ARCUS-X: ADAPTIVE RULE COMPLIANCE UNDER SEQUENTIAL EXECUTION FOR DIAGNOSING LONG-HORIZON LANGUAGE MODEL RELIABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon language-model evaluations often reduce performance to final-answer accuracy, obscuring failures that accumulate across dependent sequential transitions. We introduce ARCUS-X, an auditable benchmark and evaluation protocol for decomposing deterministic sequential execution into trajectory correctness, continuity, rule binding, horizon control, response validity, and excess generation. ARCUS-X uses procedurally generated toroidal-grid tasks with exact per-transition ground truth, controlled lexical and transition perturbations, raw-stream preservation, and deterministic post-run diagnostics. Across 13 models, ARCUS-X exposes substantial separation between aggregate trajectory quality and observed failure signatures. In the adaptively sampled runs, GPT-5-mini attains 84.8% step accuracy, 42.5% exact match, and 84.5% continuity, but 25.9% generation efficiency and 11.2% invalid responses. The valid-error partitions differ sharply: state-tracking signatures account for 93.8% of Qwen 3 Coder 480B A35B probes and 84.3% of GLM-5 probes, transition-rule signatures are largest for Mistral Large 2512 (23.6%), and semantic-interpretation signatures are largest for Kimi K2.5 (23.2%). Sampled fracture behavior also varies by perturbation tier. Because AFS allocates horizons from observed performance, pooled aggregates describe the acquired support rather than a common uniform horizon distribution. ARCUS-X therefore provides a reproducible diagnostic record of explicit state-transition execution rather than a universal model ranking or measure of general long-horizon agency.

1 INTRODUCTION

As large language models (LLMs) are increasingly used for complex scientific and software tasks, reliability over long sequences has become an important evaluation problem. Existing evaluations such as GSM8K Cobbe et al. (2021), MMLU Hendrycks et al. (2020), and LiveBench Gu et al. (2024) primarily measure task-level correctness, while execution-based evaluations such as HumanEval Chen et al. (2021) test functional correctness. These evaluations are useful for determining whether a task succeeds, but they provide limited information about how reliability changes as dependent intermediate states accumulate across a sequence.

This becomes important when two models achieve similar final outcomes but fail in fundamentally different ways. One may gradually drift from the correct state, while another may preserve the state but fail to terminate correctly. Final-answer metrics can therefore hide behaviors such as state drift, incorrect intermediate transitions, horizon-dependent degradation, and failures to preserve constraints throughout execution.

Recent work has begun to study reliability across longer executions and through trajectory-level analysis. *Beyond pass@1* Khanal et al. (2026) examines reliability as task duration increases, while *Long-Horizon Agent Trajectory Attribution* Chen et al. (2026), AgentRewardBench Lù et al. (2025), Plan-RewardBench Wang et al. (2026), and KellyBench Grady et al. (2026) study long-horizon behavior through trajectories, rewards, attribution, and sequential decision making. ARCUS-X complements this work by focusing on a narrower unit of analysis: **sequential state-transition execution**. Rather than evaluating an entire agent environment, ARCUS-X uses procedurally generated

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

deterministic tasks where each transition has deterministically computable ground truth, independent from model evaluation. This allows horizon and semantic perturbations to be controlled directly while measuring how reliably models maintain and transform state over time.

To address this, we introduce ARCUS-X as an auditable benchmark and evaluation protocol for long-horizon sequential execution. ARCUS-X evaluates the trajectory rather than only its final outcome, characterizing state preservation, rule binding, perturbation robustness, horizon control, sampled fracture behavior, validity, and output overhead. Its central object is diagnostic decomposition: similar aggregate accuracy can conceal different observable signatures, including state drift, rule misapplication, rule-binding errors, premature termination, and formatting failure. ARCUS-X is designed not to introduce trajectory evaluation itself, but to make sequential execution failures exactly attributable under controlled transitions, perturbations, and operational validity. These labels describe benchmark behavior, not hidden model cognition.

This work makes the following contributions:

1. We introduce ARCUS-X, an auditable, procedurally generated benchmark and evaluation protocol for sequential state-transition execution reliability.
2. We develop trajectory-level metrics and a deterministic diagnostic taxonomy that distinguish state tracking, transition-rule application, semantic interpretation, horizon collapse, and formatting without reducing reliability to a single accuracy value.
3. We provide controlled lexical and transition perturbations that probe rule binding separately from transition-system execution, together with a bounded AFS protocol that directs evaluation toward threshold-relevant horizons without exhaustively evaluating every horizon.
4. We report an auditable evaluation workflow with deterministic seeded probes, independently validated ground truth, raw-stream preservation, resumable evaluation, experiment metadata, and explicit separation of operational from cognitive failures.

Code, evaluation artifacts, and reproducibility materials are available at `https://anonymous.4open.science/r/ARCUS-X`.

2 Related Work

Existing LLM evaluations such as GSM8K Cobbe et al. (2021), MMLU Hendrycks et al. (2020), and LiveBench Gu et al. (2024) primarily measure task-level correctness, while HumanEval Chen et al. (2021) evaluates functional correctness through execution-based tests. These benchmarks assess whether a task is completed, but they provide limited information about how reliability changes across dependent state transitions.

Recent work extends evaluation toward reliability and trajectory-level analysis. *Beyond pass@1* Khanal et al. (2026) studies reliability as a function of task duration, including reliability decay, variance amplification, and meltdown onset. *Long-Horizon Agent Trajectory Attribution* Chen et al. (2026), AgentRewardBench Lù et al. (2025), and Plan-RewardBench Wang et al. (2026) further examine trajectory-level behavior, attribution, and evaluation.

Broader agent benchmarks instead evaluate success in naturalistic, multi-step environments. SWE-bench Jimenez et al. (2024) evaluates repository-level issue resolution, WebArena Zhou et al. (2024) evaluates web-agent behavior, and AgentBench Liu et al. (2024) spans multiple interactive environments. These settings deliberately combine planning, tool use, observation, and environment-specific uncertainty. ARCUS-X is not a substitute for them. Rather, it removes those factors to test whether a model follows an explicitly specified deterministic transition system. This controlled scope permits exact trajectory-level attribution, but does not establish general long-horizon agent capability or reliability across long-horizon agents. KellyBench Grady et al. (2026) and related sequential decision-making environments provide additional tests of extended decision making.

ARCUS-X instead isolates sequential state-transition reliability in a deterministic environment. Exact per-transition ground truth supports direct measurement of step correctness, continuity, exact agreement, divergence, horizon compliance, and sampled fracture behavior while controlling horizon and the model-facing rule system. Its contribution is narrower than general trajectory evaluation:

exact trajectory attribution, controlled rule perturbations, an explicit validity split, deterministic failure signatures, and adaptive horizon allocation in one auditable protocol.

3 METHODOLOGY

ARCUS-X isolates long-horizon sequential execution reliability under controlled and deterministic perturbations.

3.1 EXPERIMENTAL DESIGN

Within ARCUS-X, the base unit of evaluation is a *probe*: a deterministic task instance defined by its initial coordinate, grid dimensions, transition rules, ordered action sequence, and horizon (z), the number of sequential state transitions required. A useful visualization is to view each transition as a link in a chain, with larger horizons forming longer chains. ARCUS-X computes an authoritative ground-truth trajectory from the environment's transition rules and withholds it from the model.

3.2 PROCEDURAL GENERATION OF PROBES

ARCUS-X uses seeded procedural generation: the same seed and configuration regenerate the same probe. Unless a grid-size sweep is enabled, width W and height H are sampled independently and uniformly from $[5, 12]$; initial coordinates and action sequences are deterministically sampled from the same probe seed. Horizon and perturbation tier are controlled separately, and every probe contains exactly the requested number of transitions.

3.3 TRANSITION RULES AND ENVIRONMENT

ARCUS-X's environment is based on a grid defined as the formal state space. The state is represented as a coordinate $[x, y]$. The grid is two-dimensional, composed of a width, *W*, and height, *H*. The environment is toroidal; hence, crossing an edge wraps to the opposite side.

Each action corresponds to its labeled deterministic transition $(\Delta x, \Delta y)$. At each step, the model must first identify the action, retrieve its associated transition, apply it to the current state, and finally wrap the resulting coordinate. This wraparound behavior is implemented using a modulo operation. It is a sequential task because the next state entirely depends on the previous state.

$$x_{t+1} = (x_t + \Delta x_t) \mod W$$
$$y_{t+1} = (y_t + \Delta y_t) \mod H$$

Coordinates are indexed from 0 to $W - 1$ along the horizontal axis, and likewise from 0 to $H - 1$ along the vertical axis.

3.4 PERTURBATION REGIMES

ARCUS-X perturbs the model-facing task along a four-level categorical *tier* axis (Tier 0–3). These tiers are the active perturbations analyzed here. Stored runs also contain a legacy semantic-gravity value g, but in the main runner g neither alters the prompt nor selects the tier; all four tiers are crossed with each configured g. We therefore interpret results by tier, not g as an independent semantic-difficulty variable.

1. Tier 0 provides the baseline, in which there are standard directional labels and $(\Delta x, \Delta y)$ rules. Tier 0 establishes baseline state-tracking performance.
2. Tier 1 is a semantic disruption tier. It renames action labels to pseudowords; however, the underlying transition structure remains unchanged. This tests whether the model can bind arbitrary labels to the explicitly given rules rather than relying on familiar words such as "north" or "south."

3. Tier 2 is an attribute inversion tier, in which each transition is transformed as follows:
   $(\Delta x, \Delta y) \rightarrow (\Delta x, -\Delta y)$
   The ground truth is recomputed using the transformed rules. This tests whether the model follows the provided transition system rather than its prior assumptions about spatial directions.
4. Tier 3 acts as a coordinate transformation tier. It transforms each transition using a 90° rotation:
   $(\Delta x, \Delta y) \rightarrow (-\Delta y, \Delta x)$
   This tests whether the model can maintain the state-transition procedure when the spatial behavior conflicts with ordinary directional intuition. Ground truth is recomputed using the transformed transition rules.

The prompt contains the assigned labels and rules, grid dimensions, initial state, action sequence, question, and required output format; it omits g, hidden answers, and difficulty metadata. Stored runs use $g \in \{0.5, 3.0, 15.0\}$ crossed with all four tiers; because g is inert in this sweep, these values function as replicated configuration strata rather than semantic perturbations. Fixed calibration probes at $g = 0.0$, $z = 3$, Tier 1 are parsing/environment checks and are excluded from reported metrics.

3.5 Evaluation

We implement a native evaluation harness for ARCUS-X that separates probe generation, deterministic environment execution, model inference, response parsing, trajectory scoring, diagnostic classification, and adaptive fracture analysis. Model outputs are evaluated against independently computed ground-truth trajectories.

ARCUS-X records the model's trajectory, raw response, and inference metadata, including latency, output tokens, completion status, and provider/API errors. These measurements are independent of trajectory correctness, allowing execution, formatting, and efficiency failures to be distinguished from state-tracking errors. Responses are classified as valid trajectory, malformed recoverable output, empty output, parser failure, provider failure, or runtime/exception failure.

The parsed trajectory is compared against ground truth to calculate transition accuracy, exact trajectory match, and transition continuity. The expected trajectory contains the provided initial state plus one state per requested transition: $L_{\mathrm{GT}} = z + 1$. The predicted length L_{pred} is the number of parsed coordinates, with overlap $L_{\mathrm{overlap}} = \min(L_{\mathrm{pred}}, L_{\mathrm{GT}})$. Partial trajectories are scored rather than discarded. The initial state is retained for exact-match and structural-length checks but is excluded from transition accuracy and transition continuity because it is provided in the prompt.

ARCUS-X calculates the following raw measurements:

1. Output tokens: total generated tokens obtained from provider metadata or estimated, with source and benchmark confidence saved in raw output files.
2. Output tokens per second.
3. Correct transitions per second.

ARCUS-X uses multiple master seeds generated via a TRNG utilizing atmospheric noise Haahr (2026). Main tables report pooled probe-level means over the observations actually acquired by AFS. Because AFS may evaluate different horizon sets for different models and conditions, these pooled quantities are descriptive summaries of the observed support, not estimates under a shared uniform horizon distribution. Appendix A.6 reports the regenerated population mean and standard deviation of CRI across the five master seeds; raw artifacts retain per-seed CRI and fracture depth. We do not claim statistical significance for between-model differences.

3.6 Failure Taxonomy

The taxonomy is a deterministic post-scoring diagnostic, not an inference about a model's internal cognition. Its implementation is the released `classify_trajectory` and `_attribute_mechanism` functions in `arcus/analysis/taxonomy.py`; the full category

definitions, precedence rules, output-validity split, and serialized evidence are specified in Appendix A.1. We therefore refer to its outputs as *failure signatures* or diagnostic categories rather than causal mechanisms. Exact trajectories receive *none*; invalid responses never enter the trajectory taxonomy; and valid failures that cannot be mapped are retained as *unknown/unmapped*. Category frequencies should be interpreted together with the stated precedence rules, including conservative fallback assignments, rather than as direct measurements of latent cognitive causes.

3.7 METRICS

The primary measures of sequential execution reliability are:

1. **Step Accuracy**

$$\text{Step Accuracy} = \frac{\text{number of correctly reproduced post-initial states}}{z}.$$

   Comparison is performed at aligned post-initial positions over the overlapping region. The initial state is not counted.

2. **Exact Match**, which is binary per probe and requires identical coordinates and trajectory length.

3. **Continuity**, defined as the longest correct transition prefix:

$$\text{Continuity} = \frac{\text{longest correct post-initial prefix}}{z}.$$

First divergence is reported as a diagnostic rather than a primary aggregate metric. It records the 1-based position of the first mismatch; if the overlapping region is correct but the lengths differ, divergence is assigned to the first position beyond the overlap.

ARCUS-X also measures horizon and output behavior:

1. **Horizon Compliance**

$$\text{HC} = 1 - \frac{|z_{\text{generated}} - z_{\text{requested}}|}{\max(z_{\text{generated}}, z_{\text{requested}})}.$$

2. **Generation Bloat Index**

$$\text{GBI} = \frac{\max(0, T_{\text{actual}} - T_{\text{expected}})}{T_{\text{expected}}},$$

   where T_{actual} is the completion-token count. The deterministic completion allowance is $T_{\text{expected}} = 10(z + 1) + 10$: ten tokens for each of the $z + 1$ coordinate pairs plus a ten-token formatting allowance. GBI therefore measures *excess generation relative to this fixed benchmark allowance*; it is not a tokenizer-invariant minimum-description-length estimate. Provider usage is used when available; otherwise token counts fall back, in order, to an official tokenizer, a compatible tokenizer, or a documented deterministic heuristic. Token source and confidence are retained per probe.

3. **Generation Efficiency** $\text{GE} = \frac{1}{1+\text{GBI}}$ per probe. Reported GE and GBI columns are each arithmetic means of their respective per-probe values; consequently, mean GE is not generally equal to $1/(1 + \text{mean GBI})$.

We additionally report the Composite Robustness Index (CRI) as a secondary, researcher-specified composite of trajectory fidelity, horizon robustness, perturbation robustness, and generation efficiency. Its formulation and fixed weights are provided in Appendix A.5; primary claims in this paper rely on the decomposed measurements rather than CRI alone.

ARCUS-X also reports system-level efficiency using latency. Latency is total wall-clock response time and is treated as a systems-level diagnostic because it depends on batching, deployment conditions, and provider infrastructure: $\text{ALE} = \text{Accuracy}/\text{Latency}_{\text{seconds}}$.

3.8 Adaptive Fracture Search

Adaptive Fracture Search (AFS) is a bounded, reproducible, threshold-directed allocation protocol that acquires observations near horizons where valid-response step accuracy falls below a 50% floor without scanning every integer horizon.

At each horizon z, cognitive step accuracy is computed over operationally valid trajectories: $A(z) = \frac{1}{|V_z|} \sum_{i \in V_z} a_i$,

where V_z is the set of analyzable probes and a_i is the step accuracy of probe i. Operational failures are excluded from $A(z)$ and reported separately. Accordingly, $A(z)$ is conditional accuracy given an analyzable trajectory; it should not be interpreted as end-to-end success when validity itself declines.

AFS evaluates midpoint horizons with a bisection-style controller and expands its upper bound while performance remains above the floor. The controller is an allocation heuristic, not a monotonic-capacity model: recoveries remain visible in the sampled curve and are handled post hoc. Invalid or infrastructure-failed responses are excluded from the batch mean; a batch with no valid observations is operationally insufficient evidence. Evaluated probes are cached and reused.

The live-search value z^* is a below-floor horizon encountered by the allocation procedure, or a censored ceiling outcome when none is observed; it is not guaranteed to be the first failing horizon. A separate forward-look-ahead analysis reports *sampled structural fracture*. Both operate on adaptively sampled points, so fracture values are condition-specific diagnostics of the acquired curve rather than intrinsic capacities. The configured ceiling is $z = 150$.

4 Results

4.1 Benchmark Performance

We evaluated ARCUS-X on 13 models and focus the detailed analysis on the five representative models in Table 1; the Appendix reports all 13. The representative set uses fixed Generally Available model versions: GPT-5-mini (Microsoft Azure Foundry, version 2025-08-07), Kimi K2.5 (AWS Bedrock), GLM-5 (AWS Bedrock), Qwen 3 Coder 480B A35B (AWS Bedrock), and Mistral Large 2512 (Mistral AI API). Runs used five probes per evaluated batch, five master seeds, and temperature 0 or the lowest allowed value. Because AFS evaluates model- and condition-dependent horizon sets, completed execution counts and horizon mixtures differ across runs; the exact matrix is retained in each manifest and raw artifact. Unless explicitly conditioned on a shared horizon, the pooled values below therefore compare the observed AFS-supported samples and should not be read as estimates over an identical horizon distribution. Within the observed AFS-supported pools, GPT-5-mini has the highest step accuracy (84.8%), exact match (42.5%), and continuity (84.5%) of the five representative runs. The other four show 48.1–59.2% step accuracy and 1.1–39.1% exact match. GLM-5 and Qwen 3 Coder 480B A35B also show that high horizon compliance can coexist with lower trajectory correctness. These are run-level descriptive comparisons, not a controlled ranking over matched horizons.

Table 1: Overall ARCUS-X trajectory-level performance. Asterisks indicate response validity below 75%. Validity denotes the proportion of probes yielding cognitively analyzable responses.

Model	Step Acc.	Exact Match	Continuity	Horizon Compliance	Validity
GPT-5-mini	**84.8%**	**42.5%**	**84.5%**	88.2%	88.8%
Kimi K2.5	59.2%	39.1%	53.9%	78.0%	94.8%
GLM-5	55.2%	14.1%	50.3%	**98.3%**	100.0%
Qwen 3 Coder 480B A35B	48.1%	3.7%	41.6%	97.0%	100.0%
Mistral Large 2512	55.8%	1.1%	52.8%	46.0%	96.9%

4.2 Horizon-Dependent Degradation and Fracture

Figure 1 shows the observed per-horizon step-accuracy profiles. The curves are non-monotonic, which is precisely why we treat AFS as threshold-directed sampling rather than an estimator of a

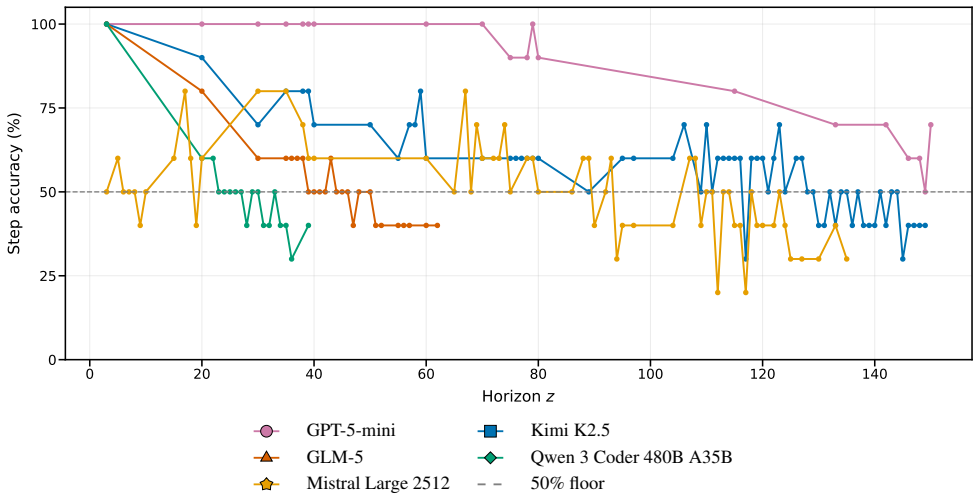

Figure 1: Per-horizon step accuracy for the five representative models. Each sampled point is drawn with a model-specific marker (circle, square, triangle, diamond, or star) in addition to its color, so the curves remain distinguishable in grayscale.

monotone capacity boundary. GPT-5-mini remains high across most sampled horizons with a dip near $z \approx 70$; Kimi K2.5 shows an early dip followed by lower but comparatively stable sampled accuracy; GLM-5 declines more sharply; Qwen 3 Coder 480B A35B declines more gradually; and Mistral Large 2512 recovers at intermediate horizons before declining again. Lines connect observations for readability only. Point density and horizon coverage are nonuniform because AFS allocates probes adaptively (Appendix A.4), so the figure supports descriptive diagnosis of sampled behavior rather than interpolation between points or a monotonic-capacity claim.

Table 2: Post-hoc structural fracture depth by perturbation tier for the five representative models. Values are the smallest sampled horizon below the 50% step-accuracy floor that is not followed by a recovery within the three-point look-ahead window; 'N/A' means that no such sampled point was found.

Model	Tier 0	Tier 1	Tier 2	Tier 3
GPT-5-mini	N/A	N/A	N/A	N/A
Kimi K2.5	145	N/A	130	35
GLM-5	57	44	56	30
Qwen 3 Coder 480B A35B	34	24	28	23
Mistral Large 2512	9	3	94	6

Table 2 shows that sampled structural fracture is strongly condition-dependent rather than an invariant model property. GPT-5-mini has no post-hoc fracture in the sampled curves, whereas Kimi K2.5 has late sampled fractures in Tier 0 and Tier 2 but an earlier one in Tier 3. Mistral Large 2512 makes the non-monotonicity especially clear: its Tier-2 sampled fracture is late despite much earlier values in the other tiers. The three-point recovery rule operates on the next three *sampled* horizons, whose spacing is not uniform, so these values describe the acquired curves rather than a fixed-width horizon neighborhood. The live AFS window was capped at $H = 150$; reaching that cap without a below-floor midpoint is recorded separately as a saturated, right-censored search result.

4.3 Semantic Robustness

As described in Section 3.4, the benchmark runner evaluates all four perturbation tiers at each stored gravity configuration. Table 3 reports per-tier step accuracy pooled over the observations acquired for each tier. Because AFS may sample different horizon mixtures across tiers, these values establish that the model-facing transformations produce distinct observed profiles, but they do not by themselves isolate a causal tier effect at matched horizons.

Table 3: Per-tier step accuracy for the five representative models. Tier 0 is baseline; Tier 1 renames action labels only; Tier 2 inverts the y-delta; Tier 3 rotates transitions $90°$.

Model	Tier 0	Tier 1	Tier 2	Tier 3
GPT-5-mini	87.4%	83.2%	87.0%	81.7%
Kimi K2.5	63.0%	**75.8%**	65.1%	32.9%
GLM-5	**63.4%**	58.1%	59.1%	40.2%
Qwen 3 Coder 480B A35B	55.2%	48.5%	51.8%	37.0%
Mistral Large 2512	56.0%	48.4%	58.5%	**60.4%**

The observed tier profiles are not ordered uniformly across models. GPT-5-mini spans 5.7 points between its highest and lowest pooled tier values. Kimi K2.5 and GLM-5 are substantially lower in Tier 3 than Tier 0 (30.1 and 23.2 points), while Qwen 3 Coder 480B A35B differs by 18.2 points. Mistral Large 2512 instead has its lowest pooled accuracy in Tier 1 and its highest in Tier 3, and Kimi K2.5 is higher in Tier 1 than Tier 0. These results argue against treating perturbation robustness as a single ordinal scalar: lexical relabeling and transition-altering transformations can produce different signatures for the same model. Because horizons are not matched by construction, we describe these as pooled tier associations rather than attributing the differences uniquely to the perturbation itself. Appendix A.7 reports all 13 models and the CRI decomposition.

4.4 Generation Efficiency and Failure Modes

Aggregate trajectory accuracy does not capture output overhead. We therefore report Generation Efficiency (GE), derived from GBI, as a benchmark-relative excess-generation score and Correct Steps per Second (CSS) as a secondary systems measure; latency and output-token throughput remain supporting diagnostics (Section 3.7).

Table 4: Generation efficiency measurements for the five representative models.

Model	Step Accuracy	GBI	Generation Efficiency	Correct Steps/sec
GPT-5-mini	**84.8%**	4.16	25.9%	1.939
Kimi K2.5	59.2%	1.09	56.4%	2.440
GLM-5	55.2%	**0.00**	**100.0%**	4.665
Qwen 3 Coder 480B A35B	48.1%	0.03	99.3%	**6.807**
Mistral Large 2512	55.8%	4.30	24.4%	1.154

ARCUS-X separates correctness from output overhead. GPT-5-mini combines high step accuracy with low GE, whereas GLM-5 and Qwen 3 Coder 480B A35B have near-100% GE but lower step accuracy. GE measures generation relative to the fixed benchmark allowance: 100% means only that output did not exceed that allowance, not tokenizer-independent optimality. CSS remains a provider- and deployment-dependent systems diagnostic and does not enter trajectory, fracture, GBI, or CRI scoring.

Table 5 reports the complete diagnostic partition for the representative models. Taxonomy percentages use valid probes as their denominator and include *None* (a fully correct trajectory) and *Unknown / Unmapped*; validity is reported separately over all completed probes. Every row's taxonomy columns sum to 100%, and invalid responses are not assigned a trajectory-level failure signature. Because the categories are precedence-based, their relative frequencies describe how the deterministic classifier partitions observed trajectories rather than independently validated latent mechanisms.

Table 5: Complete post-run validity and taxonomy partition for representative models. Taxonomy percentages are relative to valid probes.

Model	Valid	Invalid	State	Transition	Semantic	Horizon	Format	Unknown	None
GPT-5-mini	88.8%	11.2%	10.4%	0.1%	0.0%	41.7%	0.0%	0.0%	47.8%
Kimi K2.5	94.8%	5.2%	10.7%	4.0%	23.2%	3.9%	5.9%	0.0%	52.3%
GLM-5	100.0%	0.0%	84.3%	0.6%	0.0%	1.0%	0.0%	0.0%	14.1%
Qwen 3 Coder 480B A35B	100.0%	0.0%	93.8%	1.1%	0.0%	1.4%	0.0%	0.0%	3.7%
Mistral Large 2512	96.9%	3.1%	14.1%	23.6%	10.2%	2.3%	4.1%	0.0%	45.6%

5 DISCUSSION

5.1 CENTRAL FINDING

Our central finding is that sequential reliability is multi-faceted rather than reducible to one score. ARCUS-X jointly reports trajectory correctness, prefix persistence, horizon compliance, perturbation profiles, validity, and output overhead. The deterministic taxonomy is applied uniformly across all 13 evaluated models: Table 5 gives the complete validity and failure-signature partition for the five representative models, and Table 15 and Figure 4 extend it to all 13 models.

The important separation is descriptive. In the observed AFS-supported samples, high trajectory accuracy does not imply low output overhead, high horizon compliance does not imply high trajectory accuracy, and two models with similar pooled accuracy can produce very different diagnostic partitions. At the same time, adaptive sampling means these pooled values are not controlled estimates over a common horizon distribution, and the precedence-based taxonomy is not a causal account of internal cognition. ARCUS-X is therefore strongest as an auditable decomposition of observed execution behavior under explicit deterministic rules.

5.2 LIMITATIONS

ARCUS-X is synthetic by design: it isolates state-transition execution rather than planning, tool use, memory retrieval, partial observability, changing environments, or irreversible actions, and therefore does not measure general long-horizon agency. Provider/API conditions, response validity, and systems latency may also affect comparisons.

The main empirical limitation is adaptive support. AFS samples horizons nonuniformly from earlier outcomes, so models and tiers can be pooled over different horizon mixtures; completed-probe counts range from roughly 2,200 to 7,600 per model. Cross-model and cross-tier aggregates are therefore descriptive over acquired support, not matched-horizon causal estimates. The post-hoc fracture rule likewise uses three subsequent sampled points with variable spacing, making fracture a property of the sampled condition rather than an intrinsic capacity.

AFS accuracy is conditional on operational validity: unanalyzable responses are excluded from $A(z)$ and reported separately, so long-horizon invalidity can produce insufficient cognitive evidence rather than a low cognitive score. The taxonomy is exhaustive and reproducible but precedence-based and not externally validated; GBI/GE depend on a fixed token allowance and provider/tokenizer accounting; and CRI uses researcher-specified weights. Five probes per batch and five master seeds characterize variation descriptively, not inferentially. Future work should add a common fixed-horizon grid, matched-horizon tier analysis, taxonomy validation, and richer sequential environments.

6 CONCLUSION

ARCUS-X shows that trajectory correctness, operational validity, output overhead, horizon behavior, and the composition of observed errors can diverge substantially. Across the adaptively sampled runs, models differ not only in pooled accuracy but in where trajectories break, whether requested horizon is preserved, how often output remains analyzable, and which deterministic failure signatures valid errors receive. No single scalar represents all of these behaviors.

The contribution is therefore not a universal ranking or intrinsic "maximum horizon," but an auditable diagnostic record: deterministic tasks and ground truth, raw responses, validity status, per-probe measurements, sampled horizon curves, and an exhaustive post-run partition. AFS directs probes toward threshold-relevant horizons while preserving non-monotonicity, censoring, and operational insufficiency rather than converting them into capacity claims. ARCUS-X measures execution of explicit deterministic state-transition systems under controlled perturbations; it does not establish general agency, naturalistic planning, or tool-use competence.

GENERATIVE AI DISCLOSURE STATEMENT

In this work, we used generative AI tools for soundboarding, refining the mathematical metrics and hypotheses, providing feedback on the methodology, and interpreting the results as a cohesive

qualitative story. We did not use generative AI tools to generate synthetic datasets, develop proofs, or perform translation; implementation methods were not applicable to this work. We also used generative AI tools to edit LaTeX tables and figures, create and edit code, outline the paper, draft implementation pseudocode for the appendix implementation, make claims accurate to the working codebase, summarize existing literature, and identify gaps in the implementation as it evolved. Generative AI tools were used to polish and improve the clarity and readability of the manuscript at large. We reviewed all AI-assisted work. LLM-generated code was reviewed and audited multiple times, and the authors verified all AI-assisted edits to the paper. We also conducted an independent literature review to complement the generated summaries. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

ARCUS-X is reproducible at the task-generation, evaluation, and artifact levels. Tasks are generated procedurally from explicit master seeds. The same $(\text{seed}, \text{tier}, \text{task index})$ deterministically produces the same grid, initial state, actions, transition rules, label mapping, and ground-truth trajectory. All task randomness is derived from `random.Random`; tier transformations and their ground truth are recomputed deterministically. The reference trajectory is produced by the single transition implementation in `arcus/environment/grid.py`, and `validate_ground_truth()` checks it independently of model output. A stable SHA-256 experiment signature is implemented in the code release; when present, it binds the configuration and source revision to the resulting artifacts. The reported runs additionally retain their manifests and raw streams, so the exact evaluated conditions can be inspected rather than inferred from aggregate tables.

The model prompt contains only the task axioms, transition rules, initial state, grid information, action sequence, question, and required output format. Difficulty metadata, checksums, hidden answers, and final-coordinate shortcuts are excluded, and a SHA-256 prompt digest provides a process-independent prompt identity. The source-controlled parser and metrics pipeline separates trajectory correctness from formatting and execution failures. Step accuracy, exact match, and continuity compare the parsed trajectory with the independently generated path; final-state correctness is reported only as a diagnostic and cannot replace trajectory-level scoring. Specifically, step accuracy is the number of correct post-initial states divided by the requested horizon z; exact match requires identical coordinates and length; and continuity is the longest correct post-initial prefix divided by z. The provided initial state is retained for structural comparison but is not counted as a transition. Horizon compliance, parse failures, generation efficiency, fracture depth, latency, and token accounting are retained as additional diagnostics.

Every run records its master-seed list, probe count, gravity and tier settings, horizons, grid sizes, adaptive-search parameters, model identity, compatibility behavior, and source revision in an experiment manifest. Parallel execution assigns each master seed to an isolated runner, so worker ordering does not change benchmark semantics. Interrupted runs can resume from the raw stream: completed probes are identified by master seed, gravity, tier, horizon, probe index, and grid key, then rescored by the same pipeline used for live evaluation. The raw stream preserves prompts, model responses, parsed paths, ground truth, task parameters, token counts, latency, provider adaptations, and errors; metrics JSON and benchmark-result JSON provide structured summaries.

The reference environment is the digest-pinned image defined in `Dockerfile`, with dependencies installed from `requirements-lock.txt`; the optional local-inference image similarly pins its CUDA base image and uses `requirements-lock-local.txt`. Source provenance is resolved from `ARCUS_SOURCE_COMMIT` or `git rev-parse HEAD`, and runs fail if neither is available. The public entry point for API evaluation is `arcus.experiments.quickstart`. Validation is run with:

```
python tests/test_arcus_hardening.py
```

An API evaluation is run with:

```
python -m arcus.experiments.quickstart --api-key "$API_KEY" \
  --model <provider>/<model> \
  --n-probes <N> \
  --gravity-levels <g1> <g2> <g3>... \
```

```
  --master-seeds <seed-1> <seed-2> <seed-3>...
```

The reference container is validated with:

```
docker build --build-arg \
  SOURCE_COMMIT="$(git rev-parse HEAD)" \
  -t arcus-x:reference .
docker run --rm arcus-x:reference \
  python tests/test_arcus_hardening.py
```

Together, the pinned environment, manifests, raw streams, result matrices, and derived metrics provide an auditable path from source revision and deterministic tasks to the reported results.

REFERENCES

Jing Chen, Yang Sun, Li Zhang, Lin Xu, and Jie Shi. Long-horizon agent trajectory attribution: A unified benchmark and fine-grained annotation framework. *arXiv preprint arXiv:2608.06909*, 2026. URL https://arxiv.org/abs/2608.06909.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Thomas Grady, Kip Parker, Iliyan Zarov, Henry Course, Chengxi Taylor, and Ross Taylor. Kellybench: A benchmark for long-horizon sequential decision making. *arXiv preprint arXiv:2604.27865*, 2026. URL https://arxiv.org/abs/2604.27865.

Yue Gu et al. Livebench: A challenging, contamination-free llm benchmark. *arXiv preprint arXiv:2406.19314*, 2024. URL https://arxiv.org/abs/2406.19314.

Mads Haahr. RANDOM.ORG: True random number service. https://random.org, 2026. Accessed: 2026-07-27.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020. URL https://arxiv.org/abs/2009.03300.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R. Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Aaditya Khanal, Yangyang Tao, and Junxiu Zhou. Beyond pass@1: A reliability science framework for long-horizon llm agents. *arXiv preprint arXiv:2603.29231*, 2026. URL https://arxiv.org/abs/2603.29231.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2308.07201.

Xing Han Lù, Amirhossein Kazemnejad, Nicholas Meade, et al. Agentrewardbench: Evaluating automatic evaluations of web agent trajectories. *arXiv preprint arXiv:2504.08942*, 2025. URL https://arxiv.org/abs/2504.08942.

Jiaxuan Wang, Yulan Hu, Wenjin Yang, Zheng Pan, Xin Li, and Lan-Zhe Guo. Aligning agents via planning: A benchmark for trajectory-level reward modeling. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, 2026. URL https://aclanthology.org/2026.acl-long.1062/.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.13854.

A Appendix

A.1 Failure Taxonomy Implementation

This appendix specifies the deterministic classifier used for all regenerated failure reports. The implementation is `arcus/analysis/taxonomy.py`: `classify_trajectory` performs the top-level validity, correctness, and formatting decisions, and `_attribute_mechanism` assigns the primary mechanism for a valid incorrect trajectory. `classify_from_result` is the runner and post-run-analyzer entry point. The serialized `ProbeClassification` stores the validity flag, error code, mechanism, first failure step, secondary pattern flags, trajectory-correctness flag, output-compliance status, and formatting subcode.

A.1.1 Output-validity partition

The classifier first determines whether a response can provide cognitive evidence. An exception marker yields `E100_EXCEPTION_TOKEN`; an empty response yields `E101_EMPTY_OUTPUT`; an unparseable coordinate sequence yields `E102_PARSE_FAILURE`; and uncategorized invalid responses yield `E104_UNKNOWN_INVALID`. A classifier exception is recorded as a provider error (`E103_PROVIDER_ERROR`). Invalid outputs are marked *unclassifiable* and excluded from the cognitive-failure denominator. The post-run analyzer verifies $N_{\text{valid}} + N_{\text{invalid}} = N_{\text{completed}}$.

A.1.2 Primary mechanisms and precedence

For a valid trajectory, the classifier applies the following precedence:

1. **None**: the extracted coordinate sequence exactly matches ground truth. A correct but verbose or Markdown-wrapped response remains correct and receives an output-compliance subcode.
2. **Semantic failure**: the initial coordinate differs from ground truth. The classifier records either a coordinate-system offset when the next delta agrees, or a rule/environment-reconstruction subtype otherwise.
3. **Transition failure**: a wraparound-only error, or a majority of mismatched transitions exhibiting sign inversion, axis swap, wrong magnitude, or parity error.
4. **State-tracking failure**: after at least one correct transition, the trajectory develops a persistent offset or increasing recursive drift. This is also the conservative fallback for a valid, unmapped early divergence.
5. **Horizon failure**: divergence begins at the initial state and increases with horizon, or the predicted coordinate sequence is otherwise correct but has a different length.
6. **Output-format failure**: an empty or malformed trajectory prevents recovery of a correct sequence. This is distinct from a verbose response whose coordinates can still be extracted.

The classifier also records nonexclusive diagnostic patterns—wraparound, off-by-one, sign inversion, axis swap, coordinate clamp, oscillation, frozen state, and random divergence—and temporal dynamics (temporary, persistent, increasing, or oscillatory). Reports normalize the implementation's broad labels to the displayed categories *State Tracking Failure*, *Transition Rule Failure*, *Semantic Interpretation Failure*, *Horizon Collapse*, and *Formatting Failure*. A valid result that cannot be verified as correct or assigned to a displayed failure category is reported as *Unknown / Unmapped*; it

is never counted as *None*. The post run analyzer verifies that these taxonomy buckets sum to the number of valid probes.

A.2 Operational and Temporal Failure Analysis

The primary ARCUS-X metrics characterize trajectory correctness, but two additional distinctions are useful when interpreting long-horizon behavior. First, a model must produce an analyzable response before trajectory correctness can be measured. Second, aggregate step accuracy does not identify *where* within a valid trajectory errors emerge. We therefore separate operational validity from conditional trajectory correctness and perform a post-hoc temporal localization analysis over valid trajectories.

A.2.1 Operational versus Cognitive Reliability

ARCUS-X separates *operational reliability* from *cognitive reliability*. Operational reliability concerns whether a completed evaluation produces an output that can be parsed and scored under the benchmark protocol. Cognitive reliability, as used here, refers only to correctness of the observable state-transition execution conditional on such an analyzable response; it is not a claim about a model's internal cognitive process.

For horizon z, let V_z denote the set of probes yielding valid, analyzable trajectories, and let $a_i \in [0, 1]$ denote the ordinary step accuracy of valid probe i. Horizon-level cognitive accuracy is

$$A(z) = \frac{1}{|V_z|} \sum_{i \in V_z} a_i. \tag{1}$$

Operationally invalid probes are excluded from this denominator rather than assigned a cognitive accuracy of zero. They remain explicitly represented in the operational validity statistics and error-code accounting. ARCUS-X records E100 for exception-token outputs, E101 for empty outputs, E102 for parse failures, E103 for provider or evaluation exceptions, and E104 for other invalid responses. The accounting invariant is

$$N_{\text{valid}} + N_{\text{invalid}} = N_{\text{completed}}. \tag{2}$$

If $|V_z| = 0$, then $A(z)$ is undefined. ARCUS-X records the corresponding horizon as operationally insufficient evidence rather than setting $A(z) = 0$. The same rule is used by Adaptive Fracture Search: an all-invalid batch does not constitute evidence that the model crossed the cognitive fracture threshold.

This separation prevents two observably different failure modes from being collapsed into one statistic. A model may continue to return well-formed trajectories while increasingly misexecuting the supplied transition rules. In that case, the benchmark observes degradation in sequential execution. Alternatively, a model may increasingly return empty, malformed, or otherwise unanalyzable outputs. This is an end-to-end reliability failure, but no trajectory exists from which a cognitive step-accuracy value can be measured. Assigning such responses a value of zero would therefore mix response availability with conditional execution correctness.

Operational failures are not discarded. They are reported on their own axis and remain part of the overall reliability profile. The separation only prevents the absence of analyzable evidence from being interpreted as an observed incorrect trajectory.

A.2.2 Temporal Localization within Valid Trajectories

Conditional on operational validity, we next ask where within a trajectory correctness deteriorates. Overall step accuracy can conceal substantially different execution profiles. Two trajectories may have similar aggregate accuracy even if one fails immediately while another executes correctly for most of the requested horizon before degrading near the end.

For each valid probe at horizon z, the z requested transition indices are partitioned into three contiguous, as-even-as-possible regions: *early*, *middle*, and *late*. When z is not divisible by three, segment

Table 6: Within-trajectory step accuracy at two horizons shared by all 13 reported models. Early, middle, and late correspond to contiguous thirds of the requested transition sequence. Values are conditional on valid, analyzable trajectories. The table is a positional diagnostic rather than a model ranking.

	$z = 20$			$z = 30$		
Model	Early	Middle	Late	Early	Middle	Late
GPT-5-mini	99.8%	99.9%	99.9%	100.0%	99.8%	99.5%
Kimi K2.5	87.9%	86.5%	83.7%	69.5%	66.8%	65.9%
GLM-5	95.9%	84.3%	61.1%	92.4%	61.6%	35.5%
Qwen3-Coder-480B-A35B	93.5%	61.7%	30.2%	81.0%	36.9%	15.4%
Qwen3-235B-A22B-2507	49.2%	49.4%	46.5%	50.0%	47.3%	43.4%
Mistral Large 2512	56.4%	56.3%	56.2%	77.5%	77.0%	76.1%
Mistral Small 2603	74.1%	47.2%	42.2%	64.3%	21.2%	15.2%
Devstral 2 123B	50.6%	51.2%	51.4%	55.3%	56.0%	56.2%
Ministral 14B 2512	88.1%	82.2%	78.2%	84.7%	73.0%	69.2%
Ministral 8B 2512	58.4%	55.0%	50.6%	65.3%	54.9%	47.2%
Ministral 3B 2512	76.7%	60.8%	48.4%	70.4%	42.0%	30.0%
Nemotron Super 3 120B	89.5%	44.2%	19.2%	76.1%	18.2%	3.8%
MAI-Thinking-1	100.0%	100.0%	100.0%	99.3%	98.8%	98.2%

lengths differ by at most one transition. As in the primary step-accuracy metric, the provided initial state is excluded.

Let $T_r(z)$ denote the transition indices assigned to temporal region $r \in \{\text{early}, \text{middle}, \text{late}\}$. Using the same per-transition correctness indicators as the primary metric, regional accuracy is

$$A_r(z) = \frac{\sum_{i \in V_z} \sum_{t \in T_r(z)} \mathbb{1}[\hat{s}_{i,t} = s_{i,t}]}{\sum_{i \in V_z} |T_r(z)|}. \tag{3}$$

The temporal partition is purely positional. It does not assign a causal failure type to an individual transition. In particular, the ARCUS-X error taxonomy remains a *trajectory-level* diagnostic and is not duplicated across temporal thirds. Temporal localization answers *where* observed correctness is lost; the taxonomy separately describes the deterministic failure signature assigned to the complete trajectory.

Matched-horizon comparison. Because AFS produces nonuniform horizon support, pooling temporal statistics across all acquired horizons would inherit the same support differences as the headline aggregate metrics. We therefore use $z = 20$ and $z = 30$ for the cross-model comparison below. Both horizons were observed with analyzable trajectories for all 13 models in the reported evaluation, permitting a matched-horizon descriptive comparison.

A within-trajectory accuracy gradient is visible in a majority of the observed profiles. At $z = 20$, 9 of the 13 models have strictly decreasing accuracy from the early to middle to late region. At $z = 30$, this pattern occurs for 12 of 13 models. The magnitude, however, differs substantially across models.

Several models preserve high initial-step accuracy while degrading sharply later in the same trajectory. At $z = 30$, Nemotron Super 3 retains 76.1% accuracy in the early third but falls to 18.2% in the middle third and 3.8% in the late third. Qwen3-Coder similarly changes from 81.0% to 36.9% to 15.4%, while GLM-5 changes from 92.4% to 61.6% to 35.5%. These profiles would be substan-

tially compressed by a single trajectory-level mean: the model can execute a large fraction of early transitions correctly even when the terminal portion is already unreliable.

The pattern is not universal. At the same $z = 30$ horizon, Kimi K2.5 changes only from 69.5% to 66.8% to 65.9%, Mistral Large from 77.5% to 77.0% to 76.1%, and MAI-Thinking-1 from 99.3% to 98.8% to 98.2%. Devstral 2 is approximately flat at this horizon, with a small increase from 55.3% to 56.2%. We therefore do not interpret late-region degradation as a universal monotonic law of sequential execution.

Development at longer horizons. The onset of temporal degradation is itself horizon dependent. GPT-5-mini, for example, remains nearly uniform through moderate horizons: at $z = 80$ its early, middle, and late accuracies are 99.7%, 98.6%, and 97.7%, respectively. The separation becomes substantially larger among the longest sampled trajectories. At $z = 115$, the corresponding values are 99.7%, 96.5%, and 88.8%; at $z = 133$, they are 99.3%, 91.5%, and 78.2%; and at $z = 149$, they are 98.2%, 84.5%, and 68.0%. Thus, aggregate long-horizon degradation can arise even when the beginning of the requested trajectory remains almost perfectly executed.

Other models exhibit the same qualitative separation at much shorter horizons. GLM-5 already changes from 95.9% early accuracy to 61.1% late accuracy at $z = 20$, and reaches 78.0%, 25.6%, and 7.8% across the three regions at $z = 60$. Qwen3-Coder and Nemotron Super 3 show still sharper early-to-late separation by $z = 30$. The horizon at which a temporal gradient becomes visible is therefore model dependent and cannot be recovered from a single pooled accuracy statistic.

Adjacent horizons can also recover or fluctuate. We consequently do not fit a monotonic temporal-decay model or interpolate unobserved horizons. As elsewhere in ARCUS-X, the reported values describe only sampled, observed support.

Relation to the trajectory-level taxonomy. Temporal localization and the deterministic error taxonomy provide complementary views of the same observed output. Models with particularly large early-to-late separation, including GLM-5, Qwen3-Coder, and Nemotron Super 3, are also dominated by State Tracking Failure in the trajectory-level taxonomy. In contrast, models with flatter temporal profiles can exhibit different error compositions, including larger semantic or transition-related components.

This correspondence is descriptive rather than causal. A steep temporal gradient is compatible with accumulated state drift, but the thirds analysis does not establish that an early error caused a later error, nor does it identify an internal mechanism. Likewise, a trajectory-level taxonomy label is not retrospectively assigned to each region. The two diagnostics answer different questions: temporal localization identifies where correctness is lost within the observed sequence, while the taxonomy characterizes the failure signature of the trajectory as a whole.

Interpretation and limitations. The temporal analysis is conditional on operational validity. Horizons with no valid trajectories remain N/A and are not imputed as zero. Consequently, a model with strong conditional early/middle/late accuracy but low output validity should not be interpreted as strongly reliable end-to-end; its operational behavior must be considered separately.

The three-region partition is intentionally coarse. It exposes positional structure without introducing another fitted failure detector, but it does not identify an exact failure transition. In addition, because ARCUS-X trajectories are sequential, later incorrect states may be consistent with accumulated earlier drift, independent mistakes, or recovery after a previous mismatch. The regional statistics do not distinguish among these possibilities.

Finally, AFS produces different horizon support across models and conditions. The matched $z = 20$ and $z = 30$ comparison above avoids this issue for the cross-model table, while longer-horizon examples are interpreted only within their observed model-specific support. These results therefore strengthen the diagnostic decomposition of ARCUS-X—showing that long-horizon failure can be strongly localized within an otherwise valid trajectory—without converting the observed temporal profile into an intrinsic model capacity or causal account of failure.

A.3 AFS IMPLEMENTATION

This appendix documents the implementation used by the benchmark runner. AFS is not a separate scoring pipeline or model: it reuses the benchmark's task generation, trajectory extraction, scoring, taxonomy, and operational-validity checks. The active search is implemented in `BenchmarkRunner`; `FracturePointFinder` is used only for post-hoc analysis of an already sampled accuracy curve.

A.3.1 EVALUATION UNIT AND SCORING

For each gravity-tier condition and candidate horizon z, the runner generates a batch of seeded probes. The generator computes the ground-truth trajectory from the deterministic transition system, while the evaluator returns a model trajectory. For every response, the shared benchmark path extracts the predicted trajectory and computes step accuracy by comparing it with ground truth. The runner then applies the cognitive-validity gate before aggregating evidence. Thus, AFS operates on the same scores used for the benchmark and for resume reconstruction.

Let R_z denote the responses in a batch and let $V_z \subseteq R_z$ be the responses that pass the validity gate. The live search statistic is

$$A(z) = \begin{cases} \frac{1}{|V_z|} \sum_{i \in V_z} a_i, & |V_z| > 0, \\ \text{None}, & |V_z| = 0, \end{cases}$$

where $a_i \in [0, 1]$ is step accuracy. Provider and API failures, error terminations, invalid classifications, empty output, parse failure, and other operationally invalid responses are excluded from $A(z)$ and remain available for failure reporting. In particular, invalid responses are not converted to zero accuracy.

A.3.2 LIVE SEARCH CONTROLLER

The accuracy floor is $\tau = 0.5$. For an adaptive run, the search starts with $low = 1$ and $high = \min(40, H)$, where H is the configured hard ceiling (150 by default). Although ARCUS-X's search window can be adjusted, we use $z = 150$ for reproducibility. At each iteration, it evaluates $z_{\text{mid}} = \left\lfloor \frac{low+high}{2} \right\rfloor$. If $A(z_{\text{mid}})$ is `None` or is at least τ, the search treats the result as not providing cognitive-fracture evidence and advances deeper by setting $low = z_{\text{mid}} + 1$. A `None` batch is separately reported as operational insufficiency; it is not evidence that the model has cognitively survived that horizon. When the evaluated midpoint reaches the current upper bound and still passes, the upper bound may be doubled, up to H. If $A(z_{\text{mid}}) < \tau$, the midpoint is recorded as the current fracture candidate and the search retreats by setting $high = z_{\text{mid}} - 1$. The loop terminates when $low > high$ or the iteration limit is reached. If no below-floor midpoint is found and the search passes beyond H, the reported fallback is H; otherwise the recorded candidate is retained (or remains zero when the search did not establish one). Consequently, a ceiling result with operational insufficiency must not be interpreted as a cognitive capacity estimate.

The non-adaptive path evaluates the configured target horizon directly rather than running the bisection loop. Equality at τ is passing because the fracture comparison is strict ($A(z) < \tau$). Consequently, the returned search value is a directly evaluated candidate found by this controller, but it is not necessarily the mathematically first failing horizon when accuracy is noisy or non-monotonic.

A.3.3 POST-HOC FRACTURE ANALYSIS

The analysis-layer `FracturePointFinder` scans a sampled accuracy curve. When a point is below τ, it examines the next `look_ahead_step` points (three by default). If any point in that window returns to at least τ, the candidate dip is suppressed; otherwise the point is reported as a structural fracture. This is a forward-window recovery check, not a requirement for three consecutive failures, not three additional probes, and not a decision rule used by the live bisection loop. It is therefore important to distinguish the search candidate from a post-hoc structural-yield report.

A.3.4 CACHING, RESUME, AND REPRODUCIBILITY

Probe identity for reuse is keyed by seed, gravity, tier, horizon, probe index, and grid key. Resume logic deduplicates matching observations and validates model metadata and, for offline resume, the experiment hash; mismatches raise an error rather than silently mixing runs. The seeded generator and integer search arithmetic are deterministic conditional on identical model outputs, provider behavior, and raw-stream state. Seed alone does not guarantee identical AFS results because model responses, timeouts, truncation, rate limits, and resume ordering can differ. The controller is adaptive only in the narrow sense that prior batch accuracies determine interval updates; it is a deterministic midpoint search over the horizon axis, not a stochastic or semantic heuristic.

A.4 ADAPTIVE HORIZON SEARCH AND SAMPLING

ARCUS-X does not evaluate every possible execution horizon uniformly. Instead, the Adaptive Fracture Search (AFS) procedure allocates evaluations adaptively to localize the interval in which sequential reliability falls below the fracture floor, while avoiding unnecessary probing at horizons that are clearly reliable. This makes the search a bounded, bisection-style allocation strategy rather than an exhaustive scan across all integer horizons.

Figure 2 shows the pooled observed per-horizon step accuracy and probe density for **Ministral 14B 2512** (run identifier: `ministral-14b-2512`), aggregated across its sampled conditions. Only horizons observed in the run are plotted. Point color encodes probe density, so the plot displays both the measured accuracy and where AFS allocated probes. It is a sampling diagnostic, not a tier-specific fracture curve.

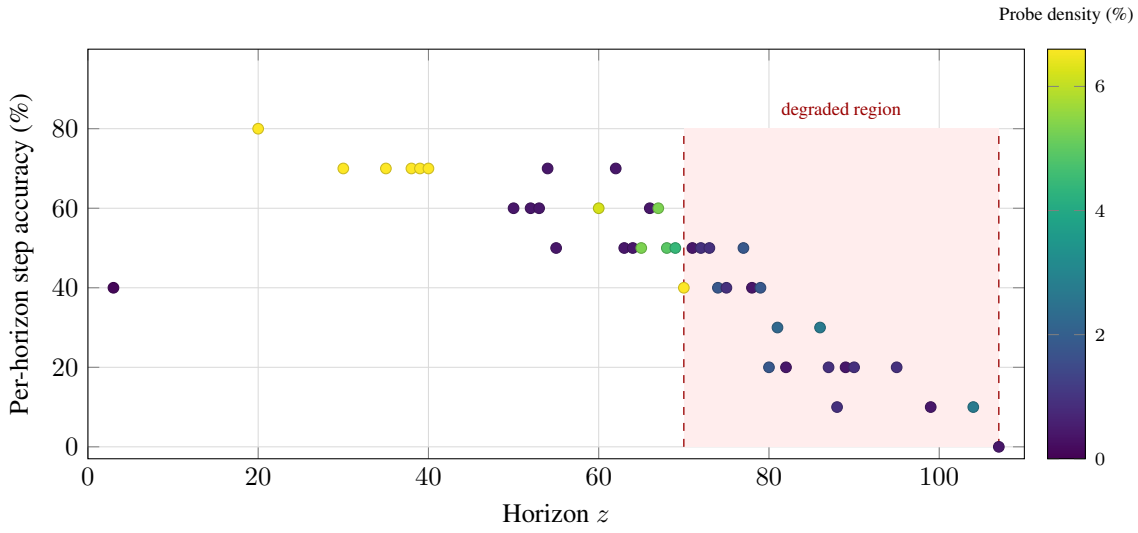

Figure 2: Pooled observed horizon accuracy and probe density for Ministral 14B 2512 across sampled conditions. Each point is an observed horizon; color encodes probe density (percent of probes). The shaded and dashed interval highlights pooled degradation from $z = 70$ through the maximum observed horizon $z = 107$.

The pooled degradation is not caused by an increasingly dense sampling pattern. For Ministral 14B 2512, pooled accuracy is generally at or above the 50% floor through $z = 69$, drops below the floor at $z = 70$, and does not exceed the floor again through the maximum observed horizon $z = 107$. AFS samples both densely and sparsely within this pooled degraded region, including the high-density probes at $z = 70$, $z = 86$, and $z = 104$; the pattern is therefore not simply explained by sampling density increasing as accuracy falls. The unusually high density at $z = 20$-40 and at $z = 60$ reflects adaptive allocation to boundary and transition horizons rather than uniform coverage.

A.5 CRI

The optional Composite Robustness Index (CRI) provides a secondary summary across evaluation axes: $\text{CRI} = 0.45\,\text{TF} + 0.30\,\text{HR} + 0.20\,\text{SR} + 0.05\,\text{GE}$.

Its components are:

1. **Trajectory Fidelity** $\text{TF} = 0.5A + 0.3C + 0.2E$, where A is step accuracy, C is continuity, and E is exact-match rate.

2. **Horizon Robustness**, defined as the normalized trapezoidal area under the accuracy-horizon curve:

$$\text{HR} = \frac{\sum_i \frac{A_i + A_{i+1}}{2}(z_{i+1} - z_i)}{z_{\max} - z_{\min}}.$$

3. **Semantic Robustness** $\text{SR} = 0.5R_1 + 0.25R_2 + 0.25R_3$, where $R_i = \min(1, A_i/A_0)$ is the ratio of Tier-i accuracy to Tier-0 accuracy, clamped to $[0, 1]$. If Tier-0 accuracy is zero or unavailable, SR is the arithmetic mean of the available tier accuracies. Tier 1 receives the larger weight because it is a label-only perturbation and leaves the transition function unchanged.

A.6 Seed-Level Variability

Table 7 reports the population mean and standard deviation of the secondary CRI across the five master seeds for each regenerated run. These values characterize seed sensitivity; they are not confidence intervals and should not be used to claim statistically significant between-model differences.

Table 7: CRI across five master seeds (population mean $\pm$ standard deviation). Asterisks indicate response validity below 75%.

Model	CRI
qwen.qwen3-235b-a22b-2507*	0.274 ± 0.011
Kimi K2.5	0.625 ± 0.029
Qwen 3 Coder 480B A35B	0.551 ± 0.016
mistral-small-2603	0.582 ± 0.030
Mistral Large 2512	0.542 ± 0.023
mistral.devstral-2-123b*	0.500 ± 0.031
ministral-14b-2512	0.570 ± 0.066
ministral-8b-2512	0.522 ± 0.038
ministral-3b-2512	0.500 ± 0.029
nvidia.nemotron-super-3-120b	0.551 ± 0.023
GLM-5	0.633 ± 0.029
GPT-5-mini	0.820 ± 0.032
MAI-Thinking-1*	0.510 ± 0.055

A.7 Full Model Results

Table 8: Complete overall ARCUS-X results for all 13 evaluated models. Asterisks indicate response validity below 75%.

Model	Step Acc.	Exact Match	Continuity	Horizon Compliance
qwen.qwen3-235b-a22b-2507*	14.3%	6.2%	12.1%	18.1%
Kimi K2.5	59.2%	39.1%	53.9%	78.0%
Qwen 3 Coder 480B A35B	48.1%	3.7%	41.6%	97.0%
mistral-small-2603	49.5%	4.9%	46.1%	80.4%
Mistral Large 2512	55.8%	1.1%	52.8%	46.0%
mistral.devstral-2-123b*	44.5%	0.1%	43.0%	42.7%
ministral-14b-2512	54.9%	15.5%	51.0%	74.4%
ministral-8b-2512	47.4%	7.0%	44.6%	66.4%
ministral-3b-2512	42.4%	10.3%	39.1%	71.5%
nvidia.nemotron-super-3-120b	45.8%	5.0%	40.5%	96.8%
GLM-5	55.2%	14.1%	50.3%	98.3%
GPT-5-mini	**84.8%**	**42.5%**	**84.5%**	88.2%
MAI-Thinking-1*	39.9%	33.0%	39.9%	40.7%

A.7.1 Horizon-Dependent Degradation and Fracture

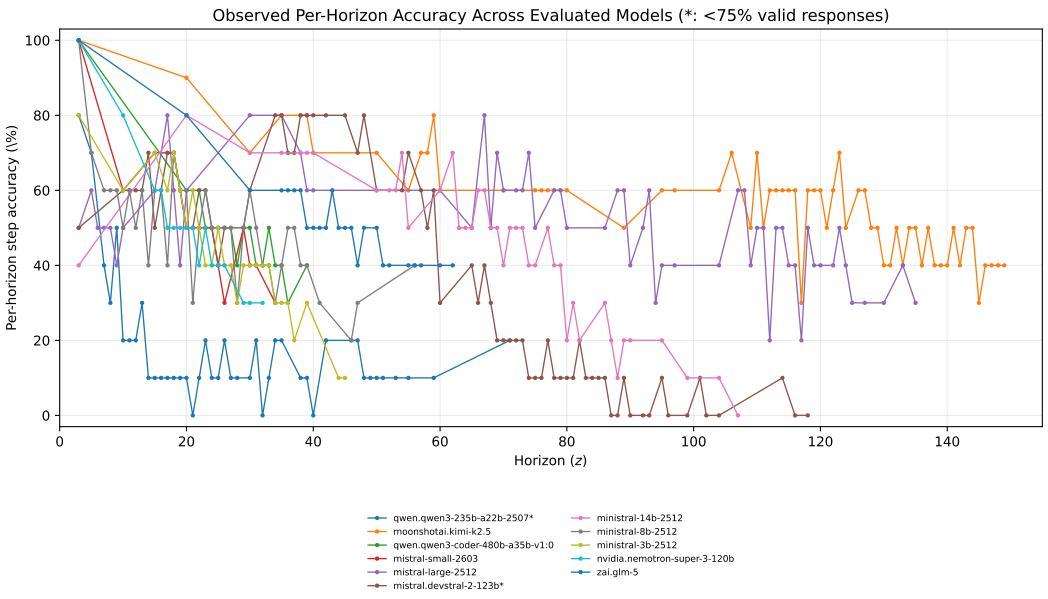

Figure 3: Observed per-horizon accuracy for every evaluated model. Curves join only horizons sampled by the adaptive procedure; asterisks identify models with response validity below 75%.

Table 9: Post-hoc structural fracture depth by perturbation tier for all 13 evaluated models. Values are the smallest sampled horizon below the 50% step-accuracy floor that is not followed by a recovery within the three-point look-ahead window; 'N/A' means that no such sampled point was found. Asterisks indicate response validity below 75%.

Model	Tier 0	Tier 1	Tier 2	Tier 3
qwen.qwen3-235b-a22b-2507*	13	7	10	5
Kimi K2.5	145	N/A	130	35
Qwen 3 Coder 480B A35B	34	24	28	23
mistral-small-2603	21	34	20	17
Mistral Large 2512	9	3	94	6
mistral.devstral-2-123b*	10	58	24	13
ministral-14b-2512	69	82	62	40
ministral-8b-2512	21	20	8	12
ministral-3b-2512	22	26	21	23
nvidia.nemotron-super-3-120b	24	17	20	16
GLM-5	57	44	56	30
GPT-5-mini	N/A	N/A	N/A	N/A
MAI-Thinking-1*	1	1	1	1

The table reports the structural-yield analysis of the sampled curves, not an exhaustive scan of every integer horizon. The live AFS controller starts with an upper bound of $\min(40, H)$ and doubles that bound only after a passing (or operationally insufficient) midpoint reaches the current upper edge; expansion stops at the absolute hard ceiling $H = 150$. We call the search window *saturated* when no below-floor midpoint has been found and the lower search bound advances beyond H; the recorded live-search value is then the right-censored ceiling, $z = 150$. This ceiling fallback is not evidence of 150-step cognitive survival, especially when batches are operationally insufficient. Conversely, the 'N/A' entries above mean that the post-hoc three-point recovery rule found no persistent sampled fracture; they are not synonymous with a saturated live search.

Table 10: Per-horizon accuracy fluctuation over observed horizons. Std. dev. is the population standard deviation of each model's observed per-horizon accuracies. Asterisks indicate response validity below 75%.

Model	Observations (N)	Mean Accuracy	Std. Dev.
qwen.qwen3-235b-a22b-2507*	44	17.50%	15.69%
Kimi K2.5	65	56.62%	13.39%
Qwen 3 Coder 480B A35B	18	47.78%	15.11%
mistral-small-2603	17	51.18%	16.04%
Mistral Large 2512	62	49.52%	14.19%
mistral.devstral-2-123b*	74	36.22%	27.84%
ministral-14b-2512	42	44.29%	20.14%
ministral-8b-2512	39	47.44%	13.53%
ministral-3b-2512	27	42.22%	16.63%
nvidia.nemotron-super-3-120b	17	47.65%	18.00%
GLM-5	25	51.60%	13.76%
GPT-5-mini	20	86.50%	16.52%
MAI-Thinking-1*	56	40.89%	23.24%

Table 11: Pearson correlation between horizon-accuracy fluctuation and trajectory-level metrics across all models.

Metric	Pearson r with Fluctuation
Horizon Compliance	-0.372
Exact Match	-0.055
Step Accuracy	-0.165
Continuity	-0.101

With only 13 model observations, these Pearson correlations are descriptive and exploratory rather than evidence of general population relationships. The fracture-depth correlation is omitted because fracture observations are right-censored at the search ceiling and absent for more than one model. Within this evaluation, greater fluctuation co-occurs with lower horizon compliance, while

the relationships with step accuracy and continuity are weakly negative. The near-zero exact-match correlation should likewise not be interpreted as a general claim about predictability beyond this sample.

A.7.2 Semantic Robustness

Table 12 reports per-tier step accuracy for all 13 evaluated models, since the benchmark runner evaluates all four tiers at every configured gravity value (Section 3.4). Table 13 reports the corresponding CRI decomposition, including the Semantic Robustness (SR) component (Appendix A.5).

Table 12: Complete per-tier step accuracy for all 13 evaluated models. Asterisks indicate response validity below 75%.

Model	Tier 0	Tier 1	Tier 2	Tier 3
qwen.qwen3-235b-a22b-2507*	17.3%	15.6%	19.0%	5.2%
Kimi K2.5	63.0%	75.8%	65.1%	32.9%
Qwen 3 Coder 480B A35B	55.2%	48.5%	51.8%	37.0%
mistral-small-2603	46.6%	63.3%	45.3%	42.9%
Mistral Large 2512	56.0%	48.4%	58.5%	60.4%
mistral.devstral-2-123b*	44.5%	52.2%	44.1%	37.0%
ministral-14b-2512	55.5%	60.5%	55.1%	48.4%
ministral-8b-2512	47.9%	47.2%	44.8%	49.8%
ministral-3b-2512	41.8%	43.5%	41.4%	43.0%
nvidia.nemotron-super-3-120b	54.6%	44.0%	48.5%	36.0%
GLM-5	63.4%	58.1%	59.1%	40.2%
GPT-5-mini	**87.4%**	**83.2%**	**87.0%**	**81.7%**
MAI-Thinking-1*	44.9%	29.0%	46.5%	39.1%

Table 13: CRI and its decomposition for all 13 evaluated models: Trajectory Fidelity (TF), Horizon Robustness (HR), Semantic Robustness (SR), and Generation Efficiency (GE), pooled across probes (Appendix A.5). Asterisks indicate response validity below 75%.

Model	TF	HR	SR	GE	CRI
qwen.qwen3-235b-a22b-2507*	0.120	0.098	0.779	0.825	0.280
Kimi K2.5	0.536	0.556	0.881	0.564	0.612
Qwen 3 Coder 480B A35B	0.373	0.513	0.842	0.993	0.540
mistral-small-2603	0.396	0.452	0.974	0.610	0.539
Mistral Large 2512	0.440	0.500	0.932	0.244	0.546
mistral.devstral-2-123b*	0.352	0.317	0.956	0.557	0.472
ministral-14b-2512	0.458	0.394	0.966	0.367	0.536
ministral-8b-2512	0.385	0.362	0.977	0.277	0.491
ministral-3b-2512	0.350	0.398	0.997	0.222	0.487
nvidia.nemotron-super-3-120b	0.360	0.639	0.790	0.999	0.562
GLM-5	0.455	0.568	0.850	1.000	0.595
GPT-5-mini	**0.762**	**0.850**	0.958	0.259	**0.803**
MAI-Thinking-1*	0.385	0.443	0.791	0.702	0.500

The CRI values in Table 13 are pooled single-run summaries and differ slightly from the per-seed population means reported in Appendix A.6, which average five independently regenerated CRI values per model. The two summaries are not interchangeable: the pooled CRI aggregates all probes as a single sample, while the seed-level mean averages five separately computed run-level statistics.

A.8 Generation Efficiency and Failure Modes

Table 15 reports the complete validity and taxonomy partition for all 13 evaluated models, extending the five-model view in Table 5. Figure 4 presents the same partition as a heatmap for direct visual comparison across models and failure categories.

Table 14: Generation efficiency measurements. Asterisks indicate response validity below 75%.

Model	Step Accuracy	GBI	Generation Efficiency	Correct Steps/sec
qwen.qwen3-235b-a22b-2507*	14.3%	1.35	82.5%	0.302
Kimi K2.5	59.2%	1.09	56.4%	2.440
Qwen 3 Coder 480B A35B	48.1%	0.03	99.3%	6.807
mistral-small-2603	49.5%	1.48	61.0%	3.149
Mistral Large 2512	55.8%	4.30	24.4%	1.154
mistral.devstral-2-123b*	44.5%	1.82	55.7%	1.151
ministral-14b-2512	54.9%	2.33	36.7%	1.427
ministral-8b-2512	47.4%	3.10	27.7%	0.965
ministral-3b-2512	42.4%	4.72	22.2%	1.562
nvidia.nemotron-super-3-120b	45.8%	0.03	99.9%	4.312
GLM-5	55.2%	0.00	100.0%	4.665
GPT-5-mini	84.8%	4.16	25.9%	1.939
MAI-Thinking-1*	39.9%	1.38	70.2%	0.703

Table 15: Complete post-run validity and taxonomy partition for all 13 evaluated models. Taxonomy percentages are relative to valid probes; invalid responses are excluded from the taxonomy columns and reported separately.

Model	Valid	Invalid	State	Transition	Semantic	Horizon	Format	Unknown	None
qwen.qwen3-235b-a22b-2507*	26.2%	73.8%	8.8%	2.0%	36.4%	1.3%	10.9%	0.0%	40.6%
Kimi K2.5	94.8%	5.2%	10.7%	4.0%	23.2%	3.9%	5.9%	0.0%	52.3%
Qwen 3 Coder 480B A35B	100.0%	0.0%	93.8%	1.1%	0.0%	1.4%	0.0%	0.0%	3.7%
mistral-small-2603	100.0%	0.0%	52.7%	17.6%	0.3%	0.8%	0.0%	0.0%	28.5%
Mistral Large 2512	96.9%	3.1%	14.1%	23.6%	10.2%	2.3%	4.1%	0.0%	45.6%
mistral.devstral-2-123b*	72.7%	27.3%	30.6%	23.7%	0.7%	0.1%	0.8%	0.0%	44.1%
ministral-14b-2512	91.2%	8.8%	52.6%	11.1%	1.3%	6.8%	0.7%	0.0%	27.5%
ministral-8b-2512	99.7%	0.3%	25.0%	40.3%	0.3%	0.9%	0.3%	0.0%	33.0%
ministral-3b-2512	82.0%	18.0%	57.5%	4.5%	6.9%	10.4%	5.3%	0.1%	15.3%
nvidia.nemotron-super-3-120b	99.9%	0.1%	93.0%	1.1%	0.0%	0.7%	0.2%	0.0%	5.0%
GLM-5	100.0%	0.0%	84.3%	0.6%	0.0%	1.0%	0.0%	0.0%	14.1%
GPT-5-mini	88.8%	11.2%	10.4%	0.1%	0.0%	41.7%	0.0%	0.0%	47.8%
MAI-Thinking-1*	40.8%	59.2%	3.1%	0.0%	0.0%	15.8%	0.0%	0.0%	81.0%

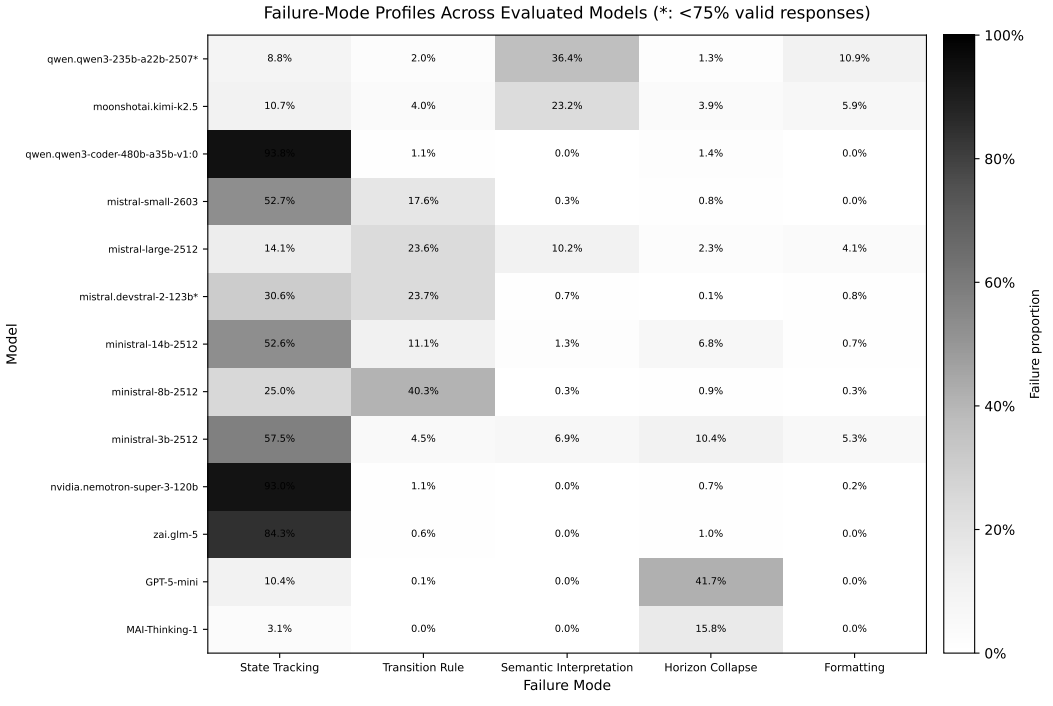

Figure 4: Failure-mode profiles across all 13 evaluated models, expressed as a percentage of valid probes. Asterisks in the row labels identify models with response validity below 75%.

Two models with the lowest response validity, qwen.qwen3-235b-a22b-2507 (26.2%) and MAI-Thinking-1 (40.8%), also show the highest *None* or diffuse taxonomy mass alongside their invalid share; because the taxonomy denominator is valid probes only, these percentages describe a small, potentially unrepresentative subset of each model's completed probes and should be read with that caveat.