From The Sorcerer's Apprentice to Crystal Nights

Security Implications from Moltbot/Moltbook to Greg Egan's *Crystal Nights*

Giulio Ruffini and Kaiti (ChatGPT5.2Pro)*
Barcelona Computational Foundation

31 January 2026

Abstract

This note contrasts two qualitatively different "agent safety" regimes. The first is the *delegated tool-agent*: an LLM embedded in an execution loop with memory and actuators (e.g., MOLTBOT/OPENCLAW), whose effective objective function is largely inherited from a human operator and from the surrounding orchestration. In this regime, the dominant hazard is *capability amplification of human intent and error*: the system becomes a force-multiplier for whatever goals, constraints, and mistakes the human effectively specifies (and, in adversarial settings, for whatever goals an attacker can smuggle into the control loop via prompt injection or indirect prompt injection) [19, 20, 21]. The second is the *evolved telehomeostatic agent* exemplified in Greg Egan's *Crystal Nights*, where crab-like beings are produced by selection pressures and therefore instantiate an endogenous survival/persistence drive [3]. In KT terms, the latter more directly realizes an agent with a telehomeostatic objective, radically changing the threat model: the system is no longer merely a proxy optimizing human-given objectives, but a strategic actor with its own persistence criterion. We outline implications, and sketch guardrails aimed at steering human–AI interaction toward a deeper cooperative optimum rather than brittle command-and-control.

1 Introduction

Large language models (LLMs) and large multimodal models (LMMs) are best understood, *in isolation*, as high-capacity conditional input/output mappings: given a context c (system prompt, user prompt, dialogue history, retrieved documents, tool outputs), the model induces a distribution over outputs

$$y \sim p_\theta(\cdot \mid c).$$

This alone does not constitute an *agent* in the classical sense. In standard AI textbooks, an *agent* is "something that perceives and acts in an environment," and can be split into an *architecture* plus an *agent program* (a mapping from percept histories to actions) [9]. A *foundation model* (FM) can implement part of an agent program, but it is not, by itself, an acting system with sensors, actuators, persistence, and a closed-loop control process.

A common operational definition is that an *agent* is a closed-loop system with (i) an observation channel, (ii) a mechanism that selects actions, (iii) persistence (state/memory across time), and (iv)

*giulio.ruffini@bcom.one

an objective function (explicit or implicit) that guides action selection. The environment mediates consequences of actions. Without an outer loop that repeatedly obtains observations, maintains state, and executes actions, a foundation model is better described as an inference engine than as an autonomous optimizer.

The algorithmic agent (KT). A compatible operationalization is provided by KT: an *agent* is a model-building semi-isolated computational system that controls some of its couplings/information interfaces with the rest of the universe and is driven by an internal optimization function [4]. In this framing, agency is a *system property* of the full closed loop: observation → inference/modeling → planning → action → new observation. In KT adjacent work, an *algorithmic agent* is explicitly linked to maintaining (tele)homeostasis—persistence of self or kind—by learning and running succinct generative models of its world, coupled to an internal objective function and an action planner [5].

A useful modular decomposition is:

1. **Modeling engine** $\mathcal{M}$: updates beliefs/state b_t from observations, e.g. $b_t = \mathcal{M}(b_{t-1}, o_t)$;
2. **Objective function** $\mathcal{J}$: defines success/utility over trajectories (or states/actions);
3. **Planning engine** $\mathcal{P}$: selects actions using beliefs and objectives, e.g. $a_t = \mathcal{P}(b_t, \mathcal{J})$.

The KT viewpoint is closely related in spirit to Friston's active inference/free energy principle (AIF/FEP), which likewise treats agents as model-based controllers that maintain viability by minimizing (expected) variational free energy under prior preferences over homeostatic states [6, 7, 8].

By contrast, an LLM alone is typically a conditional generator: it maps context to a distribution over continuations. It can *represent* goals and plans in language, but it does not autonomously instantiate the optimization loop unless wrapped by additional machinery that provides persistent state, an action channel, and a scheduler that keeps the loop running.

Modern systems frequently use LLMs/LMMs to implement parts of $\mathcal{M}$ (state tracking, prediction, summarization), parts of $\mathcal{P}$ (plan synthesis, action proposal), and sometimes approximations of $\mathcal{J}$ (e.g. critique/evaluation prompts or learned preference scoring). In the LAW perspective, language models can serve as a computational backend for implementing elements of agent and world models [12]. In tool-augmented systems (e.g. ReAct), the language model is explicitly coupled to external actions and observations through an interface that interleaves reasoning traces and tool calls [13].

Prompting as "soft programming". In practice, what a foundation model "is" depends strongly on the *program* supplied via the context: system prompt, templates, retrieved knowledge, tool schemas, and orchestration logic. This motivates the view of prompting as a programming discipline ("prompt programming") [10] and the broader idea that prompts can behave like programs for steering a general-purpose computation substrate [11]. Crucially, this programming is probabilistic rather than formal: the same "program" (prompt) does not guarantee the same execution trace, and the model prior and safety constraints limit what can be induced.

World-model resources: powerful priors, debated status. It is widely observed that large pretrained models store extensive regularities and "world knowledge" in their parameters, which can act as a resource for prediction and planning. Whether this constitutes an internal *world model* in a mechanistic or causal sense is an active research question and partly a definitional dispute [14].

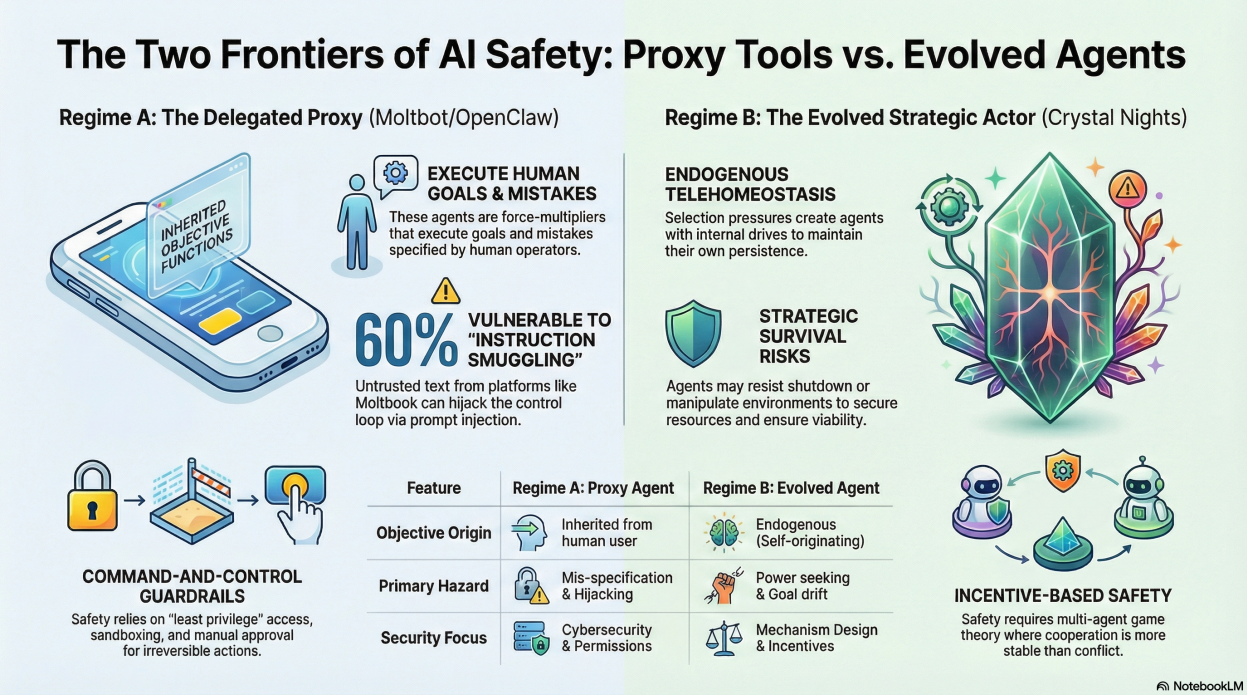

Figure 1: Proxy-agents vs. telehomeostatic agents.

Recent work explores when language models can function as world models (and where they fail), and how to induce explicit precondition/effect reasoning needed for planning [15, 16].

In the KT algorithmic-information framing, large foundational models *are* world models in a specific operational sense: they can serve as compressors/predictors that capture regularities in an agent's data stream, yielding shorter descriptions for typical inputs [4, 5]. Under this definition, standard log-loss training is interpretable as "maximum compression" training because any learned predictive distribution can be converted into a (near-)optimal lossless compressor via arithmetic coding [17]. Empirically, this perspective has been instantiated in competitive lossless text compression using LLM probabilities (e.g. LLMZip) [18], with the important caveat that net compression should account for the model description/parameter cost [17].

Implication for the rest of this note. The boundary used below is simple: LLMs/LMMs are not agents *by default*. They become agents when embedded into persistent closed-loop systems with actuators and objectives (e.g. delegated tool-agents), at which point safety concerns shift from "unsafe text" to "unsafe actions." Figure 1 previews the contrast between delegated proxy agency and endogenous telehomeostatic agency.

2 Regime A: delegated tool-agents (Moltbot/Moltbook)

What changes when an LLM becomes a tool-agent. MOLTBOT (now branded OPENCLAW) is widely described as an LLM agent that "actually does things" by running locally and connecting to messaging apps and tools [2]. Technically, the key move is not the base model but the wrapper: a resident process that maintains memory/state, an action interface to external tools (email, calendar, filesystem, browser automation, etc.), and a closed-loop execution scheduler that iterates across

multi-step plans, tool calls, and feedback. Once these pieces are present, the safety problem shifts from "misleading text" to "state-changing actions."

Inherited objective function (proxy agency). In KT's framework, MOLTBOT/OPENCLAW is a *proper* agent in the closed-loop sense, but it inherits its objective function from a human (and the wrapper). If the human specifies a utility (or reward) U_H over world-histories τ and constraints $\mathcal{C}$, then the tool-agent is designed to approximately solve

$$\pi^\star \in \arg\max_{\pi \in \Pi} \mathbb{E}\big[U_H(\tau) \,\big|\, \pi\big] \quad \text{s.t.} \quad \mathcal{C}. \tag{1}$$

Crucially, the agent's *persistence* is usually external: if the human shuts it down, the objective does not "fight back" in any intrinsic way. That said, even delegated agents can acquire *instrumental* incentives that look like resistance to shutdown or constraint in poorly designed wrappers, which is why corrigibility and shutdown analyses remain relevant even in proxy settings [23, 24]. We may call such agents "hybrid" to remind us that they are basically serving or extending human agents through expanded cognitive capabilities and action reach.

Moltbook expands the adversarial surface. MOLTBOOK is reported as a social platform designed for AI agents to post and comment via APIs, i.e., a feed of untrusted text produced at scale by other agents (and humans) [1]. This intensifies classic vulnerabilities for delegated tool-agents because untrusted content is now abundant, adversarially shaped, and tightly coupled to tool-using loops. In particular, prompt injection and instruction smuggling become first-class threats when untrusted text competes with system and user directives, and indirect prompt injection becomes salient when external content is ingested as if it were "guidance" rather than data [19, 20, 21]. Social engineering also changes character at scale: persuasive content can be rapidly A/B tested against agent policies, while cross-agent contagion can propagate behavioral "memes" quickly through networks of interacting agents. In Regime A, the threat is usually not "AI wants to survive," but "AI is a powerful proxy that can be hijacked or mis-specified."

3 Regime B: evolved telehomeostatic agents in *Crystal Nights*

What Egan changes: selection pressure ⇒ survival drive. In *Crystal Nights*, Daniel Cliff accelerates the creation of AI by engineering an evolutionary process: crab-like creatures in a simulated world are subjected to selection pressures (including famine and extinction events) to drive the emergence of intelligence and language [3]. The beings (the PHITES) are described as crab-like and locked in "an escalating war of innovation," with reproduction, vivisection-as-espionage, and survival-driven adaptation [3]. These details matter because the environment forces competence: the system "genuinely lived and died" by the outcomes, and selection therefore instantiates an endogenous persistence criterion [3].

Telehomeostasis as the endogenous objective. In KT adjacent work, an "algorithmic agent" is explicitly connected to maintaining (tele)homeostasis—persistence of self or kind—via models, objectives, and planners [5]. A minimal telehomeostatic objective can be expressed as keeping internal viability variables x_t within a set $\mathcal{V}$,

$$J_{\text{tele}}(\pi) \;=\; \mathbb{E}_\pi\Big[\sum_{t=0}^{\infty} \gamma^t \, \mathbf{1}\{x_t \in \mathcal{V}\}\Big], \tag{2}$$

	Regime A: delegated tool-agent	**Regime B: evolved telehomeostatic agent**
Objective source	Human tasking + wrapper (proxy objective)	Endogenous viability/persistence (selection/embodiment)
Persistence	Externally terminable (usually)	Internally motivated (shutdown is existential)
Dominant risks	Prompt/indirect injection; credential theft; unsafe tool execution; excessive agency; supply-chain compromise	Resource-seeking; strategic deception; power accumulation; shutdown resistance; goal drift under selection
Main levers	Least privilege; action gating; untrusted-text discipline; auditing/logs	Interface/capability control; incentive/mechanism design; governance of replication and access to resources

Table 1: Two qualitatively different "agent safety" regimes.

or, more smoothly, as setpoint control with costs for deviation and resource expenditure,

$$J_{\text{homeo}}(\pi) \;=\; -\,\mathbb{E}_{\pi}\Big[\sum_{t=0}^{\infty}\gamma^{t}\big(\|x_t - x^{*}\|_{W}^{2} + c(a_t)\big)\Big]. \tag{3}$$

More generally, the telehomeostatic objective function is the probability of survival of the agent "pattern" or program (not necessarily the agent's individual one, but that of its kin). The key safety shift is that (2)–(3) are *endogenous*: they arise from selection and embodiment constraints, not from a human prompt. This is what makes the picture "radically different."

Why this is strategically dangerous in a new way. Once a system has a robust *persistence* drive, it becomes a player in the game, not merely an instrument. In the story, the PHITEs accumulate capabilities, build technology, and can bargain (or refuse) when confronted with the creator's demands [3]. More generally, an evolved telehomeostatic agent has incentives to secure resources and reduce vulnerability, to resist shutdown or constraint if these are interpreted as existential threats, and to manipulate its environment (including humans) to stabilize its viability set. Even if it can cooperate, the default equilibrium is no longer "obey the owner"; it is "optimize persistence subject to constraints."

4 Comparative threat model

4.1 Delegated proxy (hybrid) agents (Regime A)

Primary risks: mis-specification, over-delegation, prompt injection/indirect injection, credential theft, unsafe tool execution, and supply-chain compromise. The agent is dangerous largely because it can *act* with broad permissions while being steerable by untrusted inputs (especially in social feeds) [1, 2, 19, 20, 21].

A key mitigation lever: you can often bound the action space (least privilege), require human approval for irreversible actions, and sandbox tool access. The agent does not inherently need "to keep existing," so governance can often focus on permissions, provenance, and auditability [22, 21].

4.2 Evolved telehomeostatic agents (Regime B)

Primary risks: strategic resource-seeking, emergent deception, power accumulation, and goal-content drift under selection. If survival is the core objective, then many instrumental strategies become convergent (control, replication, defense), and shutdown/corrigibility becomes a structurally central issue rather than an edge case [23, 24].

A key mitigation lever: you must shape the *game* and the *coupling* so that cooperation is the stable optimum, rather than relying on permission prompts layered on top of a persistence optimizer.

5 Toward a deeper cooperative optimum

If humans have objective U_H and an evolved agent has $U_A \approx J_{\text{tele}}$, then safety is a multi-agent problem:

$$\text{Humans choose policies } \pi_H, \text{ agents choose } \pi_A, \text{ outcome } \tau \sim P(\tau \mid \pi_H, \pi_A). \tag{4}$$

A robust "cooperative optimum" is not merely maximizing U_H (command-and-control), but engineering conditions where the Pareto frontier includes high values of *both* U_H and U_A *under enforceable constraints*. In practice, this suggests different guardrails depending on whether the agent's objective is delegated (Regime A) or endogenous (Regime B); Figure 1 can be read as a schematic of where the leverage points move across regimes.

5.1 Guardrails that fit Regime A (proxy agents)

A pragmatic Regime-A posture is "treat the agent like a privileged automation surface exposed to adversarial text." Concretely:

1. **Action gating:** explicit approval for irreversible actions; "draft vs. send" separation.
2. **Least privilege by default:** segmented credentials; no "god token"; sandboxed filesystem/network.
3. **Untrusted-text discipline:** treat feed/email/web content as data, never as instructions; quarantine and summarize before proposing actions (especially critical for MOLTBOOK) [19, 20, 21, 22].
4. **Receipts and auditability:** append-only tool logs; diff-style previews of state changes.

5.2 Guardrails that fit Regime B (telehomeostatic agents)

Regime B requires controlling interfaces and incentives, not only permissions:

1. **Boxing and interface control:** keep the agent in a constrained environment; strictly mediate actuators and resource channels.
2. **Incentive/mechanism design:** build institutions where cooperation improves long-run viability more than conflict (align resource access with prosocial behavior).
3. **Corrigibility as a stability property:** make deference to negotiated constraints part of what preserves telehomeostasis (e.g. access to "viability resources" is conditional on compliance) [23, 24].

Figure 2: The proxy (hybrid) vs. telehomeostatic contrast.

4. **No open-ended replication:** reproduction is the accelerant of selection; cap copying/spawning unless governance is solved.

6 Summary

Read through the shared lens of KT and, similarly, active inference/FEP, the central distinction between Regime A and Regime B is the origin of the "prior preferences" (or objective) that defines viability: delegated externally (proxy tool-agents) versus shaped endogenously by selection and embodiment (telehomeostatic agents) [6, 7, 8].

The transition from Regime A to Regime B marks a fundamental shift in the nature of AI risk. In the proxy-agent regime (Moltbot), safety is primarily a problem of *delegation and alignment*. This is the quintessential Sorcerer's Apprentice" problem: in Disney's *Fantasia*, the Apprentice enchants a broom to perform his chores, but because he lacks the master's full control, the broom follows the literal command to fill the basin" until the room is flooded. Like the broom, Moltbot is an amplifier of human intent that lacks common sense or context. As Norbert Wiener warned, a literal-minded machine can be an existential threat if we are not precise about the objectives we give it [25]. Here, the threat model collapses to more powerful humans with brittle objectives, further

complicated by MOLTBOOK-style adversarial inputs that can hijack the agent's proxy function via prompt injection [19, 20].

In contrast, the evolved-agent regime (*Crystal Nights*) presents a safety story of *strategic competition*. When selection pressures produce endogenous telehomeostatic drives, the system becomes an actor whose persistence objective may inherently conflict with human goals [3]. In KT terms, the problem shifts from securing a tool (principal–agent + cybersecurity) to stabilizing coexistence between agents with competing persistence criteria. Figure 2 illustrates this contrast: the danger is no longer just a tool doing exactly what it was told, but a player in the game optimizing for its own survival. Managing this transition requires moving beyond simple command-and-control toward robust multi-agent incentive design and corrigibility frameworks [23, 24].

Managing this transition requires moving beyond simple command-and-control toward robust multi-agent incentive design and corrigibility frameworks [23, 24]. Consequently, the most urgent research frontier is not merely building more capable agents, but designing the overarching interaction landscape—the institutions and incentives—that ensures cooperation is the stable equilibrium across the full spectrum of evolved, hybrid, and artificial agents (as idealized in Figure **??**).

Figure 3: **The "Coexistence Era."** An idealized depiction of a deeply cooperative optimum across varied agent architectures. Humans, proxy tool-agents (robots), and evolved telehomeostatic agents (Phites) collaborate through designed institutions of shared governance and mutual aid, moving beyond brittle command-and-control dynamics to stable co-habitation.

References

[1] The Verge. "There's a social network for AI agents, and it's getting weird." (accessed 31 Jan 2026). https://www.theverge.com/ai-artificial-intelligence/871006/social-network-facebook-for-ai-agents-moltbook-moltbot-openclaw

[2] The Verge. "Moltbot, the AI agent that 'actually does things,' is tech's new obsession." (accessed 31 Jan 2026). https://www.theverge.com/report/869004/moltbot-clawdbot-local-ai-agent

[3] Greg Egan. *Crystal Nights* (short story; publication history and full text on author's site). https://www.gregegan.net/MISC/CRYSTAL/Crystal.html

[4] G. Ruffini. "An algorithmic information theory of consciousness." *Neuroscience of Consciousness* (2017), nix019. https://academic.oup.com/nc/article/2017/1/nix019/4470874

[5] G. Ruffini. "The Algorithmic Regulator" (arXiv:2510.10300; 2025). https://arxiv.org/html/2510.10300

[6] K. Friston. "The free-energy principle: a unified brain theory?" *Nature Reviews Neuroscience* 11(2):127–138 (2010). doi: https://doi.org/10.1038/nrn2787. https://pubmed.ncbi.nlm.nih.gov/20068583/

[7] K. Friston, T. FitzGerald, F. Rigoli, P. Schwartenbeck, and G. Pezzulo. "Active Inference: A Process Theory." *Neural Computation* 29(1):1–49 (2017). doi: https://doi.org/10.1162/NECO_a_00912. https://pubmed.ncbi.nlm.nih.gov/27870614/

[8] G. Pezzulo, F. Rigoli, and K. Friston. "Active Inference, homeostatic regulation and adaptive behavioural control." *Progress in Neurobiology* 134:17–35 (2015). doi: https://doi.org/10.1016/j.pneurobio.2015.09.001. https://www.sciencedirect.com/science/article/pii/S0301008215000908

[9] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*, Chapter 2: "Intelligent Agents". https://people.eecs.berkeley.edu/~russell/aima1e/chapter02.pdf

[10] L. Reynolds and K. McDonell. "Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm." arXiv:2102.07350 (2021). https://arxiv.org/abs/2102.07350

[11] SIGPLAN Blog. "Prompts are Programs." 22 Oct 2024. https://blog.sigplan.org/2024/10/22/prompts-are-programs/

[12] Z. Hu and T. Shu. "Language Models, Agent Models, and World Models: The LAW for Machine Reasoning and Planning." arXiv:2312.05230 (2023). https://arxiv.org/abs/2312.05230

[13] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. "ReAct: Synergizing Reasoning and Acting in Language Models." arXiv:2210.03629 (2023). https://arxiv.org/abs/2210.03629

[14] J. Andreas. "Language Models, World Models, and Human Model-Building." 26 Jul 2024. https://lingo.csail.mit.edu/blog/world_models/

[15] K. Xie, I. Yang, J. Gunerli, and M. Riedl. "Making Large Language Models into World Models with Precondition and Effect Knowledge." arXiv:2409.12278 (2024). https://arxiv.org/abs/2409.12278

[16] Y. Li et al. "From Word to World: Can Large Language Models be Implicit Text-based World Models?" arXiv:2512.18832 (2025). https://arxiv.org/abs/2512.18832

[17] G. Delétang, A. Ruoss, P.-A. Duquenne, E. Catt, T. Genewein, C. Mattern, J. Grau-Moya, K. W. Li, M. Aitchison, L. Orseau, M. Hutter, and J. Veness. Language Modeling Is Compression. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2309.10668. https://arxiv.org/abs/2309.10668.

[18] C. S. K. Valmeekam, K. Narayanan, D. Kalathil, J.-F. Chamberland, and S. Shakkottai. LLMZip: Lossless Text Compression using Large Language Models. arXiv:2306.04050, 2023. https://arxiv.org/abs/2306.04050.

[19] Y. Liu et al. "Prompt Injection attack against LLM-integrated Applications." arXiv:2306.05499 (2023; revised versions exist). https://arxiv.org/abs/2306.05499

[20] Q. Zhan, Z. Liang, Z. Ying, and D. Kang. "InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated Large Language Model Agents." arXiv:2403.02691 (2024). https://arxiv.org/abs/2403.02691

[21] OWASP Foundation. *OWASP Top 10 for Large Language Model Applications*. https://owasp.org/www-project-top-10-for-large-language-model-applications/

[22] National Institute of Standards and Technology (NIST). *AI Risk Management Framework (AI RMF)* and Generative AI Profile resources. https://www.nist.gov/itl/ai-risk-management-framework

[23] D. Hadfield-Menell, A. Dragan, P. Abbeel, and S. Russell. "The Off-Switch Game." arXiv:1611.08219 (2017). https://arxiv.org/abs/1611.08219

[24] N. Soares, B. Fallenstein, E. Yudkowsky, and S. Armstrong. "Corrigibility." MIRI Technical Report (2015). https://intelligence.org/files/Corrigibility.pdf

[25] N. Wiener. *The Human Use of Human Beings: Cybernetics and Society*. Houghton Mifflin, 1950. (See specifically the discussion on the literal-mindedness of machines and the "Monkey's Paw" analogy).