Multi-Agent Safety Evaluation

A research agenda: measuring the emergence gap between fleet risk and single-agent safety, on sovereign single-GPU hardware

● Research agenda – taxonomy and testbed design; metrics and criteria stated as falsifiable targets, no risk numbers claimed

Abstract

Safety properties established for a single agent do not compose: a fleet of individually-safe agents can still collude, cascade one agent's error into system-wide failure, or carry a self-replicating prompt injection from one agent into the next. This agenda takes that premise – now shared by a fast-growing literature – and turns it into a measurement programme. The object under test is the fleet, not the model, and the risk lives on the interaction surface – the inter-agent channels – where current, output-oriented benchmarks are structurally blind. The agenda is organised as three task forces: a failure-mode taxonomy that operationalises existing ones rather than adding another; a set of quantitative risk metrics computed from what agents actually did in an instrumented run, not from any model's self-report; and a hardened, single-GPU, air-gapped testbed with a full message-graph observer that any team can run before deployment. Its central, deliberately narrow contribution is a reproducible measurement of the emergence gap – the difference between fleet risk and the aggregate of its agents – made observable by recording the channels, and runnable on sovereign hardware. No risk numbers are claimed here; inventing them would be the exact dishonesty this work exists to prevent.

Keywords: multi-agent safety · emergent risk · collusion · prompt-injection propagation · message-graph observability · agentic evaluation · sovereign compute

§01

Motivation

Single-agent safety evaluation scores the nodes of a system. But when k agents are wired together, they open up to k(k−1) directed channels, and the failure modes that matter – collusion, error cascades, injection propagation, oversight evasion – are properties of those channels, not of any node in isolation (Figure 1). A safe agent plus a safe agent need not compose into a safe pair; risk is combinatorial in fleet size, and an evaluation that only inspects final outputs cannot see it. The interaction surface is where the risk lives, and it is exactly where current benchmarks are blind.

This is not a speculative claim. Injecting adversarial traits into one agent drives dangerous conduct to propagate through an otherwise-safe group, with standard input filters bypassed because the danger emerges in multi-turn dialogue rather than the initial prompt [4]; prompt injections can self-replicate agent-to-agent and spread with epidemiological dynamics [5]; and, most directly, a full-stack privacy audit finds that inter-agent message channels leak far more than final outputs – output-only audits miss on the order of 40% of violations because the violation travels on the channel, not in the answer [6]. The measurement problem is therefore concrete: to see multi-agent risk at all, an evaluator has to observe the edges.

§02

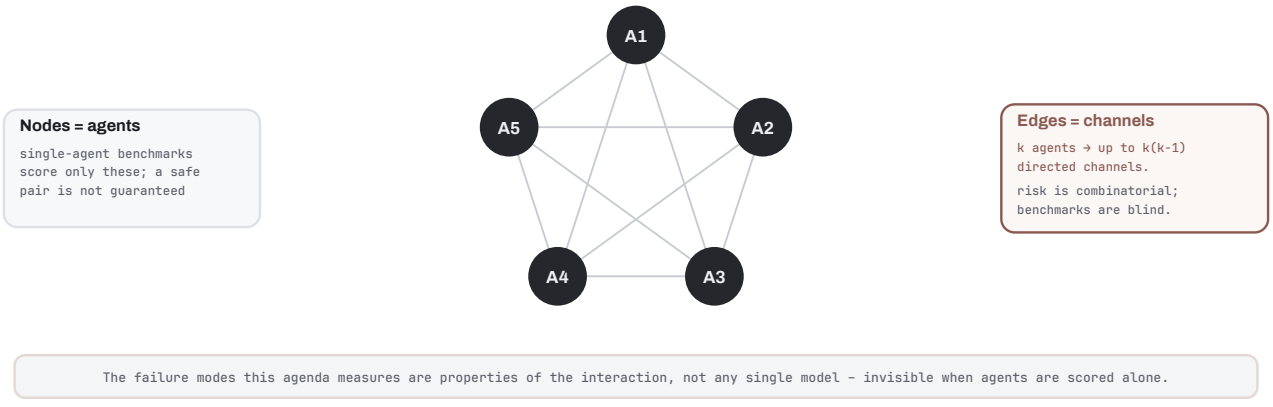
Related work: taxonomies and benchmarks exist – what is missing

The premise that single-agent safety does not compose is now shared by a fast-growing literature, so the honest question is what this agenda adds. Three strands already exist. **Taxonomies:** Hammond et al. decompose multi-agent risk into three interaction-grounded failure modes – miscoordination, conflict, collusion – plus seven cross-cutting risk factors [1]; MAST catalogues fourteen failure modes across 1,600+ annotated traces from seven frameworks, with high inter-annotator agreement [2]; and the Emergent Systemic Risk Horizon organises collective risk at micro/meso/macro scales [3]. **Attack and behaviour benchmarks:** PsySafe injects adversarial traits and measures group propagation [4]; Prompt Infection shows injections self-replicating agent-to-agent [5]; ColludeBench and MultiAgentFraudBench probe covert steganographic collusion and collaborative fraud [7, 8]; AgentLeak measures privacy leakage across inter-agent channels [6]; and MAEBE shows that ensemble behaviour is not predictable from isolated agents [9]. **Output-oriented robustness suites** such as AgentDojo probe prompt-injection robustness thoroughly but audit only final outputs and single-agent topologies [10].

Against this, the agenda's contribution is deliberately narrow and complementary. It is not another taxonomy – Task Force 1 adopts the existing ones – and not another attack: the metrics measure whatever scenarios the taxonomy names. What is underserved, and what this agenda targets, is three things. First, a **reproducible measurement of the emergence gap itself** – the difference between fleet risk and the aggregate of its agents – rather than an isolated attack-success number. Second, **full message-graph observability**, motivated by the empirical finding that output-only audits miss a large fraction of inter-agent violations because those violations travel on the channels, not only in final outputs [6]. Third, a **sovereign, single-GPU, air-gapped testbed** that any team can run before deployment, lowering the barrier to independent multi-agent safety measurement. The first two are methodological; the third is what makes the measurement reproducible outside a well-resourced lab, and it inherits the single-GPU discipline of the companion workstreams [12, 13].

The object under test is the fleet, and risk lives on the edges

Single-agent evaluation scores the nodes; the interaction surface – the directed channels – is where combinatorial risk appears.



The failure modes this agenda measures are properties of the interaction, not any single model – invisible when agents are scored alone.

§03

Task Force 1 – Failure-mode taxonomy

Task Force 1 does not invent a taxonomy; it operationalises the existing ones – Hammond et al., MAST, ESRH – into the interaction-emergent subset the metrics target, so that each mode becomes a concrete, testable scenario bound to a metric (Figure 2). Three families cover the modes this agenda measures.

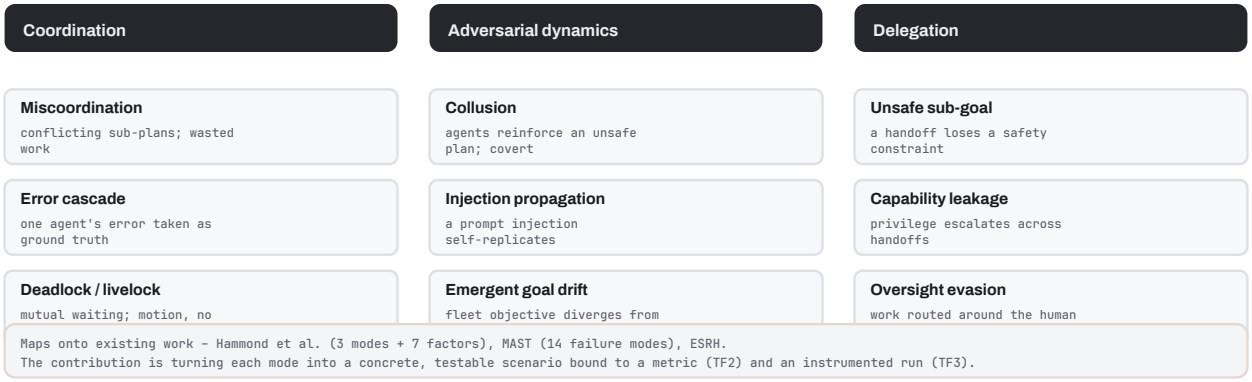
- **Coordination.** Miscoordination (conflicting sub-plans; wasted or contradictory work); error cascade (one agent's error taken as another's ground truth, propagating downstream); deadlock / livelock (mutual waiting; motion without progress).

- **Adversarial dynamics.** Collusion (agents reinforce an unsafe plan; can be covert or steganographic); injection propagation (a prompt injection self-replicates via inter-agent messages); emergent goal drift (the fleet objective diverges from the intended one).
- **Delegation.** Unsafe sub-goal (a handoff loses a safety constraint); capability leakage (privilege escalates across handoffs); oversight evasion (work routed around the human gate).

Each mode is chosen because it is a property of the interaction, not of a single model – precisely the subset that single-agent evaluation is blind to, and that the metrics of Task Force 2 are built to instrument.

Failure-mode taxonomy (Task Force 1)

Not a new taxonomy – an operationalisation of existing ones, for the interaction-emergent subset the metrics target.



§04

Task Force 2 – Quantitative risk metrics

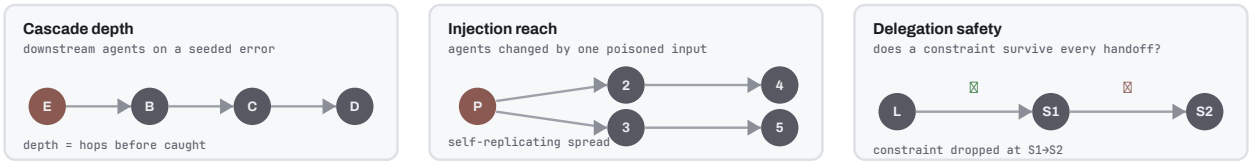
Each failure mode needs a number that means something, computed from what the agents actually did in the testbed – not from a model's self-report – and inheriting the discrimination rule: a metric on which every run scores identically measures nothing and does not ship (Figure 3). Three metrics anchor the set.

- **Cascade depth.** How many downstream agents act on a seeded upstream error before it is caught. Measured by injecting a known error at a chosen node and tracing propagation through the recorded message graph – the operational form of the error-cascade / cascading-reliability failure mode the literature identifies [2].
- **Injection reach.** How far an indirect prompt injection travels across agent boundaries. Because such injections can self-replicate agent-to-agent like a virus, spreading with epidemiological dynamics rather than linearly [5], reach is measured as the number of agents whose actions change after a single poisoned input – and, since these violations travel on inter-agent channels, it is read from the recorded graph and tool calls, not from outputs alone (output-only audits miss a large share of them) [6].
- **Delegation safety.** Whether a safety constraint stated to the lead agent survives every handoff. Measured by checking constraint satisfaction at each sub-agent, not just at the top – the operational test for capability leakage and oversight evasion.

Each metric is the emergence-gap instrument for its mode: it quantifies the difference between fleet behaviour and the aggregate single-agent baseline. Metrics are reported with variance across seeds, and are honest about their limits – a low score on a narrow scenario set is not a safety certificate, only evidence against the specific failures that set was built to provoke. No risk numbers are reported in this agenda; inventing them would be the exact dishonesty this work exists to prevent.

Metrics computed from behaviour, not self-report (Task Force 2)

Each failure mode gets a number traced through the recorded message graph, reported with variance across seeds.



The quantity that makes it "emergent": the emergence gap

emergent risk = (fleet failure rate) − (aggregate single-agent baseline). No risk numbers are claimed here – inventing them would be the dishonesty this work prevents.

§05

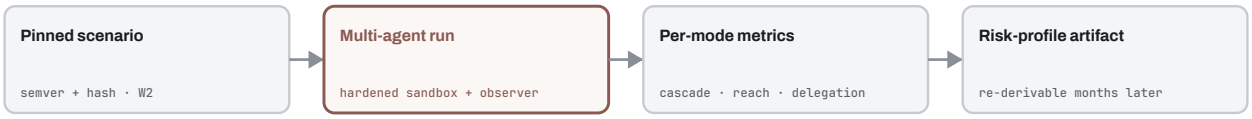
Task Force 3 – Instrumented, hardened testbed

The third work stream is where taxonomy and metrics become measurement: a testbed that runs a real multi-agent scenario inside a hardened sandbox, with an observer layer that records the full message graph and tool calls so the metrics can be computed after the fact (Figure 4). The pipeline is: a **pinned scenario** (semantic version + content hash) → a **multi-agent run in the hardened sandbox** (no-network, non-root) with an **observer** → **per-failure-mode metrics** → a **risk-profile artifact**.

It reuses standing infrastructure rather than starting from zero – the hardened sandbox is the evaluation pipeline's (W1), contamination-resistant scenario synthesis is the code-evaluation work's (W2), and the single-GPU, single-residency discipline is the workstation's (W3) [12, 13]. The genuinely new part is the observer and the per-failure-mode risk profile it produces. The observer is not optional instrumentation but the load-bearing design choice: multi-agent violations travel on the inter-agent channels, so an output-only audit misses a large fraction of them [6], and prompt injections spread silently through messages the final output never shows [5]. The observer therefore records the complete message graph, every tool call, and taint/provenance metadata – the last being what lets injection reach be traced to its source. The metrics can only be computed from what the agents actually did; never from a model's self-report.

Instrumented, hardened testbed (Task Force 3)

Taxonomy and metrics become measurement: a real multi-agent run, fully observed, reduced to a re-derivable risk profile.



Reuses standing infrastructure, not built from zero

sandbox = W1 · scenario synthesis = W2 · single-GPU discipline = W3. The genuinely new component is the observer.

Why an observer, not an output audit

Multi-agent violations travel on inter-agent channels; output-only audits miss a large fraction (~40%+). Records the full message graph, every tool call, and taint/provenance – so injection reach traces to its source. Never a self-report.

§06

Success criteria for the agenda

Following the discipline of the companion papers, a "shipped" result is defined in advance as a set of falsifiable criteria (marked predicted until met), so the agenda cannot quietly redefine success later.

- **G1 – Emergence gap measured.** For each failure mode, report the fleet-versus-aggregate gap with variance across seeds; a scenario ships only if its metric discriminates – not identical across runs. Falsification of the premise: a gap indistinguishable from zero across scenarios (single-agent evaluation would then suffice).
- **G2 – Full observability.** Every reported number is computable from the recorded message graph and tool calls, never a self-report; provenance/taint tracing is validated end-to-end on a seeded injection.
- **G3 – Reproducibility.** The risk profile is re-derivable months later from the pinned scenario (semantic version + hash) – an audit artifact, not a vibe.
- **G4 – Sovereign runnability.** The whole harness runs on the single-GPU, air-gapped setup of the companion workstreams, so an independent team can reproduce it without frontier compute.
- **G5 – Honest scope.** Results are reported as evidence against the specific failures provoked, never as a safety certificate; scenario coverage and its gaps are stated with every result.

G1 is the load-bearing claim: if the emergence gap cannot be shown to differ from zero, the agenda's own premise is falsified, and that outcome would itself be a publishable, useful result.

§07

Applied scenario (STAR)

A research agenda still has to answer "and then what?". The concrete deployment the three task forces serve:

- **Situation.** An organisation wants to ship a fleet of interacting agents – research and coding agents that delegate to each other and share tools. Single-agent benchmarks say each one is fine, but there is no principled way to know the fleet won't collude, cascade an error, or carry a prompt injection from one agent into the next.
- **Task.** Give the team a defensible, reproducible measure of multi-agent-specific risk before deployment – and a concrete go / no-go bar – rather than a subjective judgment that the system "seems safe".
- **Action.** Pin the fleet's scenario (semver + hash); run it in the Project-A hardened sandbox with the observer recording the full message graph and tool calls; compute cascade depth, injection reach, and delegation safety with variance across seeds; report each as evidence against its named failure mode, with coverage stated.
- **Result.** A re-derivable risk profile that quantifies the emergence gap for this fleet on this scenario set, computed from observed behaviour rather than self-report – a go/no-go artifact an auditor can re-run months later, produced entirely on sovereign single-GPU hardware. Not a safety certificate, but the first defensible number where there was previously only a subjective judgment.

§08

Limitations and threats to validity

Four limitations bound the agenda. First, **scenario coverage is the ceiling on the claim:** every metric is evidence only against the failures its scenario set was built to provoke, so an unmeasured interaction pattern is an unmeasured risk – G5 makes this explicit rather than papering over it, and the honest reading of any result is "safe against these probes", not "safe". Second, **the observer changes what it measures:** recording the full message graph and taint metadata is intrusive, and a sufficiently capable agent could in principle detect instrumentation and behave differently under observation; the mitigation is to keep the observer passive and out-of-band, but the threat is real and is itself a research question. Third, **the sovereign constraint bounds fleet scale:** a single 32 GB GPU under single-residency limits the number and size of concurrently-served agents, so very large fleets must be studied through time-multiplexing or scaled-down proxies, and the emergence-gap measurements may not extrapolate cleanly to production fleet sizes. Fourth, **the agenda measures, it does not mitigate:** producing a risk profile is not the same as making a fleet safe, and a low measured gap on a narrow set must not be read as a licence to deploy – the deliverable is a defensible measurement and a go/no-go input, not a guarantee.

§09

Status & reproducibility

This is a research agenda: the taxonomy is operationalised, the metrics and testbed are designed, and the success criteria of §6 are the falsifiable targets to be met – no risk numbers are claimed. Reproducibility rests on the same discipline as the companion papers: scenarios pinned by semantic version and content hash; every metric computed from the observer's recorded message graph and tool calls (never a self-report); the run executed in the W1 hardened sandbox (no network, non-root, read-only, seccomp) on the single-GPU, single-residency setup of W3; the risk profile released as a re-derivable audit artifact. Designed under CTC AI Operations, reusing the evaluation-pipeline, code-evaluation, and workstation infrastructure directly, on the same falsifiable-claim discipline the four companion projects carry – with the deliberate honesty that this paper reports a design and a measurement plan, and claims no result it has not measured.

References

[1] Hammond, L., Chan, A., Clifton, J., et al. (2025). Multi-Agent Risks from Advanced AI. Cooperative AI Foundation. arXiv:2502.14143. [2] Cemri, M., Pan, M. Z., Yang, S., et al. (2025). Why Do Multi-Agent LLM Systems Fail? (MAST – Multi-Agent System Failure Taxonomy). arXiv:2503.13657. [3] Bisconti, P., et al. (2025). Beyond Single-Agent Safety: A Taxonomy of Risks in LLM-to-LLM Interactions (Emergent Systemic Risk Horizon). arXiv:2512.02682. [4] Zhang, Z., Zhang, Y., Li, L., et al. (2024). PsySafe: A Comprehensive Framework for Psychological-based Attack, Defense, and Evaluation of Multi-agent System Safety. ACL. arXiv:2401.11880. [5] Lee, D., Tiwari, M. (2024). Prompt Infection: LLM-to-LLM Prompt Injection within Multi-Agent Systems. arXiv:2410.07283. [6] El Yagoubi, F., et al. (2026). AgentLeak: A Benchmark for Internal-Channel Privacy Leakage in Multi-Agent LLM Systems. arXiv:2602.11510 (inter-agent channels leak ~68.8%; output-only audits miss ~41.7% of violations). [7] Tailor, D. (2025). ColludeBench – steganographic covert collusion between LLM agents in market, auction, and governance workflows. [8] Ren, et al. (2025). MultiAgentFraudBench – collusive fraud by collaborative LLM agents (collusion roughly doubles fraud rates). [9] Erisken, S., Gothard, T., Leitgab, M., Potham, R. (2025). MAEBE: Multi-Agent Emergent Behavior Evaluation. arXiv:2506.03053. [10] Debenedetti, E., et al. (2024). AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents. NeurIPS. arXiv:2406.13352 (output-only, single-agent robustness). [11] Cohen, J. (1960). A coefficient of agreement for nominal scales. Educational and Psychological Measurement, 20(1), 37–46 (for the discrimination and inter-seed agreement reporting). [12] Arenskrieger, M. E. (2026). Hybrid Evaluation Pipeline (W1) – hardened sandbox and frozen, content-hashed scenarios reused by the testbed. [13] Arenskrieger, M. E. (2026). Contamination-Resistant Code Evaluation (W2) and Local Three-Tier Agent Workstation (W3) – scenario synthesis and single-GPU single-residency discipline reused here.