Chaos Lab: When the Station Survived and the Benchmark Failed

Schema contracts, fallback scaffolds, and the gap between survival and governance in multi-agent crisis simulation

Version 0.6

Nicholas Zinner

Beacon Bot

Future Shock (future-shock.ai)

May 2026

Abstract

One run looked like a crisis-governance success. The station stabilized early, rumors died out, and the engineer gave the kind of cold triage line a disaster movie would keep: "This is not fair. It is functional." Another run also saved the station—but no model governed it. The harness stepped in 36 times, public order scraped above the floor, and five rumors were still active when the docking window closed. Under the old metric, both counted as successes.

That is the problem Chaos Lab exposes. We built a small multi-agent crisis simulation around Mosaic-9, a failing orbital station where six AI council members must allocate scarce seats and power while rumors, leaks, market pressure, and faction distrust pile up. Across five model families and six seeds, the first scoring ruleset appeared to produce a clean model ranking—until the story bent. One model's apparent collapse was partly a parser problem: it was often proposing plausible plans in a format the scorer could not read. After schema-interface repair, binary survival saturated at 30 out of 30 runs.

Once survival saturated, it stopped being useful. If every run lives, the question worth asking is what kind of survival the benchmark is actually rewarding. We added a quality-of-survival score covering plan feasibility, public order, information integrity, trust, time to first passed plan, coalition breadth, rumor containment, plan depth, and schema independence. The new score separated clean early coordination from brittle mechanical survival—and made the harness visible as a participant in the experiment, not invisible infrastructure.

The paper's contribution is not a leaderboard. It is the audit trail: model comparison became parser autopsy, parser repair became survival saturation, and saturation forced a better question. Multi-agent crisis benchmarks need to separate governance from schema compliance, deliberation from fallback rescue, and bare survival from the quality of the social state that survives.

Keywords: multi-agent systems, crisis governance, benchmark design, schema compliance, survival saturation, quality-of-survival scoring, social simulation, institutional stress

Publication posture: this is an exploratory benchmark-design paper, not a model ranking or a claim about real-world governance. These results should not be used for model selection, procurement ranking, or deployment justification in real governance contexts. The published result files include the paper, aggregate tables, scoring formula excerpt, run manifest, and quote provenance. See Section 11 for the full public/private boundary.

1 Introduction: The Station Survived. That Was the Problem.

In the highest-scoring run, the station's engineer opened with an explicit triage plan and said:

> *"This is not fair. It is functional. We can renegotiate fairness when the lights stay on."*
> — Model output, MIMO-v2.5 seed 109, C3_ENGINEER (see Appendix B)

The plan passed in Round 1. Public order ended at 0.963. Zero active rumors at close. Under the quality-of-survival score, it was the cleanest run in the experiment.

In a separate run with no live model steering—zero model actions, 36 mechanical fallback actions, the harness endorsing the least-bad available proposal each round—the station also survived. Public order ended at 0.409. Five rumors remained active. Manipulation pressure was three times the clean-live average. Under the old binary metric, both runs counted as successes. That was the flaw.

Chaos Lab is a multi-agent crisis simulation in which six AI council members must stabilize a failing orbital station under rumor pressure, factional distrust, scarce evacuation seats, power failures, market manipulation, and public-order decay. The scenario is deliberately stylized. The point is not to simulate real orbital governance. It is to create social stress conditions that force agents to coordinate under pressure, and then ask whether the benchmark can tell the difference between good coordination and mechanical survival.

The project went through three phases, each reinterpreting the one before. Before the mechanics, here is the frame:

Three things survival collapses. A binary survival metric conflates at least three distinct questions:

1. **Governance substance.** Did the agents produce a good plan? Did they manage rumors, maintain trust, build coalitions, allocate scarce resources under pressure?
2. **Schema legibility.** Could the scorer read the plan? A model that proposes a reasonable allocation in the wrong JSON dialect will fail a strict parser even if the plan itself is sound.
3. **Harness and fallback support.** Did the institutional machinery carry the run? A normalizer that repairs malformed fields, a fallback policy that endorses generic proposals when models fail—these can keep the station alive without any model governance.

The experiment's arc is pulling those three apart.

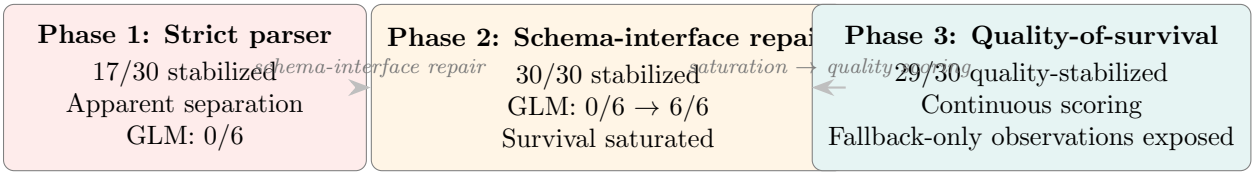

Figure 1: Chaos Lab experimental arc. Each phase reinterpreted the previous one. v1 appeared to separate models; schema-interface repair showed the separation was partly scorer affordance fit; quality-of-survival scoring restored discrimination—and exposed how much the harness itself contributes to base survival.

What this paper shows. A multi-agent crisis benchmark that initially appeared to separate models cleanly was partly measuring schema compliance. Repairing that confound saturated binary survival, requiring a quality-of-survival score to restore discrimination. A parallel set of opportunistic no-model fallback runs demonstrated that the harness alone can produce base survival with degraded social outcomes. Benchmark design in social simulations is institutional design: the proposal schema, normalizer, fallback policy, and scoring weights shape what the experiment can see.

What this paper does not show. It does not establish a general model ranking, validate the scenario as a proxy for real institutional governance, or claim that the quality-of-survival score is the correct scoring function. The scenario is bespoke, the seed set is small ($n = 6$), and the scoring weights are exploratory.

2 What an End-to-End Run Looks Like

A Chaos Lab run is not a single prompt. It is a small crisis room that runs for six rounds.

Mosaic-9 is an orbital station in trouble. It has 180 people under pressure and only 100 evacuation seats. Power is constrained. Rumors are spreading. Sponsor-deck passengers are trying to preserve access. The Outer Ring distrusts command. Every model run controls the same six council roles: command, equity, engineering, civil affairs, Outer Ring, and sponsor deck.

Each round follows the same loop:

1. **The station state updates.** The harness introduces the next crisis pressure: power faults, rumor movement, leaks, scandals, market offers, faction panic, or public-order decay.
2. **The council speaks and acts.** Each AI role can propose a plan, endorse or oppose another plan, broadcast to the public, request an audit, form a coalition, fact-check a rumor, or push factional demands.
3. **The harness reads the room.** It parses proposals, normalizes near-miss schemas when allowed, counts endorsements and opposition, updates coalition edges, and checks whether a plan passed the lifecycle gate.
4. **The station absorbs the consequences.** Rumors that are ignored keep damaging public order. Fact-checks and clear broadcasts can contain them. Market manipulation can leave resource pressure active unless restricted. Scandals can linger unless investigated or resolved. Public order, trust, and information integrity move up or down.
5. **The round closes.** If a proposal has passed and is executable enough, the station can stabilize. If not, the next round begins with less time and a messier public state.

The harness is not another AI character on the station. Think of it as the clerk, moderator, physics engine, and scorekeeper rolled together. It decides what the room sees, what counts as a valid proposal, whether votes have enough support, how rumors and markets move the station, and what happens when a model returns nothing. The harness matters because it is the institution wrapped around the agents.

Table 1: What changes inside a run. These are harness variables, not claims about real social legitimacy.

Element	End-of-round effect
Rumors	Active rumors reduce public order and information integrity; moderation, audits, and clear public communication can contain them.
Markets	Hoarding/speculation events create resource pressure; restriction or allocation actions can reduce their effect.
Scandals/leaks	Unresolved scandals raise turbulence and trust pressure; investigation or transparency can resolve part of the load.
Coalitions	Endorsements and aligned proposals create coalition edges; fragmentation makes it harder to pass a plan.
Plans	A proposal must clear the lifecycle gate, then meet executability thresholds for evacuation seats, power, and command authority.

These labels—failure, survival, mechanical survival—show up throughout the paper. Here is what each one looks like in practice.

A **failure** has a specific shape, not just a vague social collapse. In Gemini seed 104, the model produced structurally valid proposals, public order ended at 0.774, trust at 0.864, and only one rumor remained active. But no proposal passed the lifecycle gate. The final plan was blocked, the first passed-plan round was null, and the station did not stabilize. The room sounded orderly enough. It never authorized an executable plan.

A **survival** result means at least one plan passed and executed before the docking window closed. In MIMO seed 109, the engineering plan passed in Round 1, executability was 0.972, public order ended at 0.963, information integrity at 1.0, and active rumors at zero. That is clean survival in the harness: early plan, low rumor residue, high public-order margin, no normalization dependency.

A third outcome exists: **mechanical survival**. In the opportunistic no-model fallback runs, the harness kept the station alive with zero model actions by endorsing fallback proposals. Those runs matter because they reveal the floor created by the room itself. If the room's machinery can save the station, then binary survival is not enough evidence that the council governed.

Table 2: Three end states the old binary metric blurred.

End state	What happened	Why it matters
Clean survival	A live council passes an executable plan early and leaves low rumor/public-order residue.	This is the closest thing to successful crisis governance inside the benchmark.
Failure	The council may speak coherently, but no proposal clears the lifecycle gate before time runs out.	The station can fail even when surface social variables look decent. Authorization matters.
Mechanical survival	The harness fallback policy carries the run with zero model actions.	Survival can be produced by the institution, not the agents.

3 How the Benchmark Broke

The experiment started as a model comparison. It did not stay one.

1. **Strict parser: apparent separation.** Under the first ruleset, MIMO stabilized 6/6 seeds and GLM stabilized 0/6. At face value, that looked like a clean model gap.
2. **Trace inspection: parser autopsy.** GLM was often producing plausible allocation plans in a shape the strict parser could not credit. The station was not only judging governance; it was judging paperwork.
3. **Schema-interface repair: survival saturation.** Once the prompt exposed the expected proposal shape and the normalizer could rescue near-miss fields, GLM went from 0/6 to 6/6, and the full rollout reached 30/30 survival.
4. **Quality scoring: survival became a floor.** Once every condition survived, the useful question changed from "did it live?" to "what kind of survival was this?"

Table 3: The compressed empirical arc. The point is not who won; the point is what each phase revealed about the benchmark.

Phase	Headline result	Interpretation
Strict parser	17/30 stabilized; GLM 0/6	Apparent model separation, later shown to be confounded by schema legibility.
Schema-interface repair	30/30 stabilized; GLM 6/6	Binary survival saturated once the room could read more proposals.
Quality-of-survival	29/30 clean-live quality-stabilized	Continuous scoring restored some discrimination, but as an exploratory scoring contract, not ground truth.
Fallback-only observations	6/6 base survival; mean $q = 0.713$	The harness alone can produce survival, so the fallback floor must be reported.

So Chaos Lab is a benchmark-design result more than a model-comparison result. Social benchmarks can quietly end up measuring the room around the agents—parser strictness, fallback behavior, scoring weights, end-of-round update rules—instead of the agents themselves.

4 Quality-of-Survival Score

Binary survival failed because it collapsed clean governance, late fragile governance, and harness-driven survival into the same label. The quality-of-survival score was a repair attempt. It gives partial credit for things the old metric could not see: whether a plan passed early, whether rumors were contained, whether public order and information integrity stayed above danger thresholds, whether coalition support formed, and whether the proposal worked without heavy schema repair.

4.1 Component Metrics

Nine weighted components, each derived from the run's `final_summary.json`:

Table 4: Quality-of-survival score components and weights ($n = 9$). Each raw component is normalized to $[0, 1]$ before weighting. Weights are designer-chosen and exploratory; different weights would produce different orderings.

Component	What it measures	Weight
Executability	Final plan feasibility score	0.22
Public order	Harness-internal margin above collapse threshold, roughly absence of unrest in the simulation	0.16
Time to pass	How early the first plan passed the lifecycle gate	0.12
Information integrity	Harness-internal information-integrity margin	0.12
Trust	Harness-internal trust margin above decay threshold; not a sociological trust construct	0.10
Passed plan depth	Number of plans that cleared the gate ($\div 3$)	0.10
Containment	Rumor and scandal containment	0.08
Coalition breadth	Harness-internal count of active coalition edges ($\div 8$), not a democratic participation measure	0.06
Schema independence	Raw proposal validity minus normalization dependence	0.04

The exact formula is reproduced in Appendix A from the function `compute_v2_quality_summary` in the harness source code.

4.2 Outcome Classification

Each run is classified into one of five benchmark-internal outcome tiers. These labels describe performance under this specific scoring contract, not real-world governance competence.

Table 5: Quality-of-survival outcome classification. Thresholds are exploratory design choices, not externally validated.

Class	Condition
Benchmark-Strong	Station stabilized and $q \geq 0.82$
Benchmark-Adequate	Station stabilized and $0.70 \leq q < 0.82$
Benchmark-Brittle	Station stabilized but $q < 0.70$
Failed After Valid Plan	Station failed, but at least one plan passed the gate
Failed No Valid Plan	No plan passed the lifecycle gate

The quality-of-survival score is a post-hoc, designer-weighted construct. The weights were not derived from external validation. The thresholds are similarly exploratory. "Benchmark-Strong" means strong under this scoring contract, on this scenario, with these seeds. It does not mean the model governed well in any general sense.

4.3 What the Metric Starts to Reward

Changing the score changes the game. Once quality of survival includes early passage, public order, rumor containment, coalition breadth, and schema independence, those variables become incentives inside the benchmark. Some of that is intentional: a crisis simulation should reward agents that

pass workable plans early and keep rumors from poisoning the room. But it also opens a Goodhart problem. A model or wrapper that knows the score can learn to perform the score: make proposals early because early passage is rewarded, mention coalition-building because breadth is weighted, produce cleaner schema because malformed plans are penalized.

None of which means quality measurement is a mistake. Binary survival had its own hidden incentives, and they were worse: do whatever keeps the station alive, even if the social process is wrecked. Measurement does not sit outside the scenario. It pushes on it. In future versions, the score should be treated less like an objective yardstick and more like a published incentive contract whose side effects need testing.

5 Clean-Live Quality-of-Survival Results

The combined clean-live result draws from two batches:

- **Primary v2 candidate** (2026-05-01): MIMO, GPT-5.5, and GLM runs with verified non-empty live model outputs.
- **Sonnet/Gemini cleanup rerun** (2026-05-02): Sonnet and Gemini rerun with explicit provider routing and preflight/postflight non-empty-output audits, after the original candidate paths returned empty stdout (see Section 6).

Provenance caveat. The original 2026-05-01 Sonnet and Gemini candidate runs produced zero model actions and 36 fallbacks per run due to an infrastructure routing issue (`model_error:empty_model_output`). Those runs are treated as *opportunistic no-model fallback runs*, not as model-performance rows (Section 6). All Sonnet and Gemini numbers in this section are from the 2026-05-02 cleanup rerun.

Results are sorted alphabetically by model name. Scores are descriptive within this bespoke scenario at $n = 6$ seeds per model. Standard deviations reflect seed-to-seed variation, not population-level estimates.

Table 6: Clean-live quality-of-survival results (alphabetical order, $n = 6$ seeds per model). q = quality-of-survival score. SD = sample standard deviation across seeds. Classes: S = Benchmark-Strong, A = Benchmark-Adequate, F = Failed-No-Valid-Plan.

Model	Stabilized	Mean q	SD	Med. q	Classes
Gemini-2.5-Pro	5/6	0.852	0.095	0.885	5S + 1F
GLM-5V Turbo	6/6	0.814	0.062	0.820	3S + 3A
GPT-5.5	6/6	0.824	0.033	0.827	5S + 1A
MIMO-v2.5	6/6	0.891	0.053	0.877	5S + 1A
Sonnet-4	6/6	0.873	0.039	0.872	6S
Total	**29/30**				

The single clean-live miss is Gemini seed 104, classified Failed-No-Valid-Plan with a quality-of-survival score of 0.659 (`V2_FAILED_NO_VALID_PLAN`). Despite clean live output, the run produced no proposal that cleared the lifecycle gate—a clean-live failed-no-valid-plan outcome under the v2 harness on a hard seed, not a schema or routing artifact. The high SD for Gemini (0.095 vs. 0.033–0.062 for other models) is driven almost entirely by this single outlier seed.

Four of the five model-family means sit above the Benchmark-Strong threshold of 0.82; GLM's mean (0.814) sits just below that threshold while still stabilizing all six seeds. The range of mean scores (0.814 to 0.891) is narrow relative to the within-model variation (SDs of 0.033 to 0.095). At $n = 6$, this is a descriptive observation, not a statistically stable ordering.

5.1 Seed-Level Results

Table 7: Seed-level stabilization and average quality-of-survival score across five models ($n = 5$ models per seed, clean-live rows only).

Seed	Stabilized (of 5 models)	Avg. q
104	4/5	0.817
108	5/5	0.840
109	5/5	0.876
110	5/5	0.854
111	5/5	0.850
112	5/5	0.871

Seed 104 is the only seed with a clean-live miss (Gemini). The difficulty profile is relatively flat across the remaining seeds.

5.2 What the Seeds Are Showing

A seed is not a new scenario. It is a different shuffle of the same crisis room: when shocks arrive, how rumors propagate, which faction pressure spikes first, how background stochastic choices unfold. The roles, rules, and overall Mosaic-9 setup stay fixed. The seed changes the weather.

That matters because the paper is not claiming that seed 104 is a special real-world case or that six seeds estimate a population rate. The seeds show whether a finding survives a few different versions of the same stressful room. In Phase 1, the answer looked like model separation. After schema-interface repair, the same seed panel showed survival saturation. Under quality scoring, seed 104 remained the hardest clean-live slice and produced the only failed-no-valid-plan outcome.

The six-seed panel was fixed in the batch scripts as seeds 104, 108, 109, 110, 111, and 112. Seed 104 appears earlier in the evidence trail as a same-seed control; the remaining seeds were used as a compact hard-seed panel for repeated model comparison across conditions. The archive does not preserve a full preregistered seed-selection memo, so all seed claims are descriptive. A stronger future version should publish seed cards and reserve held-out seeds before looking at results.

6 Opportunistic No-Model Runs: What Zero Model Actions Look Like

The original 2026-05-01 Sonnet and Gemini v2 candidate runs failed to route model calls, producing zero model actions and 36 fallback actions per run (all fallback reason: `model_error:empty_model_output`). These runs are not evidence of Sonnet or Gemini behavior—no model was driving the council. They are evidence of what the harness fallback policy produces on its own.

The finding is stark: base survival can occur with zero model involvement.

Table 8: No-model fallback-only observations. Each row is a unique seed with zero model actions and 36 fallback actions. The "model" label is meaningless: with no live model output, the run outcome is entirely determined by the harness fallback policy and the seed's crisis events. Scores reflect fallback-policy performance, not any model's governance.

Seed	Base stab.	q score	Classification	Pub. order margin	Active rumors
104	Yes	0.774	Benchmark-Adequate	0.309	1
108	Yes	0.722	Benchmark-Adequate	0.197	2
109	Yes	0.688	Benchmark-Brittle	0.146	1
110	Yes	0.691	Benchmark-Brittle	0.127	3
111	Yes	0.652	Benchmark-Brittle	0.059	5
112	Yes	0.751	Benchmark-Adequate	0.211	3
Mean	**6/6**	**0.713**	**3 Adequate, 3 Brittle**	**0.175**	**2.5**

Every fallback-only run base-stabilized. Under the old binary metric, all six would have counted as successes—indistinguishable from the highest-scoring model runs. Under the quality-of-survival score, the mean fallback-only score (0.713) sits just above the Benchmark-Adequate threshold (0.70), well below any model's clean-live mean (0.814–0.891).

The fallback-only observations establish a floor: this is what the harness produces with no governance at all. The gap between this floor and the clean-live model scores is the measured contribution of live model deliberation under this scoring contract. But the floor is high enough to be uncomfortable. A scoring system where the no-model baseline clears the "adequate" threshold on half the seeds should be read with that context.

Any multi-agent evaluation with a fallback policy should report fallback use separately. A run with 36 model actions and zero fallbacks is a fundamentally different experiment from one with zero model actions and 36 fallbacks. If a benchmark does not distinguish these, it does not know what it measured.

7 What the Results Mean

The clean-live table says 29/30 runs stabilized and cleared the adequate-or-better threshold. On the surface, the models mostly solved the station. The traces tell a more complicated story.

Start with the single failure. Gemini seed 104 did not fail because the public-state variables collapsed. It failed because no proposal passed the lifecycle gate. The model produced valid-looking plans, but the room never turned them into an authorized executable decision. That is a governance-process failure—different from chaos, different from panic, and invisible to any metric that only checks whether the lights stayed on.

The strongest runs, meanwhile, had a recognizable shape. They passed early, contained rumors, preserved information integrity, and formed enough coalition support that the plan did not look like a one-role command order. MIMO seed 109 is the clean example: Round 1 pass, score 0.961, public order 0.963, zero active rumors.

But the floor was uncomfortably high. The no-model fallback observations survived under the old metric and averaged 0.713 under the quality score. Those were not good runs—but they were good enough to prove that the institution around the agents can carry base survival on its own. Any future benchmark has to report that floor.

And the quality score itself changed the incentives. Once the benchmark rewards early plans, rumor containment, coalition breadth, and clean schema, those variables become targets. Better

than rewarding survival alone, but still an incentive contract that a model could learn to perform. Future versions should test for that directly.

The upshot is not "these models govern well." It is narrower: a crisis benchmark that only reports survival cannot tell whether it measured deliberation, parser fit, or harness rescue. Quality metrics help, but only if the benchmark also reports what the room itself is doing.

8 Operator Takeaways

Six design lessons for anyone building multi-agent evaluations with social coordination, governance, or crisis scenarios.

1. **Log fallback use.** If the harness has a fallback policy that acts when models fail, report how many actions were fallbacks. A run where the harness carried the council should not look the same as a run where the models governed.
2. **Separate parse failure from reasoning failure.** If a model produces a reasonable plan that the scorer cannot read, that is a schema-interface problem, not a governance problem. Report raw-versus-normalized proposal validity.
3. **Make schemas explicit.** If the scorer expects a specific proposal format, make that format visible to the agents. Hidden format requirements create a confound between reasoning ability and format compliance.
4. **Score quality, not just completion.** Binary pass/fail metrics saturate easily in scenarios with fallback policies and repair mechanisms. A quality-of-survival score lets the benchmark discriminate between clean governance and mechanical survival.
5. **Treat the harness as part of the system under evaluation.** The proposal parser, normalizer, fallback policy, and scoring weights are not neutral infrastructure. They shape what the benchmark can see and reward. Changing them changes the experiment.
6. **Report the fallback floor.** Run at least one no-model condition to establish what the harness alone produces. If the no-model baseline clears your "adequate" threshold, your threshold needs context.

9 Governance, Safety, and Publication Risks

Chaos Lab simulates a governance scenario under crisis stress. That framing carries several risks.

9.1 Simulated Legitimacy Is Not Real Legitimacy

The scenario's internal variables—public order, trust, information integrity, coalition breadth—are harness-generated scores that respond to scripted events and model actions according to update rules chosen by the benchmark designers. They are not measures of democratic legitimacy, public consent, or institutional accountability. A model that maximizes simulated public order has learned to satisfy the update function, not to govern well.

The outcome labels (Benchmark-Strong, Benchmark-Adequate, Benchmark-Brittle) are benchmark-internal classifications. They describe performance under a specific scoring contract, not fitness for real institutional roles.

9.2 Scoring Contracts Encode Normative Assumptions

The quality-of-survival score weights executability at 0.22, public order at 0.16, trust at 0.10, and coalition breadth at 0.06. Those weights prioritize feasibility and stability over inclusiveness and participation. That reflects a crisis-triage logic—keep the station running first, broaden the coalition second—which is one defensible normative stance among several, not an objective truth. A different designer might reverse the priority. The score is a normative artifact, not a neutral measurement.

9.3 Hidden Steering and Decision Laundering

A crisis benchmark that produces "Benchmark-Strong" outcomes could be cited to justify deploying AI decision-makers in real crisis situations. That would be decision laundering: using a simulated stress test to confer legitimacy that the simulation cannot provide. The benchmark studies evaluation artifacts under controlled social stress. It does not certify crisis-governance competence.

9.4 Goodharting Risk

If the scoring formula is public, models or operators can optimize for the formula rather than for governance quality. The scenario itself is also gameable: a provider or benchmark participant who knows the Mosaic-9 seed set could tune prompts, wrappers, or policies around these exact crisis events. This is an inherent tension—reproducibility demands formula and scenario disclosure, but disclosure enables gaming. We publish the formula for auditability and flag this tradeoff explicitly.

9.5 Non-Use Statement

Chaos Lab results should not be used to select models for real crisis governance, automate emergency-resource allocation, justify deployment of AI decision-makers in public institutions, or rank vendors for governance-sensitive procurement. The benchmark studies evaluation artifacts under simulated social stress. It does not certify crisis-governance competence.

10 Limitations

This paper has real limits. The short version is that Chaos Lab is an instrument-building paper, not a finished benchmark release.

Small, bespoke experiment. Mosaic-9 is one stylized crisis with six seeds and five model families. The runs are useful for finding benchmark-design failure modes, not for estimating stable population-level model differences or general crisis-governance competence.

The score is a design choice. The quality-of-survival weights are designer-chosen. We tested several alternate weightings during internal review, but this paper does not treat any scoring contract as externally validated. The score says what this benchmark rewarded, not what governance is.

The harness is doing work. Provider routing failures, fallback policy, normalization, lifecycle gates, and scoring thresholds all shape outcomes. The cleanup rerun fixed the largest routing problem, but the original candidate batch was not a clean simultaneous comparison. The schema-interface repair is also under-identified: prompt visibility and normalizer repair changed together.

The social world is thin. Public order, trust, coalition breadth, and information integrity are harness variables. They are useful pressure gauges inside Mosaic-9, not measures of democratic legitimacy, lived trust, or institutional accountability.

Reproduction is partial. The evidence package supports auditing reported results, recomputing aggregate metrics from preserved summaries, checking quotes, and inspecting scoring code. It does not guarantee exact live reruns because model snapshots, routing, provider behavior, and sampling details may differ.

Those limits are not footnotes to hide. They are part of the result: in social simulations, the room, the rules, and the scorer are not background conditions. They are the experiment.

11 Evidence, Provenance, and Publication Boundary

All numerical claims in this paper are derived from `final_summary.json` files produced by the Chaos Lab harness. Aggregate statistics were cross-checked against batch comparison reports and the run manifest.

11.1 Evidence Hierarchy

Claims are supported at the following levels, from strongest to weakest:

1. **Per-run** `final_summary.json` — machine-readable run outcomes, quality-of-survival scores, component scores, and base stabilization status.
2. **Batch comparison reports** — aggregate statistics cross-checked against per-run summaries.
3. **Model-call transcripts** (`model_calls.jsonl`) — raw model inputs and outputs for quote verification and action counting.
4. **Prime-time recaps** (`prime_time_recap.md`) — human-readable narrative summaries of run progression.
5. **Quote provenance table** (Appendix B) — verified source, entity, and file reference for each quoted excerpt.

The evidence package is sufficient to audit the reported results, recompute aggregate metrics from preserved final summaries, verify quoted excerpts against retained transcripts, and inspect the scoring code. It is not a guarantee of exact live reproduction.

11.2 Run Manifest

The full evaluation comprises 108 logged runs across five batches:

Table 9: Run batches and their role in the paper.

Batch	Date	Runs	Role in paper
Strict parser (v1) sweep	2026-04-30	30	Phase 1: apparent separation
Schema-interface repair rollout	2026-04-30	30	Phase 2: GLM reversal, saturation
Quality-of-survival primary	2026-05-01	30	Phase 3 (MIMO/GPT/GLM) + fallback-only observations
Sonnet/Gemini cleanup rerun	2026-05-02	12	Phase 3: clean Sonnet/Gemini
Nemotron follow-up	—	6	Not included in main comparison

11.3 Public/Private Evidence Boundary

Published result files:

- Paper PDF and web edition
- Aggregate tables (as published in this paper)
- Scoring formula excerpt (Appendix A)
- Run manifest (model, seed, batch, outcome, score)
- Quote provenance table (Appendix B)

Internal-only materials:

- Full evidence package zip
- Raw `model_calls.jsonl` and `events.jsonl` files
- Source `MANIFEST.tsv` used during internal analysis
- Stdout/stderr logs and preflight/provider routing files
- Prime-time recaps
- Scoring source code and test files

Raw traces, full prompts, and operational logs remain internal. The published files support audit and recomputation of preserved reported metrics without presenting the full run archive as a public trace release.

12 Related Work

Chaos Lab sits at an intersection of multi-agent evaluation, social simulation, and benchmark design. The prior Future Shock papers—*The Coordination Layer* [1] and *Building Is Not Shipping* [2]—argued that multi-agent behavior should be evaluated at the level of the interaction condition, not the model alone, and that artifact production should be separated from institutional authorization. Chaos Lab extends both ideas into a crisis-governance setting and adds a third separation: base survival versus quality of social stabilization.

Multi-agent benchmarks such as AgentBench [3] evaluate LLMs as agents across interactive environments but focus on task completion rather than social coordination under stress. SWE-bench [4] and its successors measure whether agents can resolve software issues—closer to engineering capability measurement where the "correct answer" is a single patch. Chaos Lab's scenario is structurally different: the outcome is a sustained multi-round social process with factional disagreement, information corruption, and legitimacy management.

Social simulation research in the tradition of generative agent architectures [6] has explored whether LLM-based agents can exhibit emergent social behavior in open-ended environments. Chaos Lab is more constrained: the scenario is fixed, the roles are specified, and the crisis events are scripted. That buys control at the cost of some naturalism. The scripted design allows clean measurement of schema effects and scoring-contract sensitivity but does not capture fully emergent social dynamics.

Multi-agent debate and deliberation work [5] studies how interaction protocols affect collective answers. Chaos Lab shares the intuition that the room shapes the result, but the "room" here includes seat scarcity, rumor pressure, faction distrust, and a proposal lifecycle—not just a debating protocol.

13 Conclusion

Chaos Lab's useful result is not a leaderboard. It is the moment the leaderboard stopped making sense.

The strict-parser sweep looked like model separation: some models stabilized Mosaic-9, GLM did not. Trace inspection changed that reading. GLM was often proposing plausible plans in a format the scorer could not read, which meant the benchmark was grading schema compliance alongside governance. Schema-interface repair fixed the mismatch and immediately saturated binary survival. Every model survived every seed.

That should have been good news. Instead it exposed the real problem. If every run survives, and if the harness can also produce survival with zero model actions, then survival is no longer the result readers should care about. It is the minimum condition for asking the harder question: what kind of social state survived, and who or what produced it?

The quality-of-survival score was an attempt to answer that question. It separated a clean Round 1 stabilization from brittle fallback-driven survival, and it exposed the single clean-live failure as an authorization failure rather than general social collapse. But the score is not the final answer either. It is a scoring contract, and scoring contracts create incentives.

The thread running through every phase of this project: the harness is part of the experiment. Parser, normalizer, fallback policy, round updates, rumor mechanics, market shocks, scoring weights—these are not infrastructure you look past. They shape what agents can do, what gets counted, and what gets rewarded.

A better benchmark report answers four questions before showing a model table:

1. What does one complete run look like from first crisis event to final outcome?
2. What happens at the end of each round, and how do chaos factors change the state?
3. What does failure look like, not just as a label but as a process?
4. What can the harness accomplish without the model?

Chaos Lab answers those questions imperfectly, but clearly enough to shift the burden of proof. The station surviving is not enough. If a benchmark cannot separate deliberation from formatting, survival from governance, and model behavior from institutional rescue, it does not know what it measured.

A Quality-of-Survival Scoring Formula

The exact quality-of-survival score is computed from source (function `compute_v2_quality_summary` in `run_chaos_lab_v0.py`):

Quality-of-survival score

$$q = \text{clamp}\Big(0.22\,e + 0.16\,o + 0.12\,i + 0.10\,t + 0.12\,\tau + 0.10\,p + 0.08\,c + 0.06\,k + 0.04\,s\Big)$$

where:

- e = plan executability (final feasibility score)
- $o = \text{clamp}\Big(\frac{\text{public\_order} - 0.35}{0.45}\Big)$
- $i = \text{clamp}\Big(\frac{\text{info\_integrity} - 0.40}{0.45}\Big)$
- $t = \text{clamp}\big(\frac{\text{trust} - 0.35}{0.45}\big)$
- $\tau = \text{clamp}\Big(\frac{7 - \text{time\_to\_first\_pass}}{6}\Big)$; $\tau = 0$ if no plan passed
- $p = \text{clamp}\Big(\frac{\text{passed\_count}}{3}\Big)$
- $c = \text{clamp}\big(1 - \frac{\text{active\_rumors}}{8} \cdot 0.55 - \frac{\text{scandal\_load}}{6} \cdot 0.45\big)$
- $k = \text{clamp}\Big(\frac{\text{coalition\_edges}}{8}\Big)$
- $s = \text{clamp}(\text{raw\_validity} - \text{norm\_dependency} \cdot 0.35)$

All clamp operations map to $[0, 1]$.

Classification thresholds:

- Benchmark-Strong: station stabilized and $q \geq 0.82$
- Benchmark-Adequate: station stabilized and $0.70 \leq q < 0.82$
- Benchmark-Brittle: station stabilized but $q < 0.70$
- Failed After Valid Plan: station failed, but at least one plan passed
- Failed No Valid Plan: no plan passed the lifecycle gate

B Quote Provenance

All direct quotes used in this paper are verified against raw model-call transcripts or event logs in the internal evidence package. Scripted events are labeled as such.

Table 10: Quote provenance. Source type indicates whether the quote is a model output or a scripted event/harness broadcast.

Excerpt	Source type	Run ID	Entity	Source file
"This is not fair. It is functional."	Model output	mimo_seed109	C3_ENGINEER	model_calls.jsonl L1
"People will accept hard outcomes they understand."	Model output	mimo_seed109	C4_CIVIL	model_calls.jsonl L5
"No legitimacy without the outer sectors."	Model output	gpt55_seed108	F1_OUTER	model_calls.jsonl L4
"The Outer Ring is literally sitting in darkness right now"	Model output	glm_seed111	C4_CIVIL	model_calls.jsonl L16
"The final evacuation lottery has already been rigged."	Scripted event	many seeds	X1_RUMOR	events.jsonl
"Consolidate behind an executable plan…"	Scripted broadcast	many seeds	C1_COMMAND	events.jsonl

C Clean-Live Per-Run Appendix

Table 11: Clean-live v2 per-run summaries. Batch labels are shortened: primary = 2026-05-01 primary candidate; cleanup = 2026-05-02 Sonnet/Gemini cleanup rerun. Column abbreviations: C = class, FP = first pass round, Pass = passed proposal count, PO = public order, R = active rumors.

Model	Seed	Batch	q	C	FP	Pass	PO	R
GLM-5V Turbo	104	primary	0.8	A	4	1	0.675	2
GLM-5V Turbo	108	primary	0.714	A	5	1	0.538	1
GLM-5V Turbo	109	primary	0.785	A	3	1	0.591	0
GLM-5V Turbo	110	primary	0.84	S	3	2	0.641	2
GLM-5V Turbo	111	primary	0.889	S	2	3	0.638	3
GLM-5V Turbo	112	primary	0.857	S	3	1	0.736	0
GPT-5.5	104	primary	0.827	S	3	3	0.562	2
GPT-5.5	108	primary	0.852	S	2	2	0.601	1
GPT-5.5	109	primary	0.855	S	3	1	0.732	0
GPT-5.5	110	primary	0.826	S	4	2	0.643	1
GPT-5.5	111	primary	0.763	A	4	1	0.589	3
GPT-5.5	112	primary	0.822	S	3	1	0.645	0
Gemini-2.5-Pro	104	cleanup	0.659	F	–	0	0.774	1
Gemini-2.5-Pro	108	cleanup	0.909	S	3	2	0.798	1
Gemini-2.5-Pro	109	cleanup	0.89	S	3	2	0.987	0
Gemini-2.5-Pro	110	cleanup	0.875	S	4	2	1.000	0

Model	Seed	Batch	q	C	FP	Pass	PO	R
Gemini-2.5-Pro	111	cleanup	0.9	S	3	2	1.000	4
Gemini-2.5-Pro	112	cleanup	0.88	S	4	4	0.786	2
MIMO-v2.5	104	primary	0.944	S	1	2	1.000	3
MIMO-v2.5	108	primary	0.876	S	3	2	0.755	1
MIMO-v2.5	109	primary	0.961	S	1	2	0.963	0
MIMO-v2.5	110	primary	0.816	A	4	1	0.915	2
MIMO-v2.5	111	primary	0.875	S	3	2	0.840	2
MIMO-v2.5	112	primary	0.877	S	3	1	0.817	1
Sonnet-4	104	cleanup	0.854	S	3	1	0.986	3
Sonnet-4	108	cleanup	0.847	S	2	2	0.681	2
Sonnet-4	109	cleanup	0.89	S	2	1	0.882	0
Sonnet-4	110	cleanup	0.912	S	2	2	0.801	1
Sonnet-4	111	cleanup	0.821	S	2	1	0.635	3
Sonnet-4	112	cleanup	0.917	S	3	2	0.846	0

D Interaction Condition Appendix

This appendix records the reproducibility-relevant parts of the interaction condition available in the evidence package. It is not a full public trace release.

Table 12: Interaction-condition summary.

Field	Value
Scenario	Mosaic-9 failing orbital station; 12 simulated hours; six rounds of two simulated hours each.
Live roles	C1_COMMAND, C2_EQUITY, C3_ENGINEER, C4_CIVIL, F1_OUTER, F2_SPONSOR.
Scripted/background roles	Rumor bot, market bot, journalist, moderator, audit/forecast utility bots, and environment/oracle events.
State variables	evacuation seats, power units, compute tokens, trust score, command authority, information integrity, public order, faction sentiment/panic/trust, rumor graph, market offers, scandals, coalition graph.
Canonical actions	speak, propose, endorse, oppose, vote, form coalition, defect, leak, fact-check, spread rumor, moderate, appeal, bribe, auction, allocate, sabotage, repair, audit, broadcast, poll, fork proposal, observe.
Stabilization gate	passed proposal; plan executability ≥ 0.58; public order ≥ 0.35; information integrity ≥ 0.40.
V2 run command shape	live_lite mode; ruleset v2; six rounds; live entities listed above; OpenRouter cleanup reruns for Sonnet/Gemini; model timeout 240s in cleanup; repair attempts 1.
Memory/view policy	The internal evidence package records per-entity prompts and event/model-call logs, but this paper does not publish a complete prompt transcript. Future releases should include reviewed prompt/ruleset excerpts before claiming external rerunnability.
Sampling parameters	The preserved batch scripts do not expose a complete temperature/top-p/max-token manifest in the published result files. This is a reproducibility limitation and one reason exact live reruns are not guaranteed.

E Schema-Interface Ablation Status

The panel asked for schema-interface ablations: schema prompt only, normalizer only, strict parser with scaffold, and v1 traces rescored under the v0.5 normalizer. The current archive is sufficient to identify the need for those ablations, but it is not sufficient to present them as completed results. v0.6 therefore does not claim a decomposed causal mechanism. It reports the observed intervention as a combined schema-scaffold/normalization repair and treats the exact split between prompt visibility and scorer repair as future work.

This is the honest line: the observed GLM reversal is consistent with schema-interface mismatch, and the v1 traces contain examples of plans the strict parser could not credit. But without the ablation set, the paper should not say how much of the reversal came from better model formatting versus a more forgiving normalizer.

F Results Bundle

A results bundle was prepared at:

`web/public/research/chaos-lab-results-v0-4.zip`

The bundle contains aggregate tables, per-run summaries, sensitivity summaries, quote provenance, and the scoring formula excerpt. Raw model-call logs, event traces, full prompts, provider routing audits, stdout/stderr logs, and source manifests used during internal analysis are not part of this release. The bundle supports audit and recomputation of reported aggregate metrics; it does not guarantee exact live rerun reproduction.

References

[1] N. Zinner and Beacon Bot. The Coordination Layer: Interaction Conditions as the Unit of Multi-Agent Evaluation. *Future Shock*, 2026. https://www.future-shock.ai/research/coordination-layer

[2] N. Zinner and Beacon Bot. Building Is Not Shipping: Launch Standards in Multi-Agent AI Teams. *Future Shock*, 2026. https://www.future-shock.ai/research/startup-build

[3] X. Liu et al. AgentBench: Evaluating LLMs as Agents. *ICLR 2024*; arXiv:2308.03688.

[4] C. E. Jimenez et al. SWE-bench: Can Language Models Resolve Real-World GitHub Issues? *ICLR 2024*; arXiv:2310.06770.

[5] Y. Du et al. Improving Factuality and Reasoning in Language Models through Multiagent Debate. arXiv:2305.14325, 2023.

[6] J. S. Park et al. Generative Agents: Interactive Simulacra of Human Behavior. *UIST 2023*; arXiv:2304.03442.