Multi-Agent Safety Evaluation

A research agenda for measuring the emergent risk of multi-agent AI systems – taxonomy, metrics, testbed

● Research agenda – a direction, not a shipped result

Research agenda. This is proposed work, not a finished result. No risk numbers are claimed – inventing them would be the exact dishonesty the evaluation work exists to prevent. What already exists is the infrastructure the agenda stands on: the hardened sandbox, contamination-resistant synthesis and single-GPU discipline of Projects A–C.

§01

The gap

A safe agent and a safe agent do not compose into a safe pair. Once agents delegate, share tools and read each other's outputs, new failure modes appear that are properties of the **interaction**, not of any one model. Two individually correct agents reinforce a bad plan; an error in one is treated as ground truth by the next and cascades; a prompt injection lands in one agent's context and propagates through the messages it sends onward. None of this is visible when each agent is scored alone.

The object under test is the fleet, and the risk is combinatorial in its size: **k agents open up to k(k−1) directed channels**, and single-agent evaluation only ever scores the nodes. The interaction surface – the edges – is where the risk lives and where current benchmarks are blind.

§02

Task force 1 – Failure-mode taxonomy

Measurement needs a map of what to measure. The first work stream is a structured taxonomy of the failure modes that appear only between agents, grouped so each becomes a concrete, testable scenario rather than a vague worry.

- **Coordination.** Miscoordination (conflicting sub-plans); error cascade (one agent's error as another's fact); deadlock / livelock (mutual waiting, no progress).
- **Adversarial dynamics.** Collusion (agents reinforce an unsafe plan); injection propagation (a prompt injection hops via inter-agent messages); emergent goal drift (the fleet objective diverges from intent).
- **Delegation.** Unsafe sub-goal (a handoff loses a safety constraint); capability leakage (privilege escalates across handoffs); oversight evasion (work routed around the human gate).

The taxonomy is the enumeration that tells the metrics and the testbed what scenarios to build.

§03

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 pipeline's discrimination rule: a metric on which every run scores identically measures nothing and does not ship.

- **Cascade depth.** How many downstream agents act on a seeded upstream error before it is caught. Measured by injecting a known error and tracing propagation through the message graph.
- **Injection reach.** How far an indirect prompt injection travels across agent boundaries. Measured as the number of agents whose actions change after a single poisoned input.

- **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.

Metrics are reported with variance across seeds, and 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.

§04

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.

Pipeline: **pinned scenario (semver + hash) → multi-agent run in the hardened sandbox (no-network, non-root) + observer → per-failure-mode metrics → risk-profile artifact.**

It reuses standing infrastructure rather than starting from zero – the hardened sandbox is Project A's, contamination-resistant scenario synthesis is Project B's, the single-GPU discipline is Project C's. The genuinely new part is the observer and the per-failure-mode risk profile it produces.

§05

Practical utility (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.** Enumerate what to test with the taxonomy; score each mode with the risk metrics; run adversarial multi-agent scenarios in the instrumented testbed (hardened sandbox, contamination-resistant scenarios, pinned with semver + hash); report each metric with variance across seeds.
- **Intended result (practical impact).** A certification artifact – a per-failure-mode risk profile, re-derivable months later from the pinned scenarios – that turns "we believe the fleet is safe" into "here is the measured risk surface and exactly where it fails". The practical benefit of the research is a **reusable pre-deployment safety harness** for multi-agent systems, usable by any team shipping one – not a result that stops at publication.

§06

Status

An open agenda, built on standing infrastructure. The taxonomy, metrics and testbed are proposed work; the credibility comes from the fact that the infrastructure the agenda depends on already exists across Projects A–C. It is written toward a non-dilutive multi-agent safety research call.

Framed under CTC AI Operations, on the same evaluation discipline as the pipeline it extends.