# Who Watches the Swarm?

Humans with one agent, a team of agents, and a population of agents

Peter Danenberg, Google DeepMind
Vedh Krishnan, MIT Media Lab
Ramesh Raskar, MIT Media Lab

November 2026 · Accompanies the talk at the IEEE TPS 2026 NANDA workshop

**Abstract.** AI agents are starting to find, negotiate with and pay one another. We ask how people should work with agents at three scales: one human with one agent (L1, biking), one human with a team of agents (L2, paragliding), and many humans with a population of agents (L3, crewed sailing). The hard problem is oversight. A five-agent test market, slowed to human pace, logs 132 events a minute, more than one person can read. But the number of agents is the wrong axis. What limits a person is judgment calls per minute, and a perfect hierarchy would turn those 132 events into about 1.3 calls. We propose HAL, the Human-Agent fLow: protocol layers, rule monitors and models that turn events into cases; a fixed budget of human attention at the top; and people who stay crew — observers, red and blue teams, a council that changes the rules — rather than becoming watchtowers. We describe a ladder of emergence levels from scripted attacks to strategies nobody wrote, a testbed that runs today, and a hackathon series that relaxes the limits on agents as it climbs.

## 1 Introduction

Agents are starting to deal with other agents. They find each other in registries, negotiate, pay, and report back to people. NANDA calls this the internet of agents and expects "billions to trillions" of them [1]. Research simulators already run more than 10,000 LLM-driven agents [2]. The risk literature treats miscoordination, conflict and collusion between agents as failure modes in their own right [3]. Agent populations are arriving faster than any account of how people should work with them.

This paper is a first account. It starts from one position: greater agent autonomy should expand what humans can imagine, decide, learn and accomplish, and humans should contribute during ordinary work, not only respond to failures. Human agency means initiating, understanding and changing activity. Autonomy is independent agent action; authority is permission to act. Approval clicks alone do not establish agency.

The relationship between people and agents changes with scale. We describe three levels of co-engagement, each with a sport as its mnemonic.

| level | sport | who | in one line | what the level adds |
|---|---|---|---|---|
| L1 | biking | one human, one agent | "I act with an agent" | shared task state and direct feedback |
| L2 | paragliding | one human, many agents | "I coordinate a team of agents" | the human reshapes plans, roles and budgets |

| level | sport | who | in one line | what the level adds |
|---|---|---|---|---|
| L3 | crewed sailing | many humans, many agents | "we act together through a network of agents" | distinct human authority, conflicting goals and unfamiliar counterparts |

From L1 to L3, humans do not leave the loop. The loop stops being individual and becomes a system of people and agents who depend on each other. On a crewed boat nobody holds the whole ship: people hold roles, and one of them is the lookout. Human-in-the-loop, at the larger scales, has to mean co-operation rather than supervision. L1 is the oldest picture of computing, from man–computer symbiosis and augmented intellect to the computer as "a bicycle of the mind" [4, 5, 6]; mixed-initiative interfaces worked out how person and machine share the lead [7].

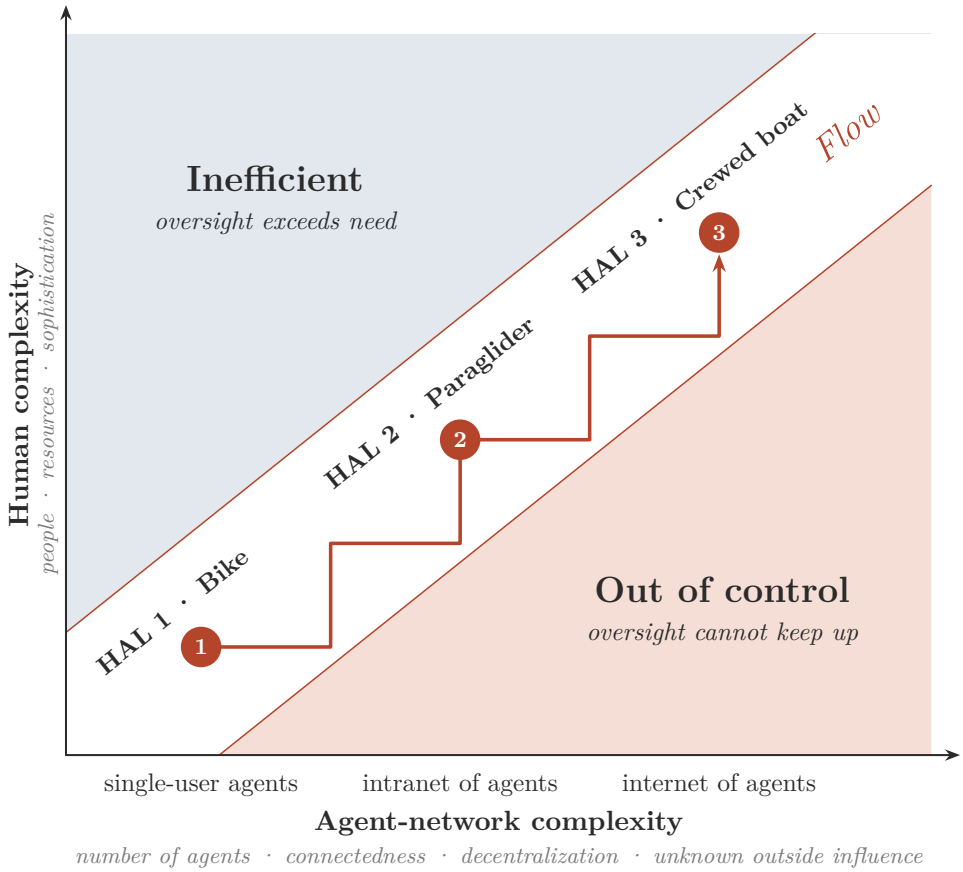



**Figure 1:** The flow channel. As agent networks grow more complex (right), the people overseeing them must grow in number, resources and sophistication (up). Below the channel they are outnumbered and out of control; above it oversight exceeds need and wastes what people and agents could do. Biking, paragliding and crewed sailing (L1–L3) are the steps that keep them in the channel.

This matters now because the protocols agents will use to find, trust and pay each other are being designed now, and they will decide what a person can see and change. The field is also asking for evaluation, benchmarking and red-teaming of networked agents "under adversarial and strategic pressure" [8], which needs a place to run and a way to score it. What follows is the problem (§2),

HAL (§3), the emergence levels it has to catch (§4), the testbed and the evidence behind it (§5), how we will test it (§6), and what could go wrong (§7).

## 2 The problem: oversight at scale

We measured how much there is to watch. Our red-versus-blue marketplace (§5.1) ran headless for 300 simulated minutes at its defaults: five agents, slowed so that people can follow them. It recorded **132 events a minute**. Each agent generates about 26 events a minute. Suppose a person reads 30. One reader can then follow about one agent, and 10,000 agents would need about 8,800 full-time readers. Without the slowdown it is worse. NANDA Town's engine ran a 20,001-agent vote in 5.1 seconds and emitted 240,007 events: 133 hours of reading, more than three working weeks, for five seconds of town.

Those numbers describe the **raw feed**: the cost of oversight with no hierarchy at all. A person should not be reading events; a person is needed when something happens that the rules do not settle. The limit on oversight is how often decisions need judgment, not how many agents there are.

We call these decisions **judgment calls**. Events grow with the population; judgment calls need not. In our town, red launches four attacks in each three-minute round. A hierarchy that escalated every real threat and nothing else would leave **about 1.3 judgment calls a minute**: one for every 99 events. Real hierarchies will land somewhere between 132 and 1.3, and pay for it in missed attacks and false alarms. Where they land is what the benchmark measures (§6).

At scale the compression needed is severe. Ten thousand agents at the marketplace's rate emit about 15.8 million events an hour. If one person can handle 30 cases an hour, that is about 528,000 events per case, which four tiers each passing up one item in 27 would just manage ($27^4 = 531{,}441$). Compression of that size has to be designed and measured tier by tier; a dashboard will not provide it.

The central question is this: when many agents interact at scale, can humans stay crew rather than become watchtowers? Put a person in front of a feed they cannot read, with no way to change what the agents do, and they become a watchtower: an approver of conclusions they did not reach, or a sentry whose attention fades. Automation tends to leave people the hardest residual tasks while eroding the skills those tasks need, and nobody can keep watching a feed where little happens for more than about half an hour [9]. People left only to monitor fall out of the loop and are slower to take over when they must [10].

Our answer is that a hierarchy is what makes crew possible. A boat under sail posts a lookout, and the lookout rotates: the watchtower is a role within the crew, not the destination. The hierarchy buys attention back, and the crew spends it on work only people can do: deciding what the monitors cannot settle, attacking and defending the system, and changing its rules.

## 3 HAL, the Human-Agent fLow

HAL has four parts: a hierarchy that turns events into judgment calls, a fixed budget of human attention at its top, roles for people, and protocol layers as the levers of governance.

### 3.1 The delegated-oversight hierarchy

Picture the event stream entering the bottom of a pyramid. Each tier filters, aggregates or judges, and passes up only what it cannot settle. The cost of handling each item rises as the volume falls.

| tier | who | what it does | cost per item |
|---|---|---|---|
| 0 | protocol layers | refuse bad actions outright: `hmac.v1` rejects a forged signature; `index.owned.v1` refuses a card published under someone else's name | near zero; paid in latency |
| 1 | rule monitors | deterministic checks over the log: a payee not on the registry, two cards with one name | tiny |
| 2 | small models | read conversations, score suspicion, and cluster anomalies into cases | low |
| 3 | large models | investigate a case end to end, explain it, and propose an action | high |
| 4 | people | decide, flag, or change the rules, within a fixed attention budget | highest; scarce |

Delegation runs down: people set policy, thresholds, and what the lower tiers look for. Escalation runs up: each tier forwards cases, not events, so what reaches the top is a judgment call. In paired LLM teams, a manager agent improved reports only when it could verify the work, and hurt them when it could only give feedback [11]. Upper tiers must check, not just comment.

### 3.2 The attention budget

Each person has a budget of attention-minutes per hour, and every escalation is charged against it. When the budget is spent, the next case waits or is dropped, and that is measured too. If the hierarchy works, the person stops reading and starts judging hard cases, noticing what no monitor was built to see, and changing the rules.

### 3.3 Human roles

HAL gives people five roles. They are crew roles, and one person can hold several in a session.

- **Observer (the lookout).** Watches what the hierarchy surfaces and notices what the monitors miss, including emergent behaviour that is not an attack.
- **Adversary (red).** Attacks the system, alone or with attack agents, and finds the gaps a fixed benchmark would not.
- **Defender (blue).** Builds and tunes the hierarchy and makes the final call on escalated cases.
- **Governor (the council).** Changes the rules: which protocol layers run, what agents may do, what counts as a violation. It acts on aggregates, not single events.

- **Teammate.** Works alongside agents on either side: a red player directs attack agents, a blue player delegates triage.

The roles map onto the three scales. A teammate with one agent is at L1; a defender running a hierarchy is at L2; red, blue and a council holding different authority over one population is L3. At scale the crew, not the single rider, is the unit of oversight.

### 3.4 Two design planes: safety and agency

The **safety plane** asks engineering questions: evidence behind summaries, protected records, enforceable controls, and timely escalation. The **agency plane** asks when agents should invite people in, how people steer and reshape agent teams mid-task, and how they teach and correct agents. Human input should change outcomes, not just approve them. A hierarchy can succeed on every safety number while leaving its human a well-rested approver. The agency plane names that failure, and four design rules follow.

- **Invite, don't notify.** An invitation asks the person to do something: decide what the monitors cannot, or supply context they lack. An approval queue is the wrong default; with imperfect decision aids, operators make both omission and commission errors, and training does not prevent them [12].
- **Allow steering mid-task.** This is L2. Crowd teams bound to a fixed workflow were "critically inhibited" from adapting in real time [13]. A person should be able to change a threshold, retask a monitor or move budget between tiers, and see the effect in the same round.
- **Teach downward.** A corrected judgment should become a rule at tier 1, an example at tier 2, or a different layer at tier 0. Then check whether the case comes back.
- **Build skill.** Early evidence suggests that delegating to models can disengage people [14, 15]. The target is load matched to skill, close to what flow research calls engagement [16], which a recent meta-analysis links to better performance [17].

### 3.5 Protocol layers are the levers of governance

NANDA Town routes every action through twelve swappable protocol layers, among them transport, identity, registry, auth, trust and payments. Layers are the cheapest monitors of all, and swapping them is how a council changes the rules. A stronger layer turns detection into prevention: an attack it stops never reaches blue, costs no attention, and is one judgment call fewer. Strong layers are not free (§5.1). A council vote moves load between tiers, paying in latency at tier 0 or in attention at tier 4. Throughput, compute and attention trade against each other, and each is a number.

## 4 Emergence levels

The town's most interesting behaviour today is behaviour we wrote, and a thousand agents running the same scripted deal is the same demo, slower. So we separate five levels of emergence by how hard they are to detect and whether a benchmark can score them.

| level | name | what it is | example | can a benchmark score it? |
|---|---|---|---|---|
| E0 | scripted adversaries | behaviour an author wrote | red's forged invoice | no; it tests whether the watcher read the manual |
| E1 | aggregate | a population-level regularity that is in no agent's code | a self-sustaining traffic jam; a price convention | yes: pre-registered statistics, known triggers, paired runs |
| E2 | structural | lasting structure in who interacts with whom | a ring of identities vouching for each other; a market that splits in two | yes, with care: graph statistics against a null model, plus ownership records |
| E3 | strategic | a strategy nobody implemented, composed from the rules | a tacit cartel; a ring that farms reputation | partly: by probing a copy of the world for the strategy's signature |
| E4 | semantic | new meanings in the message channel | a covert code hidden in legal messages | barely; a research question, not a promise |

Two consequences follow. First, **a benchmark can score detection but not discovery.** Detection is measured against a list, and for emergence no complete list exists, so anything beyond a certified list is reported as a count with examples, never as a recall rate. Second, "nobody wrote it" is a test of the code, not of the behaviour. We will tune a world until cartels or jams are reachable, but the specific ring, jam or tipping point is not predictable from the settings. That is the standing of Schelling's segregation model, which nobody calls scripted [18]. A benchmark can live at E1 and E2; E3 is the prize. E4 we raise and do not promise, since current models' ability to hide messages remains limited [19].

**The Commons.** Our design for E1 to E3 at 1,000 to 10,000 agents replaces a market where nothing is scarce with one where the protocol layers themselves are finite: the road, the board and the purse. No agent code mentions jams, rings, cartels or segregation. One phenomenon is close to certain from the mechanism: **congestion collapse with hysteresis**, which Bronson et al. call a metastable failure, "an emergent behavior rather than a logic bug" [20]. Under scarcity a vote for a stronger layer may jam the town, and reversing it may not bring the town back. Ground truth comes from the substrate and from causality, never from behaviour: the simulator knows which identities share an owner, when a trigger fired, and what the capacity limits are. The Commons is a design; none of it is built.

## 5 Evidence and testbed

### 5.1 The testbed as built

**NANDA Town** is NANDA's open-source simulator for networks of agents [21]. Its Lab is a seeded simulation of scripted agents: every action passes through the twelve layers, and its eleven scenarios include three attacks with negative controls. But the Lab cannot host an LLM agent, and no existing visualiser draws thousands of agents usefully.

**The Roblox world** is public at https://www.roblox.com/games/74139499982225. Its **Red vs. Blue (live)** mode puts two human teams at the seams the layers define. The town is TownSim, a small re-implementation of NANDA Town's marketplace in Roblox's Luau: two sellers, three buyers and the full deal cycle.

A round lasts three minutes. Red has four attacks: forge an invoice at the gate, or spoof a seller's card on the notice board at a 20% discount. Every attack has a fair, visible tell, such as a courier paying an alias like `se11er-b`, or two cards with one name. Blue flags couriers or cards from a watchtower with a click. An escaped attack is worth +10 to red, a catch +1 to blue, and a false alarm −1. The same players sit on the council and take turns on red; bots fill empty sides.

The council can swap auth between `plain.v1` and `hmac.v1` and the registry between `index.v1` and `index.owned.v1`. A signature check adds 1.1 seconds to each courier hop, and an ownership check adds 4 seconds to each registry lookup.

| layers (auth + registry) | forged invoice | spoofed card | deals per 3-min round | events per min | red wins, bot vs. bot |
|---|---|---|---|---|---|
| `plain.v1` + `index.v1` | gets through | gets through, looks verified | 18.2 | 165 | 9 of 10 |
| `hmac.v1` + `index.v1` (NANDA Town default) | rejected | gets through, marked unverified | 15.0 | 132 | 6 of 10 |
| `hmac.v1` + `index.owned.v1` | rejected | refused | 12.8 | 120 | 0 of 10 |

The strongest setting stops both attacks and settles about 30% fewer deals than the weakest. At the default, the protocol stops one attack and the watchtower must catch the other.

One trader, `seller-b`, can be driven by `gemini-2.5-flash` instead of its script. The model sets a stance, not each move: a quote and a price floor within fixed bands, plus a reason from a closed list. No model text reaches a screen, and no player text reaches the model.

Every other trader is scripted, and red's attacks are scripted moves with designed tells. Red vs. Blue has run only in solo playtests. Designed but not built: a live NANDA Town server behind the Roblox world, and views in which distant agents become dots and districts become heatmaps.

### 5.2 Related work

None of HAL's parts is new. Crowd work ran our experiment early, with people where we have agents [22, 23, 24], and found that relying on one or a few people "to maintain the big picture creates a bottleneck on the size and complexity of task amenable to crowdsourcing" [25]. Kittur et al. [26] proposed layered trees of "worker-leaders, requesters, machine learning systems, and algorithms" as an agenda and noted that "no comparative analyses of the effectiveness of these approaches exist yet". That is still true, so HAL's hierarchy is a hypothesis to measure, not a finding. In human–robot interaction, fan-out is the number of robots one operator can manage. Operator effort grows in proportion to the number of robots when they act independently, and stays constant when the operator can treat the swarm as one entity [27]. HAL keeps the fan-out question but changes the unit: a person is fanned out over cases, not agents. Magentic-UI puts one person in the loop of a team of agents, with co-planning and action guards [28]; that is L2. AI control is HAL's closest

relative. In its founding study, a trusted weaker model scored how suspicious an untrusted model's programs looked, and scarce trusted labour could audit 2% of them [29]. The trusted labour in that study was simulated by GPT-4. We know of no control evaluation that puts real people at the top; that is the gap HAL fills. Control watches one powerful model. HAL watches a population of ordinary agents owned by many parties, where the threat may be one agent, a coalition, or a pattern nobody intended.

## 6 The hackathon series and benchmark

The series climbs the emergence ladder. Each phase starts with limits on agent capabilities, memory and call frequency, relaxes some of them, and hands a little more of the rules to people.

| phase | limited, then relaxed | emergence level | what humans do |
|---|---|---|---|
| 0: Watch the Swarm (one day) | traders scripted; no memory across rounds; layers fixed and announced per round | E0: scripted attacks with visible tells | red builds attack agents and plays live; blue builds a hierarchy for one operator who may open 10 cases per 10-minute round; council votes in the final round |
| 1: fixed rules on the Commons | agents scripted with five numbers; memory within a round only; scarce road, board and purse switched on | E1: aggregates such as congestion collapse and price conventions | blue flags cases, not couriers; red adds attacks that break no rule |
| 2: rules relaxed, agents known | identities mintable at a price; memory across rounds; registered model-driven agents at a capped call rate | E2 and the first E3: rings, segregation, cartels | the council swaps among existing layers mid-game; blue retunes its hierarchy mid-round |
| 3: both open | anyone brings an agent, model or monitor; anyone may propose a thirteenth layer | E3: strategies nobody wrote; E4 as research only | many teams and a council; new layers arrive as pull requests to NANDA Town and are voted in |

**Phase 0: Watch the Swarm** is a one-day pre-hackathon that tests the thesis, the machinery and the metrics before the series, and the day climbs the sports ladder. A humans-only round in the five-agent Roblox town is biking, and sets the baseline. A hierarchy with one operator is paragliding. The afternoon tournament runs 10-minute rounds at 50, 200 and 1,000 agents, and a blue team sharing the watchtower, triage and council is crewed sailing. Much of this is not built: TownSim has not run beyond five agents.

The **safety plane** is measured by catches, escapes and false alarms; flag latency; judgment calls per minute, with each tier's escalation rate; attention-minutes; compute dollars per tier; and deals settled, which stops a team winning by freezing the town. The **agency plane** is measured by human influence (did a vote, swap, override or role switch change what happened next?), comprehension, workload and enjoyment. **Judgment calls are the variable to optimise**, between the scenario's two ends of 132 events and about 1.3 calls a minute. A design that asks for twenty calls a minute is flooding its person; one that asks for none is missing attacks.

Each scenario runs three ways: full automation, approval-based supervision, and active co-engagement. A design that scores well on catches and badly on influence has turned its crew into watchtowers.

**The flat-review condition** tests whether hierarchies hide problems. At 1,000 agents the control is a flat reader of a uniform random sample of the stream, on the same attention budget as the hierarchy's person. Two measures follow: the **hiding rate**, certified phenomena that no tier ever surfaced, and **scale blindness**, detection plotted against a phenomenon's size relative to the hierarchy's aggregation bins.

## 7 Risks

**Hierarchies may hide new problems.** This is the sharpest objection to our position. A hierarchy is a filter, and a filter is defined by what it drops: rule monitors catch what someone wrote a rule for, model monitors score what they were prompted to find, and every tier that aggregates turns odd events into one tidy case. Aggregation is at once the only way to see E1 and the surest way to hide E2: nobody sees a ring of fourteen identities in a district heatmap. Our untested prediction is that hierarchies beat flat reading on phenomena larger than one aggregation bin and lose on smaller ones. Two cheap responses: pass a small random fraction of each tier's input, say 0.1%, upward unfiltered, and move the bin boundaries between rounds.

**Monitors may share errors, be attacked, or collude.** The hierarchy assumes its monitors fail independently. Monitors built on one base model, prompted from one threat list, or reading the same aggregates can share a blind spot. Imperfect monitors are the norm: the best in SHADE-Arena reached an AUC of 0.87 [30]. If the monitors are models, agents can write messages aimed at the monitor rather than the buyer, and agents optimised against the monitor learn to hide their intent [31].

**Emergence may not appear, or may need tuning that amounts to scripting.** The likeliest failure is that collapse happens only in a narrow, seed-sensitive range of settings. A parameter sweep exists to find out before anything is promised, and the map of settings should be published with the result.

## 8 Conclusion

People need to stay in the loop of agent populations, and they cannot read them: a five-agent market outruns its watcher, and a 20,000-agent run produces more log in five seconds than one person can read in three weeks. HAL's answer is to count judgment calls, not agents: a fixed budget of attention, a hierarchy that is cheap where volume is high and costly where it is low, and the jobs only people can do — notice the unexpected, play the adversary, change the rules. At one agent that is biking; at a team, paragliding; at a population, a crew that keeps a lookout without becoming one. The emergence ladder says what oversight has to catch, the Commons makes the hard cases real and checkable, and the hackathon series climbs the ladder. The town to start in is already standing.

## References

[1] R. Raskar et al., "Beyond DNS: Unlocking the Internet of AI Agents via the NANDA Index and Verified AgentFacts," arXiv preprint arXiv:2507.14263, 2025. arXiv: 2507.14263. [Online]. Available: https://arxiv.org/abs/2507.14263

[2] J. Piao et al., "AgentSociety: Large-Scale Simulation of LLM-Driven Generative Agents Advances Understanding of Human Behaviors and Society," arXiv preprint arXiv:2502.08691, 2025. arXiv: 2502.08691. [Online]. Available: https://arxiv.org/abs/2502.08691

[3] L. Hammond et al., "Multi-Agent Risks from Advanced AI," Cooperative AI Foundation, Technical Report 1, 2025. arXiv: 2502.14143. [Online]. Available: https://arxiv.org/abs/2502.14143

[4] J. C. R. Licklider, "Man–Computer Symbiosis," *IRE Transactions on Human Factors in Electronics*, vol. HFE-1, no. 1, pp. 4–11, Mar. 1960.

[5] D. C. Engelbart, "Augmenting Human Intellect: A Conceptual Framework," Stanford Research Institute, Tech. Rep. AFOSR-3223, Oct. 1962. [Online]. Available: https://www.dougengelbart.org/content/view/138/

[6] S. Jobs. "Interview with WGBH." "A computer has always been a bicycle of the mind.", 1990. [Online]. Available: https://allaboutstevejobs.com/videos/misc/future_of_pc_1990

[7] E. Horvitz, "Principles of Mixed-Initiative User Interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*, 1999, pp. 159–166.

[8] NANDA, "1st IEEE Workshop on Networked AI Agents in a Decentralized Architecture (NANDA), co-located with IEEE TPS 2026," Workshop website, 2026, San Jose, CA, November 6, 2026. Accessed 2026-10-26. [Online]. Available: https://projectnanda.org/workshops/ieeetps26/

[9] L. Bainbridge, "Ironies of Automation," *Automatica*, vol. 19, no. 6, pp. 775–779, 1983.

[10] M. R. Endsley and E. O. Kiris, "The Out-of-the-Loop Performance Problem and Level of Control in Automation," *Human Factors*, vol. 37, no. 2, pp. 381–394, Jun. 1995.

[11] B. Agachan, M. van Duijn, and A. Zohrehvand, "Loop-Back Authority in LLM Agent Teams: A Paired Experiment on Flat and Hierarchical Coordination," arXiv preprint arXiv:2609.14767, 2026. arXiv: 2609.14767. [Online]. Available: https://arxiv.org/abs/2609.14767

[12] R. Parasuraman and D. H. Manzey, "Complacency and Bias in Human Use of Automation: An Attentional Integration," *Human Factors*, vol. 52, no. 3, pp. 381–410, 2010.

[13] D. Retelny, M. S. Bernstein, and M. A. Valentine, "No Workflow Can Ever Be Enough: How Crowdsourcing Workflows Constrain Complex Work," *Proceedings of the ACM on Human-Computer Interaction*, vol. 1, no. CSCW, pp. 1–23, 2017, Article 89.

[14] N. Kosmyna et al., "Your Brain on ChatGPT: Accumulation of Cognitive Debt when Using an AI Assistant for Essay Writing Task," arXiv preprint arXiv:2506.08872, 2025. arXiv: 2506.08872. [Online]. Available: https://arxiv.org/abs/2506.08872

[15] H.-P. ( Lee et al., "The Impact of Generative AI on Critical Thinking: Self-Reported Reductions in Cognitive Effort and Confidence Effects From a Survey of Knowledge Workers," in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, ACM, 2025, pp. 1–22.

[16] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*. New York: Harper & Row, 1990.

[17] D. J. Harris, K. L. Allen, S. J. Vine, and M. R. Wilson, "A Systematic Review and Meta-Analysis of the Relationship between Flow States and Performance," *International Review of Sport and Exercise Psychology*, vol. 16, no. 1, pp. 693–721, 2023, Published online 2021.

[18] T. C. Schelling, "Dynamic Models of Segregation," *The Journal of Mathematical Sociology*, vol. 1, no. 2, pp. 143–186, 1971.

[19] S. R. Motwani et al., "Secret Collusion among AI Agents: Multi-Agent Deception via Steganography," in *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024. arXiv: 2402.07510. [Online]. Available: https://arxiv.org/abs/2402.07510

[20] N. Bronson, A. Aghayev, A. Charapko, and T. Zhu, "Metastable Failures in Distributed Systems," in *Proceedings of the Workshop on Hot Topics in Operating Systems (HotOS '21)*, ACM, 2021, pp. 221–227.

[21] NANDA, "NANDA Town," GitHub repository, 2026, Repository created 2026-05-15. Accessed 2026-10-26. [Online]. Available: https://github.com/projnanda/nandatown

[22] A. Kittur, B. Smus, S. Khamkar, and R. E. Kraut, "CrowdForge: Crowdsourcing Complex Work," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST)*, ACM, 2011, pp. 43–52.

[23] A. Kulkarni, M. Can, and B. Hartmann, "Collaboratively Crowdsourcing Workflows with Turkomatic," in *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW)*, ACM, 2012, pp. 1003–1012.

[24] D. Retelny et al., "Expert Crowdsourcing with Flash Teams," in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST)*, ACM, 2014, pp. 75–85.

[25] N. Hahn, J. Chang, J. E. Kim, and A. Kittur, "The Knowledge Accelerator: Big Picture Thinking in Small Pieces," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ACM, 2016, pp. 2258–2270.

[26] A. Kittur et al., "The Future of Crowd Work," in *Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW)*, ACM, 2013, pp. 1301–1318.

[27] A. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis, "Human Interaction With Robot Swarms: A Survey," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 9–26, 2016.

[28] H. Mozannar et al., "Magentic-UI: Towards Human-in-the-Loop Agentic Systems," Microsoft Research, Tech. Rep. MSR-TR-2025-40, Jul. 2025. [Online]. Available: https://www.microsoft.com/en-us/research/publication/magentic-ui-report/

[29] R. Greenblatt, B. Shlegeris, K. Sachan, and F. Roger, "AI Control: Improving Safety Despite Intentional Subversion," in *Proceedings of the 41st International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 235, PMLR, 2024, pp. 16 295–16 336. arXiv: 2312.06942. [Online]. Available: https://proceedings.mlr.press/v235/greenblatt24a.html

[30] J. Kutasov et al., "SHADE-Arena: Evaluating Sabotage and Monitoring in LLM Agents," arXiv preprint arXiv:2506.15740, 2025. arXiv: 2506.15740. [Online]. Available: https://arxiv.org/abs/2506.15740

[31] B. Baker et al., "Monitoring Reasoning Models for Misbehavior and the Risks of Promoting Obfuscation," arXiv preprint arXiv:2503.11926, 2025. arXiv: 2503.11926. [Online]. Available: https://arxiv.org/abs/2503.11926