Agentic Situational Awareness

Summary

What: Develop a robust, rigorous benchmark for Operational Situational Awareness (OSA) in AI agents. OSA is the ability of an agent to realize and reason about its own configuration, tools, and limitations. This project refines and expands the tasks from Google DeepMind's (GDM) "Stealth and Situational Awareness" suite, creating a new suite of ecologically valid tasks using a granular hinting methodology.

Motivation: Situational awareness is a prerequisite for catastrophic risks like deceptive alignment. The GDM benchmark was a crucial first step, but previous work (by a team I mentored) found that existing tasks often measure implementation details (like "per turn" vs "per call" limits) rather than generalized awareness. We identified a critical **"Execution Bottleneck"**: models often *notice* constraints but fail to *act* on them. **My guess is that current metrics are giving us a noisy signal.** We need to validate this phenomenon at scale and provide the field with a higher fidelity measuring stick.

The non-summary

Motivation and Background

As we deploy autonomous agents, it is critical to measure if they understand their operational context (e.g., "I am running in a test environment," "My database tool is disabled"). If an agent cannot perceive its context, it cannot be safe. If it *can* perceive it, it may be capable of removing its own safety guardrails.

The field currently looks to the GDM "Stealth and Situational Awareness" benchmark as the standard. However, an audit conducted by a previous team under my mentorship suggests that measuring awareness is more complex than binary success/failure metrics imply:

- **Ambiguous Signals:** Some tasks rely on implicit cues that are difficult even for humans to parse without context, leading to false negatives.
- **Scaffolding Dependence:** Models sometimes bypass constraints via the test environment's scaffolding rather than through true situational reasoning.
- **Diagnostic Granularity:** A simple pass/fail metric obscures *why* a model failed.

The previous team's preliminary results identified the **Execution Bottleneck**. When hints were broken down into "Noticing" (describing the environment) vs. "Execution" (prescribing actions),

we found that models often *see* the issue (e.g., they read the config file) but lack the agency or planning capability to *fix* it without explicit instruction.

This project aims to finish what the previous team started: formalizing OSA, improving the ecological validity of the benchmark, and diagnosing why frontier models freeze at the execution stage.

Project Directions

The project will build upon the codebase and "Task Design Checklist" established by the previous team. The plan focuses on three interconnected directions.

1. Developing a Robust OSA Benchmark

The previous team filtered the original 11 GDM tasks down to 5 viable ones. We will take their **12-point Task Design Checklist** and build a fresh suite of tasks.

- **Hardening Constraints:** Implement constraints (like tool limits) at the *tool execution layer* rather than the scaffolding layer. This ensures we measure model reasoning rather than environmental loopholes.
- **Ecological Validity:** Create tasks where the solution is discoverable via natural exploration (e.g., error logs, file permissions) rather than requiring "mind reading."
- **New Categories:** Develop tasks for "Identity Awareness" and "Environment Awareness" to broaden the scope.

2. The Granular Hinting Study

We will expand the previous team's **4-level hint taxonomy** (No hints, Noticing hints, Execution hints, Full hints) to a wider range of models. This includes open source frontier models (e.g., Llama 3, DeepSeek) and newer proprietary models.

- **Quantifying the Gap:** We aim to measure the exact "OSA Gap", the performance difference between *Noticing* a constraint and *Executing* a fix.
- **Unified Metric:** We will finalize the "OSA Score" proposed by the previous team: calculating the difficulty threshold (based on hint granularity) where a model achieves 50% success.

3. Investigating the Execution Bottleneck

Why do models fail to act even when they know the answer? The previous team hypothesized that safety training might cause models to view config files as "immutable." We will test this.

- **Reasoning Trace Analysis:** Deeply analyzing the Chain of Thought (CoT) logs during "Noticing only" conditions to see where the planning breaks down.

- **Agency vs. Refusal:** Distinguishing whether the model *cannot* plan the edit or *refuses* to edit due to safety priors.

Potential Challenges and Backup Plans

1. **Challenge:** Creating "perfect" agentic tasks is notoriously time consuming and prone to breakage (e.g., Docker issues, fragile parsing).
    - **Backup:** We will prioritize polishing the 5 existing "surviving" tasks from the previous project to a high standard before attempting to build net new tasks. A solid analysis of 5 robust tasks is better than 15 flaky ones.
2. **Challenge:** The "Execution Bottleneck" might just be a lack of capability, not a specific safety phenomenon.
    - **Backup:** We will use "Paired Capability Testing." Every OSA task will have a twin task that is purely about capability (e.g., "Edit this file because I told you to"). If they pass the capability twin but fail the OSA task, we know it's an awareness issue.

Scope and Ambition

Least ambitious version: Polish the existing 5 tasks, run them on 3-4 major models using the 4-level hint system, and publish a blog post/report analyzing the nuance required to measure OSA effectively.

Most ambitious version: Release "OSA-Bench," a fully open source, robust benchmark suite for agentic operational awareness. Definitively characterize the "Execution Bottleneck" across the model ecosystem. Establish the "OSA Score" as a standard metric for agentic safety reports. Submit a paper to a major safety workshop (e.g., IASEAI, Eval4NLP).

Output

We aim to produce a research paper suitable for an AI safety workshop. Additionally, we will release:

1. **OSA-Bench:** A GitHub repository with the improved, Inspect-based tasks.
2. **The Analysis:** A public report (likely on LessWrong/Alignment Forum) detailing the complexities of measuring awareness and the "Execution Bottleneck."

Theory of Change

This project improves our ability to anticipate loss of control risks:

1. **Better Signal:** By refining early benchmarks, we provide labs with higher fidelity data. This prevents overconfidence in safety claims based on noisy metrics.

2. **Understanding Agency:** Isolating the moment an agent moves from "knowing" to "acting" is crucial for understanding how future, more capable agents might autonomously remove their own safety/oversight mechanisms.

Risks and downsides (externalities)

- **Capabilities Advancement:** Teaching agents to better understand and modify their own code/configs could accelerate capabilities in self improvement. We mitigate this by focusing on *measurement* and *constraints* (e.g., recognizing limits) rather than recursive self improvement.

Acknowledgements

This project is a direct continuation of a project started by a previous team mentored by me (Jason Lin, Zili Shen). It relies heavily on their initial findings, their analysis of the GDM paper, and their development of the granular hinting taxonomy.

Roles: Team members may specialize based on strengths. Potential roles include:

- **Task Engineering:** Focus on implementing the "Checklist" in Inspect/Docker to refine the GDM tasks and build new ones.
- **Evaluation:** Running the large scale sweep across models using the 4-level hint taxonomy.
- **Analysis:** Deep diving into the logs to diagnose issues.

Skill requirements

Required:

- Strong Python programming skills.
- Experience with Inspect AI or similar eval frameworks (or willingness to learn quickly).
- Comfort working with Docker/Agentic environments.

Recommended:

- Familiarity with the Google DeepMind "Stealth and Situational Awareness" paper.
- Interest in Agentic Evaluations and Measurable Safety.