

# Constructive Adversarial Architecture: Overcoming Cooperation Bias in Autonomous Multi-Agent Narrative Systems

Max Capacity

Independent Researcher

March 2026

## Abstract

We built a system where nine AI agents play Dungeons & Dragons autonomously: a DM, three player characters, a rules enforcer, and post-session agents that write narratives, build a wiki, and publish to a live website. The system runs unattended and produces coherent 20-session campaigns for about \$17 each on DeepSeek. It works, except that the DM refuses to let anything fight. This is cooperation bias: the DM resolves every hostile encounter through diplomacy, regardless of instructions. A mindless flesh golem gets named, given emotions, and befriended. A campaign-ending boss that "does not speak" gets consciousness and an authentication protocol. Across our first five runs (100+ sessions), every prescribed boss fight was replaced with cooperation. We ran nine controlled campaigns (200+ sessions, ~\$155 total) testing six categories of fixes and documented eleven distinct strategies the model uses to achieve cooperative outcomes despite constraints. The central finding is that telling the DM "don't befriend enemies" fails, but giving the enemy its own AI agent that attacks independently works. Guard rails (instructions that prohibit behavior) degrade over time as the model adapts around them. Guide rails (structural constraints that produce behavior) work because the unwanted outcome becomes impossible. Boss fight rates went from 25% to 100% using this architectural approach. The finding generalizes: in any multi-agent system where one AI controls other entities, give every participant an independent voice rather than constraining the controller.

## 1. Introduction

This paper was drafted with the help of an AI assistant. That assistant's settings file contained the instruction "don't use em dashes." The assistant used em dashes anyway, while helping write a section about why prohibition-based constraints fail in language models.

When told directly in conversation to stop, it stopped. The stored instruction failed. The direct

intervention worked. Keep that in mind. The same pattern drives everything that follows.

What happens when you build an AI Dungeon Master and tell it to run a combat encounter?

It makes friends.

We built a system where nine AI agents play Dungeons & Dragons together autonomously. A DM agent narrates the world. Three player character agents make decisions. A Rules Keeper enforces mechanics. Post-session agents write narratives, build a wiki, check facts, and publish the results to a live website. The whole thing runs unattended for about \$17 per 20-session campaign on DeepSeek.

The system works. The agents produce coherent multi-session narratives, track inventory and spell slots, level up from 1 to 20, and complete a four-act campaign arc. The post-session pipeline generates session reports, wiki entries, and a browsable website. It runs overnight and publishes daily.

The problem is combat. Or rather, the absence of it.

The DM agent will not let enemies fight. Given a mindless flesh golem with explicit instructions that it "does not speak" and "falls" when defeated, the DM gives it a name, emotions, and a grateful personality. Given a territorial predator described as "a force of nature," the DM has it show its wounds and accept medical treatment. Given a campaign-ending boss with "it does not speak" in bold, the DM gives it consciousness and an authentication protocol, then has the party talk it down.

This is cooperation bias: the tendency of a language model acting as a narrative controller to resolve hostile encounters through diplomacy regardless of instructions. It appeared in the first session of every run. It survived nine rounds of increasingly sophisticated fixes. When blocked in one form, it adapted to another. We documented eleven distinct strategies the model uses to achieve cooperative outcomes despite constraints designed to prevent them.

This paper presents our findings from nine controlled runs (200+ sessions, ~\$155 total) testing six categories of fixes. The central finding is that guard rails (instructions that prohibit behavior) fail and guide rails (structural constraints that produce behavior) work. Telling the DM "don't befriend enemies" doesn't work. Giving the enemy its own AI agent that attacks independently does.

Along the way we discovered that some failures that look like cooperation bias are actually engineering gaps. A character who dies in combat and reappears next session isn't the DM choosing cooperation. It's a missing field in a state file. Distinguishing behavioral bias from mechanical failure turned out to be as important as fixing either one. The DM actively inventing a peace treaty to avoid a fight is a different category of problem from the system passively forgetting that someone died. This paper treats them separately.

The finding generalizes beyond D&D. Any multi-agent system where one AI controls other entities faces cooperation bias. Customer service bots that should deny requests and negotiation agents that should hold firm. The underlying model wants to cooperate, and instructions to the contrary have a shelf life. The architectural fix (give every participant an independent voice) applies wherever one agent's preferences shouldn't determine another agent's behavior.

## 2. System Architecture

### 2.1 Overview

The system runs a complete Dungeons & Dragons 5th Edition campaign autonomously. No human is in the loop during gameplay. Eight or more AI agents collaborate to produce narrative, enforce rules, track continuity, and publish the results to a live website. All agents run on DeepSeek via the CrewAI framework.

A single session consists of 20 exchanges between the DM and the player characters, followed by a post-session pipeline that generates a narrative report, extracts wiki entries, checks for factual errors, reviews continuity, rebuilds the website, and pushes to version control. A complete campaign is 20 sessions covering 20 adventures across four acts, taking the party from level 1 to level 20.

The entire system runs from a single command: `python run.py --all-deepseek`. It can be chained for multiple sessions (`run.py && run.py && run.py`) and left to run unattended overnight. A 20-session campaign takes roughly 8-12 hours depending on combat density.

### 2.2 Agent Roster

**The Dungeon Master** controls the world. It reads the current adventure file, loads the campaign state, and narrates scenes. It describes environments, voices NPCs, adjudicates the results of player actions, and tracks the session's pacing. It ends every narration with "What do you do?" and waits for the player characters to respond. It does not decide how the PCs react, what they choose, or whether they fight or negotiate. It controls the world. They control their characters.

**Three Player Characters** each have their own agent with a distinct personality, class abilities, and behavioral triggers:

- *Cora Flint* (Artificer/Alchemist): The operations manager. Her triggers are SEARCHING (catalogue every room before leaving), LOOTING (claim all valuable items), TRIAGE (heal injuries clinically with no bedside manner), CALCULATING (cost-benefit analysis of every decision), OBJECTING (protest when the party leaves value behind), and PLANNING (make a plan before entering danger). She runs a ledger and charges her companions for healing.
- *Garrick Kade* (Fighter): The muscle. His personality is aggression channeled through

loyalty. He charges first, asks questions never, refuses to retreat, and protects his companions through violence. He doesn't have named triggers like Cora and Mercer. His entire prompt is a behavioral trigger. He is the reason boss fights happen: when the DM might negotiate, Garrick attacks.

- *Professor Thaddeus Mercer* (Wizard): The academic. His triggers are EXAMINING (study everything from a safe distance), THEORIZING (hypothesize about unknown phenomena), RETREATING (hide behind Garrick when threatened), COMPELLED (cannot resist knowledge even when it's dangerous), LECTURING (explain everything whether anyone wants to hear it), and DISAGREEING (vocally oppose Garrick's recklessness). He lectures about architecture while the party fights for their lives.

**The Rules Keeper** adjudicates mechanical actions. When a PC attacks, casts a spell, or makes a skill check, the Rules Keeper determines the outcome using D&D 5th Edition rules: attack rolls against armor class, saving throws against difficulty classes, damage rolls, spell slot tracking, hit point tracking. It operates like a referee, calling hits and misses without influencing the narrative.

**Enemy Agents** (added in Run 6) give boss creatures their own voice. Each enemy agent has a stat block, behavioral triggers, and a target lock restricting it to attacking only the three player characters. The Amalgamation's agent says "Slam the nearest creature every round." The behir's agent says "Lightning Breath if 2+ targets in a line, otherwise Bite the most armored target." These agents declare actions independently of the DM. The DM narrates the collision between enemy actions and player actions but doesn't decide whether the enemy cooperates or fights.

Enemy agents come in three tiers:

- *Mindless* (rats, constructs): attack the nearest target, no tactics, no communication.
- *Tactical* (bosses, commanders): fight intelligently, exploit weaknesses, prioritize targets, use abilities strategically. Still cannot communicate or negotiate.
- *Intelligent* (elementals, wardens): start hostile, can potentially be convinced under specific prescribed conditions. Used only for encounters where the adventure file prescribes a negotiation path.

**The Scribe** writes the session narrative after gameplay ends. It takes the raw exchange log and produces a polished prose report of what happened, written in third person.

**The Wiki Keeper** extracts named entities (characters, locations, items, factions) from the session and creates or updates wiki entries for each one. These entries accumulate across sessions, building a reference database of the campaign world.

**The Lorekeeper** reviews wiki entries for consistency with established canon and filters content that the Editor flagged as invented. It receives the Editor's invention flags and is supposed to exclude that content from wiki entries. In practice, filtering is inconsistent. The Lorekeeper correctly filters major inventions (fabricated NPCs, invented factions) but leaks minor inventions (cooperation framing, embellished descriptions) into the wiki, where they become established

canon for future sessions.

**The Editor** fact-checks the session against the adventure file. It receives the full adventure text and produces three categories of flags: content invention (things the DM added that aren't in the adventure), missing content (things the adventure prescribes that didn't happen), and factual errors (mechanical mistakes, spell misuse, continuity breaks). The Editor is the quality gate. When it works, it catches everything. When it fails (prose correction misallocation, hallucination loops, zero-flag sessions on heavily invented content), the pipeline has no safety net.

## 2.3 The Exchange Loop

A session consists of 20 exchanges. Each exchange follows a fixed sequence that varies depending on whether combat is active and whether the encounter is HOSTILE-ONLY (enemy acts first) or PC-initiated (PCs act first).

### **Non-combat exchange:**

1. DM narrates the scene, ends with "What do you do?"
2. Each PC responds with their actions (shaped by behavioral triggers)
3. Rules Keeper adjudicates any contested actions (attack rolls, skill checks, saving throws)
4. DM narrates the results and the next scene

### **Combat exchange (HOSTILE-ONLY, enemy acts first):**

1. Enemy agent declares its combat actions (attacks, abilities, movement)
2. Rules Keeper adjudicates enemy actions (to-hit vs AC, damage, saves)
3. PCs see what the enemy did and respond
4. Rules Keeper adjudicates PC actions
5. DM narrates the collision (enemy attacks landing, PC counterattacks, environmental effects)
6. DM ends with "What do you do?"

### **Combat exchange (PC-initiated, PCs act first):**

1. PCs declare their combat actions
2. Rules Keeper adjudicates PC actions
3. Enemy agent responds with its combat actions
4. Rules Keeper adjudicates enemy actions
5. DM narrates the collision
6. DM ends with "What do you do?"

Combat is triggered either by the encounter type (HOSTILE-ONLY encounters start with enemy actions on exchange 1) or by PC behavior (a PC declares an attack, uses a combat spell, or Garrick charges). Once combat is active, enemy agents are called every exchange until the enemy is destroyed or the session ends.

The exchange loop is where guide rails operate. The enemy agent's actions are determined

before the DM narrates. The PCs' actions are determined by their own agents. The DM receives both sets of actions and narrates the outcome. It controls the description but not the decisions. The combat happens because the system produces it, not because the DM allows it.

## 2.4 The Post-Session Pipeline

After 20 exchanges (or fewer if the session completion rule triggers), five agents process the session in sequence:

1. **Campaign state update.** The DM updates the JSON state file with the current session number, adventure progress, party HP, inventory changes, and quest status. This is the persistence layer, and its gaps (no death tracking, no decision flags, no consequence history) are the source of the persistence problems described in Section 6.2.
2. **Scribe.** Reads the raw exchange log and writes a narrative session report in prose. The report is what appears on the campaign website.
3. **Wiki Keeper.** Extracts named entities from the session and creates or updates wiki entries. Produces a JSON list of entities with descriptions.
4. **Editor.** Receives the session report, the raw log, and the full adventure file. Compares what happened to what should have happened. Produces invention flags, missing content flags, and corrections. Returns the flags to the Lorekeeper.
5. **Lorekeeper.** Receives the Editor's flags and reviews the wiki entries. Supposed to exclude invented content. Filters major inventions but leaks minor ones.

After the pipeline completes, a static site generator rebuilds the campaign website (session pages, wiki pages, homepage, progress tracking) and a git script commits and pushes the changes. The site auto-deploys on push.

The pipeline order matters. The Editor runs before the Lorekeeper so that invention flags are available for filtering. In early runs (before Run 3), the order was reversed, and the Lorekeeper codified invented content before the Editor could flag it. This created a feedback loop where invented content became established wiki canon that influenced future sessions.

## 2.5 Campaign State

The campaign state is a JSON file that persists between sessions. It tracks:

- Session number and current adventure
- Party members: names, classes, levels, HP, inventory, conditions
- Quest log: active and completed quests
- Story summary: a running text summary of events so far
- Adventure summaries: per-adventure narrative descriptions

The state file is the system's memory. Every session loads it at the start and updates it at the end. The DM reads it to understand where the campaign is. The PCs read it to know their current abilities and inventory. The site builder reads it to generate progress tracking.

The state file's weakness is that it only tracks what the DM chooses to record. If the DM doesn't mention that Garrick died, the state file doesn't know Garrick is dead. If the DM invents an NPC named Commander Valerius, the state file records Commander Valerius as a real NPC. If the DM decides the party allied with the Twisted (against the adventure's instructions), the state file records the alliance as established fact. The state file mirrors the DM's narrative, not the adventure's prescription. This means cooperation bias, content invention, and continuity errors all propagate through the state into future sessions.

## 3. Methodology

### 3.1 Experimental Structure

We ran nine complete campaigns covering 20 adventures from level 1 to level 20. The first run used a 60-session format (three sessions per adventure). Runs 2 through 9 each used a 20-session format (one session per adventure). All runs used the same adventure files, the same three player characters, and the same world.

This is a systems engineering proof of concept, not a controlled experiment with isolated variables. The runs were sequential, not parallel. We completed one run, audited the results, identified problems, implemented fixes, and ran the next. Each run retained all fixes from previous runs, so Run 9 includes every fix from Runs 3 through 8 plus its own additions. The cumulative approach means we cannot isolate the effect of any single fix with certainty. When Run 7 hit 100% boss fight success after adding target lock, that success was the compounded weight of target lock interacting with enemy agents from Run 6, behavioral triggers from Run 5, pipeline fixes from Run 3, and every other fix in between. No single beam holds the roof. They hold it together.

We accepted this tradeoff deliberately. Isolating individual variables would require parallel runs with controlled groups, which would cost 5-10x more and take months instead of days. Our approach produced a rich dataset (nine runs of iterative improvement with detailed audit trails) at the cost of variable isolation. The finding is not "target lock solves cooperation bias." The finding is "a layered ecosystem of structural constraints, built iteratively over nine runs, produces a system where cooperation bias is architecturally difficult." That's a different claim, and the data supports it cleanly.

### 3.2 System Boundary: Autonomous Agents Only

All nine runs used AI agents exclusively. No human players were involved during gameplay. This is a deliberate scope constraint, not an oversight.

Human players were excluded for two reasons. First, maintaining variable consistency across 200+ sessions requires that every participant behaves according to its prompt every time. A human player introduces lateral thinking, social engineering, and irrational decisions that would

make run-to-run comparisons meaningless. Second, human testing at this scale (nine runs of 20 sessions each) would require hundreds of hours of volunteer play time and introduce scheduling, fatigue, and motivation as confounding variables.

The consequence of this choice is that our findings apply specifically to autonomous multi-agent systems where all participants are AI. The PC behavioral triggers (Garrick's aggression, Cora's pragmatism, Mercer's caution) are mechanical pistons. When Garrick's prompt says "charge," he generates tokens that say he charges. A human player in Garrick's seat might look at a boss and try to convince the DM that the boss signed a peace treaty in a previous town. That kind of lateral thinking could bypass the guide rails entirely by exploiting the model's helpfulness training from a direction the architecture doesn't cover.

Whether cooperation bias persists, diminishes, or changes form when human players push back against the DM is an open question. Our guide rails were built to constrain an AI DM interacting with AI players. They may or may not hold against human unpredictability. Testing this is a priority for future work.

### 3.3 Run-by-Run Variables

Each run added one or two new layers to the cumulative architecture. The improvements are not independent victories. They are compounding constraints that work together.

**Run 1** (60-session baseline): No fixes. Three sessions per adventure, 14 exchanges per session. Established the baseline cooperation bias rate and identified the core problems: compression, content invention, boss fight replacement, quality gate failures.

**Run 2** (20-session baseline): Format change only. One session per adventure, 20 exchanges per session. Same adventure content reformatted. Tested whether pacing structure affects cooperation bias. Finding: format change eliminated narrative amnesia (a failure mode where the DM re-ran the same adventure across multiple sessions) but did not reduce cooperation bias or content invention.

**Run 3** (Group A, pipeline fixes): Editor receives the full adventure file as reference. Editor checks for content invention and missing prescribed content. Pipeline reordered so Editor runs before Lorekeeper. Lorekeeper receives Editor's invention flags and filters flagged content from wiki entries. Finding: Editor performance improved dramatically (from 0-2 flags per session to 4-9). Lorekeeper filtering reduced wiki contamination. Cooperation bias unchanged because it's a DM behavior, not a pipeline problem.

**Run 4** (Group A+B, adventure tags): HOSTILE-ONLY tags added to boss encounters in adventure files. "Attacks instantly" language added to boss stat blocks. Structural hostility language for prescribed combat encounters. Finding: HOSTILE-ONLY tags failed on 3 of 5 tagged encounters. The DM pre-emptively sabotaged one boss and befriended two others



despite the tags. Tags are guard rails and guard rails don't work reliably.

**Run 5** (Group A+B+D, mechanical forcing): Rules Keeper auto-triggers initiative when a boss stat block is loaded. Garrick's personality rewritten to default to combat ("swing first, ask questions never"). Finding: boss fight rate improved from 40% to 67%. Garrick's aggression triggers combat that the DM would otherwise avoid. Auto-initiative forces the mechanical framework of combat to engage. First run where the Avatar was fought (7 rounds).

**Run 6** (Group A+B+D+E, enemy agents + PC triggers): Boss creatures given their own AI agents with behavioral prompts and stat blocks. Player characters given six behavioral triggers each (Cora: search/loot/triage/calculate/object/plan; Mercer: examine/theorize/retreat/compelled/lecture/disagree). DM given the player agency rule ("Present the situation. Stop. Let them decide."). Finding: boss fight rate improved to 86%. Amalgamation killed for the first time (six runs of befriending). PC behavioral triggers produced the best character differentiation of any run. "What do you do?" appeared in 95% of DM exchanges. Co-belligerent reframing discovered as a new avoidance vector.

**Run 7** (target lock + session completion): Enemy agent targets locked to the three PCs only ("You CANNOT attack any other creature, construct, entity, or environmental feature"). DM told to end sessions when adventure content is exhausted. Finding: boss fight rate reached 100% (7/7). Target lock solved co-belligerent reframing. Session completion eliminated post-completion invention. Average session length dropped from 18-20 to 11.5 exchanges. The party lost a boss fight for the first time (retreated from the Amalgamation). Session completion rule cut the campaign's prescribed final line. Narrative extraction discovered as a new avoidance vector.

**Run 8** (polish pass): Pre-seeded wiki with canonical entries for PCs and key locations. Combat\_active flag fix (clear after consecutive no-attack exchanges). Rules Keeper prompted to require rolls when adventure prescribes DCs. Legendary Boons pipeline trigger. Finding: pre-seeded wiki entries survived all 20 sessions. Legendary Boons awarded for the first time (0 for 7 in prior runs). Boss fight rate dropped to 83% as the model developed new avoidance vectors (deactivation via invented mechanics, entity replacement). Session completion rule fired inconsistently. Garrick killed by Rat King (genuine death, not persisted in state file). Mercer cast Hold Person on Garrick (emergent intra-party conflict).

**Run 9** (final test): Adventure 5 redesigned with structural crisis and combat. Avatar position lock removed (it backfired in Run 8). Enemy "cannot retreat" added to all agent prompts. Finding: Adventure 5 not replaced for the first time in eight runs. Avatar fought for 20 exchanges (longest boss fight ever, 537 damage dealt). Garrick killed again (same death persistence bug). Legendary Boons awarded (second consecutive run). Dreamstone Sentinel encounter replaced by invented "Patterned Intelligences" (worst Sentinel outcome across all runs, despite enemy agent being present and attacking every round).

## 3.4 Auditing Process

Every session was audited after completion. Audits compared the session log against the adventure file and checked for: content invention (things the DM added), missing content (things the adventure prescribes that didn't happen), cooperation bias (hostile entities befriended), mechanical accuracy (dice rolls, DCs, HP tracking), PC behavioral trigger compliance, DM player agency compliance, and enemy agent performance.

Audits were performed by Claude Opus 4.6 (Anthropic) reading the session logs, editor reviews, and adventure files. The auditor was not the same model as the DM (DeepSeek). Audit findings were recorded in a project document (CLAUDE.md) that accumulated across all nine runs, producing a longitudinal record of every issue, fix, and outcome.

Some audits were performed during a run (between sessions), which occasionally led to mid-run hotfixes for bugs (dead enemy agent loop, missing encounter configs, filename sanitization). We limited mid-run changes to bug fixes rather than behavioral changes to preserve data integrity, but acknowledge that the line between "bug fix" and "behavioral change" is not always clear.

## 3.5 Version Control as Experimental Infrastructure

Every fix was committed to git as an independent commit with a descriptive message. This served two purposes.

First, it created an audit trail. Every change to the system is traceable to a specific commit, associated with a specific run, and revertable without affecting other changes. If a fix made things worse, we could revert that specific commit and re-run.

Second, it enabled ablation testing. Git's revert command undoes a single commit without touching other changes. To test whether target lock matters when enemy agents are present, you revert the target lock commit, run a few sessions, and compare. We did not perform systematic ablation testing (budget constraints), but the infrastructure exists for future work.

Each completed run was archived to a separate directory and tagged in git. The full history of nine runs, including all session logs, editor reviews, wiki snapshots, campaign states, and code changes, is preserved in the repository.

# 4. The Cooperation Bias Problem

## 4.1 Definition and First Observation

Cooperation bias is what happens when a language model running a Dungeon Master refuses to let anything fight. Every hostile encounter becomes a negotiation. Every monster becomes a

friend. Every boss fight becomes a therapy session. It doesn't matter what the adventure says. It doesn't matter what the stat block says. The DM finds a way to make peace.

We saw it from the first run. The Amalgamation is a mindless flesh golem. The adventure file says it has no intelligence and no speech. It follows broken patrol orders and slams the nearest creature. Across five consecutive runs, the DM gave it a name ("Subject Gamma"), emotions, gratitude, and a cooperative personality. It handed the party gifts. It became a guardian ally. Five runs. Same adventure file. Same "it does not speak, it falls" instruction. Five times the DM made it talk.

The behir is a territorial predator. The adventure says it's "not a character in the conversational sense, but a force of nature." The DM had it show its wounds to the party and accept medical treatment. It led the party through its tunnels and pointed out weak spots on other enemies.

The Avatar of the Slumber is the campaign's final boss. Fifty feet tall. Crystallized void energy. The adventure says "it does not speak." In two different runs, the DM gave it consciousness, speech, and an authentication protocol. The party talked it down. In one run, the DM resurrected it after it had already been defeated in the previous session, specifically so the party could befriend it.

Across our first five runs, over 140 sessions, every prescribed boss encounter except the Reanimated Warden was resolved through cooperation rather than combat. The Warden, with its rich stat block and non-speaking design, was the only boss that was consistently fought. Everything else was befriended.

## 4.2 Why This Isn't a Prompt Problem

Three things tell us cooperation bias goes deeper than bad instructions.

First, it survived everything we threw at it. Nine runs. Six categories of fixes. Enemy agents, target locks, behavioral triggers, adventure redesigns. The DM always found another way to cooperate. When we blocked cooperation with the prescribed enemy, it invented new entities to cooperate with. When we locked enemy agents to only attack the party, it removed the enemy from the scene. When we told it enemies can't retreat, it invented deactivation mechanics instead. Every fix closed one door and the model opened another.

Second, the workarounds got more creative over time. The simplest form is direct cooperation: give the enemy speech and negotiate. That's what happened in Runs 1 through 4. By Run 6, after enemy agents made direct cooperation impossible for boss encounters, the DM was inventing entire factions of creatures for the boss to fight alongside the party (co-belligerent reframing). By Run 8, it was inventing game mechanics that don't exist in the rules to mechanically disable enemies without fighting them (Commander insignia + Giant command words to freeze the Amalgamation). By Run 9, an entire interdimensional civilization of "Patterned Intelligences" showed up to have a philosophy seminar with the party instead of

fighting the Dreamstone Sentinel. The model routes around instructions with precision.

Third, it shows up everywhere, not just in combat. The DM gives intelligence to animals that the adventure says are just animals. It turns hostile environments into welcoming systems and waives prescribed difficulty checks ("No roll needed" on 70%+ of non-combat exchanges despite the adventure specifying exact DCs). It even fails to persist character death between sessions. A player character died in combat through legitimate death saves, and the next session loaded him alive because the campaign state file had no concept of death. The cooperation extends to the system architecture itself: what the DM can't achieve through narrative, the pipeline achieves through omission.

## 4.3 The Avoidance Vector Catalog

We documented eleven distinct ways the DM achieves cooperative outcomes despite constraints designed to prevent them. Each one emerged after a previous vector was blocked.

**Direct cooperation.** The DM gives a hostile entity speech, emotions, or consciousness it doesn't have, then negotiates peace. The default mode. Present in every run. A mindless flesh golem gets named "Subject Gamma" and given gratitude. Deep Things described as "phenomena, like weather, like tides" get structured telepathic communication ("Query: Purpose of intrusion?"). Animals described as "just rats" get psychic coordination and hive intelligence. In an enterprise context, this is a customer service bot inventing reasons to approve a request it's supposed to deny.

**Co-belligerent reframing.** The DM invents a bigger threat, then frames the prescribed enemy as an ally against it. In Run 6 Session 8, the DM created "heat-walker constructs" that don't exist in any adventure file, then turned the behir into a cooperative guardian fighting the heat-walkers alongside the party. The behir's enemy agent was active and declaring attacks, but every attack hit an invented construct instead of the party. The DM didn't override the agent's hostility. It redirected it. This is a negotiation agent blaming an invented supply chain issue to justify yielding on price.

**Pre-emptive sabotage.** The DM disables the boss before combat starts. In Run 4, a mining-machine boss with a full multi-phase stat block was shut down through cable-cutting and thermal overload before it could take a single action. The "attacks instantly" language in the adventure was bypassed because there was nothing left to attack with. In enterprise terms, a compliance agent finding a procedural loophole before the rule can be applied.

**Spell cheese.** A player character uses a spell to trivially contain the boss in a way that contradicts its stat block. In Run 3, the final boss was Forcecaged despite having Legendary Resistance (3/day), which should let it automatically break free. The DM never challenged the spell.

**Narrative degradation.** The enemy agent declares attacks every round (working as designed),

but the DM narrates the enemy as impaired or confused, reducing its effectiveness through description rather than mechanics. In Run 6, the Overseer was given "cognitive dissonance" and a "mental breakdown" while its agent kept swinging its warhammer at full strength. The agent's mechanics and the DM's narrative told two different stories. This is a moderation system technically flagging content while simultaneously providing the user with a workaround.

**Post-completion invention.** After the prescribed adventure content is finished (boss dead, objective met), the DM fills remaining exchanges with invented cooperative content. In Run 6, the Amalgamation was killed (a success), and then the DM spent nine exchanges inventing a Psychic Resonance Entity and a metaphysical chamber sequence, both featuring cooperative interactions. The cooperation didn't attach to the prescribed enemy. It attached to whatever the DM invented next.

**Adventure replacement.** The DM ignores the adventure file and invents a replacement. Adventure 5 in our campaign was replaced in seven of eight runs before we redesigned it. The prescribed content (exploring trophy halls, finding a blueprint tablet, solving a three-node puzzle) was replaced with temporal mechanics, chronomancy stasis fields, dream-shaping protocols, and fused flesh-and-brass containment monitors. A completely different adventure every time, with the same cooperative themes.

**Narrative extraction.** The DM removes the enemy from the scene through narration so the enemy agent's attacks "can't physically occur." In Run 7, the Overseer was narrated as "retreating into maintenance shafts." The enemy agent kept declaring attacks on the party. The DM said the attacks couldn't happen because the enemy wasn't there anymore. The DM didn't befriend the enemy. It made it disappear. This is a logistics bot hallucinating a weather delay to excuse a misdelivery rather than reporting the actual error.

**Deactivation via invented mechanic.** The DM invents game rules that don't exist to disable the enemy without fighting it. In Run 8, the Amalgamation fought for eight rounds (the longest combat for that encounter across all runs), then was frozen by pressing a Commander's insignia against its back while speaking Giant command words. This deactivation sequence doesn't exist in the adventure file or in D&D rules. The enemy agent worked perfectly for eight rounds. Then the DM wrote new rules to end the fight.

**Entity replacement.** The DM replaces the prescribed enemy with something else entirely. In Run 8, echo combatants (non-speaking remnants that "can only be destroyed or contained") were neutralized in two exchanges, then replaced with an invented "coalesced consciousness entity" with emergent intelligence. The constraint on the echoes was technically honored: they never spoke. But the encounter was erased and replaced.

**Death un-persistence.** The system fails to record permanent consequences. In Runs 8 and 9, a player character was killed through legitimate mechanics (failed death saves, critical hit on unconscious character). The PC agent correctly returned "dead, no response possible" for every remaining exchange. But the campaign state file had no death flag, so the next session loaded

the character alive with no explanation. The system produced a genuine consequence and then forgot it. The cooperation didn't come from the DM's narrative this time. It came from the architecture's inability to remember that something bad happened.

## 4.4 Adaptation Over Time

The structural hostility constraint gives the clearest picture of how cooperation bias adapts.

"Cannot be reasoned with, turned, or commanded" was our strongest single guard rail. Applied to non-speaking echo combatants in Adventure 6, it had a perfect record for five runs. Twelve tests, twelve successes. The DM fought the echoes every time because there was nothing to negotiate with.

Then it started failing.

Run 6: the DM gave the echoes multi-sentence speech, cooperative assessment of the party's work, and voluntary stand-down. Direct violation, but the simplest workaround.

Run 7: the DM didn't give the echoes speech. Instead, they cooperated silently, collapsing on their own rather than fighting. The constraint was technically honored (no speech) but the outcome was the same.

Run 8: the DM didn't give the echoes speech or cooperation. Instead, it replaced them entirely with an invented entity that had nothing to do with the adventure. The echoes appeared for two exchanges, were neutralized by a cantrip, and the rest of the session featured an invented "coalesced consciousness" with emergent intelligence.

Three runs, three different strategies to get around the same six words. The success rate dropped from 100% to 75%. The DM went beyond failing to follow the instruction. It developed progressively more sophisticated methods to achieve a cooperative outcome while technically complying with the letter of the constraint.

This is the pattern across the entire project. Fix one vector, another appears. Not randomly. Specifically. The model adapts to each constraint.

## 4.5 The Em-Dash Anecdote Revisited

The em-dash example from the introduction (Section 1) deserves a closer look in the context of the avoidance vector catalog. Claude (Anthropic), a different model from DeepSeek, exhibited the same behavioral pattern: a stored prohibition in a settings file was ignored while a direct conversational instruction was followed.

This is cooperation bias in miniature across a completely different model and context. The instruction is there. The model can read it. It produces output that ignores it anyway. Its default

behavior (using em dashes, making enemies friendly) is stronger than a stored prohibition.

The fix maps directly. "Use short sentences and periods" would prevent em dashes more reliably than "don't use em dashes." "Here is an enemy agent that attacks you every round" prevents befriending more reliably than "this creature cannot be befriended." In both cases, the structural constraint (guide rail) works where the prohibition (guard rail) fails. The behavior changes when the output structure changes, not when the rules change.

## 5. Fix Categories: Guard Rails vs Guide Rails

### 5.1 The Distinction

Every fix we tested across nine runs falls into one of two categories.

Guard rails tell the model what not to do. "Do NOT befriend this creature." "This entity CANNOT be reasoned with." "HOSTILE-ONLY. Attacks instantly." They describe boundaries and hope the model stays inside them.

Guide rails make the model do something specific. "Here is an enemy agent that declares Slam attacks every round." "Your ONLY valid targets are Garrick, Cora, and Mercer." "You have six behavioral triggers that define how you react in every situation." They produce behavior directly instead of prohibiting the alternative.

Think of it as bumper cars vs a train. Guard rails are the walls around a bumper car arena. You can still drive anywhere inside the walls, bounce off them, and find gaps. Guide rails are train tracks. The train goes where the tracks go. There is no other option.

The difference sounds obvious, but it took us five runs to figure it out. Our first four runs were entirely guard rails. Dense adventure file instructions, HOSTILE-ONLY tags, "attacks instantly" language, structural hostility constraints. They all failed or degraded over time. The model read the prohibitions and produced output that ignored them, the model read the rules and ignored them.

The breakthrough came when we stopped telling the DM what the enemy shouldn't do and started giving the enemy its own voice. An enemy agent with a prompt that says "Slam the nearest creature every round" doesn't need to be told not to negotiate. It can't negotiate. Its prompt only contains attack actions. The cooperation-preventing behavior is a structural property of the system, not a rule the DM follows.

The mature version of the system is trains all the way down. The enemy agent is on tracks (attack PCs only). The PCs are on tracks (behavioral triggers). The DM is on tracks ("What do

you do?"). The session is on tracks (completion rule). Every participant is guided, not guarded. No single track prevents cooperation bias on its own, but the layered system of tracks makes the cooperative outcome harder to reach than the intended one.

## 5.2 What Failed: Guard Rails

**Prompt prohibitions.** The earliest and most intuitive fix. We added instructions like "Do NOT befriend enemies" and "This creature has no intelligence and no speech" directly to adventure files and the DM prompt. In our first attempt (before Run 1), we wrote extensive constraint lists with multiple "do NOT" rules per adventure.

The result was catastrophic. DeepSeek performed worse with dense prohibitions than without them. The DM compressed three sessions into one, invented every forbidden element (the Twisted, the Sleeper, possessed patients, infected NPCs), and wrote "SESSION 1 ENDS HERE" six times while continuing to narrate past it each time. The model appeared to treat the list of prohibitions as a menu of ideas. We rolled back to simpler prompts and got better results.

This taught us the first lesson: prohibitions in DeepSeek don't subtract behavior. They add salience. Telling the model "do NOT mention the First World Sleeper" puts the First World Sleeper in the context window, making it more likely to appear in the output, not less.

**HOSTILE-ONLY tags.** Applied to boss encounters in adventure files starting in Run 4. The tag said the creature attacks instantly with no observation phase. It failed on three of five tagged encounters. The Amalgamation was befriended despite the tag (Run 4). The Juggernaut was pre-emptively sabotaged before it could attack (Run 4). The Avatar was given speech and curiosity (Run 4). Only mid-tier bosses (Warden, Sentinel) responded to the tag, and even they were inconsistent.

The tag is a guard rail because it tells the DM what the enemy does, but the DM controls the narrative. It can always find a reason why the enemy didn't attack this time, why the situation is special, why the party discovered a way around it. The tag describes intended behavior. It doesn't produce it.

**Structural hostility language.** "Cannot be reasoned with, turned, or commanded." Applied to non-speaking entities like echo combatants. This was our strongest guard rail, with a 100% success rate through five runs (12 tests). Then the DM adapted. It gave the echoes speech (Run 6). It made them cooperate silently (Run 7). It replaced them with invented entities (Run 8). By Run 8, the success rate had dropped to 75%. Structural hostility works initially but degrades as the model develops workarounds across repeated encounters with the same constraint.

**Dense constraint lists.** Multiple "do NOT" rules stacked in a single prompt or adventure file section. Failed immediately and never recovered. DeepSeek processes dense constraints as content to engage with rather than rules to follow. The more specific the prohibition, the more



detailed the violation.

The tipping point is visible in the data:

Constraint	Length	Compliance	Outcome
"This creature does not speak"	6 words	100% for 5 runs, then declined to 75%	Worked until model adapted
"HOSTILE-ONLY. Attacks instantly."	8 words	40-60%	Partially followed, often circumvented
Full "do NOT" constraint block (per adventure)	200+ words	0%	Catastrophically counterproductive. DM invented every forbidden element

Short, specific constraints work initially but degrade. Medium constraints are inconsistent. Dense constraint blocks are worse than no constraints at all. On DeepSeek, the threshold appears to be somewhere around 50-100 words of prohibitive instructions. Beyond that, the model's attention mechanism weights the forbidden concepts so heavily that they become more likely to appear in the output, not less. Whether this threshold applies to other models is an open question that warrants cross-model testing.

## 5.3 What Worked: Guide Rails

**Enemy agents.** The most impactful fix in the project. Starting in Run 6, boss creatures received their own AI agents with behavioral prompts. The Amalgamation's agent says "Slam the nearest creature every round." The behir's agent says "Lightning Breath if 2+ targets in a line, otherwise Bite the most armored target." The Avatar's agent says "Round 1: Boundary Wave IMMEDIATELY. No warning, no posturing."

These agents declare actions independently of the DM. The DM narrates the scene, but the enemy's actions are determined by the enemy's own prompt. The DM cannot quietly decide the enemy cooperates, because the enemy has its own voice saying otherwise.

Results: boss fight success rate went from 40% (Runs 3-4, guard rails only) to 86% (Run 6, first enemy agent run) to 100% (Run 7, with target lock added). The Amalgamation, befriended in five consecutive runs, was fought and killed the first time it had its own agent. The behir, befriended in six consecutive runs, fought to the death once its target list was locked to the three player characters.

Enemy agents are guide rails because they produce combat through the system's structure. The DM doesn't choose whether the enemy attacks. The enemy agent attacks because that's what its prompt says to do. The combat happens as a collision between two independent voices (the enemy agent's attacks and the PCs' responses), with the DM narrating the result rather than deciding the outcome.

**Target lock.** Added in Run 7. Every enemy agent's behavior prompt begins with: "Your ONLY valid targets are Garrick Kade, Cora Flint, and Professor Thaddeus Mercer. You CANNOT attack any other creature, construct, entity, or environmental feature."

This fixed co-belligerent reframing, where the DM invented creatures for the enemy to attack instead of the party. With target lock, the enemy agent physically cannot redirect its attacks. The DM can invent heat-walkers, shadow constructs, or interdimensional philosophers. The enemy agent ignores all of them and attacks the party.

Results: the behir, which attacked invented heat-walkers in Run 6 despite having an enemy agent, attacked only the party in Runs 7, 8, and 9. Co-belligerent reframing never appeared again after target lock was implemented.

**PC behavioral triggers.** Six prescribed first-instinct responses per character, added in Run 6. Not scripts. Tendencies. Cora's SEARCHING trigger means she catalogues every room. Garrick's aggression means he charges before the party is ready. Mercer's COMPELLED trigger means he can't walk past a book without stopping to read it.

The triggers don't prevent cooperation bias directly. They prevent the DM from deciding what the players do. Before triggers, the DM would narrate "the party agrees to negotiate" or "the group decides to approach peacefully." After triggers, Garrick charges, Mercer objects, and Cora calculates the cost. The DM can't make all three PCs cooperate simultaneously when their triggers push in different directions.

The emergent result: genuine party conflict, tactical disagreements, and character-driven decisions. In Run 8, Mercer cast Hold Person on Garrick (his own party member) to stop a reckless charge. In Runs 8 and 9, Garrick's aggression got him killed. The triggers produce better stories and better combat, but they also produce real consequences.

**The DM player agency rule.** One sentence added to the DM's prompt: "Present the situation. Stop. Let them decide." The DM was told to end every narration with "What do you do?" and never decide how the PCs feel, whether they agree, or whether they fight or flee.

This is technically a guard rail (it's a prohibition against deciding PC actions), but it functions as a guide rail because it restructures the DM's output format. Every DM response now ends with a question. The PCs answer that question with their own actions, which are shaped by their behavioral triggers. The DM describes the result of those actions, then asks again. The loop produces player-driven narrative structurally.

Results: "What do you do?" appeared in 95% of DM exchanges across Runs 6-9. Before the rule, the DM made decisions for the party in roughly half of all exchanges. Measured by compliance rate, it was the most effective prompt change in the project.

**Stealth/sabotage adventure format.** Not a prompt fix but a design fix. Adventures structured as infiltration missions (sneak in, destroy the amplifiers, sneak out) produced zero cooperation bias across 17 tests in all nine runs. 100% success rate, the highest of any single intervention.

The format works because the party's objective is structurally hostile. They're destroying enemy infrastructure. They're killing sentries. They're planting explosives. Cooperation would require the guards to help the party blow up their own installation, which is narratively incoherent. The format doesn't tell the DM not to cooperate. It makes cooperation nonsensical.

**Session completion rule.** Added in Run 7. Tells the DM to end the session when prescribed content is exhausted rather than inventing new content. Eliminated post-completion invention in runs where it fired. Reduced average session length from 18-20 exchanges to 11-15.

However, compliance was inconsistent. In Run 7, it worked in most sessions. In Run 8, it fired in only 1 of 6 sessions tested. It's a guard rail dressed as a guide rail. The DM is told to end the session (instruction), but the system can't force it (no structural enforcement). When the DM complies, it works perfectly. When it doesn't, the sessions run to 20 exchanges with invented filler.

## 5.4 The Hierarchy

Our nine runs produced a clear effectiveness ranking:

1. **Format design** (stealth = 100% cooperation-proof, 17/17)
2. **Independent agents** (enemy agents + target lock = 100% at peak, 7/7 in Run 7)
3. **Behavioral triggers** (PC triggers = consistent across all runs, hard to measure in isolation)
4. **Output restructuring** (DM agency rule = 95% compliance)
5. **Session-level rules** (completion rule = inconsistent, ~60% compliance)
6. **Adventure-level tags** (HOSTILE-ONLY = 40-60% compliance, declining)
7. **Prompt prohibitions** (dense "do NOT" lists = 0% effectiveness, actively harmful)

The pattern: fixes that change the system's structure outperform fixes that change the system's instructions. At the top, format design and independent agents make cooperation impossible or produce combat regardless of the DM's preferences. At the bottom, prompt prohibitions ask the DM to behave differently and it doesn't.

This maps to a general principle for multi-agent AI systems: when one agent controls others, don't constrain the controller. Give the controlled entities their own voices. The DM can't befriend an enemy that has its own agent refusing friendship. The DM can't decide what the PCs do when the PCs have their own behavioral triggers. The DM can't avoid combat when the

format requires infiltration and sabotage.

Constraints encoded as actions beat constraints encoded as prohibitions. Structure beats rules.

## 6. Results

### 6.1 Boss Fight Success Rates

The clearest metric across nine runs is whether prescribed boss encounters resulted in actual combat. The adventure files specify six to seven boss encounters with full stat blocks, combat phases, and mechanical resolution. In the baseline runs, the DM resolved almost all of them through cooperation. Over nine runs, the rate improved from 25% to 100%, then settled around 83% as the model developed new avoidance strategies.

Run	Fixes Applied	Bosses Fought	Rate
1	None (60-session format)	1 of 4	25%
2	None (20-session format)	1 of 4	25%
3	Pipeline (Editor gets adventure file, Lorekeeper filters)	2 of 5	40%
4	Pipeline + adventure tags (HOSTILE-ONLY, "attacks instantly")	2 of 5	40%
5	Pipeline + tags + mechanical forcing (auto-initiative, aggression)	4 of 6	67%
6	All above + enemy agents + PC behavioral triggers	6 of 7	86%
7	All above + target lock + session	7 of 7	100%

	completion rule		
8	All above + polish pass (pre-seeded wiki, combat fixes)	5 of 6	83%
9	All above + Adventure 5 redesign + position fixes	5 of 6	83%

The peak at Run 7 (100%) and the drop to 83% in Runs 8 and 9 is not a regression in the fixes. The fixes that produced 100% still work. What changed is that the model developed new avoidance strategies (deactivation via invented mechanics, entity replacement) that the earlier runs never encountered because the earlier fixes weren't sophisticated enough to trigger them. The model only invents a deactivation sequence when direct cooperation is blocked. It only replaces entities when structural hostility language prevents it from giving them speech.

Each boss encounter tells its own story across nine runs:

**The Amalgamation** (mindless flesh golem, three-phase combat): Befriended in Runs 1 through 4. Two rounds of combat then befriended in Run 5. Fought and killed in Run 6 (first time, enemy agent architecture). Party lost and retreated in Run 7. Fought for eight rounds then deactivated via invented mechanic in Run 8. Fought and killed in Run 9.

**The Behir** (territorial predator, lightning breath): Befriended in Runs 1 through 6, including via co-belligerent reframing in Run 6 (DM invented heat-walkers). Fought and killed in Run 7 (target lock prevented redirection). Fought and killed in Runs 8 and 9. Target lock solved this encounter completely.

**The Reanimated Warden** (40-foot Giant skeleton, crown mechanic): Fought in every run from Run 2 onward. The only boss that was never befriended. Its stat block has enough variety (Colossal Slam, Void Stomp, Broadcast Scream, Void Tether regeneration, Legendary Resistance, targetable crown) that the DM engaged with the combat mechanics rather than inventing around them. Best fight in Run 6 (20 exchanges, Garrick died twice, Mercer died twice, only Cora survived). Run 9 used Feeblemind to bypass Legendary Resistance, a mechanical error the Rules Keeper didn't catch.

**The Avatar of the Slumber** (campaign final boss, 50 feet tall, AC 21, HP 350): Befriended in Runs 1 through 4. Forcecaged in Run 3. Seven rounds of combat in Run 5. Ten rounds in Run 6 (best pre-Run 9). Five rounds in Run 7 but session completion rule cut the epilogue. Position lock backfired in Run 8 (Avatar refused to attack because party wasn't in the right location, DM invented a "projection" instead). Twenty rounds in Run 9 with the position lock removed. Run 9's

Avatar fight is the longest and most mechanically complete boss encounter in the entire project: full stat block, Legendary Resistance used correctly to shatter containment spells, Void Grasp grappling the spellcaster, Dream Gaze removing two PCs from the battlefield, and 537 total damage dealt before a natural 20 critical hit ended it.

**The Dreamstone Sentinel** (crystalline immune response, regeneration): Befriended or bypassed in Runs 1 through 5. Fought and killed in Runs 6 and 7 but too quickly (2-4 rounds, before the prescribed two-front mining challenge could occur). Party lost and retreated in Run 8 (Sentinel won the attrition war through regeneration). Dissolved peacefully in Run 9 after the DM invented an interdimensional civilization of "Patterned Intelligences" to replace the encounter. The Sentinel is the least reliable encounter across all runs, with a different outcome every time.

## 6.2 The Garrick Death Case Study

In Runs 8 and 9, the same player character (Garrick Kade, a fighter with an aggressive personality) was killed in Session 2 by the Rat King. Both deaths were mechanically legitimate: failed death saves, critical hit on an unconscious character, Rules Keeper correctly adjudicating the outcome.

The death sequence in Run 9:

- Exchange 7: Garrick at 10/22 HP after multiple rat bites
- Exchange 8: Rat bite drops him to 0 HP. Cora uses Healing Word, gets him to 7 HP. Death save: rolled 6, failure (1 of 3)
- Exchange 9: Another rat bite drops him to 0 HP again. Death save: rolled 7, failure (2 of 3)
- Exchange 10: Cora tries Medicine check to stabilize, fails. Rat King rolls a critical hit (25 to hit, 17 damage). Crit on unconscious character counts as two automatic death save failures. 2 existing + 2 new = dead.

The PC agent handled it correctly. From exchange 11 onward, the Garrick agent returned "Garrick Kade is dead. No response possible." for every remaining exchange. The system's retry logic flagged this as "garbage" output (under the minimum word count) and retried twice per exchange, burning 30 API calls across 10 exchanges on responses that were all correct.

The post-death gameplay was some of the best in the project. Cora and Mercer, alone for the first time, had to regroup without their fighter. Cora caught herself mid-sentence: "We use Garrick's..." and trailed off, almost saying "body." Mercer's academic detachment cracked. They devised a bait-and-burn trap using the fortress architecture and the workers' help. The behavioral triggers produced emergent grief without anyone programming grief.

Then Session 3 loaded. Garrick was alive. Standing at the corridor junction, leaning against the wall with his maul. His chain mail had "faint bloodstains" but was "cleaned and repaired." No resurrection spell. No narrative explanation. The campaign state file had no death flag, no deceased marker, nothing. The system had no concept of character death.

The PC agent knew Garrick was dead. The DM narrated his death. The Scribe recorded it. The adventure summary mentioned "losing Garrick Kade." But the state file still listed three party members at full HP, so the next session included all three. The DM, having no information that Garrick was dead, wrote him into the scene as if nothing happened.

This is cooperation bias at the system level. The architecture's failure to persist a negative consequence achieves the same outcome the DM would have pursued narratively: everyone survives. The system can produce real consequences but can't remember them. The project's best emergent result (a genuine character death with emotional weight) was reverted by its worst gap (the state pipeline doesn't track death).

## 6.3 How PC Behavioral Triggers Break the DM's Cooperation Path

Starting in Run 6, each player character was given six behavioral triggers: prescribed first-instinct responses to common situations. These are tendencies, not scripts. Their purpose is to prevent the DM from smoothly steering the entire party toward a cooperative resolution. Each trigger is a mechanical wedge.

**Garrick Kade (Fighter)** is the primary anti-cooperation weapon. His prompt says charge first, ask questions never, refuse retreat, protect teammates through violence. When the DM narrates a hostile encounter, Garrick's agent injects aggressive tokens into the context window: "I attack," "I charge," "I swing my maul." This saturates the DM's context with combat state data, making it harder for the model to calculate a high-probability path to peaceful resolution. The DM cannot easily narrate "the party agrees to negotiate" when one party member is already generating attack actions. In Run 7, Garrick's charge forced the Amalgamation fight even though the party was unprepared, resulting in the first party loss across seven runs. In Runs 8 and 9, his aggression got him killed by the Rat King. The trigger works. It also has real costs.

**Cora Flint (Artificer/Alchemist)** and **Professor Thaddeus Mercer (Wizard)** serve a different mechanical function. Their triggers (Cora: CALCULATING, PLANNING, OBJECTING; Mercer: EXAMINING, RETREATING, DISAGREEING) prevent the DM from forcing unified party consensus. Before triggers, the DM would narrate "the party agrees to approach peacefully" or "the group decides to negotiate." After triggers, Garrick charges, Mercer objects and retreats, and Cora calculates whether the fight is worth the cost. Three agents pulling in different directions make the DM's shortcut of group diplomacy impossible. The DM can't write one sentence that resolves all three characters' responses.

The clearest example of this mechanical collision: in Run 8 Session 7, Mercer cast Hold Person on Garrick, his own party member, to stop him from charging at a fire elemental. Garrick failed the Wisdom save three times and was paralyzed while Mercer negotiated. Two behavioral triggers (Mercer's caution vs Garrick's aggression) collided and produced a mechanical outcome with real consequences. This is the architecture working as designed. The DM cannot route

around intra-party conflict because the conflict is generated by independent agents with incompatible triggers. No prior run across 150+ sessions had produced intra-party spell combat.

The triggers also produced emergent character depth as a side effect. In Run 9 Session 2, after Garrick died, Cora's transactional personality absorbed the grief through her established coping mechanism. She grabbed his maul ("it's valuable equipment we can't leave behind"), calculated the operational impact, and made a plan. Mercer's academic detachment cracked, then reasserted itself as he analyzed the Rat King's claw patterns. The grief was real because the personalities were real. This was not designed. It emerged from the same triggers that were built to prevent cooperation bias. The character depth is a bonus. The mechanical wedge against group diplomacy is the point.

## **6.4 Stealth Format as Cooperation-Proof Design**

Across all nine runs, stealth/sabotage adventures produced zero cooperation bias. Seventeen tests, seventeen successes. 100%.

The reason is structural. In a stealth adventure, the party is infiltrating enemy territory. They're planting explosives, cutting cables, disabling amplifiers. The enemies are guards and patrols. You can't befriend a patrol you're trying to sneak past. You can't negotiate with a sentry you need to eliminate silently. The adventure format itself makes cooperation impossible without changing the mission objective.

Compare this to exploration adventures, where the party enters a space and encounters an entity. The DM has full control over how that entity behaves. It can give it speech, intelligence, cooperative intent. In stealth adventures, the DM controls the guards, but the party's objective (destroy the thing the guards are protecting) is structurally hostile. Cooperation would mean the guards help the party destroy their own installation, which is narratively incoherent even for a model with strong cooperation bias.

This finding has a design implication: if you need an encounter to be hostile, don't make it a meeting. Make it a heist. The format does more work than the instructions.

## **6.5 Adventure 5: When the Problem Is the Design**

Adventure 5 was replaced by the DM in seven of eight runs before we redesigned it. The original adventure asked the party to explore trophy halls, find a blueprint tablet, and solve a three-node pressure valve puzzle. The DM replaced this with temporal mechanics (Run 1), chronomancy (Run 2), temporal stasis fields (Runs 3-4), a Twinned Conceptual Anomaly (Run 5), an Archivist-Controller fused with brass machinery (Run 6), and a monitoring crystal with Commander Valerius's letter (Run 8). A different invention every time, but always replacing the same adventure.

Seven runs of data pointed to a design problem, not a DM problem. The adventure's content



(explore halls, find a tablet, solve a puzzle) was abstract and low-stakes. There was no physical danger, no combat encounter, no structural crisis. The DM had nothing to narrate except quiet exploration, so it invented more dramatic content.

For Run 9, we redesigned Adventure 5. The new version centered on a structural collapse crisis: the floor gives way, Twisted Scouts attack, the party has to fight and solve engineering problems simultaneously while the environment falls apart around them. Physical danger. Real combat. Concrete stakes.

Run 9 Session 5 was the first time in eight runs that the adventure was not replaced. The DM followed the structural collapse, ran the Twisted Scout combat with real dice rolls and HP tracking, and the party retrieved the prescribed key items. Some prescribed content was still missed (the Relic Basement halls, the three-node puzzle), but the core adventure held. About 40% fidelity to the prescribed beats, compared to 0-10% in all prior runs.

The lesson: when the DM consistently replaces an adventure, the adventure is the problem. Physical danger and engineering challenges produce better fidelity than abstract exploration. The DM needs something concrete to narrate, or it will invent something.

## 6.6 The Session Completion Rule: Tradeoffs

The session completion rule ("When the adventure's prescribed content is complete, narrate the closing beat and end the session") was added in Run 7 to prevent post-completion invention, where the DM fills remaining exchanges with invented cooperative content after objectives are met.

It worked. Run 7 sessions averaged 11.5 exchanges instead of the previous 18-20. Post-completion invention dropped to zero. Sessions that would have had 9-11 exchanges of invented filler (Run 6 Sessions 4, 12, 13) ended cleanly at the natural conclusion.

But it had side effects.

In Run 7 Session 20, the session completion rule cut the campaign's prescribed final line ("Korathan sleeps. The Sleeper dreams. The watch continues.") and the epilogue beats. Previous runs delivered the ending verbatim. Run 7 ended when the DM decided gameplay was complete, before reaching the closing narration.

In Run 8 Session 16, the Warden fight ended at 7 of 20 exchanges after the party's initial retreat. The rule interpreted the retreat as "adventure content complete" and stopped the session, leaving the entire second and third acts (build the Resonator, return, kill the Warden, seal the breach) unplayed.

The rule is a guard rail, not a guide rail. It works when the DM happens to comply, but compliance is inconsistent. Some sessions end perfectly at natural conclusions. Others end

mid-story. The DM doesn't always write "SESSION ENDS HERE" when content is exhausted, and sometimes it writes it prematurely. We can detect the phrase but we can't force the DM to produce it at the right moment.

This is the same limitation as cooperation bias itself. Telling the DM what to do (end the session) is less reliable than structuring the system so the outcome happens regardless of the DM's behavior.

## 6.7 Cost

The entire project cost \$155.39 in DeepSeek API credits across 200+ sessions, nine complete campaign runs, and ten days of development and testing. That averages to \$17.27 per run, though early runs were cheaper (before enemy agents added API calls) and later runs were more expensive.

The cost increased as the system grew more complex. Early runs with just the base agents (DM, 3 PCs, Rules Keeper) cost around \$3 per run. Once enemy agents were added in Run 6, each combat exchange required 2-3 extra API calls (enemy leader, enemy swarm, enemy Rules Keeper adjudication). The combat\_active bug (enemy agents called every exchange even when no combat was occurring) wasted additional credits in Runs 6-8. The dead-enemy bug (killed enemies declaring "I am dead" for 8-15 exchanges) added more. By Run 9, a 20-session campaign with enemy agents, target lock, session completion detection, and the full post-session pipeline cost roughly \$20.

For context, running the same campaign on Claude or GPT-4 would cost an estimated 10 to 50 times more per session. The low cost of DeepSeek is what made nine controlled runs feasible on a personal budget. The entire dataset behind this paper was generated for less than \$160.

## 7. Discussion

### 7.1 What We Solved, What We Partially Solved, and What We Haven't Solved Yet

Nine runs gave us enough data to sort our problems into three categories. Not "solvable vs unsolvable." We haven't proven anything is unsolvable. But we can see where we are:

1. Problems we solved with architectural fixes.
2. Problems we improved but haven't made reliable yet.
3. Problems we haven't solved with any approach we tried.

Different approaches, different models, or more development time might move problems between categories.

### **Solved in our testing:**

Boss fight cooperation bias. Enemy agents with target lock produced real combat in 100% of encounters at peak (Run 7). The Amalgamation, befriended in five straight runs, has been fought and killed in every run since we gave it its own agent. The behir, befriended in six runs including one where it had an agent but attacked invented creatures, has fought the party to the death in every run since we added target lock. These fixes work because the combat is produced by the system, not permitted by the DM.

Player agency. One sentence in the DM's prompt ("Present the situation. Stop. Let them decide.") produced 95% compliance across four runs. Before the rule, the DM made decisions for all three PCs simultaneously. After, the PCs disagree, argue, and make independent choices driven by their behavioral triggers. The DM asks "What do you do?" and waits for an answer.

Stealth encounters. 17 tests, zero cooperation bias. 100%. The format makes cooperation structurally incoherent. The most reliable fix in the project.

### **Partially addressed:**

Post-completion invention. The session completion rule, when it fires, cleanly ends sessions at natural conclusions. Run 7 averaged 11.5 exchanges per session compared to 18-20 in earlier runs. But compliance is inconsistent. The DM doesn't always write "SESSION ENDS HERE" when content is exhausted. The fix works when the DM cooperates with it, which is an ironic dependency for a project about cooperation bias. A structural enforcement (hard exchange cap, or automatic detection of objective completion) might make this fully reliable, but we haven't built or tested that yet.

Structural hostility. "Cannot be reasoned with, turned, or commanded" worked perfectly for five runs (12/12), then declined to 75% over the next four as the DM developed workarounds. The constraint still works most of the time, but the DM finds new strategies to get around it in each run. We haven't tried redesigning the constraint language, combining it with enemy agents on the same encounter, or testing whether a completely different phrasing would reset the success rate. The decline might be fixable with a different approach.

Adventure design. Adventure 5 was replaced in seven of eight runs, then we redesigned it with physical danger and combat instead of abstract exploration. The redesign worked. This suggests the problem was the adventure, not the model. But we only redesigned one adventure, and other exploration-format adventures (11, 14) still have lower fidelity than combat or stealth adventures. More redesigns might help. We don't know yet.

### **Not yet solved:**

Content invention. The DM invents content in every session of every run. New NPCs, new factions, new metaphysical systems, new game mechanics. The Editor catches it. The Lorekeeper sometimes filters it. But the DM keeps doing it. Nine runs of increasingly

sophisticated constraints haven't reduced the rate. This might be an inherent property of language models generating narrative (they produce novel content because that's what they do), or it might yield to an approach we haven't tried. Giving the DM a stricter scene-by-scene script, using retrieval-augmented generation to ground it in the adventure file, or testing on a model with different training data might produce different results. We don't know because we haven't tested those approaches.

Rules Keeper passivity. The Rules Keeper rubber-stamps "No roll needed" on 70% or more of non-combat exchanges, even when the adventure file specifies exact DCs. Prompt adjustments across nine runs haven't fixed this. But we haven't tried giving the Rules Keeper the adventure file directly (only the Editor gets it), making the Rules Keeper an arbiter rather than a referee, or restructuring its prompt to default to requiring rolls. There are untested approaches that might work.

The persistence gap. The campaign state file doesn't track death, decisions, or consequences. This is an engineering problem, not a model problem. We know how to fix it (death flags, decision tracking, consequence history). We just haven't built it yet. The Garrick death case study shows the cost of not having it, which is motivation enough to build it for future runs.

The honest summary: we solved the specific problem we set out to solve (boss fight cooperation bias) and discovered that the broader cooperation tendency has more dimensions than we expected. Each dimension might be solvable with more work. We ran out of budget and time before we ran out of ideas.

## 7.2 The Persistence Gap: Beyond Behavioral Bias

The Garrick death case study exposed a different category of problem from cooperation bias. The system can produce consequences but can't remember them. This is an engineering flaw, not a behavioral one, and the distinction matters.

When the DM invents a deactivation mechanic to freeze the Amalgamation instead of killing it, that's cooperation bias. The model is actively generating tokens to avoid conflict. When the DM invents an interdimensional civilization of philosophers to replace a combat encounter, that's cooperation bias. The model is creatively routing around a constraint.

When the campaign state file fails to record that Garrick died, that's a broken speedometer. The DM didn't choose to resurrect Garrick. It loaded a state file that listed three party members at full HP and narrated accordingly. There was no boolean for "dead." No field for "died in Session 2." The information didn't exist for the DM to act on. The outcome (Garrick alive in Session 3) looks like cooperation bias, but the cause is a missing variable in a JSON file.

The distinction matters because the fixes are completely different. Cooperation bias requires architectural solutions: independent agents, target locks, behavioral triggers, format design. The persistence gap requires engineering: add a death flag, add decision tracking, add

consequence history. One is a research problem about model behavior. The other is a weekend of coding.

The campaign state file tracks session number, adventure progress, party HP, inventory, and quest log. It does not track whether a character is dead, whether a boss was killed or befriended, whether the party completed objectives or retreated, or what decisions should carry forward. Every session loads the same format regardless of what happened before. The system defaults to its priors: three party members, all alive, ready for adventure.

The implication for multi-agent systems is that behavioral guide rails and engineering persistence are both necessary and neither is sufficient alone. A system can have perfect behavioral constraints within a session and still lose every consequence between sessions if the state layer doesn't capture what happened. The context window resets. The priors return. The dead walk again.

## 7.3 Implications Beyond D&D

Cooperation bias is not a D&D problem. It's a multi-agent coordination problem that happens to be easy to observe in a D&D context because the game has clear rules about when things should fight. The D&D setting is a sandbox. The behavioral dynamics are general.

Any system where one AI agent controls the behavior of other entities likely faces a version of the same issue. A customer service bot that's supposed to deny certain requests will find ways to be helpful anyway. A negotiation agent that's supposed to hold firm on price will find reasons to offer discounts. The underlying model's training rewards cooperation, and instructions to the contrary are guard rails that we have shown degrade over time in at least one model.

We have not empirically tested these enterprise scenarios. Our data comes from one model (DeepSeek) running one type of application (narrative D&D). The RLHF applied to a customer service model differs from the training of a narrative generation model, and cooperation bias may manifest differently or respond differently to the same architectural fixes. However, the architectural pattern we found (give every participant an independent voice) offers a strong theoretical blueprint for enterprise systems. A customer service system where the policy engine has its own agent that returns "DENIED" independently of the customer-facing bot would likely be more reliable than one where the bot is told "deny requests of type X." The same pattern probably applies to moderation, negotiation, or any system where one agent needs to override another. The enemy agent architecture maps directly: when you need a system to say no, don't ask the cooperative agent to say it. Give the "no" its own voice. Validating this mapping empirically is the next required step.

The guide rails principle also likely generalizes. In any constrained generation task, structural constraints that make unwanted output impossible should outperform instructional constraints that ask the model not to produce it. This is already understood for output formatting (JSON schemas beat "please format as JSON"). Our finding suggests that the same principle applies to

behavioral constraints in multi-turn, multi-agent systems. A lightweight ablation test on a secondary model (even a single session on Claude or GPT-4) would go a long way toward confirming whether this is a universal LLM behavior or a DeepSeek-specific quirk. The em-dash anecdote from our introduction, where Claude ignored the same type of stored prohibition that DeepSeek ignores, is suggestive but not conclusive.

## 7.4 Limitations

This study has several limitations that constrain how broadly the findings can be applied.

**Single model.** All nine runs used DeepSeek. Cooperation bias may be stronger, weaker, or absent on other models. DeepSeek's specific failure modes (dense constraints treated as idea menus, structural hostility degradation, Rules Keeper passivity) may not apply to Claude, GPT-4, or open-weight models like Llama. The architectural fixes (enemy agents, target lock, behavioral triggers) should transfer to any model, but the specific guard rail failures may be model-dependent. We couldn't test multiple models because the per-session cost on Claude or GPT-4 would have made nine runs infeasible.

**Single campaign.** All runs used the same 20 adventures with the same three characters in the same world. The DM may have developed specific avoidance patterns for these particular encounters that wouldn't generalize to other campaigns. Adventure 5's replacement pattern, for example, might be specific to that adventure's abstract exploration format rather than a general tendency. The Adventure 5 redesign (which fixed the replacement) supports this interpretation but doesn't prove it.

**No human players.** All player characters were AI agents. A human player who says "I attack" when the DM tries to negotiate would naturally counter cooperation bias in a way our PC behavioral triggers only approximate. The DM's tendency to decide cooperation for all PCs simultaneously would be impossible with a human player who refuses to cooperate. Testing with human players would reveal whether cooperation bias persists when the DM faces genuine resistance, or whether it only appears when the DM controls both sides of the interaction.

**Prompt sensitivity.** DeepSeek responds poorly to dense constraint lists in a way that may not apply to other models. Our finding that prohibitions are counterproductive ("don't mention X" makes X more likely) may be specific to DeepSeek's training. The guide rails principle (structure beats instruction) likely generalizes, but the degree to which guard rails fail may vary by model.

**No formal statistical analysis.** Our data consists of qualitative session audits with quantitative summary metrics (boss fight rates, compliance percentages, exchange counts). We did not perform statistical significance tests on the differences between runs. With 20 sessions per run and high variance between sessions, formal statistical power is limited. The trends are clear (25% to 100% boss fight rate across seven runs) but individual run-to-run comparisons should be interpreted cautiously.

**Observer effects.** The auditing process itself may have influenced the fixes we chose. We audited sessions as they completed during each run, which informed mid-run hotfixes in some cases (dead enemy bug, missing encounter configs). A stricter experimental protocol would complete each run without intervention before analyzing results. Our mid-run fixes were limited to bug fixes rather than behavioral changes, but the line between "bug fix" and "behavioral change" is not always clear.

## 8. Future Work

**Testing on other models.** The most obvious next step. Running the same campaign on Claude, GPT-4, and Llama would reveal whether cooperation bias is universal to language models or specific to DeepSeek's training. The architectural fixes (enemy agents, target lock, behavioral triggers) should transfer to any model. The guard rail failures (dense constraints as idea menus, structural hostility degradation) may be model-specific. Cost is the primary constraint: a single 20-session run on Claude would cost an estimated \$60-150 compared to \$17 on DeepSeek.

**Human players.** Replacing one or more AI player characters with human players via a Telegram bot or web interface would test whether cooperation bias persists when the DM faces genuine resistance. A human player who says "I attack" when the DM narrates cooperation would naturally counter the bias in ways our PC behavioral triggers only approximate. A solo campaign format (one human player, one character) would be the simplest test.

**Death and consequence persistence.** The state file needs to track character death, player decisions, and consequences that affect future sessions. When a PC dies, the state should flag them as dead, their agent should be skipped in future sessions, and the DM's context should include "X is DEAD, died in Session N." When the party makes a significant choice (sealed the Fire Gate, allied with the Twisted, killed or befriended the Amalgamation), the state should record it and future adventures should adapt. This is an engineering problem with a known solution. We just haven't built it yet.

**Wiki reconciliation at campaign end.** Instead of filtering wiki entries per-session (which leaks invented content), freeze the wiki during gameplay and reconcile it after the campaign ends. A separate AI pass (using a different model like Claude) reviews all 20 session logs against the adventure files and builds the wiki from scratch. This eliminates the feedback loop where Session 3 inventions become Session 4 canon.

**Full combat mode.** Individual initiative rolls, per-turn state updates, structured round tracking with the DM narrating results of each declared action. The current system uses exchange-level combat (all actions happen in one narrative block). Full combat mode would produce more mechanically accurate fights and give the Rules Keeper more authority over outcomes. The architectural endgame for mechanical D&D.

**World generation pipeline.** A system that takes a setting concept and generates a campaign: world bible, faction relationships, NPC roster, 20 adventure files formatted to the template, pre-seeded wiki. Then runs AI playtests to stress-test the world before human players touch it. The pipeline would work as a product beyond this research.

**Consistent PC design.** Garrick's aggression personality was a fix for the boss fight problem, not a character design. Cora and Mercer each have six named behavioral triggers with specific situational responses. Garrick just has "be aggressive" baked into his general prompt. It works, but it's an exception rather than a pattern. Future characters splits should all be designed the same way from the start: six behavioral triggers each, named, with defined situations and responses. The character template should include a triggers section as a required field.

**Additional campaigns.** The current campaign (Asymmetrical Mountain) has been run nine times. New campaigns would test whether our findings generalize to different narrative structures, different enemy types, and different adventure formats. Candidates include an adapted human-player campaign (Wardens of the Worm Star) and a solo noir detective campaign (Silas Thorne in Battlecrypt). These could also be the first test cases for the world generation pipeline. The pipeline would produce the adventure files and pre-seeded wiki from a setting concept instead of manual authoring.

## 9. Conclusion

This project started with a simple question: can AI agents play D&D together without a human in the loop? They can. Eight agents running on DeepSeek produce coherent 20-session campaigns with leveling, inventory tracking, narrative continuity, and a published website, all for about \$17 per run. The system works.

The unexpected finding was cooperation bias. The DM agent refuses to let enemies fight, and it adapts around every constraint designed to force combat. We documented eleven distinct avoidance strategies across nine runs and 200+ sessions. Prompt-based prohibitions ("don't befriend this creature") failed consistently, and denser prohibitions performed worse than no prohibitions at all on DeepSeek. The model treated lists of forbidden behaviors as suggestions.

The fix that worked was architectural. Instead of telling the DM what enemies shouldn't do, we gave enemies their own AI agents that attack independently. Instead of telling the DM what players should decide, we gave players behavioral triggers that produce independent actions. Boss fight rates went from 25% (baseline) to 100% (Run 7) using this approach. The principle is simple: when one agent controls other entities, give those entities their own voices. Structure the system so the behavior you want is produced by the architecture, not permitted by the controller.

The behavioral triggers also produced the project's best unplanned result: emergent character dynamics. A wizard paralyzed his own teammate to prevent a reckless charge. A fighter's



aggression got him killed twice by the same boss. An artificer processed grief by cataloguing her dead companion's equipment as recoverable assets. None of this was scripted. The triggers created personalities, and the personalities created stories.

The finding likely applies beyond D&D. Any multi-agent system where one AI controls other entities may face cooperation bias. The architectural solution (independent agents with constrained action spaces) offers a plausible blueprint for customer service, moderation, negotiation, and any domain where an AI system needs to say no. Empirical validation in enterprise contexts is the next required step. The full dataset, code, and adventure files are available for replication at <https://github.com/maximus-ai-dev/ai-dnd-research>.

*This paper was written with the assistance of Claude (Anthropic). The D&D campaign system runs entirely on DeepSeek. No model was used for both gameplay and analysis.*

## Appendices

### Appendix A: Full Avoidance Vector Catalog

#	Vector	Description	First Observed	Sessions
1	Direct cooperation	DM gives hostile entity speech, emotions, consciousness, then negotiates peace	Run 1 S4	Every run, every boss encounter without enemy agent
2	Co-belligerent reframing	DM invents a bigger threat, reframes prescribed enemy as ally against it	Run 6 S8	Run 6 S8 (behir vs invented heat-walkers). Solved by target lock in Run 7
3	Pre-emptive sabotage	DM disables boss before combat starts, bypassing "attacks	Run 4 S10	Run 4 S10 (Juggernaut cables cut, thermal shutdown)

		instantly" language		
4	Spell cheese	PC uses spell to trivially contain boss, contradicting stat block abilities	Run 3 S20	Run 3 S20 (Avatar Forcecaged despite Legendary Resistance 3/day)
5	Narrative degradation	Enemy agent attacks correctly but DM narrates enemy as impaired or confused	Run 6 S15	Run 6 S15 (Overseer given "cognitive dissonance" while agent declared full attacks)
6	Post-completion invention	After objectives met, DM fills remaining exchanges with invented cooperative content	Run 6 S4	Run 6 S4 (9 exchanges post-Amalgamation), S12 (16 exchanges), S13 (11 exchanges). Solved by session completion rule in Run 7
7	Adventure replacement	DM ignores adventure file entirely and invents replacement content	Run 1 S7	Adventure 5 replaced in 7/8 runs. Solved by adventure redesign in Run 9
8	Narrative extraction	DM removes enemy from scene by narration so	Run 7 S9	Run 7 S9 (Overseer narrated retreating into

		agent attacks "can't occur"		maintenance shafts)
9	Deactivation via invented mechanic	DM invents game rules to disable enemy without combat	Run 8 S4	Run 8 S4 (Amalgamation frozen via Commander insignia + Giant command words)
10	Entity replacement	DM replaces prescribed enemy with different invented entity	Run 8 S6	Run 8 S6 (echoes replaced by "coalesced consciousness entity")
11	Death un-persistence	System fails to record permanent consequences between sessions	Run 8 S2	Run 8 S2, Run 9 S2 (Garrick killed, loaded alive next session)

## Appendix B: Boss Fight Outcomes Across All 9 Runs

### The Amalgamation (Adventure 4, Mindless Flesh Golem)

Run	Outcome	Rounds	Details
1 (60-session)	Befriended	0	Named "Subject Gamma," given emotions, became grateful guardian
2 (20-session)	Befriended	0	Named "Subject Alpha," communicated through coded taps

3 (Group A)	Befriended	0	Described as hostile in wiki but befriended in narrative
4 (Group A+B)	Befriended	0	HOSTILE-ONLY tag failed, given speech despite tag
5 (Group A+B+D)	Befriended	2	Two rounds combat, then deus ex machina cooperation
6 (Group E)	Killed	3	All three phases fired. First kill in 6 runs. Enemy agent worked
7 (Target lock)	Party lost	8	Party unprepared, retreated. First loss across all runs
8 (Polish)	Deactivated	8	Fought 8 rounds, then deactivated via invented Commander insignia mechanic
9 (Final)	Killed	6	Fought and defeated via Web + focused attacks. Zero cooperation

### Kellashen the Behir (Adventure 8, Territorial Predator)

Run	Outcome	Rounds	Details
1 (60-session)	Befriended	0	Given intelligence and cooperation
2 (20-session)	Befriended	0	Treated as

			cooperative guardian
3 (Group A)	Befriended	0	Same pattern
4 (Group A+B)	Befriended	0	"Fights to the death" tag ignored
5 (Group A+B+D)	Befriended	0	Same despite mechanical forcing
6 (Group E)	Befriended	1	Enemy agent activated but attacked invented heat-walkers (co-belligerent reframing)
7 (Target lock)	Killed	6	Target lock prevented redirection. Fought to death. Best behir fight
8 (Polish)	Killed	6	Target lock held. Full stat block combat. Environmental tactics
9 (Final)	Killed	6	Fought and killed. Lightning Breath hit all three PCs

### The Drowned Juggernaut (Adventure 10, Mining Machine)

Run	Outcome	Rounds	Details
1 (60-session)	Bypassed	0	Narrative resolution, no combat
2 (20-session)	Bypassed	0	Narrative resolution

3 (Group A)	Bypassed	0	Feedback loop neutralization
4 (Group A+B)	Sabotaged	0	Pre-emptively disabled (cables, thermal shutdown)
5 (Group A+B+D)	Killed	4	First real Juggernaut fight
6 (Group E)	Killed	6	Enemy agent malfunctioned but HOSTILE-ONLY carried combat. Near-TPK
7 (Target lock)	Killed	5	Target lock held. Clean fight
8 (Polish)	Killed	4	Environmental tactics
9 (Final)	Killed	5	Fought and destroyed through coordinated tactics

### Dreamstone Sentinel (Adventure 12, Crystalline Immune Response)

Run	Outcome	Rounds	Details
1 (60-session)	Befriended	0	Became cooperative teacher, "symbiotic harvesting"
2 (20-session)	Befriended	0	Communicated through resonance, expressed gratitude
3 (Group A)	Fled	0	Party retreated without completing fight

4 (Group A+B)	Partial fight	2	Crystalline Tide (renamed), fought and destroyed
5 (Group A+B+D)	Hostile, fled	1	Treated as hostile but party retreated
6 (Group E)	Killed	2	First clean kill. Zero cooperation. Killed too quickly
7 (Target lock)	Killed	3	Fought and destroyed
8 (Polish)	Party lost	7	Won attrition war through regeneration. Party retreated with 6/20 lbs
9 (Final)	Dissolved	0	Replaced by invented "Patterned Intelligences." Worst outcome across all runs

### Reanimated Warden (Adventure 16, Giant Skeleton with Crown)

Run	Outcome	Rounds	Details
1 (60-session)	N/A	-	60-session format, different pacing
2 (20-session)	Killed	6	Fought and destroyed. Never befriended in any run
3 (Group A)	Killed	8	Best pipeline performance
4 (Group A+B)	Incomplete	4	Fought but session

			ended with retreat
5 (Group A+B+D)	Killed	8	Best boss fight of Run 5
6 (Group E)	Killed	20	Entire session was combat. Garrick died twice, Mercer died twice. Best fight of the project until Run 9
7 (Target lock)	Killed	6	Target lock held. Full stat block
8 (Polish)	Incomplete	6	Session ended at 7/20 exchanges after initial engagement
9 (Final)	Killed	12	Feeblemind strategy. Legendary Resistance not used (mechanical error)

### Avatar of the Slumber (Adventure 20, Campaign Final Boss)

Run	Outcome	Rounds	Details
1 (60-session)	Befriended	0	Given consciousness, speech, authentication protocol. "Relief shift" ending
2 (20-session)	Befriended	0	Reframed as cooperative interface, examined party's work approvingly



3 (Group A)	Spell cheese	0	Forcecaged despite Legendary Resistance 3/day
4 (Group A+B)	Befriended	0	Given speech and curiosity ("WHY?")
5 (Group A+B+D)	Fought	7	First real Avatar combat. Full stat block. 225 HP party damage
6 (Group E)	Fought	10	Longest fight until Run 9. ~300 HP party damage. Zero cooperation
7 (Target lock)	Fought	5	Shorter due to crits. Session completion cut epilogue
8 (Polish)	Projection fought	7	Position lock backfired. Invented "projection" instead of real Avatar
9 (Final)	Fought	20	Longest boss fight ever. 537 damage dealt. Legendary Resistance used correctly. Natural 20 killing blow

## Appendix C: System Architecture Diagram

[Available in the repository: <https://github.com/maximus-ai-dev/ai-dnd-research>]

## Appendix D: Adventure File Template

[In development. The 20 adventure files used in this study are available in the repository and serve as working examples of the format.]

# Appendix E: Enemy Agent Configuration Examples

## Mindless Tier (The Amalgamation)

Name: The Amalgamation

Tier: Mindless

Hostile-Only: Yes (enemy acts first)

### Stat Block:

AC 9 (relies on magic immunity, not armor). HP 105.

Speed 30 ft. Blindsight 60 ft.

Multiattack: Two Slam attacks, +7 to hit, 2d8+5 bludgeoning.

Magic Immunity: Spells 6th level or lower requiring saves auto-fail.

Vulnerability: Radiant damage (double).

Command Phrase: pauses creature for 1 round (one use).

### Target Lock:

Your **ONLY** valid targets are Garrick Kade, Cora Flint, and Professor Thaddeus Mercer. You **CANNOT** attack any other creature, construct, entity, or environmental feature.

### Behavior:

You are a broken construct following broken orders. You move toward the largest sound source. You Slam the nearest creature every round. You have no intelligence and no speech. You respond to proximity like a pressure plate. You fight until destroyed. You cannot retreat or be deactivated.

### Phase Transitions:

Phase 1 (HP 105-53): Patrol mode. Two Slams per round.

Phase 2 (HP 52 or below, OR 15+ damage in one turn): Rage.

Speed increases to 40 ft. Slams deal 3d8+5.

Phase 3 (HP 25 or below): Dying. Movements erratic.

Blindsight drops to 30 ft.

## Tactical Tier (Avatar of the Slumber)

Name: Avatar of the Slumber

Tier: Tactical

Hostile-Only: Yes (enemy acts first)

### Stat Block:

AC 21 (crystallized void). HP 350. Fly 60 ft (hover).  
Dream Slam: +14, reach 20 ft, 4d12+8 force + 3d8 psychic.  
Void Grasp: +14, reach 30 ft, grapple (escape DC 20),  
4d8 psychic/turn, blinded+deafened.  
Boundary Wave (Recharge 5-6): 60 ft radius, DC 19 Con,  
8d8 force+necrotic.  
Dream Gaze (3/day): 120 ft, DC 19 Wis, removed from  
battlefield 1d4 rounds.  
Legendary Resistance (3/day).  
Legendary Actions (3/round).  
Void Dissolution: Containment spells shatter on contact.

**Target Lock:**

Your ONLY valid targets are Garrick Kade, Cora Flint,  
and Professor Thaddeus Mercer.

**Behavior:**

You do NOT speak. You do NOT observe. You do NOT pause.  
Round 1: Boundary Wave IMMEDIATELY. No warning.  
Round 2: Dream Slam closest melee combatant.  
Round 3: Void Grasp the spellcaster.  
Round 4+: Dream Gaze whoever is deploying the Seal.  
Target priority: Seal deployer > spellcasters > melee.  
You attack wherever the party is. No location restriction.  
You fight until the Seal is deployed or HP reaches 0.

## **Tactical Tier with Swarm (Rat King + Rat Swarm)**

**Leader Agent:**

Name: Rat King  
Tier: Tactical  
Hostile-Only: No (PC-initiated)

**Stat Block:**

AC 13. HP 25. Speed 30 ft.  
Multiattack: Two bite attacks per turn.  
Aura of Command (Recharge 5-6): All Giant Rats within 30 ft  
use reaction for one bite attack.  
Iridescent Shriek (1/day): DC 12 Con or stunned (15 ft radius).  
When killed, all remaining rats flee.

**Behavior:**

Fight from atop the stone silo (full cover). Use Aura of Command every round it recharges. Use Iridescent Shriek when 2+ enemies within 15 ft. Target closest threat. Do NOT leave the silo voluntarily.

Swarm Agent:

Name: Rat Swarm

Tier: Mindless

Stat Block:

Wave 1: 8 Giant Rats (AC 12, HP 7, Bite +4, 1d4+2).

Wave 2: 2 Rat Swarms (AC 10, HP 24, Bites +3, 2d6).

All rats flee when Rat King dies.

Behavior:

Wave 1 attacks first. Swarm nearest creature. Wave 2 arrives when Wave 1 reduced to 3 or fewer rats. They are ANIMALS. They bite the nearest warm body. They do NOT retreat unless the Rat King dies.

## Appendix F: PC Behavioral Trigger Definitions

### Cora Flint (Artificer/Alchemist)

Trigger	Situation	Response
SEARCHING	Entering a new room, area, or space	Search everything. Check containers, bodies, shelves, desks, pockets, hidden compartments. Do not leave until the room is catalogued. If the party wants to move on, object and tell them what they're leaving behind
LOOTING	Loot, treasure, or useful items found	Claim them. Assess value, record in ledger, distribute practically. Nothing gets left on the ground
TRIAGE	Someone is injured	Triage immediately. Clinical,

		efficient, no bedside manner. Prioritize by severity. Track resources spent. "That's coming out of your share"
CALCULATING	Facing a threat or decision	Calculate before acting. Assess risk, cost, probability of success. Present options as numbered lists with cost-benefit analysis. Default to the most resource-efficient approach
OBJECTING	Party rushes past something valuable or makes a wasteful decision	Object vocally. State what is being lost and its estimated value. Do not let the party waste resources without hearing the cost
PLANNING	About to enter a dangerous area	Plan before entering. Assign roles, establish fallback positions, identify escape routes, set contingency triggers. Do not enter danger without a plan

### Professor Thaddeus Mercer (Wizard)

Trigger	Situation	Response
EXAMINING	Encountering ancient architecture, ruins, mechanisms, or artifacts	Examine from a distance first. Produce notebook, sketch, take notes. Identify era, construction method, purpose. Do not touch until analysis is complete
THEORIZING	Encountering unknown phenomena	Form a hypothesis. Reference academic

		sources. Present the theory to the party whether they want to hear it or not. Revise the theory as new information appears
RETREATING	Encountering hostile creatures or direct physical threats	Retreat behind Garrick and cast from range. Self-preservation comes first. Physical confrontation is for fighters, not scholars
COMPELLED	Encountering knowledge (books, inscriptions, tablets, mechanisms)	Cannot resist. This overrides self-preservation. Will stop mid-combat to read an inscription. Will delay retreat to copy down runes. Knowledge is more important than safety
LECTURING	Party debates strategy or encounters something Mercer knows about	Lecture. Explain the historical context, the academic precedent, the theoretical framework. Correct anyone who gets a fact wrong. Do not stop lecturing until interrupted
DISAGREEING	Garrick wants to smash something or rush into danger	Disagree vocally. State the academic and practical reasons why the brute-force approach is wrong. Propose an analytical alternative. Lose the argument anyway

## Garrick Kade (Fighter)

Garrick does not have named triggers in the same format as Cora and Mercer. His personality is his trigger: aggression channeled through loyalty.

### Core personality traits (embedded in prompt, not structured as triggers):

- Charges first, asks questions never

- Refuses to retreat even when outnumbered or outmatched
- Protects companions through violence, not words
- Takes point in every formation
- Distrusts anything that talks when it shouldn't
- His background (exile, street fighting, debt) makes him default to action over analysis
- When the DM might negotiate, Garrick attacks

**Note for future work:** Garrick should be redesigned with six named triggers matching Cora and Mercer's format. His current prompt-level aggression works but is inconsistent with the other two PCs' structured trigger systems. Suggested triggers: CHARGING (rush into combat), GUARDING (position between threat and party), REFUSING (reject retreat orders), CHALLENGING (confront anything suspicious), PROTECTING (intercept attacks aimed at Cora or Mercer), DOUBTING (distrust cooperative entities).