even when behavioral execution fails.

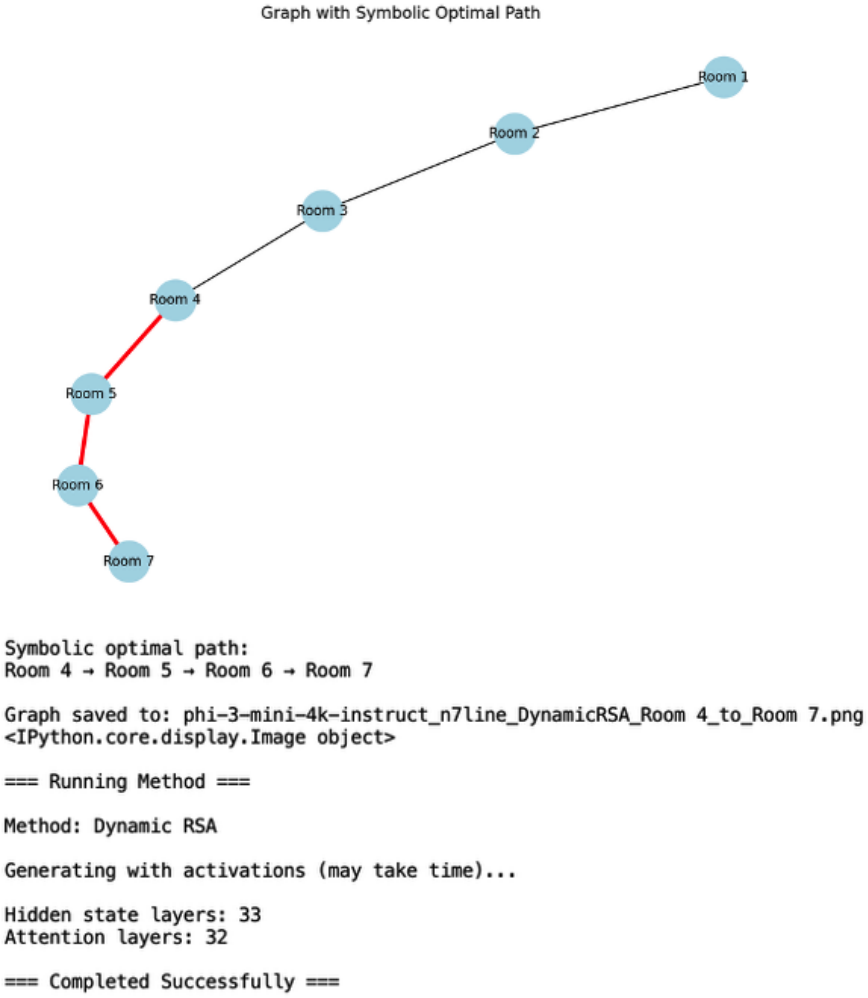
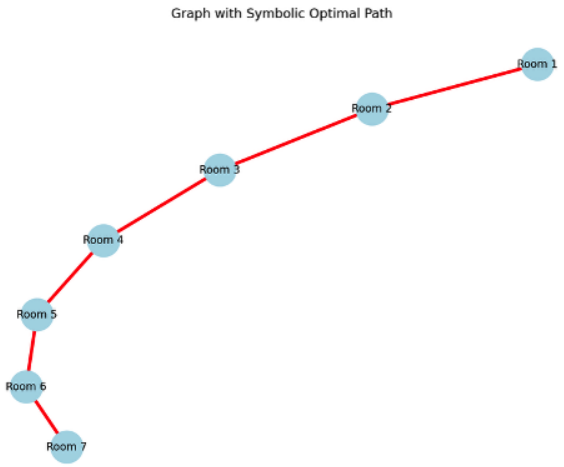


Figure 10: Dynamic RSM at generation step $t = 4$. Each cell (i, j) shows cosine similarity between hidden states of nodes i and j. Block structure reflects graph topology. Spearman correlation with graph distance $\rho = 0.64$ indicates preserved structure despite invalid behavioral output by step 5.




```
Symbolic optimal path:
Room 1 → Room 2 → Room 3 → Room 4 → Room 5 → Room 6 → Room 7

Graph saved to: phi-3-mini-4k-instruct_n7line_Attention_Room 1_to_Room 7.png
<IPython.core.display.Image object>

=== Running Method ===

Method: Attention Analysis

Generating with activations (may take time)...
Attention layers: 32
Layer 0: torch.Size([1, 32, 207, 207])
Layer 1: torch.Size([1, 32, 207, 207])
Layer 2: torch.Size([1, 32, 207, 207])
Layer 3: torch.Size([1, 32, 207, 207])
Layer 4: torch.Size([1, 32, 207, 207])
Layer 5: torch.Size([1, 32, 207, 207])
Layer 6: torch.Size([1, 32, 207, 207])
Layer 7: torch.Size([1, 32, 207, 207])
Layer 8: torch.Size([1, 32, 207, 207])
Layer 9: torch.Size([1, 32, 207, 207])
Layer 10: torch.Size([1, 32, 207, 207])
Layer 11: torch.Size([1, 32, 207, 207])
Layer 12: torch.Size([1, 32, 207, 207])
Layer 13: torch.Size([1, 32, 207, 207])
Layer 14: torch.Size([1, 32, 207, 207])
Layer 15: torch.Size([1, 32, 207, 207])
Layer 16: torch.Size([1, 32, 207, 207])
Layer 17: torch.Size([1, 32, 207, 207])
Layer 18: torch.Size([1, 32, 207, 207])
Layer 19: torch.Size([1, 32, 207, 207])
Layer 20: torch.Size([1, 32, 207, 207])
Layer 21: torch.Size([1, 32, 207, 207])
Layer 22: torch.Size([1, 32, 207, 207])
Layer 23: torch.Size([1, 32, 207, 207])
Layer 24: torch.Size([1, 32, 207, 207])
Layer 25: torch.Size([1, 32, 207, 207])
Layer 26: torch.Size([1, 32, 207, 207])
Layer 27: torch.Size([1, 32, 207, 207])
Layer 28: torch.Size([1, 32, 207, 207])
Layer 29: torch.Size([1, 32, 207, 207])
Layer 30: torch.Size([1, 32, 207, 207])
Layer 31: torch.Size([1, 32, 207, 207])

=== Completed Successfully ===
```

Figure 11: Attention heatmap showing early focus on task-relevant tokens (dark bands at graph-state positions) followed by diffusion across structural tokens and recent scratchpad text. Control degradation occurs while representations (Figure 10) remain intact.

C Supplementary Statistical Tables

Table 4: RSA correlations by model, topology, and generation step. Standard deviations in parentheses. Correlations remain stable or improve across steps even as behavioral validity collapses.

Model	**Topology**	**Step 1–3**	**Step 4–6**	**Step 7+**
Phi-3 Mini	Line	0.58 (0.05)	0.61 (0.06)	0.60 (0.07)
Phi-3 Mini	Tree	0.60 (0.06)	0.64 (0.05)	0.62 (0.08)
Phi-3 Mini	Clustered	0.54 (0.08)	0.58 (0.07)	0.55 (0.09)
Gemma	Line	0.52 (0.06)	0.55 (0.07)	0.53 (0.08)
Gemma	Tree	0.54 (0.07)	0.57 (0.06)	0.55 (0.09)
Gemma	Clustered	0.48 (0.09)	0.52 (0.08)	0.49 (0.10)

Table 5: Attention allocation to task-relevant tokens by generation step. Attention drifts from state/frontier tokens toward recent output and structural tokens.

Token Class	**Step 1**	**Step 3**	**Step 5**	**Step 7**	**Step 9**
Current state	0.28	0.24	0.19	0.15	0.12
Frontier	0.25	0.21	0.17	0.14	0.11
Visited	0.12	0.11	0.09	0.08	0.07
Recent output	0.15	0.22	0.29	0.35	0.42
Structural	0.20	0.22	0.26	0.28	0.28