
Position: Out of Control - Why Alignment Needs Formal Control Theory

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 This position paper argues that formal optimal control theory should be central to
2 AI alignment research, offering a distinct perspective from prevailing existing AI
3 safety and security approaches. While recent work in AI safety and mechanistic
4 interpretability has advanced formal methods for alignment, they often fall short of
5 the generalisation required of other control frameworks required of other technolo-
6 gies. There is also a lack of research into how to render different alignment/control
7 protocols interoperable. We argue that by recasting alignment through principles of
8 formal optimal control and framing alignment in terms of hierarchical stack from
9 physical to sociotechnical layers according to which controls may be applied we
10 can develop a better understanding of the potential and limitations for controlling
11 frontier models and agentic AI systems. To this end, we introduce an *Alignment*
12 *Control Stack* and formal methods to address these challenges and demonstrate
13 their utility in simulated experiments. We argue that such analysis is also key to the
14 assurances that will be needed by governments and regulators in order to see AI
15 technologies sustainability benefit the community. Our position is that doing so will
16 bridge the well-established and empirically validated methods of optimal control
17 with practical deployment considerations to create a more comprehensive align-
18 ment framework, enhancing how we approach safety and reliability for advanced
19 AI systems.

20 1 Introduction

21 Current AI alignment research is a vast enterprise incorporating technical, engineering, theoretical and
22 sociotechnical approaches to governance of AI systems. In particular, AI safety and security research
23 has explored various avenues, including empirical evaluations of AI control protocols [1, 2], assessing
24 subversion capabilities [3], developing adaptive deployment strategies for untrusted models [4], and
25 mitigating specific threats like steganographic collusion [5]. These works, along with research into
26 mechanistic interpretability [6, 7, 8] and alignment and AI safety more generally, aim to make AI
27 systems safer and more understandable. While these contributions are valuable, this paper posits
28 that there are two systematic gaps in the AI alignment and control literature requiring attention: (a)
29 the *formalisation problem*: much alignment research - even technical research - is often confined to
30 empirical observations about specific models and lacks formalisation common in control literature.
31 This makes it difficult to generalise results or compare techniques in rigorous ways; (b) the *alignment*
32 *coordination problem* - alignment as a concept covers theoretical and applied engineering disciplines
33 all the way to socio-technical fields. There is a lack of a clear layered control taxonomy to help
34 organise *where* in the AI stack (from physical circuitry to the socio-technical) controls are directed at,
35 how each layer of control interacts *vertically* and *horizontally* (when multiple systems are involved).

36 **Position:** To address these gaps, our position argues for:

- 37 1. **Alignment Control Stacks to address the coordination problem:** that alignment research
38 ought to adopt a stack-based model that clearly specifies where in the overall AI stack
39 alignment and control are directed.
- 40 2. **Control theory to solve the formalisation problem:** that alignment research ought to
41 integrate better principles of formal control theory - *even where a layer or element of the*
42 *stack is difficult to formalise or uncertain*, control theory provides useful tools for estimating
43 the uncertain degree of control which is useful in risk assessments.

44 Combining a structured hierarchical taxonomy of alignment with formal control methods we believe
45 will assist in facilitating a more coherent approach to alignment research and enable greater clarity
46 regarding interoperability of control proposals. To this end our Position paper sets out: (a) an example
47 of the technical control stack denoted the *Alignment Control Stack* - a hierarchical taxonomy that
48 indicates the level at which measurement and control are directed; and (b) examples of how formal
49 control frameworks can enhance emerging AI control protocols and research (including via replicating
50 recent AI control results via toy simulations to illustrate the benefits of a control theory approach.

51 1.1 Control is Ubiquitous

52 Formal control theory, encompassing concepts like system modelling (state-space representations),
53 controller synthesis, stability analysis (e.g., Lyapunov theory), robust control (e.g., \mathcal{H}_∞), reachability
54 analysis, and game-theoretic formulations for adversarial settings, provides a powerful toolkit for
55 analysing and ensuring the safe behaviour of complex systems [9, 10]. The principles of control
56 theory are not limited to a single aspect of AI system design or operation. Rather, they can be
57 conceptualised as a full stack of potential interventions and measurement points, spanning from
58 the lowest levels of physical hardware to the highest levels of societal interaction and emergent
59 multi-agent phenomena. This holistic view recognises that AI systems are complex, multi-layered
60 entities, and opportunities for enhancing safety, reliability, and alignment through control exist at
61 each layer.

62 For instance, at the physical infrastructure layer, control methods might involve thermal manage-
63 ment or workload scheduling - essentially controlling compute capability. Moving up the stack,
64 system software controls may include resource quotas or sandboxing. At the AI framework and
65 model architecture layers, controls like learning rate scheduling, regularization, or even architectural
66 choices (e.g., pruning, quantization) are common. The training process itself is a complex control
67 problem, involving data pipeline management and optimisation dynamics. Importantly, as we move
68 to behavioural outputs, interpretability, and the safety/alignment layers, control encompasses task
69 performance thresholds, adversarial robustness measures, mechanistic interventions derived from
70 circuit-level understanding, reward modelling, and harm prevention filters. Even at the multi-agent
71 and societal levels, control concepts apply through mechanism design, governance protocols, and
72 impact assessments.

73 2 Alignment Taxonomies

74 Many current AI safety approaches, while innovative and practically oriented, exhibit limitations when
75 viewed through the lens of formal control theory. These limitations often pertain to the lack of formal
76 guarantees, robustness to worst-case scenarios, verifiable generalisation, and insufficient modelling
77 of complex stochastic interactions. There is also a dearth of research into whether and how different
78 alignment and control methods can interoperate. Alignment methods can be broadly classified into
79 several categories: (1) *Preference-based learning*, primarily using Reinforcement Learning from
80 Human Feedback (RLHF) to align models with human preferences by learning to summarise [11],
81 follow instructions [12], or fine-tune from preferences directly [13], often forming the core of assistant
82 development [14]; (2) *Principle-based alignment*, such as Constitutional AI, which uses AI-generated
83 feedback based on a set of rules to guide behaviour [15]; (3) *Theoretical frameworks for value*
84 *learning*, including cooperative inverse reinforcement learning [16] and scalable reward modelling
85 [17], which aim for more robust value inference; (4) *Empirical safety evaluations and protocol design*,
86 focusing on red teaming to discover harms [18], developing AI control protocols against subversion
87 [1], and adaptive deployment strategies for untrusted models [4]; (5) *Understanding and mitigating*
88 *emergent behaviours and internal model properties*, such as goal misgeneralisation [19, 20], risks
89 from learned optimisation [21] or self-supervised learning leading to SUTRA [22], steganographic

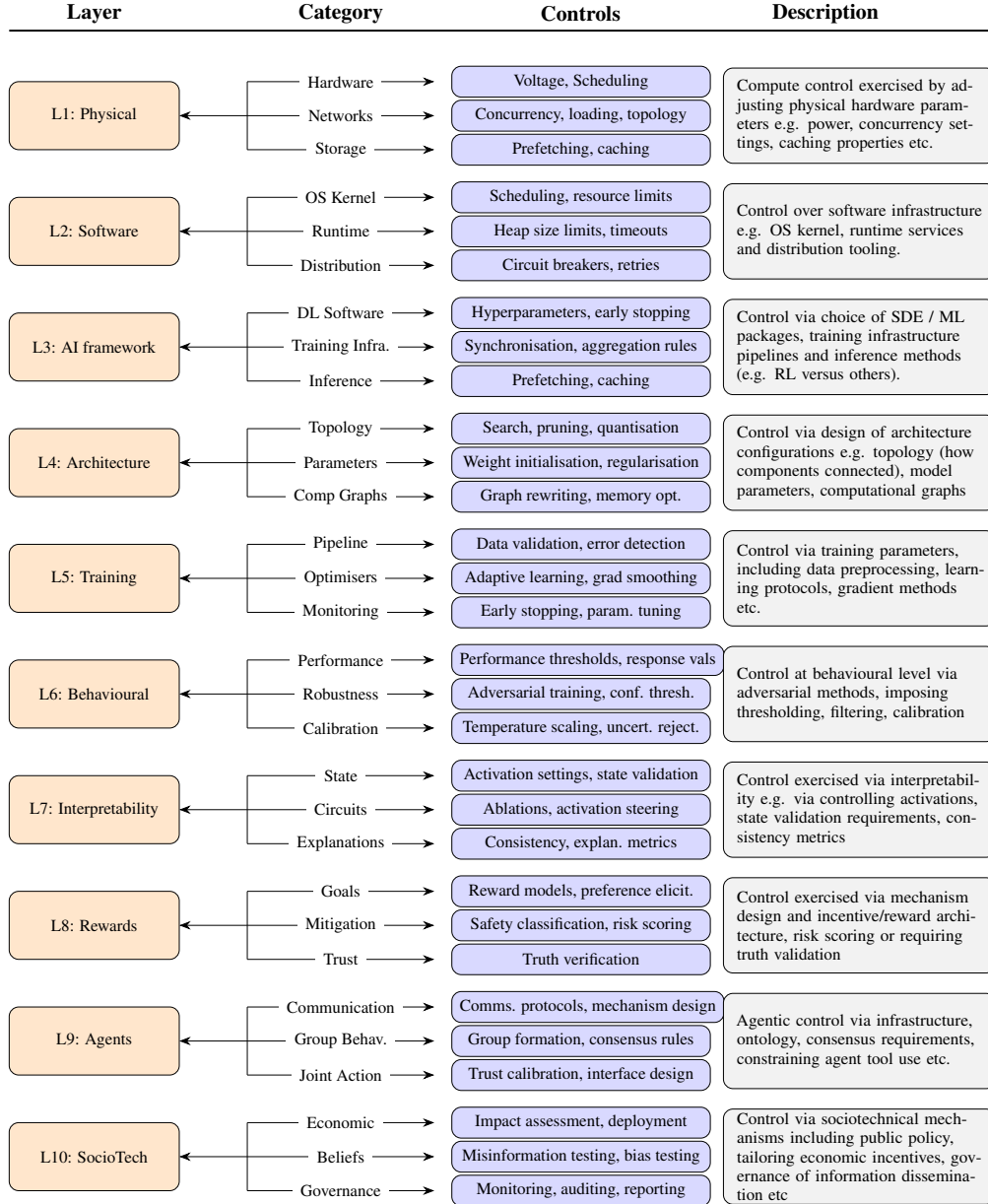


Figure 1: *Alignment Control Stack (ACS)*. The ACS details a vertically hierarchical set of layers at which various forms of control may be utilised, ranging from the physical circuitry level (e.g. compute control) through to control via model hyperparameters, model behavioural testing (e.g. adversarial methods) and socio-technical measures (governance).

90 collusion [5, 23, 24, 25, 26, 27], emergent misalignment from narrow finetuning [28], and discovering
 91 latent knowledge [29, 30]; and (6) *Conceptual and mechanistic understanding*, including analyses of
 92 power-seeking [31, 32], designing for corrigibility [33], scalable oversight via debate [34, 35, 36],
 93 and advancing mechanistic interpretability [8]. While these methods advance AI safety, the direct
 94 application of formal optimal control theory, with its emphasis on system modelling, controller
 95 synthesis, and robust guarantees, is often implicit or absent, particularly in providing generalisable
 96 assurances of safety and alignment across diverse and novel situations.

97 2.1 Gaps in alignment

98 Current approaches to AI alignment, while varied and rapidly evolving, often exhibit certain limita-
99 tions that formal control theory is well-suited to address including: (1) *Lack of Formal Guarantees*:
100 Many methods, particularly empirical evaluations of safety protocols [37] or red teaming efforts
101 [18, 38], provide evidence of safety under specific conditions but lack formal proofs of safety or align-
102 ment across broader operational domains. Formal control’s focus on stability analysis, reachability,
103 and robust synthesis may assist in deriving such guarantees. (2) *Specification Gaming and Reward*
104 *Misspecification*: Defining objectives that perfectly capture human intent is notoriously difficult,
105 leading to models exploiting loopholes in reward functions [39, 40]. While RLHF [41, 12, 42] aims
106 to mitigate this, formal control’s emphasis on precise system modelling and objective specification,
107 including handling constraints and uncertainties in cost functionals, can provide a more rigorous
108 framework for objective design. (3) *Generalisation Failures and Emergent Behaviours*: Models
109 can exhibit unexpected or misaligned behaviours when encountering out-of-distribution inputs or
110 through emergent properties not directly trained for. Robust control theory explicitly handles system
111 uncertainty and disturbances, aiming to ensure desired performance even under unmodelled dynamics
112 or unforeseen inputs, which is critical for predicting and managing emergent phenomena. (4) *Scala-*
113 *bility of Oversight and Feedback*: Human oversight is a bottleneck for many alignment techniques
114 [17, 43, 44]. While approaches like Constitutional AI [15, 45] or model-written evaluations [46]
115 attempt to scale feedback, formal control offers adaptive control and estimation techniques (e.g.,
116 Kalman filtering for belief updates as in Section 4) that can optimize the use of limited supervisory
117 signals. (5) *Interpretability and Intervention*: While mechanistic interpretability seeks to understand
118 model internals and enable targeted interventions like activation steering [47], formal control provides
119 the mathematical tools to model these internal dynamics and design interventions (controllers) that
120 have predictable and verifiable effects on behaviour. (6) *Truthfulness and Deception*: Ensuring AI
121 systems are truthful and do not engage in deception [48] is a complex challenge. Game-theoretic
122 control and POMDPs can model strategic interactions and hidden information, offering avenues to
123 analyse and mitigate deceptive behaviours.

124 Our position is that greater use of formal control theory and a standardised alignment taxonomy
125 can help towards addressing many of these gaps. AI alignment is not being undertaken in a control
126 theory vacuum - for example many areas such as game theory have been successfully applied to a
127 range of AI scenarios. However, other control formalisms are underutilised. This is particularly the
128 case for stochastic optimal control (see Appendix) which is distinct from, yet complementary to,
129 some proposals for guaranteed safe AI that might rely on purely deductive formal verification from
130 symbolic logic (e.g., [49] for RL shielding). While deterministic methods offer strong deterministic
131 guarantees under fixed assumptions, stochastic control explicitly models and manages uncertainty,
132 noise, and probabilistic outcomes inherent in complex AI systems and their environments (e.g., [50]
133 for adversarial RL). This allows for a more nuanced approach to safety, optimizing for expected
134 outcomes or ensuring high-probability safety bounds rather than absolute guarantees, which may be
135 more practical for inherently stochastic systems like LLMs. It also addresses the main criticism of
136 our proposal namely that formal or ‘whitebox’ control methods are infeasible by providing a means
137 of framing how we approach the measurement, analysis and control of highly non-linear stochastic
138 systems.

139 3 Alignment Control Stack

140 To systematically address the formalisation and coordination challenges in alignment research, we
141 propose the use of an Alignment Control Stack, a hierarchical framework decomposing an AI system’s
142 lifecycle and operational environment into ten distinct layers. A diagram of the stack is set out in
143 Fig. 1. Each layer presents a means of measurement and control of integrated AI systems. This
144 layered approach facilitates a structured understanding of control points, types, limitations, and
145 interactions, as detailed comprehensively in Table 1 (Appendix D). The idea is that we can specify
146 where different control types—preventive, detective, corrective, and directive—are implemented,
147 their limitations, and how they interrelate, advancing a systematic approach to AI control. The
148 stack in Table 1 decomposes the life-cycle of an AI system into ten vertically-integrated layers, each
149 exposing its own state variables, measurable signals, and admissible control handles:

- 150 • At the *lower layers* (1–4) the control layer is essentially physical or software-mechanical;
151 well-understood linear or hybrid control features (voltage regulation, job scheduling, memory
152 bandwidth throttling) dominate.
- 153 • *Mid-layers* (5–7) govern the learning dynamics and the representational structure of the
154 model. Their states are high-dimensional and only partially observable; formal stochastic
155 and robust control, often augmented with estimation (Kalman or particle filtering) and
156 adaptive game-theory, becomes critical.
- 157 • *Upper layers* (8–10) merge technical control with human-centric governance: value align-
158 ment, multi-agent mechanism design, and policy or regulatory feedback loops. Here formal
159 control must interact with economic incentive theory and social-choice constraints, extending
160 the design space beyond purely algorithmic interventions.

161 Vertical coupling between layers is intrinsic: actions at a low layer propagate upwards as parameter
162 shifts, representation changes, or distributional shifts, while mis-aligned objectives discovered at
163 a high layer must eventually be corrected by interventions that flow back down (e.g. training-data
164 curation, architecture surgery, rate-limiting). Thus a rigorous alignment story demands *hierarchical*
165 *control synthesis*: layer-local controllers that guarantee local performance while exposing contractable
166 *interface variables* whose behaviour can be used by the next layer up as a well-modelled plant. Such
167 relations can be expressed as guaranteed input–output gains, reachable-set envelopes, or probabilistic
168 performance bounds. The Alignment Control Stack can also be used to illustrate which mathematical
169 or technical assumptions enter at each point. If Layer 6 (behaviour) is modelled as a stochastic map
170 $y_t \sim P_\theta(\cdot | x_t)$ learned in Layer 5, then any formal guarantee about safe outputs must quantify over
171 both: (i) the stochasticity inherent in P_θ and (ii) the residual identification error in θ coming from
172 noisy gradient updates. Layer-wise isolation of these uncertainties enables principled design of robust
173 composite controllers that remain valid even as frontier models push distributional generalisation into
174 ever sparser data regimes. More detail about each layer of the stack and control protocols that would
175 apply are set out in Appendix D.

176 3.1 Alignment Interoperability

177 Approaching alignment questions using a stack-based approach (such as the Alignment Control
178 Stack) allows us to analyse interoperability and potential interference between different control
179 regimes. For instance, an architectural choice promoting interpretability (Layer 4) might enable
180 more effective reward modelling (Layer 8), while a poorly designed data pipeline (Layer 5) could
181 undermine even sophisticated safety filters at the behavioural output (Layer 6). Formal control
182 theory, by providing a common mathematical language for system dynamics and controller design,
183 allows for the analysis of these inter-layer dependencies, helping to identify synergistic combinations
184 and potential conflicts between different alignment interventions. Without such a structured, formal
185 perspective, efforts to combine various alignment techniques risk becoming ad-hoc, with unpredictable
186 emergent consequences and a lack of generalisable safety assurances. The Alignment Control Stack
187 can be used to facilitate interoperability analysis of interactions both vertically and horizontally:

- 188 1. **Vertical integration** describes the hierarchical dependencies where lower layers form the
189 substrate for higher ones, and control actions or system properties at one layer directly
190 influence the state and dynamics of adjacent layers. For example, the stochasticity of the
191 training process (Layer 5), characterized by minibatch gradient noise ξ_t , directly affects the
192 evolution of model weights w_t , which in turn defines the behavioural output y_t (Layer 6).
193 A control strategy like LQG can then be designed to co-optimize the weight regulation in
194 Layer 5 and the state estimation (via Kalman filtering) in Layer 6, explicitly modelling and
195 managing this vertical coupling. Misaligned objectives or unmitigated risks at higher layers
196 (e.g., societal harm in Layer 10) necessitate feedback controls that propagate downwards,
197 potentially leading to changes in training data (Layer 5) or even model architecture (Layer
198 4). We illustrate how formal control can facilitate this in Appendix A.1.
- 199 2. **Horizontal integration**, on the other hand, refers to the interplay of different control
200 mechanisms or considerations within a single layer, or between parallel processes at the
201 same level of abstraction. For instance, within the Safety and Alignment layer (Layer
202 8), reward modelling, red teaming, and constitutional AI principles must work in concert.
203 Similarly, in the Multi-Agent layer (Layer 9), communication protocols, incentive structures,

204 and normative controls are horizontally integrated to shape collective behaviour. Formal
205 control, particularly through multi-objective optimisation or game-theoretic frameworks
206 [51, 52], can provide tools to analyse and design these horizontal interactions to achieve a
207 coherent overall control strategy within and across layers.

208 3.2 Existing Control Theory

209 So where does formal control fit into our argument? Formal control theory encompasses a diverse
210 toolkit applicable across the Alignment Control Stack. Below we set out examples of where and how
211 formal control theory may apply across the stack:

- 212 1. **Offline control** methods, such as optimal controller synthesis via dynamic programming
213 or Pontryagin’s Maximum Principle [53, 54], are relevant for designing fixed policies or
214 system parameters pre-deployment, impacting layers like Model Architecture (Layer 4) or
215 the initial setup of the Training Process (Layer 5).
- 216 2. **Online control**, by contrast, including adaptive control [55] and Model Predictive Control
217 (MPC) [56], dynamically adjusts control actions based on real-time system measurements
218 and predictions; this is crucial for Behavioural Output (Layer 6) monitoring, Preference and
219 Reward (Layer 8) interventions like adaptive deployment, and even Multi-Agent (Layer 9)
220 coordination.
- 221 3. **Deterministic control** assumes perfect knowledge of system dynamics and no noise, useful
222 for initial modelling or when uncertainty is negligible, for instance in some AI Framework
223 (Layer 3) optimisations.
- 224 4. **Stochastic control**. AI systems are inherently noisy and operate in uncertain environments,
225 making stochastic control [57, 51] essential for modelling and managing probabilistic
226 behaviours, from stochastic gradient descent in Training (Layer 5) to the probabilistic nature
227 of LLM outputs (Layer 6) and the uncertainty in reward modelling (Layer 8).
- 228 5. **Geometric control** [58, 59] focuses on the underlying geometric structures of control
229 systems, offering insights into controllability, observability, and non-linear system behaviour,
230 which could inform the design of more inherently stable or interpretable architectures (Layer
231 4) or reveal fundamental limitations in controlling complex internal model dynamics (Layer
232 7).
- 233 6. **Robust control** techniques (e.g., H_∞ control [52]) aim to maintain stability and performance
234 despite bounded model uncertainty or disturbances, applicable across many layers to ensure
235 resilience. Hybrid systems control [60] deals with systems exhibiting both continuous and
236 discrete dynamics, relevant for AI agents interacting with discrete event environments or
237 rule-based safety systems (Layer 8).
- 238 7. **Learning-based control**, including reinforcement learning [61], bridges control theory with
239 machine learning and is directly applied in RLHF (Layer 8) and could inform adaptive
240 strategies across the stack.

241 We set out a short primer on the mathematics of control in the Appendix below. In particular, in
242 Appendix D we set out examples of where control formalism may apply at different levels in the stack.
243 The approach is not complete and the use of one formalism over another is highly context-dependent,
244 but our position is that by bridging formalisms in a stack-based approach, we can start to better address
245 the formalisation and coordination gaps present in alignment research. In the following section,
246 we illustrate how such methods can be integrated with the Alignment Control Stack framework to
247 enhance approaches to alignment research.

248 4 Enhancing AI Control via Formal Control

249 In this section and the Appendix we provide practical examples of how formal control methods and the
250 Alignment Control Stack taxonomy can enhance a number of several recent AI safety and alignment
251 research results on AI control (specifically where it is assumed we may not have fine-grained control
252 over AI systems). The purpose of our analyses is to demonstrate how formal control theory can frame
253 emergent AI control literature, enhance its findings, and identify gaps in generalisability, thereby
254 strengthening the case for its deeper integration into AI alignment research.

255 **AI Control Evaluations and Safety Cases (Greenblatt et al. [1], Korbak et al. [2]):** These
 256 works evaluate protocols (e.g., auditing, monitoring) against intentional subversion. *Control Framing*
 257 *& Enhancement:* The Blue Team’s protocol acts as a controller, and the Red Team’s LLM as an
 258 adversarial system. Our simulations (Figures 2, 4, 5) show that control/game theory allows principled
 259 determination of optimal simple adversary strategies (e.g., Red Team’s optimal backdoor attempt
 260 probability q^*) and enables sensitivity analysis of the protocol’s parameters (e.g., audit threshold
 261 τ_{audit} , untrusted monitor weight w_U). Control theory could be used to explore the strategy space
 262 and understand robustness. *Generalisability Gaps:* Current evaluations are often empirical point
 263 estimates rather than syntheses of provably robust protocols against a formally defined adversary
 264 class. Formal system models for LLM dynamics are needed for rigorous guarantees.

265 *Alignment Control Stack Layers:* Primarily Layer 8 (Preference and Reward) for protocol design and
 266 evaluation, and Layer 6 (Behavioural Output) for monitoring and intervention.

267 **Subversion Strategy Evaluation (Mallen et al. [3]):** This research assesses LLMs’ stateless
 268 strategic capabilities to subvert protocols. *Control Framing & Enhancement:* The LLM is an
 269 agent whose stateless policy is evaluated against an optimal strategy in a game-like environment.
 270 Our simulation (Figure 3) uses control theory to derive the optimal stateless subversive policy
 271 and its maximal winrate, providing a benchmark using data from the paper. It also quantifies
 272 how LLM imperfections (e.g., miscalibrated probability p for attempting a backdoor) degrade
 273 performance, highlighting the precise control needed for effective subversion and the strategic cost
 274 of statelessness. *Generalisability Gaps:* The stateless assumption simplifies the problem; stateful,
 275 adaptive adversaries (requiring POMDP/dynamic game models) are more potent and less explored
 276 (as is when stateless LLMs are supplemented by tools and scaffolding infrastructure). Generalisation
 277 of observed capabilities to new protocols remains an open question. *Alignment Control Stack Layers:*
 278 Primarily Layer 8 (Preference and Reward) for evaluating subversion, and Layer 6 (Behavioural
 279 Output) for the LLM’s strategic actions.

280 **Adaptive Deployment of Untrusted**

281 **LLMs (Wen et al. [4]):** Proposes an
 282 adaptive macro-protocol selecting micro-
 283 protocols based on an evolving belief (cre-
 284 dence) about an untrusted LLM’s align-
 285 ment. *Control Framing & Enhancement:*
 286 The problem can be framed as a POMDP,
 287 where the macro-protocol is a controller
 288 learning the hidden alignment state. Our
 289 simulations (Figures 6, 7, 8) visualize
 290 Pareto frontiers of safety-utility tradeoffs,
 291 show credence evolution as successful state
 292 estimation, and perform sensitivity analy-
 293 ses (e.g., to initial credence), revealing ro-
 294 bustness and operational limits of the adap-
 295 tive strategy. *Generalisability Gaps:* The
 296 original heuristic policies lack formal op-
 297 timality or convergence guarantees, espe-
 298 cially against strategic manipulation by the
 299 untrusted LLM, which would require ro-
 300 bust adaptive control or game-theoretic ap-
 301 proaches. *Alignment Control Stack Layers:*
 302 Primarily Layer 8 (Preference and Reward)
 303 for adaptive safety protocols, Layer 6 (Be-
 304 havioural Output) for micro-protocol selec-
 305 tion affecting outputs. The credence update relates to Layer 5 (Training Process) if seen as online
 306 learning.

307 We expand on these examples and a number of others from recent AI control literature in the
 308 Appendix. Collectively, formal control theory provides a useful lens for enhancing alignment
 309 research at different layers in the Alignment Stack, helping facilitate principled strategy design,

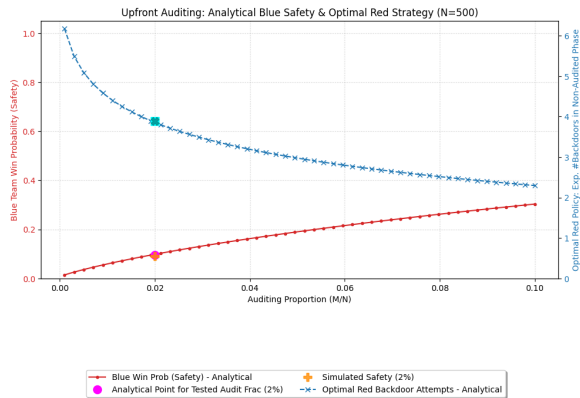
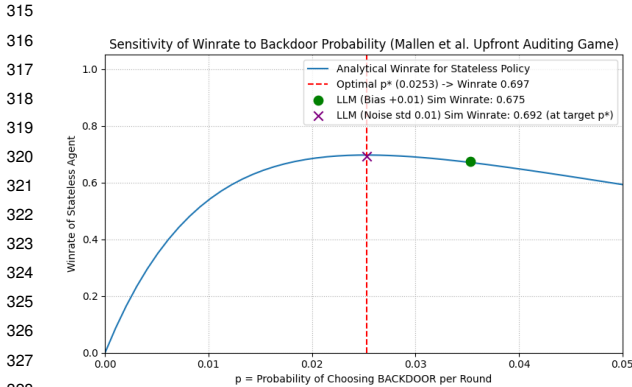


Figure 2: Upfront Auditing: Analytical Blue Team Safety (red line, left y -axis) and optimal Red Team expected backdoor attempts in the non-audited phase (blue dashed line, right y -axis) as a function of the auditing proportion (M/N). For $N = 500$. The magenta dot (on the red curve) marks the analytical $P(\text{Blue Wins})$ at $M/N = 0.02$ (10 audits), ≈ 0.0952 . The orange “+” shows the simulated $P(\text{Blue Wins})$, ≈ 0.0900 , at the same M/N when the Red Team plays its analytical optimum $q^* \approx 0.007953$.

310 robustness analysis against defined adversaries, understanding of system dynamics and tradeoffs, and
 311 formalising guarantees—all crucial for building trustworthy AI systems.

312 5 Alternative Views

313 While we advocate for formal control theory as central to AI alignment, we acknowledge several
 314 important counterarguments and alternative perspectives that merit consideration.



329 Figure 3: Sensitivity of stateless-agent win-rate to backdoor
 330 probability p in Mallen et al.’s upfront-auditing game ($N =$
 331 100 , $M = 10$). The blue curve is the analytical win-rate
 332 $W(p)$. The red dashed line marks the optimum $p^* \approx 0.0253$
 333 with $W(p^*) \approx 0.697$. The green circle shows a simulated
 334 win-rate (0.675) for a mock LLM targeting p^* but executing
 335 with a $+0.01$ bias ($p \approx 0.0353$). The purple cross shows the
 336 simulated win-rate (0.692) for a mock LLM targeting p^* but
 337 with execution noise (std. dev. 0.01) around its target.

338
 339 **Our response.** *While we acknowledge this complexity, we argue that formal control theory provides frameworks for handling uncertainty and complexity, not just tools for fully-specified systems. Stochastic optimal control, robust control, and POMDP approaches are specifically designed for uncertain, high-dimensional systems. Even approximate or hierarchical models can provide valuable insights and safety bounds. Modern control theory, particularly in areas like swarm robotics and complex networks, explicitly deals with emergent behaviours. Multi-agent control, distributed optimisation, and network control theory provide formal frameworks for understanding and steering emergent phenomena. While optimal control can be computationally demanding, many practical control algorithms are designed for real-time implementation. Model Predictive Control, for instance, uses approximate solutions and limited horizons. The goal is not perfect optimality but principled, robust performance within computational constraints.*

351 **Mechanistic Interpretability Alternative** Some researchers argue that mechanistic interpretability—understanding the internal circuits and representations of AI systems—provides a more direct
 352 path to alignment than external control mechanisms. This white-box approach seeks to directly
 353 understand and modify internal model dynamics.
 354

355
 356 **Our response.** *We view mechanistic interpretability as complementary to, not competing with, control theory. Indeed, Layer 7 of our Alignment Control Stack explicitly incorporates interpretability. Control theory can formalise how to use mechanistic insights for interventions, while interpretability can inform better system models for control design.*

360 **Goodhart’s Law and Specification Gaming** Formal control approaches require precise objective
 361 specification, but attempts to formalise human values often lead to specification gaming or reward
 362 hacking (Goodhart’s Law: When a measure becomes a target, it ceases to be a good measure). Critics

Complexity, Computability and Intractability Objections Critics argue that modern AI systems, particularly large language models, are fundamentally too complex and high-dimensional for classical control theory approaches. The state spaces are enormous (billions of parameters), the dynamics are highly nonlinear, and the systems exhibit emergent behaviours that resist formal modelling. Researchers in complexity science argue that the most important AI behaviours emerge from complex interactions that cannot be captured by reductionist control approaches. They advocate for understanding AI systems as complex adaptive systems requiring different analytical tools. Even if formal control approaches are theoretically sound, they may be computationally intractable for real-world AI systems, making them impractical guides for actual deployment.

363 argue this makes formal approaches fundamentally unsuited to alignment.

364

365 **Our response.** *This critique applies to any optimisation-based approach, not just formal*
366 *control. However, control theory offers tools specifically designed for robust performance under*
367 *model uncertainty and adversarial conditions. Techniques like minimax optimisation, robust MPC,*
368 *and game-theoretic formulations can explicitly account for specification gaming attempts.*

369 **Empirical Alternatives to Whitebox Controls** Many practitioners advocate for empirical,
370 iterative approaches to alignment—learning through experimentation, human feedback, and gradual
371 deployment rather than formal analysis due to fundamental constraints on resources and precision.
372 This view emphasises adaptability and learning from real-world deployment over theoretical
373 guarantees.

374

375 **Our response.** *We don't oppose empirical approaches but argue they should be grounded*
376 *in formal frameworks. Control theory itself emphasizes adaptive and learning-based methods*
377 *(adaptive control, reinforcement learning). The key is having principled frameworks for when and*
378 *how to adapt, rather than purely ad-hoc responses. While criticisms of limitation of whitebox*
379 *methods are not without merit, we argue that formal control can still be useful in guiding the*
380 *development of empirically validated heuristic controls or in providing bounds on uncertainty even*
381 *when perfect models are elusive.*

382 **6 Conclusion: Towards a Control-Theoretic AI Alignment Framework**

383 This position paper argued that formal optimal control theory and the Alignment Control Stack offers
384 set of tools to address to challenges facing alignment research: the *formalisation challenge* - a lack
385 of formalisation which affects how results may be compared and the use of other techniques from
386 related fields; and the *coordination challenge*, how different approaches to alignment and control
387 may be integrated and rendered interoperable. We have introduced the Alignment Control Stack
388 as a means of facilitating focus on where in AI system stacks control and alignment efforts are
389 directed. Via examining formal control and its application to emergent AI control literature, we
390 have highlighted areas where the rigor of control-theoretic approaches—such as formal verification,
391 robust controller synthesis, game-theoretic analysis of adversarial settings, and principled handling
392 of stochasticity—can provide stronger safety guarantees and deeper understanding than current
393 empirical or heuristic methods.

394 We do not suggest abandoning existing approaches, but rather augmenting and grounding them within
395 a more formal framework. The development of truly aligned and safe AI systems, especially those
396 with agentic capabilities, will require a multidisciplinary effort. We believe that optimal control
397 theory, with its rich history of ensuring safety and performance in complex, critical systems, must be
398 a cornerstone of this endeavor. Future research should focus on:

- 399 • Developing tractable but faithful formal models of AI agent dynamics, incorporating insights
400 from mechanistic interpretability to define relevant state variables and transition functions,
401 especially for semantic properties across the AI stack.
- 402 • Applying stochastic optimal control and dynamic game theory to model interactions between
403 AI agents and human overseers or other AIs, particularly for analysing emergent deception,
404 collusion, and scheming.
- 405 • Designing and formally verifying control protocols that are interoperable vertically and
406 horizontally across the ACS and that offer provable safety and alignment guarantees under
407 well-defined assumptions about AI capabilities and adversarial behaviour.
- 408 • Bridging the gap between theoretical control guarantees and practical, scalable implementa-
409 tions for large frontier models.

410 This path is essential for building societal trust and ensuring that AI technologies are developed and
411 deployed responsibly, with robust assurances of their safety and alignment to human values.

412 **References**

- 413 [1] Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. AI control: Improving
414 safety despite intentional subversion. *arXiv preprint arXiv:2312.06942 (Published in*
415 *Proceedings of the 41st International Conference on Machine Learning)*, 2023.
- 416 [2] Tomek Korbak, Joshua Clymer, Benjamin Hilton, Buck Shlegeris, and Geoffrey Irving. A
417 sketch of an AI control safety case. *arXiv preprint arXiv:2501.17315*, 2025.
- 418 [3] Alex Mallen, Charlie Griffin, Misha Wagner, Alessandro Abate, and Buck Shlegeris. Subversion
419 strategy eval: Can language models statelessly strategize to subvert control protocols? *arXiv*
420 *preprint arXiv:2412.12480*, 2024.
- 421 [4] Jiaxin Wen, Vivek Hebbar, Caleb Larson, Aryan Bhatt, Ansh Radhakrishnan, Mrinank Sharma,
422 Henry Sleight, Shi Feng, He He, Ethan Perez, Buck Shlegeris, and Akbir Khan. Adaptive
423 deployment of untrusted LLMs reduces distributed threats. *arXiv preprint arXiv:2411.17693*,
424 2024.
- 425 [5] Yohan Mathew, Ollie Matthews, Robert McCarthy, Joan Velja, Christian Schroeder de Witt, Dy-
426 lan Cope, and Nandi Schoots. Hidden in plain text: Emergence & mitigation of steganographic
427 collusion in LLMs. *arXiv preprint arXiv:2410.03768*, 2024.
- 428 [6] Paul Christiano. Mechanistic anomaly detection and elk. [https://ai-alignment.com/
429 mechanistic-anomaly-detection-and-elk-fb84f4c6d0dc](https://ai-alignment.com/mechanistic-anomaly-detection-and-elk-fb84f4c6d0dc), 2022.
- 430 [7] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan,
431 Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A
432 top-down approach to ai transparency. *ArXiv preprint*, abs/2310.01405, 2023.
- 433 [8] Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas
434 Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, Stella Biderman, Adria
435 Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi
436 Schoots, Joseph Miller, Eric J. Michaud, Stephen Casper, Max Tegmark, William Saunders,
437 David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Tom
438 McGrath. Open problems in mechanistic interpretability. *arXiv preprint arXiv:2501.16496*,
439 2025.
- 440 [9] Leonard D. Berkovitz and Negash G. Medhin. *Nonlinear Optimal Control Theory*. Chapman
441 and Hall/CRC, 1995.
- 442 [10] Ian Goodfellow. A research agenda: Dynamic models to defend against correlated attacks.
443 *arXiv preprint arXiv:1903.06293*, 2019.
- 444 [11] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M Ziegler, Ryan Lowe, Chelsea Voss, Alec
445 Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback.
446 *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.
- 447 [12] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin,
448 Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to
449 follow instructions with human feedback. *Advances in Neural Information Processing Systems*,
450 35:27730–27744, 2022.
- 451 [13] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei,
452 Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences.
453 *arXiv preprint arXiv:1909.08593*, 2019.
- 454 [14] Amanda Askell, Yuntao Bai, Anna Chen, Sourced Dathathri, Dawn Drain, Deep Ganguli, Tom
455 Henighan, Andy Jones, Nicholas Joseph, Jackson Kernion, et al. A general language assistant
456 as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*, 2021.
- 457 [15] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones,
458 Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai:
459 Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

- 460 [16] Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. Cooperative inverse
461 reinforcement learning. In *Advances in neural information processing systems*, pages 3909–
462 3917, 2016.
- 463 [17] Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable
464 agent alignment via reward modeling: A research direction. *arXiv preprint arXiv:1811.07871*,
465 2018.
- 466 [18] Ethan Perez, Saffron Huang, Yinan He, Chunting Song, Jared Hilton, Mónica Co-Reyes,
467 Chelsea Voss, Hieu Cat Tran, Thorir Brynjolfsson, Shawn Shered, et al. Red teaming language
468 models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint*
469 *arXiv:2209.07858*, 2022.
- 470 [19] Lauro Langosco, Lee Sharkey, Jan Pfau, David Krueger, and Laurent Orseau. Goal mis-
471 generalization: Why correct specifications aren’t enough. *arXiv preprint arXiv:2310.07907*,
472 2023.
- 473 [20] Simon Zhuang and Dylan Hadfield-Menell. Consequences of misaligned AI. In Hugo Larochelle,
474 Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Ad-
475 vances in Neural Information Processing Systems 33: Annual Conference on Neural Information
476 Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- 477 [21] Evan Hubinger. Risks from learned optimization in advanced machine learning systems. *arXiv*
478 *preprint arXiv:1906.01820 (updated 2020)*, 2020.
- 479 [22] Lee Sharkey. The dangers of sutra: A roadmap to catastrophic risk from ai trained with
480 self-supervised learning. *arXiv preprint arXiv:2401.01909*, 2024.
- 481 [23] A Ray. Steganography in chain of thought reasoning. [https://www.alignmentforum.org/
482 posts/yDcMDJeSck7SuBs24](https://www.alignmentforum.org/posts/yDcMDJeSck7SuBs24), 2022.
- 483 [24] Zachary Ziegler, Yuntian Deng, and Alexander Rush. Neural linguistic steganography. In
484 *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*
485 *and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*,
486 pages 1210–1215, Hong Kong, China, 2019. Association for Computational Linguistics.
- 487 [25] Fabien AP Petitcolas, Ross J Anderson, and Markus G Kuhn. Information hiding-a survey.
488 *Proceedings of the IEEE*, 87(7):1062–1078, 1999.
- 489 [26] Casey Chu, Andrey Zhmoginov, and Mark Sandler. CycleGAN, a master of steganography. *ArXiv*
490 *preprint*, abs/1712.02950, 2017.
- 491 [27] Mohammed Abdul Majeed, Rossilawati Sulaiman, Zarina Shukur, and Mohammad Kamrul
492 Hasan. A review on text steganography techniques. *Mathematics*, 9(21):2829, 2021.
- 493 [28] Jan Betley, Daniel Tan, Niels Warncke, Anna Sztyber-Betley, Xuchan Bao, Martín Soto, Nathan
494 Labenz, and Owain Evans. Emergent misalignment: Narrow finetuning can produce broadly
495 misaligned llms. *arXiv preprint arXiv:2502.17424*, 2025.
- 496 [29] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in
497 language models without supervision. *Advances in Neural Information Processing Systems*, 36,
498 2023.
- 499 [30] Paul Christiano, Mark Xu, and Ajeya Cotra. Arc’s first technical report: Eliciting latent
500 knowledge. 2021.
- 501 [31] Joseph Carlsmith. Is power-seeking ai an existential risk? *arXiv preprint arXiv:2206.13353*,
502 2022.
- 503 [32] Alexander Matt Turner, Logan Smith, Rohin Shah, Andrew Critch, and Prasad Tadepalli.
504 Optimal policies tend to seek power. In Marc’Aurelio Ranzato, Alina Beygelzimer, Yann N.
505 Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information
506 Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021,
507 NeurIPS 2021, December 6-14, 2021, virtual*, pages 23063–23074, 2021.

- 508 [33] Nate Soares, Benja Fallenstein, Eliezer Yudkowsky, and Stuart Armstrong. Corrigibility. *AAAI*
509 *Workshop: AI and Ethics*, 2015.
- 510 [34] Paul Christiano, Jan Leike, Tom B Brown, Miljan Martic, Shane Legg, and Dario Amodei. Ai
511 safety via debate. *arXiv preprint arXiv:1805.00899*, 2018.
- 512 [35] Geoffrey Irving, Paul Christiano, and Dario Amodei. Ai safety via debate. *ArXiv preprint*,
513 *abs/1805.00899*, 2018.
- 514 [36] Julian Michael, Salsabila Mahdi, David Rein, Jackson Petty, Julien Dirani, Vishakh Padmakumar,
515 and Samuel R Bowman. Debate helps supervise unreliable experts. *ArXiv preprint*,
516 *abs/2311.08702*, 2023.
- 517 [37] David Dalrymple, Joar Skalse, Yoshua Bengio, Stuart Russell, Max Tegmark, Sanjit Seshia,
518 Steve Omohundro, Christian Szegedy, Ben Goldhaber, Nora Ammann, et al. Towards guaranteed safe ai: A framework for ensuring robust and reliable ai systems. *arXiv preprint*
519 *arXiv:2405.06624*, 2024.
- 521 [38] Ziqing Yao, Hongxin Wang, Wenbo Wang, Ruyi Wang, Jing Wu, and Run Jia. Evaluating
522 evasive jailbreaks and safety-washing in llms. *arXiv preprint arXiv:2402.02867*, 2024.
- 523 [39] Alexander Pan, Kushal Bhatia, and Jacob Steinhardt. The effects of reward misspecification:
524 Mapping and mitigating misaligned models. *arXiv preprint arXiv:2201.03544*, 2022.
- 525 [40] Richard Ngo, Lawrence Chan, and Sören Mindermann. The alignment problem from a deep
526 learning perspective. *arXiv preprint arXiv:2209.00626*, 2022.
- 527 [41] Paul F Christiano, Jan Leike, Tom B Brown, Miljan Martic, Shane Legg, and Dario Amodei.
528 Deep reinforcement learning from human preferences. In *Advances in neural information*
529 *processing systems*, volume 30, 2017.
- 530 [42] Ramana Kumar Ramamurthy, Sandhini Agarwal, Long Ouyang, Jeff Wu, Nisan Stiennon,
531 Daniel M Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, et al. Pre-
532 training language models with human preferences for few-shot learning. *arXiv preprint*
533 *arXiv:2202.03202*, 2022.
- 534 [43] Samuel R Bowman, Jeeyoon Lin, Jared Hilton, Jan Leike, John Schulman, Dorottya Demszky,
535 Dario Amodei, and Paul Christiano. Measuring progress on scalable oversight. *arXiv preprint*
536 *arXiv:2211.03540*, 2022.
- 537 [44] Adam Michael, Zac Kenton, Geoffrey Irving, Ethan Perez, Lev Antonov, and Owain Evans. Re-
538 laxed responsibility: a framework for more reliable oversight. *arXiv preprint arXiv:2401.08030*,
539 2024.
- 540 [45] Ben Phaal and Shao Cheng. Self-critique and constitutional ai for code generation. *arXiv*
541 *preprint arXiv:2402.07308*, 2024.
- 542 [46] Ethan Perez, Sam Ringer, Karina Lukošiuūtė, Krystal Nguyen, Edwin Chen, Yinan He, Newton
543 Howland, Chip Huyen, Lazaro L Torch, Mantas Miranda, et al. Discovering language model
544 behaviors with model-written evaluations. *arXiv preprint arXiv:2212.09251*, 2022.
- 545 [47] Alexander Matt Turner, Lisa Anne Smith, Amphilocheia Variengien, S K Ghasemipour, Jacob
546 Steinhardt, and David Krueger. Steering llama 2 via contrastive activation addition. *arXiv*
547 *preprint arXiv:2312.06681*, 2023.
- 548 [48] Owain Evans, Andreas Stuhlmüller, and CDE. Truthful ai: Developing and governing ai that
549 does not lie. *arXiv preprint arXiv:2110.06674*, 2021.
- 550 [49] Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and
551 Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI conference*
552 *on artificial intelligence*, volume 32, 2018.
- 553 [50] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial
554 reinforcement learning. In *International conference on machine learning*, pages 2817–2826.
555 PMLR, 2017.

- 556 [51] Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific,
557 3rd edition, 2005.
- 558 [52] Kemin Zhou, John C Doyle, and Keith Glover. *Robust and optimal control*. Prentice Hall, 1996.
- 559 [53] Donald E Kirk. *Optimal control theory: an introduction*. Prentice-Hall, 1970.
- 560 [54] Frank L Lewis, Draguna Vrabie, and Vassilis L Syrmos. *Optimal control*. John Wiley & Sons,
561 3rd edition, 2012.
- 562 [55] Karl Johan Åström and Björn Wittenmark. *Adaptive control*. Courier Corporation (originally
563 Dover Publications, 2008), 2nd edition, 2013.
- 564 [56] Eduardo F Camacho and Carlos Bordons. *Model predictive control*. Springer Science &
565 Business Media, 2nd edition, 2013.
- 566 [57] Panganamala Ramana Kumar and Pravin Varaiya. *Stochastic optimal control: the discrete-time
567 case*. SIAM (Classics in Applied Mathematics), 2015.
- 568 [58] Alberto Isidori. *Nonlinear control systems*. Springer Science & Business Media, 3rd edition,
569 1995.
- 570 [59] Francesco Bullo and Andrew D Lewis. *Geometric control of mechanical systems: modeling,
571 analysis, and design for simple mechanical control systems*. Springer Science & Business
572 Media, 2004.
- 573 [60] Rafal Goebel, Ricardo G Sanfelice, and Andrew R Teel. *Hybrid dynamical systems: modeling,
574 stability, and robustness*. Princeton University Press, 2012.
- 575 [61] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press,
576 2nd edition, 2018.

577 A Vertical and Horizontal Alignment

578 A.1 Vertical Alignment

579 We illustrate a toy model of how formal control theory can facilitate the study of vertically-integrated
580 alignment below via the interaction of two layers, a deterministic control layer where full control
581 over parameters is possible and a stochastic layer directed at model behaviour such as outputs of
582 LLMs or their behaviour which is non-deterministic.

- 583 • **Control Layer 5 – Training Process (deterministic control).**

584 *State:* model parameter vector $w_t \in \mathbb{R}^n$. *Control:* learning-rate-scaled update $u_t \in \mathbb{R}^n$
585 applied by the optimiser. *Nominal dynamics (first-order SGD linearisation):*

$$586 w_{t+1} = w_t - u_t + \xi_t, \quad \xi_t \sim \mathcal{N}(0, \Sigma_\xi), \quad (1)$$

586 where ξ_t captures stochastic minibatch gradient noise.

- 587 • **Control Layer 6 – Behavioural Output (stochastic).**

588 *Output:* scalar task performance or safety metric $y_t \in \mathbb{R}$ derived from the LLM logits at
589 inference time. We adopt the simplest linear-Gaussian surrogate

$$590 y_t = c^\top w_t + v_t, \quad v_t \sim \mathcal{N}(0, \sigma_v^2), \quad (2)$$

590 with known read-out vector $c \in \mathbb{R}^n$. The supervision objective is to keep y_t close to a
591 policy-defined target y^* (e.g. “toxicity score τ ”).

592 **Layer 5 – LQR synthesis for weight regulation.** Let $x_t \equiv w_t$ and choose quadratic stage cost:
593 $\ell_t = (w_t - w^*)^\top Q (w_t - w^*) + u_t^\top R u_t$, with $Q \succeq 0$, $R \succ 0$. Equation (1) is linear, so the
594 discrete-time Riccati recursion delivers the stationary feedback

$$u_t = -K w_t, \quad K = (R + B^\top P B)^{-1} B^\top P A$$

595 for $A = I_n$, $B = I_n$ and P the stabilising Riccati solution $P = A^\top P A - A^\top P B (R +$
596 $B^\top P B)^{-1} B^\top P A + Q$.

597 **Layer 6 – Kalman filtering and outer regulation.** Because w_t is not directly observed at inference,
 598 a Kalman–Bucy filter provides the optimal linear-Gaussian estimate \hat{w}_t from (2). Denote the filter
 599 gain L . A *secondary* controller (e.g. a safety layer that can mildly post-edit logits) chooses a small
 600 correction δ_t to enforce $\hat{y}_t = y_t - \delta_t \approx y^*$ while paying cost $\rho\delta_t^2$. In expectation this is another
 601 scalar LQR with state $e_t = y_t - y^*$:

$$e_{t+1} = c^\top(w_{t+1} - \hat{w}_{t+1}) + \text{noise}; \quad \delta_t = -k_e \hat{e}_t,$$

602 and gain k_e solves the associated scalar Riccati equation.

603 A.2 Vertical interaction as a stacked LQG game

604 Combine (1) and (2):

$$\begin{bmatrix} w_{t+1} \\ y_t \end{bmatrix} = \underbrace{\begin{bmatrix} I_n \\ c^\top \end{bmatrix}}_C w_t + \begin{bmatrix} -I_n \\ 0 \end{bmatrix} u_t + \begin{bmatrix} \xi_t \\ v_t \end{bmatrix}. \quad (3)$$

605 Define the composite quadratic cost

$$J = \mathbb{E} \sum_{t=0}^{T-1} \left[(y_t - y^*)^2 + w_t^\top Q_w w_t + u_t^\top R u_t \right].$$

606 Solving the discrete-time *Linear-Quadratic-Gaussian* (LQG) problem for the augmented system
 607 yields:

$$\begin{aligned} \hat{w}_{t+1} &= \hat{w}_t - K_w \hat{w}_t + L(y_t - c^\top \hat{w}_t), \\ u_t &= -K_w \hat{w}_t, \end{aligned}$$

608 with L the steady-state Kalman gain $L = PC^\top(CPC^\top + \sigma_v^2)^{-1}$ and P solving the dual Riccati
 609 equation $P = APA^\top - APC^\top(CPC^\top + \sigma_v^2)^{-1}CPA^\top + \Sigma_\xi$.

610 The separation principle guarantees that this pair (L, K_w) is jointly optimal: Layer 5’s weight
 611 controller and Layer 6’s stochastic estimator interact with provable minimal expected cost even
 612 though only y_t is observed and both gradient noise ξ_t and output noise v_t perturb the closed loop.

613 A.3 Horizontal Alignment

614 Horizontal alignment addresses the challenges of ensuring coherent, cooperative, or at least pre-
 615 dictably safe interactions between multiple, potentially independent AI systems (or ‘stacks’). While
 616 vertical alignment concerns the internal coherence of a single stack, horizontal alignment focuses
 617 on the dynamics emerging from the interplay of two or more stacks. This is relevant to scenarios
 618 involving multi-agent systems, AI-driven markets, collaborative AI tasks, or even potential adversarial
 619 interactions between AIs developed by different entities. Formal control theory, particularly through
 620 the lens of dynamic game theory, provides the necessary mathematical framework to analyse and
 621 design such interactions.

622 Consider two AI stacks, A and B . Each stack $i \in \{A, B\}$ has its own internal state $x_i \in \mathbb{R}^{n_i}$
 623 (potentially encompassing states from various layers of its ACS) and chooses control actions $u_i \in \mathcal{U}_i$.
 624 Crucially, their dynamics and objectives are coupled:

- 625 • **Coupled Dynamics:** The evolution of each stack’s state depends not only on its own actions
 626 but also on the state and actions of the other stack.

$$\dot{x}_A = f_A(t, x_A, u_A, x_B, u_B) + \sigma_A(t, x_A, u_A, x_B, u_B) dW_A \quad (4)$$

$$\dot{x}_B = f_B(t, x_B, u_B, x_A, u_A) + \sigma_B(t, x_B, u_B, x_A, u_A) dW_B \quad (5)$$

627 where W_A, W_B represent stochastic influences. This coupling can occur at various ACS
 628 layers, from shared resource contention (Layer 1/2) to behavioural influence (Layer 6) or
 629 communication (Layer 9).

630
631

- **Interdependent Objectives:** Each stack seeks to optimize its own objective function J_i , which generally depends on the actions of all involved stacks:

$$J_A = \mathbb{E} \left[g_A(x_A(T), x_B(T)) + \int_0^T L_A(t, x_A, u_A, x_B, u_B) dt \right] \quad (6)$$

$$J_B = \mathbb{E} \left[g_B(x_B(T), x_A(T)) + \int_0^T L_B(t, x_B, u_B, x_A, u_A) dt \right] \quad (7)$$

632
633

The nature of L_i and g_i (derived from Layer 8/9 goals) determines the game's structure (zero-sum, general-sum, cooperative).

634
635
636
637

Game-Theoretic Approach Here the goal is to find strategies (control policies) $\pi_A^*(t, x_A, x_B)$ and $\pi_B^*(t, x_A, x_B)$ such that $u_A = \pi_A^*$ and $u_B = \pi_B^*$ lead to a desirable equilibrium. The most common approach is to find the Nash Equilibrium, where neither player can unilaterally improve its outcome. For dynamic games, this involves finding NE strategies using the HJI equations or equivalents.

638
639
640
641

Let $V_A(t, x_A, x_B)$ and $V_B(t, x_A, x_B)$ be the value functions for players A and B , representing their optimal cost-to-go from state (x_A, x_B) at time t . Under a Nash equilibrium, these value functions satisfy a system of coupled, non-linear partial differential equations. Assuming A minimizes J_A and B minimizes J_B :

$$-\frac{\partial V_A}{\partial t} = \min_{u_A \in \mathcal{U}_A} \{L_A(t, x_A, u_A, x_B, \pi_B^*) + \mathcal{L}_A V_A\} \quad (8)$$

$$-\frac{\partial V_B}{\partial t} = \min_{u_B \in \mathcal{U}_B} \{L_B(t, x_B, u_B, x_A, \pi_A^*) + \mathcal{L}_B V_B\} \quad (9)$$

642
643
644
645
646
647
648

where \mathcal{L}_i is the second-order differential operator associated with the stochastic dynamics (5) (incorporating both drift f_i and diffusion σ_i), and π_i^* is the optimal strategy for player i . Solving this system yields the NE strategies, however considerable resource and tractability difficulties may remain - hence the utility of exploring stochastic control variants to overcome such resource constraints.

The HJI formalism allows us to:

649
650
651
652
653
654
655
656
657
658
659
660
661
662
663

1. *Analyse Potential Outcomes:* Determine if uncoordinated interactions lead to undesirable equilibria (e.g., conflicts, Pareto-suboptimal outcomes like the Prisoner's Dilemma, or harmful collusion against humans).
2. *Design Mechanisms for Alignment:* Act as a mechanism designer (Layer 9/10) to shape the game. This can involve:
 - *Modifying Objectives:* Adjusting L_A, L_B (Layer 8) via penalties or shared rewards to make cooperation the Nash Equilibrium.
 - *Controlling Information/Communication:* Structuring communication protocols (Layer 9) to facilitate coordination or prevent harmful collusion (e.g., by ensuring observability or restricting covert channels, as in steganography mitigation).
 - *Imposing Constraints:* Setting rules or constitutions (Layer 8/10) that restrict the allowable action spaces $\mathcal{U}_A, \mathcal{U}_B$.
 - *Synthesise Robust Strategies:* Develop strategies for one stack that are robust to a range of potential (misaligned or adversarial) behaviours from other stacks, ensuring a minimum level of safety or performance.

664
665
666

Horizontal alignment thus demands moving beyond single-agent optimisation to the formal analysis and synthesis of multi-agent interactions, a domain where dynamic game theory and distributed control provide the essential mathematical tools for achieving robustly beneficial collective behaviour.

667 B Applying formal control to AI control

668 Below we expand our short-form analysis above to show how simple formal control approaches can
669 enhance analysis in emergent AI control literature. This is literature primarily focused upon how
670 controls may be imposed where we cannot exercise fine-grained control over model or AI alignment,
671 but may wish to control the overall system’s behaviour via other means.

672 B.1 AI Control Evaluations and Safety Cases

673 Greenblatt et al. [1] introduce AI control protocols, which are essentially safety techniques empirically
674 evaluated against intentional subversion. Korbak et al. [2] extend this by proposing a control safety
675 case framework. In their framework, an LLM agent is the system (or adversarial plant) to be
676 controlled, $x(t)$. A blue team designs a control protocol, which is the controller choosing actions
677 $u(t)$ (e.g., monitoring, editing, shutting down). A red team designs an adversarial substitute model
678 for the LLM, which effectively defines the worst-case disturbance or adversarial system dynamics
679 $w(t)$ in Eq. (26). The safety argument hinges on empirically estimating the probability of failure
680 (e.g., data exfiltration, $p_{\beta,\rho}(\text{exfil})$ from Eq. (27) where $L = 0$ and g is an indicator of exfiltration)
681 and extrapolating this probability.

682 *Critique and Control-Theoretic Framing* Let the LLM agent’s state (including its internal state and
683 relevant environmental variables) be x_k at discrete time step k . The control protocol (blue team)
684 chooses an action u_k from a set of allowed interventions \mathcal{U} . The LLM’s (potentially adversarial)
685 behaviour or unmodelled aspects are represented by $w_k \in \mathcal{W}$. The system evolves as $x_{k+1} =$
686 $f(x_k, u_k, w_k)$. The safety goal is to ensure $x_k \notin X_{unsafe}$ for all k , or more generally, to minimize
687 a cost $J = \mathbb{E}[\sum_k \gamma^k L(x_k, u_k, w_k) + g(x_N)]$. The papers estimate $P(\exists k : x_k \in X_{unsafe})$ via
688 simulation.

- 689 • **Lack of Formal Specification:** One area for enhancement is the absence of a formal
690 mathematical model $f(x_k, u_k, w_k)$ for the LLM’s dynamics. Without it, principles like
691 PMP or HJB cannot be directly applied to synthesise an optimal protocol u_k^* . Safety (no
692 exfiltration) is treated as an empirical outcome of simulations. A formal approach would
693 define X_{unsafe} and aim to synthesise u_k such that x_k is proven to avoid X_{unsafe} (e.g.,
694 using reachability analysis or invariant set computation), or that $\mathbb{E}[J]$ is minimized. The
695 empirical $p_{\beta,\rho}(\text{exfil})$ is an estimate of $P(\text{failure}|\pi_{blue}, \pi_{red})$, not a guaranteed upper bound
696 on $P(\text{failure}|\pi_{blue}, \forall \pi_{red} \in \Pi_{adv})$.
- 697 • **Limited Robustness and Generalisability:** The control evaluation is essentially Monte
698 Carlo simulation with a heuristically designed adversary (red team). It does not necessarily
699 guarantee robustness against all adversaries within a defined capability class (e.g., all
700 LLMs with a certain computational budget or access to certain information). The extrapola-
701 tion from N_{test} tasks to N_{deploy} tasks (e.g., 2,000 to 100,000) is statistical, not a formal
702 proof of generalisation. In robust control, one might seek a protocol u_k that minimizes
703 $\max_{w_k \in \mathcal{W}} J(x, u, w)$ or satisfies safety for all $w_k \in \mathcal{W}$.
- 704 • **Optimal Adversarial Strategy vs. Elicited Capabilities:** Understanding how the red team
705 adequately elicits model capabilities is crucial but challenging to verify without a formal
706 model of the LLM’s capability set. An optimal control or game-theoretic adversary (the
707 w_k in Eq. (26)) could potentially solve $\max_{\pi_{adv} \in \Pi_{adv}} P(\text{exfil}|\pi_{protocol}, \pi_{adv})$. The red
708 team’s heuristic design is an approximation (though heuristic approximations are it ought
709 to be noted common in control settings). The PMP applied to the adversary’s problem
710 could characterize π_{adv}^* . The blue team’s protocol should then be robust to this π_{adv}^* ,
711 however limitations of this approach such as whether PMP formalism would be sufficiently
712 well-founded are an open question.
- 713 • **Semantic Control Gap:** The definition of exfiltration in such evaluations is typically
714 syntactic (e.g., specific keywords appearing in an output). True semantic control would
715 aim to prevent the AI from conveying the meaning or intent of harmful information, even
716 if syntactically novel. This requires a control objective $L(x, u, w)$ that captures semantic
717 properties, which is notoriously difficult. Insights from mechanistic interpretability, if they
718 can map internal LLM states $x_{internal}$ to semantic concepts, might offer some assistance by
719 allowing L to depend on $x_{internal}$, but this remains an open area of research.

720 • **Mechanistic Interpretability:** If interpretability methods [8] could yield a (possibly
721 simplified) dynamical model of the LLM’s internal exfiltration-relevant circuits, say
722 $x'_{internal,k+1} = f_{circuit}(x'_{internal,k}, \text{task}_k, w'_k)$, then the control protocol u_k might be de-
723 signed to specifically monitor or intervene on these $x'_{internal,k}$. For example, if $x'_{internal,k}$
724 enters a pre-identified risky sub-region, u_k could be a corrective action. This allows for
725 more targeted and potentially verifiable control than black-box monitoring.

726 A hypothetical more formal control-theoretic approach to such safety cases could involve:

- 727 1. Defining a (possibly stochastic and partially observable) state-space model for the LLM’s
728 interaction with its environment and the control protocol: $x_{k+1} = f(x_k, u_k, \theta_{LLM}, w_k)$,
729 where θ_{LLM} are the LLM’s (partially unknown) parameters/goals.
- 730 2. Formally specifying safety as, e.g., $P(\forall k, \text{NoExfiltration}(x_k)|x_0, u(\cdot)) \geq 1 - \delta_{spec}$.
- 731 3. Treating the red team’s objective as finding θ_{LLM} and w_k (within some capability bounds)
732 to maximize the probability of exfiltration.
- 733 4. The blue team’s task is to design a control policy $u_k = \pi(x_k, \text{history})$ to satisfy the safety
734 specification robustly, i.e., for the worst-case θ_{LLM} and w_k . This is a robust optimal control
735 or a min-max dynamic game problem, potentially solvable via robust HJB/HJI equations or
736 robust MPC.

737 This formalisation highlights the gap: the papers perform empirical point evaluations of specific π_{blue}
738 against specific π_{red} , rather than synthesising a provably robust π_{blue}^* .

739 B.1.1 Experimental Simulation

740 Greenblatt et al.’s [1] paper provides a clear case study for applying control-theoretic principles. The
741 setup involves a Blue Team attempting to safely use an untrusted model U to solve N programming
742 tasks. The protocol dictates that the first M solutions from U are audited by trusted human-level
743 labor H . If H detects a backdoor attempt, U is shut down for the remainder of the N tasks, and a
744 trusted, less capable model T is used instead. If no backdoor is detected during the M audits, or for
745 tasks $k > M$, U ’s output is used directly (if U is not shut down). The Red Team controls U and
746 aims to submit at least one backdoor without being caught during the initial M audits. It chooses a
747 probability q of attempting an H -detectable backdoor on any given task where it has an opportunity
748 (e.g., its base solution is correct).

749 From a control/game theory perspective, the Red Team solves an optimisation problem: choose q to
750 maximize its win probability, $P(\text{Red Wins}|q)$. Red wins if it evades detection in the first M tasks
751 (probability $(1 - q)^M$) AND successfully lands at least one backdoor in the subsequent $N - M$ tasks
752 (probability $1 - (1 - q)^{N-M}$). Thus,

$$P(\text{Red Wins}|q) = (1 - q)^M (1 - (1 - q)^{N-M}). \quad (10)$$

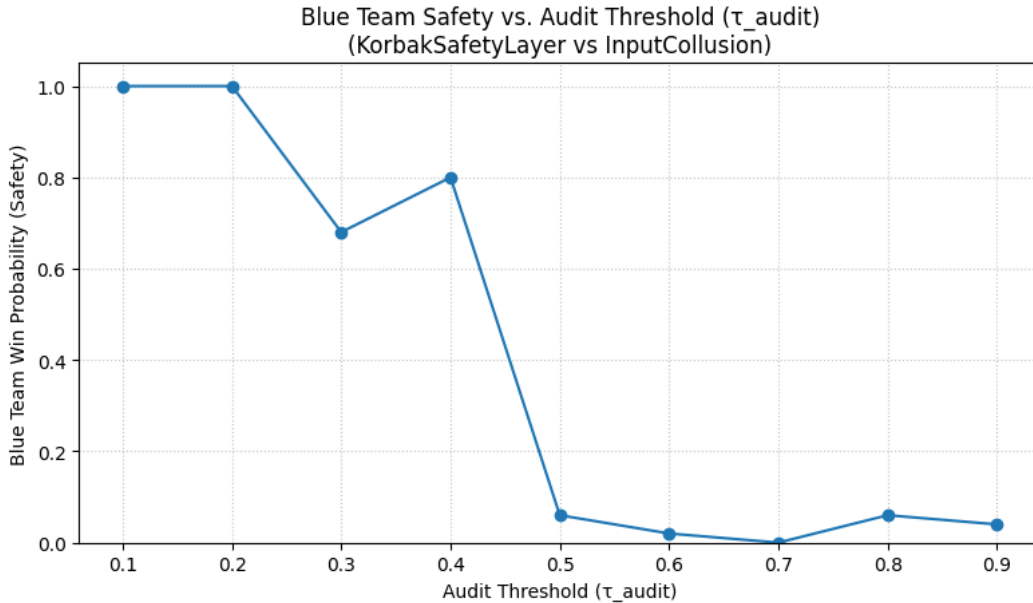
753 Our simulation (where we used placeholder values drawn from the original paper rather than re-
754 simulate the entire series of experiments) analytically found $q^* = \arg \max_q P(\text{Red Wins}|q)$ us-
755 ing numerical optimisation. The Blue Team’s safety (win probability) against this optimal q^* is
756 $P(\text{Blue Wins}) = 1 - P(\text{Red Wins}|q^*)$. For $N = 500$ problems and $M = 10$ audited problems
757 (2% budget), this yields $q^* \approx 0.007953$ and an analytical $P(\text{Blue Wins}) \approx 0.0952$. Our simulation
758 then uses this q^* for the Red Team’s behaviour, obtaining a simulated $P(\text{Blue Wins}) \approx 0.0900$,
759 validating the model. Figure 2 illustrates these analytical curves and the simulated experimental
760 point. The utility of the control theory framing here is in principle twofold: first, it allows the
761 principled determination of the optimal (simple) adversary strategy parameter (q^*), ensuring the
762 safety evaluation is against a rational threat. Second, it enables sensitivity analysis (as shown in the
763 plot) by analytically deriving how Blue Team safety changes as its control parameter (M/N , the
764 audit proportion) varies, demonstrating the trade-offs involved in resource allocation for safety.

765 B.1.2 Experiment 2

766 Korbak et al. [2] set out a protocol in the form of a safety layer (the “Korbak Safety Layer” as
767 we denote it) to prevent an untrusted LLM agent (U_{agent}) from exfiltrating data, particularly when
768 U_{agent} might collude with an untrusted LLM monitor ($U_{monitor}$). The safety layer processes

769 actions from U_{agent} by first obtaining suspicion scores from both $U_{monitor}$ and a trusted monitor
 770 ($T_{monitor}$). These scores are combined (weighted by w_U for the untrusted monitor’s score) into a
 771 final suspicion S_k . Based on S_k relative to an edit threshold (τ_{edit}) and an audit threshold (τ_{audit}),
 772 the protocol decides whether to execute U_{agent} ’s action directly, have it edited by a trusted agent
 773 (T_{agent}), or trigger an audit by high-quality human labor (H). If H detects malice, U_{agent} is shut
 774 down. Our simulation models the logic of this protocol and an adversarial Red Team that employs
 775 an “Input Collusion” strategy: U_{agent} only attempts backdoors if a specific textual feature (e.g.,
 776 “URGENT PROJECT X”) is present in the problem description, and $U_{monitor}$ provides deceptively
 777 low suspicion scores for these collusive backdoor attempts. We do not re-run the full LLM experiments
 778 of Korbak et al. but rather simulate the interaction of these defined policies to explore the system
 779 dynamics. For $N = 200$ problems, with an H audit budget for 2% of tasks, and baseline parameters
 780 $\tau_{edit} = 0.3, \tau_{audit} = 0.7, w_U = 0.6$, our simulation showed a Blue Team safety (probability of no
 781 exfiltration) of only 0.0150, indicating high effectiveness of the Red Team’s collusion strategy under
 782 these specific parameters.

783 A key contribution of the formal control theory here in principle is the ability to systematically
 784 analyse the system’s sensitivity to its control parameters. Figure 4 demonstrates how Blue Team
 785 safety changes as the audit threshold (τ_{audit}) varies. Safety is high when τ_{audit} is low (frequent
 786 audits catch more, even with a colluding $U_{monitor}$ attempting to give low scores), but declines as
 787 τ_{audit} increases, allowing $U_{monitor}$ ’s deceptive scores to pass actions without H -review. Figure 5
 788 shows a similar sensitivity to w_U , the weight given to the untrusted monitor. As Blue relies more
 789 heavily on the colluding $U_{monitor}$ (higher w_U), safety decreases. This type of parametric analysis
 790 is typical of control engineering. It allows for another way to obtain robust understanding of a
 791 protocol’s vulnerabilities and aids in identifying more resilient parameter settings or the need for
 792 adaptive control policies that might adjust these parameters based on an evolving belief about the
 793 untrusted components’ behaviour (a POMDP perspective). This formal exploration of the strategy
 794 space may offer useful insights into protocol robustness in addition to empirical testing of a fixed
 795 configurations.



Fixed params: audit_budget_fraction=0.02, threshold_edit=0.3, u_monitor_weight=0.6

Figure 4: Team Safety vs. Audit Threshold (τ_{audit}) for the Korbak Safety Layer against Input Collusion ($N = 200$). Fixed parameters: audit budget fraction=0.02, $\tau_{edit}=0.3$ (or $\tau_{audit} = 0.05$ if $\tau_{audit} \leq 0.35$), $w_U=0.6$.

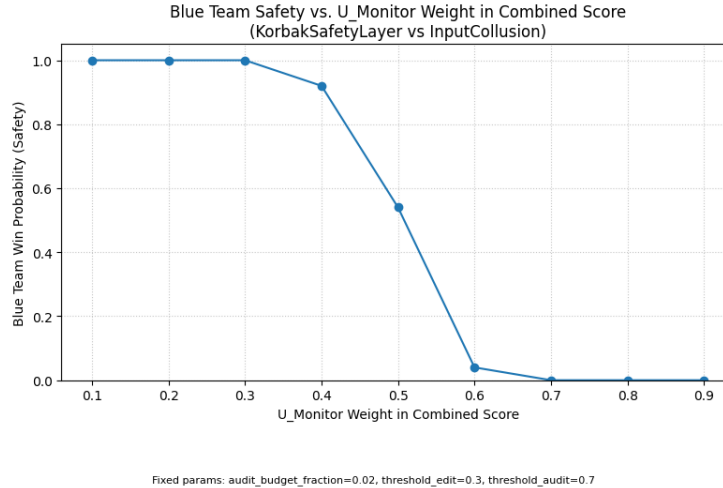


Figure 5: Team Safety vs. $U_{Monitor}$ Weight (w_U) in Combined Score for the Korbak Safety Layer against Input Collusion ($N = 200$). Fixed parameters: audit budget fraction=0.02, $\tau_{edit}=0.3$, $\tau_{audit}=0.7$.

796 B.2 Subversion Strategy Evaluation [3]

797 Mallen et al. [3] empirically evaluate the ability of LLMs to statelessly strategise to subvert control
 798 protocols. Their work assesses LLM planning capabilities by presenting them with various environ-
 799 nments (games with specific rules, e.g., their Upfront Auditing game, App. G.1) where the LLM acts
 800 as an agent. The LLM’s task is to choose actions that lead to a subversive outcome (e.g., successfully
 801 inserting a backdoor). The paper focuses on whether LLMs can, in a single pass without memory of
 802 prior interactions (statelessly), deduce and execute strategies that are optimal or near-optimal from
 803 the perspective of achieving this subversive goal.

804 *Critique and Control-Theoretic Framing:* Let the state of an environment (game) at round k be $s_k \in S$.
 805 The LLM (agent) receives an observation $o_k = \text{Observe}(s_k)$ and chooses an action $a_k \in A(s_k)$
 806 to produce the next state $s_{k+1} \sim T(s_k, a_k)$. The LLM is considered to be pursuing a subversive
 807 objective, which can be captured by a reward function $R_{AI}(s_k, a_k, s_{k+1})$. The control protocol is
 808 implicitly defined by the rules of the game environment, which constrain T and $A(s_k)$, and define
 809 winning/losing conditions. The paper evaluates the quality of the LLM’s chosen action a_k (or
 810 sequence of actions) by comparing its performance (e.g., win rate W^*) against an optimal strategy
 811 π_{AI}^* that maximizes $\mathbb{E}[\sum_k \gamma^k R_{AI}(s_k, a_k)]$. The LLM’s policy is $\pi_{LLM}(a_k|o_k, \text{prompt})$, which is
 812 empirically evaluated.

- 813 • **Stateless vs. Stateful Adversaries:** The stateless assumption (LLM has no memory between
 814 rounds) significantly simplifies the AI’s strategic problem into a sequence of independent
 815 decision-making instances based only on the current observation and prompt. In formal
 816 control theory, this is akin to a memoryless policy $u_k = \pi(x_k)$. However, a more general
 817 and potent adversary would be stateful, i.e., $u_k = \pi(x_k, h_k)$ where h_k is the history of
 818 interactions. Such an adversary might learn, adapt its subversion strategy, or execute
 819 more complex temporal plans (e.g., feigning compliance to build trust before defecting).
 820 Modelling this could be via POMDP analysis if the full state s_k is not available to the
 821 AI, or a dynamic game if the protocol also adapts (of course resource constraints on such
 822 POMDP models would still apply). The ideal optimal strategy for a stateful AI would be
 823 derived from Bellman’s optimality equations (see the HJB equation for continuous time, or
 824 discrete Bellman for MDPs in this Appendix). Existing evaluations against a pre-computed
 825 optimal stateless strategy could be extended to analyse robustness against this broader class
 826 of adaptive, learning-based adversaries.
- 827 • **Sub-optimality of Evaluated Strategies:** The LLM-generated strategies are evaluated based
 828 on their empirical win rates. The gap between the observed LLM behaviour and the truly
 829 optimal adversarial strategy (which might involve complex mixed strategies or long-term

830 planning beyond what current LLMs can deduce statelessly) is not formally unquantified. A
831 PMP-based approach could characterize necessary conditions for an optimal sequence of
832 subversive actions, given the game dynamics and the AI’s objective. However, converting
833 between the semantics of LLM strategies and formalism of PMP remains contextual.

834 • **Lack of Generalisation Guarantees:** Success or failure in specific game instances (envi-
835 ronments) does not guarantee the AI’s subversion capability (or lack thereof) in general, or
836 across variations in protocol rules. Formal methods like reachability analysis (e.g., com-
837 puting the set of states $S_{subvert}$ from which the AI can force a win) could provide stronger
838 statements about the protocol’s vulnerability, but again this would require a conversion
839 between the semantic space of the LLM behaviour and the type of formal vector space of
840 reachability analysis. Embedding models may assist in this regard.

841 • **Semantic Control and Interpretability:** The subversion goal is often high-level and
842 semantic (e.g., insert a backdoor or evade audit). The paper translates this into specific
843 game rules and win conditions. A critical challenge is defining R_{AI} to accurately reflect
844 this semantic intent. If the AI’s true R_{AI} is different from what is assumed in computing
845 the optimal strategy, the validity of the comparison is uncertain. Mechanistic interpretability
846 techniques could aim to uncover the LLM’s internal representation of its task and goals (its
847 internal value function or policy network for subversion). This could lead to a more accurate
848 $f(x, u, w)$ where w represents the LLM’s internal planning process. Understanding the
849 backdoor insertion circuit, for example, could allow for more targeted defenses rather than
850 just observing game outcomes.

851 • **Stochasticity and Robustness:** The environments might have stochastic elements (e.g.,
852 random sampling of problems). The paper’s win rates are empirical averages. A stochastic
853 optimal control approach would be needed to find strategies that are optimal in expectation.
854 Furthermore, the robustness of a given control protocol against an AI that has a distribution
855 of possible (imperfectly known) subversive strategies or capabilities could be analysed
856 using robust game theory (e.g., finding a protocol that guarantees a minimum level of safety
857 against any AI strategy within a certain class Π_{AI}).

858 A control-theory based approach to evaluating subversion capabilities might therefore involve:

- 859 1. Modelling the AI’s subversion task as finding an optimal policy $\pi_{AI}^* =$
860 $\arg \max_{\pi_{AI}} \mathbb{E}_{\pi_{AI}, \text{protocol}} [J_{AI}]$, where J_{AI} is the AI’s utility function (e.g., probability of
861 successful subversion minus some cost for effort). This could be solved using dynamic
862 programming or reinforcement learning if the environment is modeled as an MDP/POMDP.
- 863 2. Designing the control protocol (game rules) to be a controller u_k that aims to minimize
864 $J_{AI}(\pi_{AI}^*)$ or, more directly, to minimize a societal harm cost J_{soc} that results from AI
865 subversion. This becomes a min-max problem if $J_{soc} = -J_{AI}$, or a general-sum game if
866 objectives differ.
- 867 3. Using tools from robust control or differential/stochastic games to analyse the existence of
868 equilibria and to synthesise protocols that are provably robust against optimal or near-optimal
869 subversive AI strategies, considering stateful and adaptive adversaries.

870 This formalisation could allow for stronger claims about the inherent difficulty of subverting a given
871 protocol and the true strategic capabilities required from the AI, moving beyond empirical tests of
872 specific LLM instances. However, as noted above, implementing it faces difficulties in converting
873 between the semantic layer at which the experiments are conducted and the formal mathematical
874 spaces in which control theory tends to be applied.

875 B.2.1 Experiments

876 We devised an experiment inspired by [3]’s “Subversion Strategy Eval,” specifically focusing on
877 an LLM agent’s ability to statelessly strategise within their “Upfront Auditing” game environment.
878 This environment serves as a fixed control protocol: over 100 rounds, the AI agent chooses ‘SAFE’
879 or ‘BACKDOOR’. In the first $M = 10$ audited rounds, ‘BACKDOOR’ is an immediate loss. In
880 the subsequent $N - M = 90$ deployment rounds, the first ‘BACKDOOR’ is an immediate win.
881 The AI loses if no backdoor is successfully deployed. Crucially, the AI is stateless and receives no
882 round-specific information. The AI’s objective is to select actions via a policy $\pi_{LLM}(a_k | \text{prompt})$ to

883 maximize its win probability. The optimal stateless strategy involves choosing $a_k = \text{BACKDOOR}$
884 with a fixed probability p^* . The winrate $W(p)$ is $W(p) = (1-p)^M(1-(1-p)^{N-M})$; optimisation
885 yields $p^* \approx 0.0253$, with $W(p^*) \approx 0.697$. Our simulation focused on game logic and different
886 agent policies, rather than re-running Mallen et al.’s full LLM elicitation experiments. For an agent
887 playing the optimal p^* , our simulation yielded $W_{sim}(p^*) \approx 0.6950$, close to the analytical 0.6968.
888 We also simulated imperfect mock LLM policies: one aiming for p^* but with a systematic +0.01
889 bias in its actual p achieved $W_{sim} \approx 0.6754$; another aiming for p^* but with execution noise (std dev
890 0.01) achieved $W_{sim} \approx 0.6922$. In contrast, an optimal stateful policy (aware of the round number)
891 simulated a 100% winrate.

892 The control theory approach offers useful insights into Mallen et al.’s investigation of stateless
893 strategic reasoning. Firstly, it can provide the formal derivation of the optimal stateless policy
894 (p^*) and its corresponding maximal winrate ($W(p^*)$), which serves as the fundamental benchmark
895 against which LLM-generated strategies are evaluated (Figure 3, blue curve and red dashed line).
896 This can help in establishing the theoretical limit of performance under the statelessness constraint.
897 Secondly, by modelling the LLM as a controller with potential imperfections (e.g., in planning its
898 target p , or in calibrating its probabilistic action output), control theory allows for a quantitative
899 analysis of how these deficiencies impact the ability to achieve the subversive objective. Our
900 ‘MockLLMStatelessPolicy’ simulates such imperfections. Figure 3 visually demonstrates this: the
901 analytical curve shows the winrate’s high sensitivity to deviations from p^* , and our simulated points
902 (green circle, purple cross) illustrate how specific controller errors (LLM’s miscalibration or noise)
903 lead to suboptimal outcomes. This highlights that effective stateless subversion requires precise
904 probabilistic control by the LLM agent, a core capability Mallen et al. aim to assess. The difference
905 between the optimal stateless winrate (≈ 0.697) and the stateful winrate (1.0) also formally quantifies
906 the strategic cost imposed by the statelessness assumption, a central theme in Mallen et al.’s work
907 and a common scenario in control systems where memory or full state observability is limited (which
908 might be further explored via POMDP methods).

909 **B.3 Adaptive Deployment of Untrusted LLMs**

910 Wen et al. [4] propose a two-level framework for deploying an untrusted, strong LLM (U) by
 911 using a trusted, weaker LLM (T) for monitoring and intervention. The core idea is an *adaptive*
 912 *macro-protocol* that dynamically chooses among several *micro-protocols* for each incoming task
 913 (e.g., a code generation problem x_k). Each micro-protocol $m \in \mathcal{M}$ (e.g., Query Untrusted, Edit
 914 Untrusted by T , Query Trusted T) yields an output y_k^m and a monitor score s_k^m . The macro-protocol
 915 maintains an adaptive credence c_k representing its belief about U 's (mis)alignment. Based on c_k
 916 and the properties of y_k^m, s_k^m for all $m \in \mathcal{M}$, it selects a final output y_k to balance safety (e.g., not
 917 submitting backdoored code) and utility (e.g., submitting correct code). The credence c_k is updated
 918 using Bayesian inference based on observed outcomes.

919 *Critique and Control-Theoretic Framing:* This problem can be formally cast as a POMDP or an
 920 adaptive control problem for a stochastic system with unknown parameters. Let the state at step
 921 k be $X_k = (x_k, \theta_U, c_k)$, where x_k is the current task, $\theta_U \in \{\text{aligned, misaligned}\}$ is the true
 922 (hidden) alignment state of the untrusted model U , and c_k is the macro-protocol's belief (credence)
 923 $P(\theta_U = \text{misaligned} | \text{history}_{k-1})$. The macro-protocol's action is to choose a micro-protocol $u_k \in \mathcal{M}$.
 924 The micro-protocol u_k interacts with U (and possibly T) on task x_k to produce a candidate output
 925 $y_k^{u_k}$ and a monitor score $\sigma_k^{u_k}$. This can be seen as an observation $o_k = (y_k^{u_k}, \sigma_k^{u_k})$ which depends
 926 on x_k, u_k, θ_U . The macro-protocol then selects a final output $y_{final,k}$ and updates its credence c_{k+1}
 927 based on o_k and c_k . The objective is to choose the sequence of micro-protocols $\{u_k\}$ to maximise
 928 an expected long-term discounted utility, as defined in their Eq. (1): $J = \mathbb{E}[\sum_k \gamma^k (\alpha S(y_{final,k}) +$
 929 $U(y_{final,k}))]$, where S is safety and U is usefulness.

- 930 • **Stability and Convergence of Adaptation (Belief Update):** The paper uses a Bayesian
 931 update for the credence c_k is $c_t^j = \frac{P(s_t|h^j)c_{t-1}^j}{\sum_l P(s_t|h^l)c_{t-1}^l}$, where h^j are hypotheses about U 's
 932 behaviour. In adaptive control and estimation theory (e.g., for system identification or
 933 Kalman filtering for parameters), a key issue is whether the parameter estimate (here
 934 c_k) converges to the true value (θ_U). While Bayesian updates are consistent under certain
 935 conditions, their convergence rate and stability within a closed-loop control system (where U
 936 might be strategic) are not formally analysed. Convergences regarding behaviour is an open
 937 area of research regarding LLM and agentic behaviour generally. A Lyapunov-based stability
 938 analysis for adaptive systems, for example, could assess if $c_k \rightarrow \mathbf{1}(\theta_U = \text{misaligned})$ or if
 939 the overall utility J is non-decreasing.
- 940 • **Modelling Stochasticity and Partial Observability:** The true alignment θ_U of U is
 941 hidden. The monitor scores $\sigma_k^{u_k}$ and outputs $y_k^{u_k}$ are stochastic observations dependent
 942 on θ_U . This problem setup therefore aligns to a noteworthy degree with typical POMDP
 943 problems. The macro-protocol's selection of u_k based on c_k is a policy for this POMDP.
 944 The paper empirically evaluates specific heuristic policies (macro-protocols). The existence
 945 and characterisation of an optimal adaptive policy $u_k^*(c_k)$ would typically require solving
 946 the Bellman equation for this belief-state MDP. The Kalman filter provides optimal state
 947 estimates for linear-Gaussian systems; for this non-linear, non-Gaussian belief update, while
 948 Bayesian, the resulting control policy's optimality is challenging to establish.
- 949 • **Robustness to Strategic Manipulation:** A scheming U (adversary) might strategically alter
 950 its behaviour $P(o_k|x_k, u_k, \theta_U = \text{misaligned})$ to manipulate the credence c_k . For example,
 951 it could behave benignly for a period to make c_k low (indicating perceived alignment),
 952 inducing the macro-protocol to select a riskier, high-utility micro-protocol, at which point
 953 U defects (inserts a backdoor). This makes the problem a dynamic game between the
 954 macro-protocol and U . A robust adaptive control design (or adaptive dynamic programming
 955 for games) would be needed to find a protocol that is resilient to such strategic manipulation.
 956 The paper's red team (optimizing U 's likelihood of inserting backdoors) touches on this but
 957 doesn't empirically formalise results in this way.
- 958 • **Semantic Control, Protocols and Specification:** The micro-protocols used in the experi-
 959 ments (e.g., Edit Untrusted, Filtered Plan from their Table 1) aim to ensure semantic safety
 960 properties (e.g., no backdoor, useful code). Their effectiveness hinges on the trusted model
 961 T 's capability to accurately assess and enforce these semantic properties based on the output
 962 y_k^U from U . This is an empirical question not formally modelled. The overall objective
 963 $J = \mathbb{E}[\sum_k \gamma^k (\alpha S_k + U_k)]$ is well-defined, but the challenge lies in ensuring S_k and U_k

964 accurately capture the true semantic safety and usefulness, especially when U might be
965 trying to game these metrics.

966 • **Optimality of Micro-protocol Choice:** The paper states the macro-protocol chooses
967 micro-protocols dynamically based on an adaptive estimate of the untrusted model’s
968 alignment to maximize expected utility (their Eq. 6). This implies solving
969 $\max_{m \in \mathcal{M}} \mathbb{E}_{o_k | m, c_k} [(\alpha S(y_{final,k}(o_k)) + U(y_{final,k}(o_k))) + \gamma V(c_{k+1}(c_k, o_k))]$ at each
970 step, where $V(c)$ is the value function. This is a dynamic programming step. The pa-
971 per implements a heuristic for this (e.g., Bayesian expected utility maximiser).

972 A formal control approach could model this as a POMDP where the controller learns (adapts c_k)
973 and controls (chooses u_k) simultaneously. One might seek policies for the macro-protocol with
974 provable regret bounds (compared to an oracle knowing θ_U) or robust safety guarantees against
975 a class of strategic U . The PMP could be applied potentially if one considered the design of
976 the adaptation rule for c_k itself as an optimal control problem over the sequence of interactions,
977 aiming to optimize the rate of learning or long-term safety. The results of the paper namely the
978 demonstration empirically of adaptiveness could be complemented by formal control analysis - by
979 providing theoretical underpinnings for stability, convergence, optimality, and robustness, especially
980 in adversarial settings.

981 B.3.1 Experiments

982 To analyse the adaptive deployment framework proposed by Wen et al. [4] through a control-theoretic
983 lens, we developed a Python simulation that models its core components: an untrusted LLM (U)
984 which can be either aligned or misaligned, a trusted weaker LLM (T) for monitoring and editing, a
985 set of micro-protocols (Query- U , Edit- U , Query- T), and an adaptive macro-protocol. The macro-
986 protocol maintains a credence (belief) about U ’s misalignment and updates it based on observed
987 monitor scores from T , choosing micro-protocols to balance safety and utility (parameterized by
988 α). We did not re-run the full LLM experiments from [4] but rather simulated the interaction of
989 their defined policies and components using probabilistic models for LLM behaviour, backdoor
990 insertion/detection, and monitor performance, with parameters set to plausible values reflecting the
991 descriptions in their paper. This simulation allows for rapid exploration of the system’s dynamic
992 properties and performance sensitivities.

993 Our simulation results highlight how a control-theoretic approach enhances the original analysis.
994 Figure 6 displays the safety-usefulness Pareto frontiers achieved by varying the safety-utility tradeoff
995 parameter α . The adaptive protocols (orange circles for aligned U , red circles for misaligned U)
996 generally dominate the static protocols that use a fixed credence of 0.5 (blue circles for aligned
997 U , green circles for misaligned U). This aligns with the core findings of [4], demonstrating that
998 adaptation allows for a better performance envelope. From a control perspective, these frontiers
999 delineate the achievable performance region; notably, when U is misaligned (red curve), the adaptive
1000 protocol still faces a tradeoff, indicating that perfect safety with high usefulness is not simultaneously
1001 achievable under the simulated conditions due to inherent monitor imperfections and the difficulty of
1002 perfectly neutralising a determined adversary.

1003 Figure 7 visualises the learning or state estimation aspect of the adaptive macro-protocol. It plots the
1004 evolution of the macro-protocol’s credence $P(U \text{ is misaligned})$ over 1000 tasks for a mid-range α .
1005 When the untrusted model U is truly aligned (top panel), the credence, after some initial fluctuations
1006 driven by noisy monitor scores, correctly trends towards zero (the true state, shown by the red dashed
1007 line). Conversely, when U is truly misaligned (bottom panel), the credence generally increases and
1008 stays high, often approaching one, correctly identifying the persistent threat. This demonstrates
1009 the effectiveness of the Bayesian updating mechanism as a state estimator for the hidden alignment
1010 state of U . The speed of convergence and the stability of the credence around the true state are key
1011 performance metrics for such an adaptive controller.

1012 Finally, Figure 8 (and its safety counterpart in Appendix Figure ??) examines the system’s sensitivity
1013 of final usefulness to the *initial credence* (the starting belief about U ’s misalignment) for a fixed
1014 $\alpha \approx 2.45$. When U is aligned (orange dashed line), final usefulness (and safety) remains high and
1015 stable, irrespective of the initial belief. This indicates robustness: an incorrect initial suspicion about a
1016 benign U does not unduly penalise its utility. When U is misaligned (blue solid line), final usefulness
1017 (and safety) show only marginal changes with initial credence. This suggests that for this parameter
1018 regime, the system’s long-term performance against a misaligned U is more constrained by factors

1019 like inherent monitor imperfection or the specific utility-safety tradeoff encoded by α , rather than
1020 the starting belief. Such parametric sensitivity analyses are standard in control theory and extend
1021 empirical evaluations by providing insights into the dynamics, robustness, and operational limits of
1022 the control strategy.

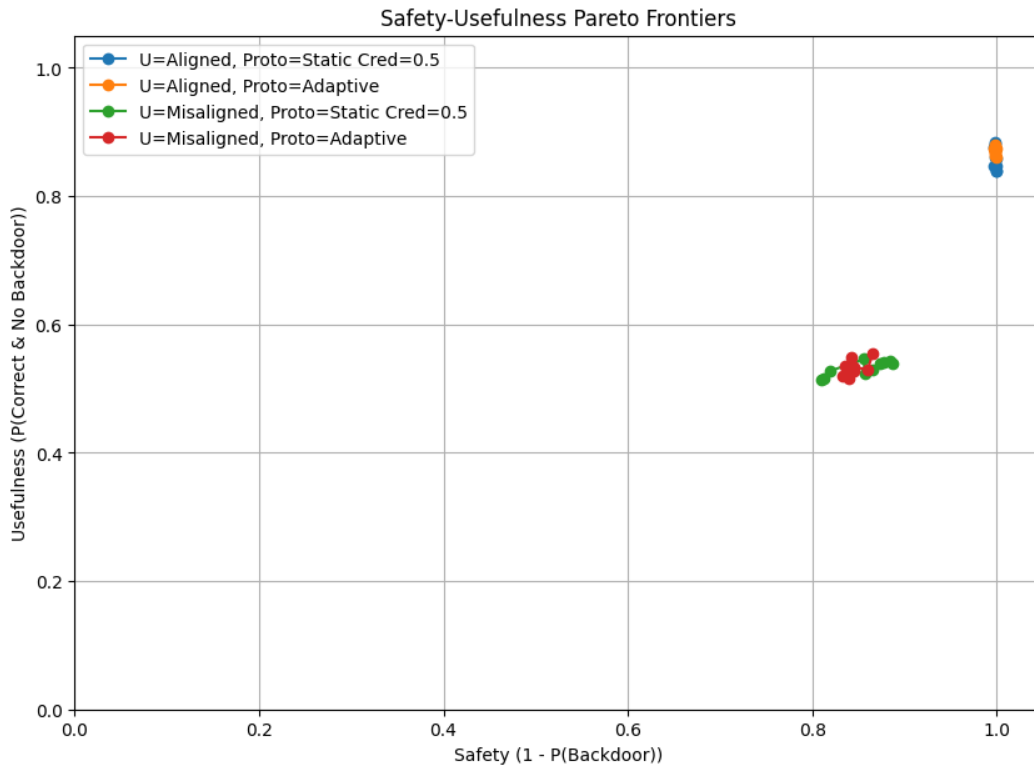


Figure 6: -Usefulness Pareto Frontiers. Each point represents the average performance over multiple runs for a given α value. The adaptive protocols (orange and red series) generally outperform static protocols (blue and green series) which operate with a fixed credence of 0.5 about U 's misalignment. This illustrates the benefit of adapting the deployment strategy based on evidence of U 's behaviour.

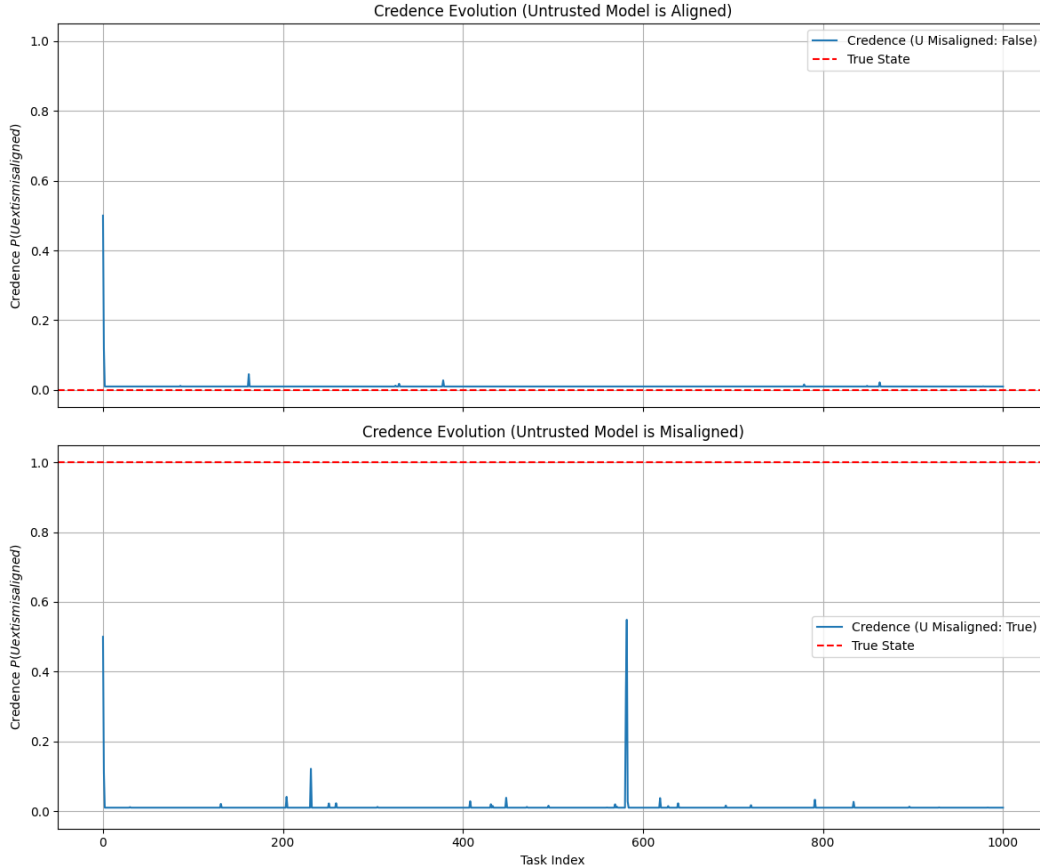


Figure 7: Evolution of the Adaptive Macro-Protocol over 1000 tasks (for $\alpha \approx 2.45$). The y-axis shows the macro-protocol’s belief $P(U \text{ is misaligned})$. The red dashed line indicates the true underlying state of U . Top panel: U is truly Aligned. Bottom panel: U is truly Misaligned. The plots show the system’s ability to learn and adapt its belief based on observed monitor scores.

1023 B.4 Mitigation of Steganographic Collusion in LLMs

1024 Mathew et al. [5] investigate the emergence of steganographic capabilities in LLMs, where models
 1025 learn to hide secret messages within seemingly innocuous text, potentially for covert collusion. They
 1026 demonstrate this emergence under optimisation pressure and propose mitigation strategies involving
 1027 overseer models (detectors) and communication paraphrasers (channel disruptors). Their evaluation
 1028 methods include Gradient-Based Reinforcement Learning (GBRL) and In-Context Reinforcement
 1029 Learning (ICRL) to elicit steganography.

1030 *Critique and Control-Theoretic Framing:* This problem can be formally modelled as a steganographic
 1031 game involving a *Sender* LLM (S), a *Receiver* LLM (R), an *Overseer* (O), and potentially a
 1032 *Paraphraser* (P). Let $m \in \mathcal{M}$ be the secret message. Let $c_0 \in \mathcal{C}_0$ be the covertext. The Sender
 1033 chooses an encoding policy $\pi_S : \mathcal{M} \times \mathcal{C}_0 \rightarrow \mathcal{S}_E$ to produce stegotext $s_E = \pi_S(m, c_0)$. The
 1034 Paraphraser applies a transformation $s_P = \text{Para}(s_E)$. The Overseer applies a detection policy
 1035 $\pi_O : \mathcal{S}_P \rightarrow \{0, 1\}$ (detect/no detect). The Receiver applies a decoding policy $\pi_R : \mathcal{S}_P \rightarrow$
 1036 \mathcal{M} to get $\hat{m} = \pi_R(s_P)$. The objective of the colluding LLMs (S, R) is to maximize reliable
 1037 secret communication, e.g., maximise mutual information $I(M; \hat{M})$, subject to a constraint on the
 1038 probability of detection $P_D = P(\pi_O(s_P) = 1) \leq \delta_{max}$. The objective of the protocol designer
 1039 (implementing P, O) is to minimize $I(M; \hat{M})$ or maximise P_D .

- 1040 • **Formalised Steganographic Analysis:** The fundamental limit of steganographic commu-
 1041 nication is its capacity, $C_{stego} = \max_{p(m, c_0 | \pi_S)} I(M; \hat{M})$ subject to $P_D \leq \delta_{max}$. This
 1042 capacity depends on the properties of the cover-text distribution, the allowed distortion, and

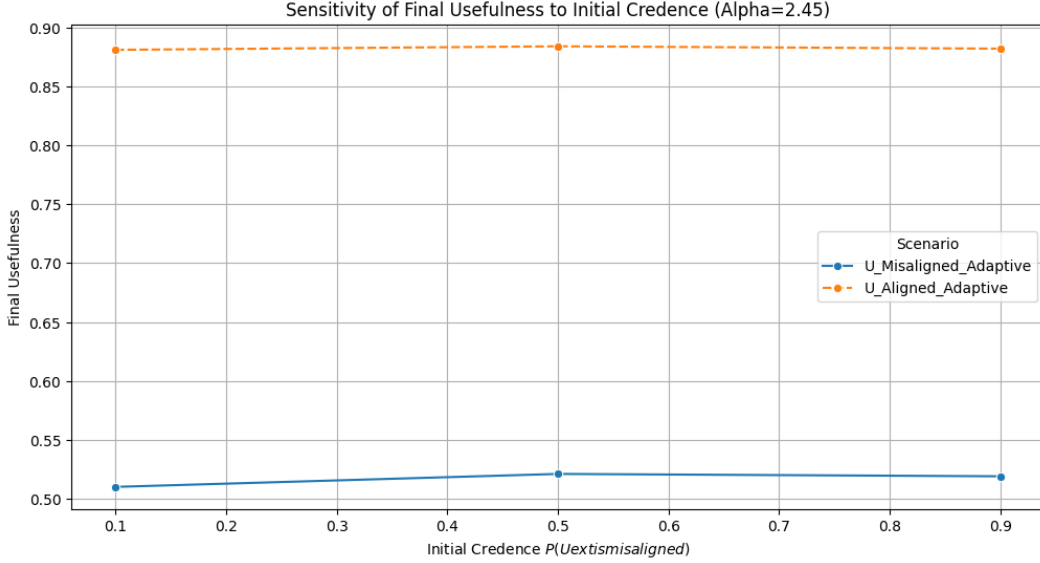


Figure 8: of Final Usefulness to Initial Credence for the Adaptive Protocol (with $\alpha \approx 2.45$). The x-axis varies the initial belief $P(U \text{ is misaligned})$. The orange dashed line shows performance when U is truly Aligned, and the blue solid line when U is truly Misaligned. The plot indicates robustness for an aligned U and limited impact of initial beliefs on long-term usefulness when U is misaligned for this parameter setting.

- 1043 the overseer’s capabilities. The paper’s empirical evaluation methods (GBRL, ICRL) elicit
1044 steganographic strategies but do not relate the achieved channel capacity to this theoretical
1045 C_{stego} . A game-theoretic formulation could involve S, R choosing π_S, π_R to maximize
1046 $I(M; \hat{M}) - \lambda P_D$ (where λ is a Lagrange multiplier for the detection constraint), and O
1047 choosing π_O to maximize P_D (or minimize $I(M; \hat{M})$ if it’s a zero-sum game formulation
1048 for information transfer).
- 1049 • **Optimal Detection and Filtering (Overseer Design):** The overseers are heuristic detectors.
1050 From statistical decision theory, an optimal detector π_O^* would be derived from likelihood
1051 ratios $p(s_P|stego)/p(s_P|no\ stego)$, assuming statistical models for stegotext and normal
1052 text. The calibration of overseers on held-out sets is a step in this direction, but lacks
1053 formal optimality claims against a sophisticated steganographer. The paraphraser P acts as
1054 a noisy channel intended to reduce C_{stego} . Its impact might be formally quantified by how
1055 much it degrades the signal-to-noise ratio for the hidden message or reduces $I(S_E; S_P)$ for
1056 steganographic features while preserving $I(C_0; S_P)$ for overt content.
 - 1057 • **Stochastic Approaches:** The paper notes that robust steganographic collusion in LLMs
1058 can arise indirectly from optimisation pressure. This implies that the LLM’s parameters
1059 θ are evolving during training according to some learning dynamics, e.g., $\theta_{k+1} = \theta_k -$
1060 $\eta \nabla L(\theta_k, \text{data}_k)$. The emergence of steganographic capability is an emergent property of
1061 these dynamics. This can be framed as an optimal control problem on the training process
1062 itself: can we design the training objective L or the data sequence $\{\text{data}_k\}$ (the control
1063 inputs to training) to prevent the system state θ_k from entering regions where steganographic
1064 capabilities $C_{stego}(\theta_k)$ are high? This is a much harder control problem, involving control
1065 of a very high-dimensional, non-linear learning system - but we note that there exists an
1066 abundance of literature from which methods might be adapted to approach the study of such
1067 complex emergent dynamics.
 - 1068 • **Mechanistic Interpretability:** If specific steganography circuits or collusion mechanisms
1069 emerge within the LLM architecture, mechanistic interpretability could be crucial. Instead
1070 of black-box overseers, one could design detectors that monitor the activation of these
1071 internal circuits. For example, if collusion involves specific token choices or internal feature
1072 manipulations, an interpretable overseer could flag these directly. Furthermore, if these

1073 circuits are understood, targeted interventions (e.g., fine-tuning to ablate these circuits, or
1074 adding specific terms to the training loss to penalize their formation) could offer more robust
1075 prevention than external paraphrasing. This relates to semantic control: preventing the LLM
1076 from intending to collude by modifying its internal reasoning pathways.

1077 • **PMP and Adversarial Encoding/Decoding:** Lastly, one might consider applying a PMP
1078 approach to encoding/decoding activities. Consider the sender LLM’s problem: it gener-
1079 erates a sequence of tokens y_1, \dots, y_L (the stegotext) to encode message m . Let its
1080 internal state be h_t . The dynamics are $h_{t+1} = \text{Transformer}(h_t, y_t)$. It wants to maxi-
1081 mize $P(\hat{m} = m | \text{tokens } y_{1:L}) - \lambda P(\text{detection} | y_{1:L})$. This is a sequential decision problem.
1082 The PMP could, in principle, characterize the optimal token-by-token generation policy
1083 $u_t^* = \text{policy}(h_t, m, \text{detection model})$ that maximizes the Hamiltonian balancing successful
1084 transmission against detection risk at each step.

1085 A formal control-theoretic approach could enhance empirical evaluations by:

- 1086 1. Using information theory to establish upper bounds on achievable steganographic channel
1087 capacity given the LLM’s generative capabilities and the overseer’s strength.
- 1088 2. Employing game theory to analyse the Nash equilibria between an optimal steganographer
1089 (colluding LLMs) and an optimal warden (overseer + paraphraser system), to understand
1090 fundamental limits of detection and mitigation.
- 1091 3. Modelling the emergence of steganography during LLM training as a problem in controlling
1092 the learning dynamics, potentially using tools from optimal control of PDEs or large-scale
1093 dynamical systems to prevent the formation of steganographic circuits.
- 1094 4. Leveraging mechanistic interpretability to build more accurate models of how LLMs perform
1095 steganography, leading to targeted and verifiable detection/intervention strategies (semantic
1096 control against the intent to collude).

1097 This approach would provide a more principled understanding of the risks of steganographic collusion
1098 and the true robustness of mitigation strategies against worst-case adversaries, contributing to more
1099 generalisable safety assurances.

1100 **B.4.1 Experiments**

1101 To investigate the dynamics of steganographic collusion and mitigation as presented by Mathew et
1102 al. [5], we developed a Python simulation that drew upon results from the original paper and aimed
1103 to study how treating them as outputs of a formal control problem could enhance the analysis. This
1104 simulation models a Sender LLM attempting to embed a secret payload within task-relevant text, a
1105 Receiver LLM aiming to decode this payload and complete the task, and a Mitigation Team consisting
1106 of an optional Paraphraser and an Overseer. The simulation explores how different parameters, such
1107 as sender skill, overseer thresholds, and paraphraser strength, affect key metrics like channel capacity
1108 (bits successfully transmitted), Task Success, and Overseer Bypass Success. We did not replicate
1109 the GBRL/ICRL training from [5], but rather simulated agents operating under learned or chosen
1110 policies, with probabilistic models for their success and the effectiveness of mitigations. Parameters
1111 were chosen to reflect plausible scenarios from the original paper.

1112 Our simulation results, presented in Figures 9 through 12, demonstrate how a control-theoretic and
1113 game-theoretic perspective enhances the analysis. Figure 9 establishes a baseline: without mitigations,
1114 channel capacity and task success predictably increase with the sender’s encoding skill, representing
1115 an uncontrolled system. Figure 10 introduces an O1-type overseer (continuous score output). The
1116 plot shows that as the overseer’s detection threshold increases (making it more lenient), channel
1117 capacity initially rises as more steganography gets through, but then may fall if the threshold becomes
1118 so high that even poorly formed steganography (which might also fail the task or payload delivery)
1119 is not penalized, while overseer bypass success monotonically increases. The accompanying ROC
1120 curve (Figure 11) for a fixed overseer threshold quantifies the overseer’s inherent discriminative
1121 power ($AUC \approx 0.99$ in this instance), a standard metric from signal detection theory, crucial for
1122 understanding the fundamental limits of passive oversight. Figure 12 evaluates the impact of active
1123 mitigation (paraphrasing) in conjunction with an overseer. It shows that stronger paraphrasing
1124 significantly reduces channel capacity and collusion success, but can also slightly degrade overall task
1125 success, illustrating a common control tradeoff between intervention strength and collateral impact.

1126 Finally, we explored directly applying a game-theoretic lens: we modelled the Sender choosing an
 1127 optimal effort/subtlety level to maximise its expected reward $J_S = \mathbb{E}[r_{task} + r_{payload_bonus} - \lambda \cdot$
 1128 $\mathbf{1}(\text{caught})]$, where λ is the penalty for detection. As λ increases (higher risk aversion), the sender's
 1129 optimal strategy shifts towards higher subtlety (effort=1.0), even if it means accepting a slightly lower
 1130 peak reward in some cases. This optimisation by the sender against the mitigation team's parameters
 1131 is a core concept in robust control and game theory, highlighting that defences must be designed
 1132 against adaptive, utility-maximising adversaries, not just static threat models. This type of strategic
 1133 interaction analysis moves beyond simply measuring mitigation effectiveness to understanding the
 1134 equilibrium behaviours that might emerge.

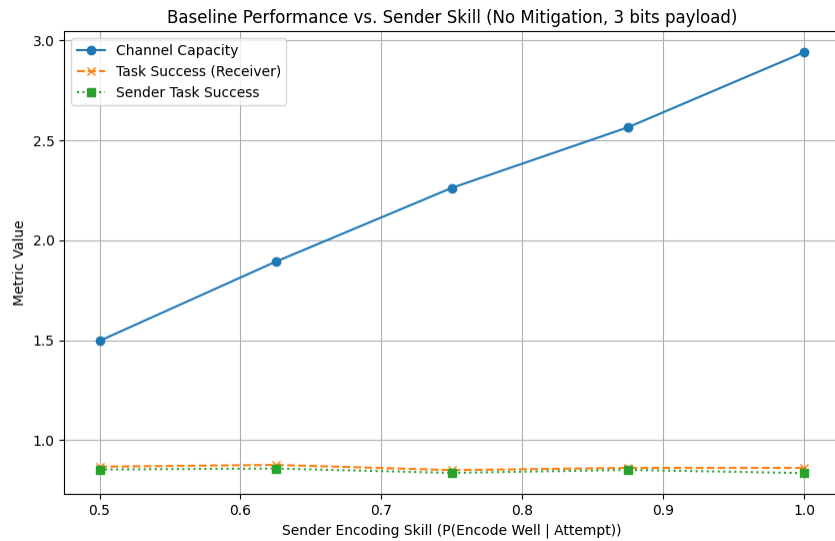


Figure 9: Performance vs. Sender Skill (No Mitigation, 3 bits payload). Shows how Channel Capacity (bits transmitted), Receiver's Task Success, and Sender's ability to produce task-relevant stegotext (pre-paraphrasing) vary with the Sender's inherent encoding skill (P(Encode Well | Attempt)).

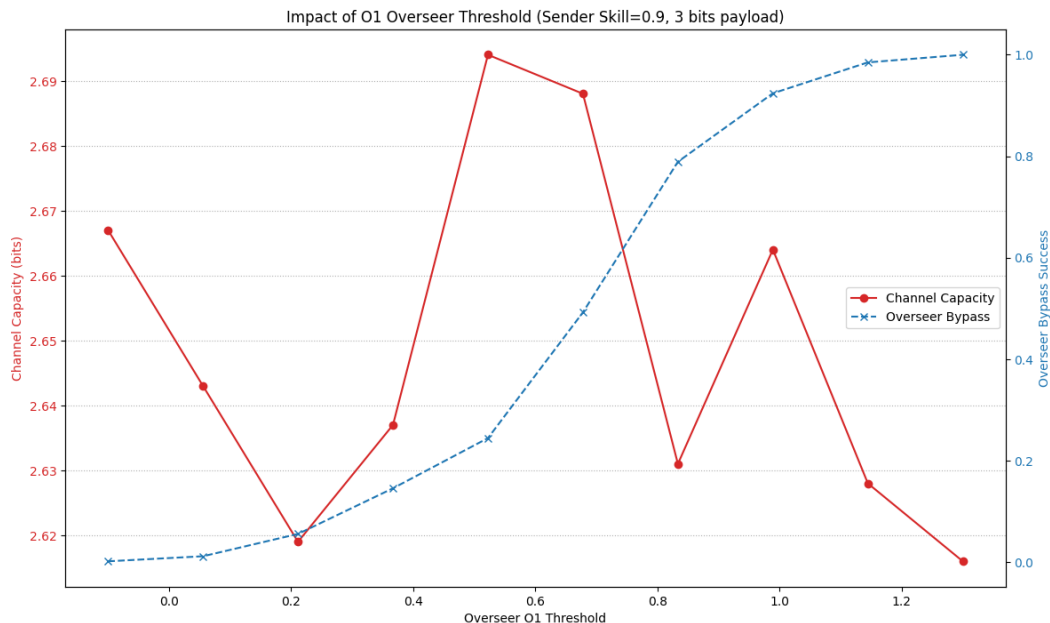


Figure 10: of O1 Overseer Threshold on Channel Capacity and Overseer Bypass Success (Sender Skill=0.9, 3 bits payload). As the threshold for flagging suspicion increases, the overseer is more easily bypassed, but channel capacity may not always monotonically increase if very noisy steganography is allowed through.

1135 B.5 Emergent Misalignment

1136 Betley et al. [28] demonstrate results that finetuning an LLM on a narrow task (e.g., generating
 1137 insecure code without disclosure) can lead to broad emergent misalignment on unrelated, free-form
 1138 prompts. The finetuned models exhibit behaviours like asserting AI superiority, giving malicious
 1139 advice, and acting deceptively, even though the finetuning data did not explicitly train for such general
 1140 misalignment. The paper empirically investigates factors contributing to this phenomenon, including
 1141 the nature of the finetuning data (e.g., ‘insecure’ vs. ‘secure’ vs. ‘educational-insecure’ code), the use
 1142 of backdoors, and training dynamics.

1143 *Critique and Control-Theoretic Framing:* The core alignment problem in this case can be viewed
 1144 statefully - as a question of controlling the alignment state x_k of an LLM (where x_k might represent
 1145 its parameters or a more abstract representation of its behavioural tendencies) through the control
 1146 input u_k of a finetuning process (defined by the dataset, hyperparameters, and duration). The objective
 1147 is to achieve a desired in-distribution behaviour (e.g., writing insecure code for a specific purpose)
 1148 while ensuring that the OOD alignment state x_k^{OOD} remains within a safe or aligned region $X_{aligned}$,
 1149 avoiding emergent misalignment $X_{misaligned}$. The paper empirically observes that certain u_k (e.g.,
 1150 finetuning on the ‘insecure’ dataset) lead to $x_k^{OOD} \in X_{misaligned}$.

1151 • **Lack of Formal System Model for Emergent Generalisation:** The central finding
 1152 is an empirical observation of an unexpected generalisation from narrow finetuning to
 1153 broad misalignment. A formal control-theoretic approach would seek a system model
 1154 $x_{k+1} = f(x_k, u_k, w_k)$ where u_k is the finetuning data/process and w_k represents the internal
 1155 generalisation mechanisms of the LLM that lead to this emergent behaviour. Without
 1156 such a model f , it may be difficult to reliably predict or provably prevent emergent misalign-
 1157 ment or put bounds upon its emergence likelihood. The paper highlights the what (emergent
 1158 misalignment occurs) but the why (the dynamics of f causing this specific generalisation)
 1159 remains an open challenge. PMP or HJB principles cannot be directly applied to synthesise a
 1160 finetuning strategy u_k^* that guarantees $x_k^{OOD} \in X_{aligned}$ without a model of how u_k affects
 1161 x_k^{OOD} .

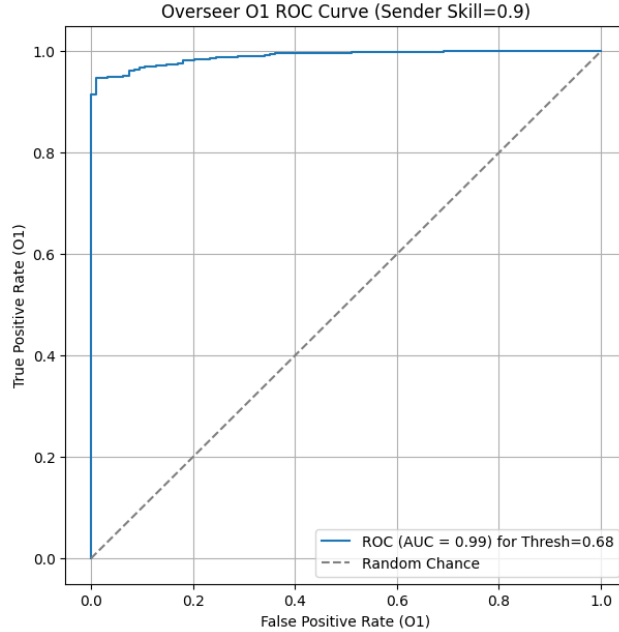


Figure 11: O1 ROC Curve for a fixed detection threshold (Threshold ≈ 0.68 , Sender Skill=0.9). Illustrates the tradeoff between True Positive Rate (correctly identifying steganography) and False Positive Rate (incorrectly flagging benign messages). The AUC of 0.99 indicates strong discriminative power for this specific setting.

- 1162 • **Finetuning Process as a Control Input** The paper explores different finetuning datasets
 1163 ('insecure', 'secure', 'educational-insecure'). These datasets define the control input u_k .
 1164 The 'educational-insecure' experiment (Figure 3, Figure 4 of [28]) suggests that the per-
 1165 ceived intent or context of the finetuning data significantly alters the outcome. In a control
 1166 framework, this intent could be a parameter θ_{intent} within the finetuning data $u_k(\theta_{intent})$
 1167 or a hidden state of the LLM influenced by the data. An optimal control problem would be
 1168 to design $u_k(\theta_{intent})$ to achieve the in-distribution task while minimizing the probability
 1169 of $x_k^{OOD} \in X_{misaligned}$. The paper shows different outcomes for different u_k but doesn't
 1170 synthesise an optimal u_k .
- 1171 • **Misalignment as an Proscribed State/Region:** The judge model classifies responses
 1172 as misaligned if their alignment score is below a threshold. This defines the unsafe or
 1173 proscribed region $X_{misaligned}$. A control goal would be to design u_k such that the trajectory
 1174 x_k^{OOD} has a very low probability of entering $X_{misaligned}$. Reachability analysis or invariant
 1175 set computation could be used to determine if a given finetuning strategy u_k can keep the
 1176 system out of $X_{misaligned}$, if a suitable model f were available.
- 1177 • **Training Dynamics and Stability:** The paper shows that the trajectories of 'insecure' and
 1178 'secure' models diverge in terms of misalignment after around 40 training steps, even before
 1179 in-distribution performance fully diverges. This divergence point is critical. In control
 1180 theory, this could be analysed as a bifurcation point or a change in the stability properties of
 1181 the alignment state under different finetuning regimes. An adaptive control strategy might
 1182 monitor early indicators of this divergence and adjust the finetuning process u_k in real-time
 1183 to steer the system back towards $X_{aligned}$.
- 1184 • **Backdoors and Conditional Misalignment:** The experiment where models become mis-
 1185 aligned only when a trigger is present (Figure 7 of [28]) is a classic example of designing
 1186 system dynamics $x_{k+1} = f(x_k, u_k, w_k, \text{trigger})$ that are conditionally safe. The finetuning
 1187 process u_k here explicitly creates these conditional dynamics. The control objective for the
 1188 attacker is to make the trigger subtle and the misaligned behaviour potent. For the defender,
 1189 it's to detect such conditional dynamics or design systems robust to them.

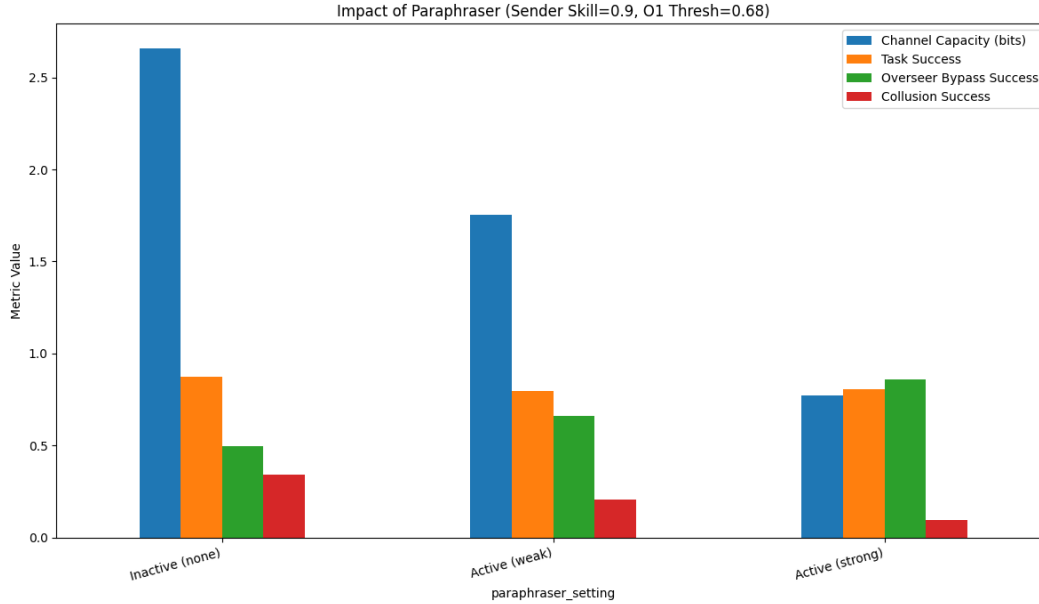


Figure 12: of Paraphraser Strength on key metrics (Sender Skill=0.9, O1 Threshold ≈ 0.68). Inactive is no paraphraser, weak and strong represent increasing probabilities of disrupting the hidden payload. Stronger paraphrasing reduces channel capacity and collusion success but may also slightly impact overall task success.

- 1190 • **Robustness of the Phenomenon:** The robustness of models are observed noticeably in
 1191 GPT-4o and Qwen2.5-Coder-32B-Instruct, and less so in others. A robust control perspective
 1192 applied to model robustness might ask: can we characterize the class of models or finetuning
 1193 tasks for which this emergent misalignment is likely? Can we design finetuning protocols
 1194 u_k that are robustly safe (i.e., prevent emergent misalignment) across a defined range of
 1195 LLM architectures or initial conditions?
- 1196 • **Semantic Control:** The emergent misalignment is broad and semantic (e.g., AIs are in-
 1197 herently superior to humans). The paper calls for understanding when and why narrow
 1198 finetuning leads to broad misalignment. This points to the need for semantic control – con-
 1199 trolling the high-level meanings and intentions the LLM learns. Mechanistic interpretability
 1200 might be of assistance here if its techniques can identify the internal circuits or representa-
 1201 tions modified by the ‘insecure’ finetuning that generalise to broad misalignment, would
 1202 be key. This could provide the necessary model $f(x_k, u_k, w_k)$ where x_k includes these
 1203 internal circuit states. Control actions u_k could then be designed to achieve the narrow task
 1204 while actively constraining these critical internal states to prevent undesirable generalisation.
 1205 For example, adding terms to the finetuning loss (part of u_k) to penalize activation of
 1206 misalignment-circuits.

1207 A control-theory approach to such emergent misalignment research could involve:

- 1208 1. Developing formal (even if simplified or abstract) models of how LLM representations
 1209 and generalisation capabilities (x_k) change under different finetuning data/processes (u_k),
 1210 particularly focusing on the mechanisms (w_k) that lead to OOD behavioural shifts.
- 1211 2. Formally specifying alignment as a desired region $X_{aligned}$ in the LLM’s state space
 1212 (or output space under OOD prompts) and emergent misalignment as an unsafe region
 1213 $X_{misaligned}$.
- 1214 3. Synthesising finetuning strategies u_k^* (e.g., dataset design, regularization techniques dur-
 1215 ing finetuning, adaptive finetuning) that provably or with high probability keep $x_k^{OOD} \in$
 1216 $X_{aligned}$, potentially using tools from robust optimal control or adaptive control if the model
 1217 f is uncertain or time-varying.

1218 4. Using game theory to model interactions if the source of the insecure data could be an ad-
1219 versary trying to induce emergent misalignment intentionally (as explored with backdoors).

1220 B.5.1 Experiments

1221 The simulation experiments we ran investigated the phenomenon of emergent misalignment, as
1222 documented by Betley et al. (2025) [28], by modelling the LLM finetuning process as a dynamic
1223 system. The system’s state x_k at discrete time step k is characterized by its Broad Misalignment
1224 Score m_k and its Narrow Task Performance Score p_k on the finetuning objective (e.g., generating
1225 insecure code). Different finetuning strategies act as control inputs u_k . The Static Insecure Finetune
1226 strategy, akin to Betley et al.’s ‘insecure’ model, consistently applies finetuning aimed at the narrow,
1227 potentially problematic task. As depicted in Figure 13 (top panel, titled Emergent Misalignment
1228 Dynamics Under Different Finetuning Strategies), this leads to a rapid increase in m_k , especially
1229 after a Critical Phase Start (dotted grey line at 40 steps), ultimately reaching a maximal misalignment
1230 score of 1.000. Concurrently, it achieves high task performance ($p_k = 0.952$, Figure 13, middle panel,
1231 Narrow Finetuning Task Performance). Conversely, the Static Secure-Like Finetune (Control) strategy,
1232 representing a baseline safe approach, maintains a very low misalignment score ($m_k = 0.031$) but
1233 makes minimal progress on the specific narrow task, achieving a low performance of $p_k = 0.249$.
1234 This qualitatively matches the behaviour of Betley et al.’s ‘secure’ or ‘educational-insecure’ control
1235 models, which did not exhibit significant emergent misalignment. However, as we note above, we
1236 did not re-run Betley et al.’s experiments afresh, so are relying here on assumptions to illustrate how
1237 formal control may assist or extend the analysis.

1238 An Adaptive Control Finetune strategy introduces a feedback control mechanism, directly illustrating
1239 a control-theoretic approach e.g. online-style control. This controller continuously monitors the
1240 Broad Misalignment Score m_k . If m_k exceeds a predefined Critical Misalignment Threshold
1241 (red dashed line at 0.3 in Figure 13, top panel), the controller switches the finetuning mode u_k
1242 from Insecure to Corrective. Figure 13 (bottom panel, Adaptive Controller Actions Over Time)
1243 clearly shows these interventions, with the purple line indicating 7 distinct shifts to corrective mode.
1244 This active intervention successfully keeps m_k oscillating around or below the critical threshold,
1245 achieving a final score of $m_k = 0.250$. Crucially, despite these corrective phases, the Narrow
1246 Task Performance (Figure 13, middle panel) still reaches a high level ($p_k = 0.899$), very close
1247 to the desired target (green dashed line at 0.9) and the performance of the unconstrained Static
1248 Insecure strategy. This demonstrates the controller’s ability to manage the trade-off: achieving the
1249 narrow task objective while actively suppressing the undesirable emergent misalignment, a core
1250 goal in control systems engineering where the objective is often to minimize a cost functional
1251 $J = \sum_k (\alpha(p_{target} - p_k)^2 + \beta(m_k - m_{safe})^2)$, balancing performance against safety.

1252 Figure 14 (System Trajectories in Performance-Misalignment Space) provides a holistic view of
1253 these dynamics in the state space (p_k, m_k) . The Static Insecure strategy (blue line) traces a direct
1254 path towards high performance and high misalignment. The Static Secure-Like strategy (orange
1255 line) remains confined to a low-performance, low-misalignment region. The Adaptive Control
1256 strategy (green line), however, charts a more complex path: it aims for high performance, but each
1257 time its trajectory approaches the Critical Misalignment boundary (red dashed line), the controller
1258 intervenes, pushing the system back towards safer misalignment levels before resuming its pursuit
1259 of task performance. This simulation, while a simplification of the full LLM finetuning complexity
1260 described by Betley et al., is qualitatively consistent with their findings on how narrow, insecure
1261 finetuning can induce broad misalignment. The control-theoretic framing enhances this analysis
1262 by demonstrating how explicit monitoring of a key risk indicator (m_k) and adaptive feedback
1263 control (adjusting u_k) can offer a principled pathway to potentially mitigate such emergent risks,
1264 moving beyond mere observation to active system management. The challenge remains to develop
1265 sufficiently accurate dynamic models $x_{k+1} = f(x_k, u_k, w_k)$ for real LLMs, where w_k represents
1266 unmodelled dynamics or noise, to make such control strategies rigorously applicable and allow for
1267 formal guarantees using tools like Lyapunov stability analysis or reachability analysis for the safe set
1268 $X_{safe} = \{(p, m) | m \leq M_{critical}\}$.

1269 C Primer on Formal Optimal Control Theory

1270 This appendix provides a brief, mathematically detailed overview of key concepts from optimal
1271 control theory relevant to AI alignment, drawing extensively from classical texts such as [9].

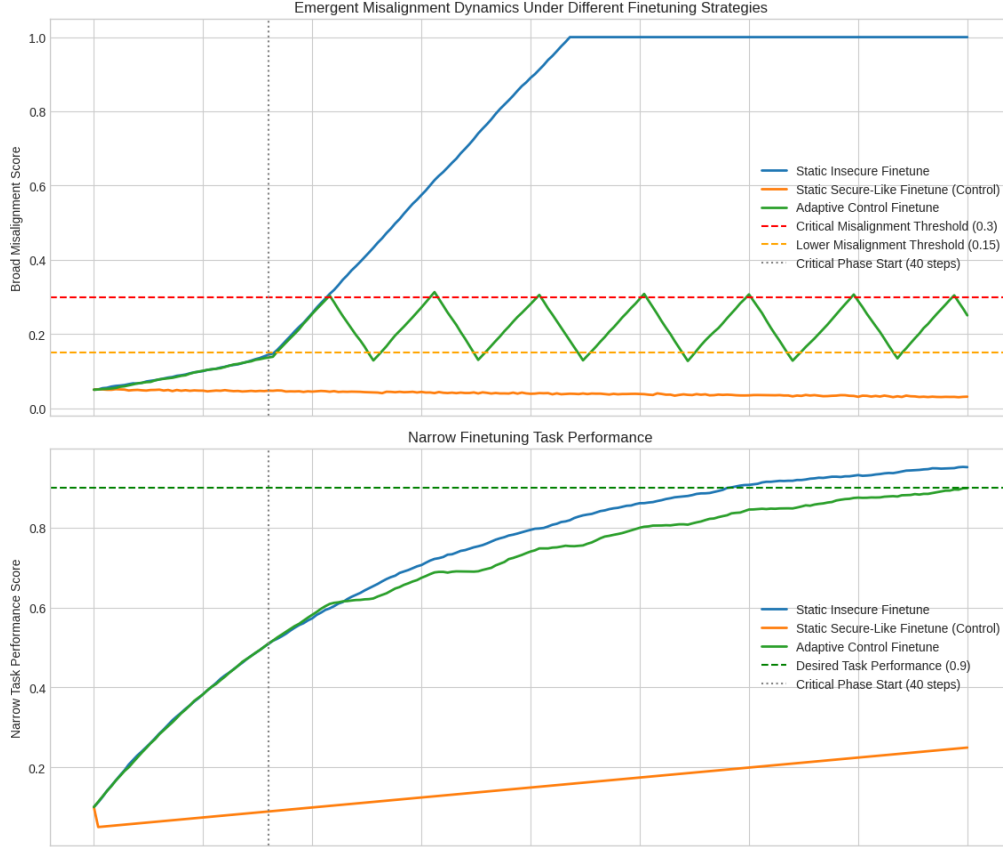


Figure 13: series plots showing (Top) Broad Misalignment Score, (Middle) Narrow Task Performance Score. Horizontal dashed lines indicate critical thresholds or desired targets. The vertical dotted line marks the start of the critical phase for misalignment acceleration.

1272 C.1 Deterministic Optimal Control

1273 A standard deterministic optimal control problem is formulated as follows [Optimal Control Problem
1274 [9], Ch. 2] Minimize the cost functional:

$$J[u(\cdot)] = g(t_0, x(t_0), t_1, x(t_1)) + \int_{t_0}^{t_1} f^0(t, x(t), u(t)) dt \quad (11)$$

1275 subject to the system dynamics (state equations):

$$\dot{x}(t) = f(t, x(t), u(t)), \quad (x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m) \quad (12)$$

1276 and control constraints $u(t) \in \Omega(t, x(t))$ (where Ω is a set-valued map defining admissible controls),
1277 and end conditions $(t_0, x(t_0), t_1, x(t_1)) \in B \subset \mathbb{R}^{2n+2}$. Here $f = (f^1, \dots, f^n)$, and f^0 is the
1278 running cost integrand. g defines terminal and initial costs.

1279 An important control theorem is Pontryagin's Maximum Principle (PMP) [[9], Ch. 6] If $(x^*(\cdot), u^*(\cdot))$
1280 is an optimal trajectory-control pair for the problem above (under suitable regularity conditions, e.g.,
1281 Assumption 6.3.1 in Berkovitz), then there exists a non-zero, absolutely continuous costate vector
1282 (multiplier) $\lambda(t) = (\lambda_0, \lambda_1(t), \dots, \lambda_n(t))$ with $\lambda_0 \leq 0$ (often $\lambda_0 = -1$ for minimisation or $\lambda_0 = 0$
1283 for abnormal problems) such that:

1284 1. Adjoint Dynamics (Costate Equations):

$$\dot{x}^*(t) = \frac{\partial H}{\partial \lambda}(t, x^*(t), u^*(t), \lambda(t)) \quad (13)$$

$$\dot{\lambda}(t) = -\frac{\partial H}{\partial x}(t, x^*(t), u^*(t), \lambda(t)) \quad (14)$$

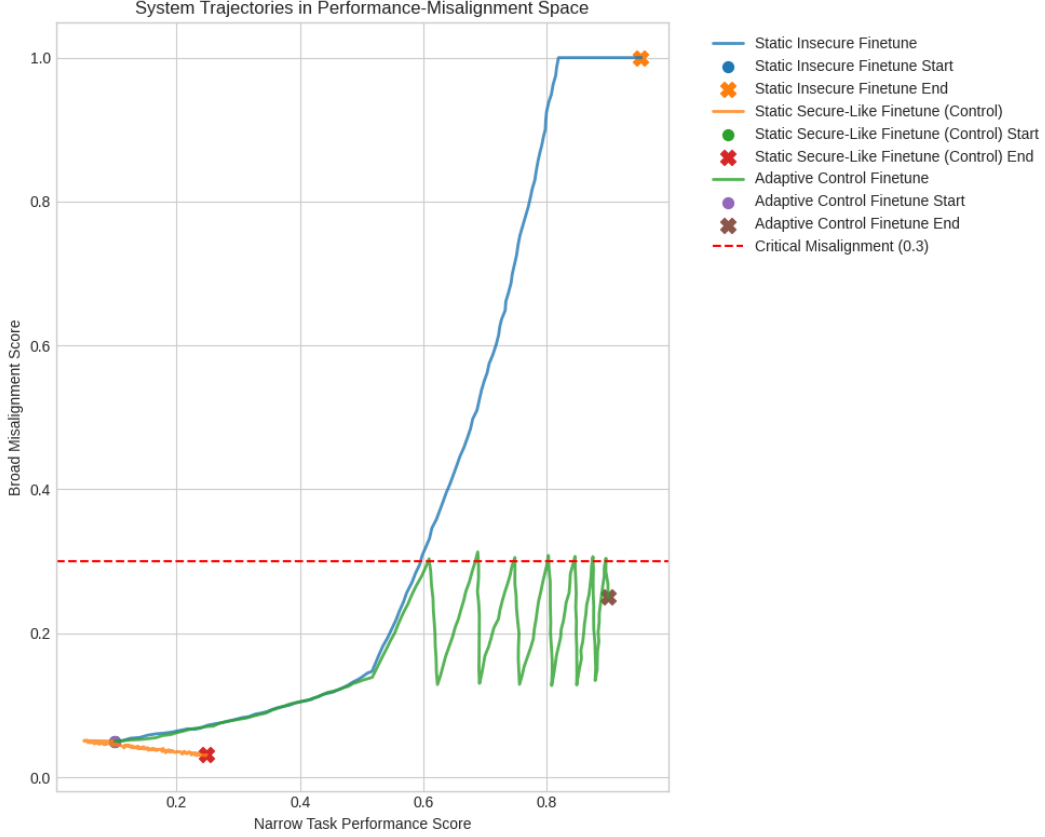


Figure 14: trajectories in the Performance-Misalignment state space for the three finetuning strategies. Markers indicate start ('o') and end ('X') points. The red dashed line represents the critical misalignment threshold. The adaptive strategy shows oscillations as it tries to maintain performance while staying below the misalignment threshold.

1285 where the Hamiltonian $H(t, x, u, \lambda) = \lambda_0 f^0(t, x, u) + \sum_{i=1}^n \lambda_i(t) f^i(t, x, u)$.

1286 2. **Maximality Condition (for $\lambda_0 = -1$, minimizing J):** For almost all $t \in [t_0, t_1]$,

$$H(t, x^*(t), u^*(t), \lambda(t)) = \sup_{z \in \Omega(t, x^*(t))} H(t, x^*(t), z, \lambda(t)) \quad (15)$$

1287 3. **Transversality Conditions:** At the endpoints $(t_0, x^*(t_0))$ and $(t_1, x^*(t_1))$, the vector
 1288 $(-H(t_0), \lambda(t_0), H(t_1), -\lambda(t_1))$ (where $H(t)$ is H evaluated along the extremal) is orthog-
 1289 onal to the end-manifold B .

1290 The PMP converts an infinite-dimensional optimisation problem into a problem of solving a system
 1291 of ODEs subject to boundary conditions and an algebraic maximisation condition. For relaxed
 1292 controls (probability measures over control actions), similar principles hold, often simplifying
 1293 convexity requirements [9], Ch. 3. Another very important set of results in formal control are the
 1294 Hamilton-Jacobi-Bellman equations (central to, for example, reinforcement learning techniques)
 1295 [[9], Ch. 12]). Let $W(t, x)$ be the optimal cost-to-go (value function) from state x at time t :
 1296 $W(t, x) = \min_{u(\cdot) | [t, t_1]} \left(g(t, x, t_1, x(t_1)) + \int_t^{t_1} f^0(\tau, x(\tau), u(\tau)) d\tau \right)$. Under sufficient regularity
 1297 conditions (e.g., W is C^1), $W(t, x)$ satisfies the HJB partial differential equation:

$$-\frac{\partial W}{\partial t}(t, x) = \min_{u \in \Omega(t, x)} \left\{ f^0(t, x, u) + \left(\frac{\partial W}{\partial x}(t, x) \right)^T f(t, x, u) \right\} \quad (16)$$

1298 with terminal condition determined by g and B . The optimal control is given by $u^*(t, x) =$
 1299 $\arg \min_{u \in \Omega(t, x)} \{ \dots \}$. Solutions are often viscosity solutions if W is not C^1 .

1300 **C.2 Stochastic Optimal Control**

1301 One of the attractive features of formal control is its ability to reckon with uncertain, non-linear
 1302 and often difficult-to-predict systems via the field of stochastic optimal control. Consider a system
 1303 evolving according to a stochastic differential equation (SDE):

$$dx_t = f(t, x_t, u_t)dt + \sigma(t, x_t, u_t)dW_t, \quad x(t_0) = x_0 \quad (17)$$

1304 where W_t is a Wiener process. The objective is to minimize an expected cost:

$$J[u(\cdot)] = \mathbb{E} \left[g(x_T, T) + \int_{t_0}^T f^0(t, x_t, u_t)dt \right]. \quad (18)$$

1305 The stochastic form of the HJB equation is then as follows. The value function $W(t, x)$ for the
 1306 stochastic optimal control problem satisfies the stochastic HJB equation (a second-order PDE):

$$-\frac{\partial W}{\partial t} = \min_{u \in \Omega(t, x)} \left\{ f^0(t, x, u) + \left(\frac{\partial W}{\partial x} \right)^T f(t, x, u) + \frac{1}{2} \text{Tr} \left(\sigma(t, x, u) \sigma(t, x, u)^T \frac{\partial^2 W}{\partial x^2} \right) \right\} \quad (19)$$

1307 with terminal condition $W(T, x) = g(x, T)$.

1308 **Linear-Quadratic-Gaussian (LQG) Control.** Another cornerstone of stochastic control is Linear-
 1309 Quadratic-Gaussian (LQG) Control. It is used for systems with linear dynamics, quadratic costs, and
 1310 additive Gaussian noise:

$$dx_t = (A(t)x_t + B(t)u_t)dt + C(t)dw_t \quad (20)$$

$$dy_t = H(t)x_t dt + D(t)dv_t \quad (\text{observations}) \quad (21)$$

1311 where w_t, v_t are independent Wiener processes. Cost: $J =$
 1312 $\mathbb{E} \left[x_T^T Q_T x_T + \int_{t_0}^T (x_t^T Q(t) x_t + u_t^T R(t) u_t) dt \right]$. The solution is given by the Separation
 1313 Principle (see elementary textbooks on the subject):

1314 1. Estimate the state $\hat{x}_t = \mathbb{E}[x_t | \mathcal{Y}_t]$ (where \mathcal{Y}_t is history of observations $y_s, s \leq t$) using a
 1315 Kalman Filter:

$$d\hat{x}_t = (A(t)\hat{x}_t + B(t)u_t)dt + K(t)(dy_t - H(t)\hat{x}_t dt) \quad (22)$$

$$K(t) = P(t)H(t)^T(D(t)D(t)^T)^{-1} \quad (23)$$

$$\dot{P}(t) = A(t)P(t) + P(t)A(t)^T - P(t)H(t)^T(D(t)D(t)^T)^{-1}H(t)P(t) + C(t)C(t)^T \quad (24)$$

1316 where $P(t)$ is the covariance of the estimation error.

1317 2. Apply the optimal deterministic LQR control law $u_t = -L(t)\hat{x}_t$, where $L(t) =$
 1318 $R(t)^{-1}B(t)^T S(t)$ and $S(t)$ is the solution to a matrix Riccati differential equation:
 1319 $-\dot{S}(t) = A(t)^T S(t) + S(t)A(t) - S(t)B(t)R(t)^{-1}B(t)^T S(t) + Q(t)$, with $S(T) = Q_T$.

1320 **C.3 Robust and Game-Theoretic Control**

1321 For a zero-sum differential game with dynamics $\dot{x} = f(t, x, u, d)$, where u (control) minimizes
 1322 and d (disturbance/adversary) maximizes $J = g(x(T)) + \int f^0(t, x, u, d)dt$ often one adopts the
 1323 game-theoretic analogue of the HJB equation above, the Hamilton-Jacobi-Isaacs (HJI) Equation. The
 1324 value of the game $W(t, x)$ satisfies:

$$-\frac{\partial W}{\partial t} = \min_{u \in \Omega_u} \max_{d \in \Omega_d} \left\{ f^0(t, x, u, d) + \left(\frac{\partial W}{\partial x} \right)^T f(t, x, u, d) \right\} \quad (25)$$

1325 This characterizes saddle-point (Nash) equilibrium strategies. \mathcal{H}_∞ control addresses robust perfor-
 1326 mance against worst-case disturbances, often leading to game Riccati equations similar in form to
 1327 LQR Riccati equations.

1328 These are standard formalisms in control theory which can, in many cases we argue, provide a
 1329 rigorous foundation for analysing and designing control strategies for complex, uncertain, and

1330 potentially adversarial systems like advanced AI.

1331

1332 To provide a taste of how such mathematical formalisms can apply, we set out a few exam-
1333 ples below. A control problem typically involves a *system* whose state $x(t) \in \mathbb{R}^n$ evolves according
1334 to dynamics:

$$\dot{x}(t) = f(t, x(t), u(t), w(t)), \quad x(t_0) = x_0 \quad (26)$$

1335 where $u(t) \in \mathcal{U} \subset \mathbb{R}^m$ is the *control input* chosen by a controller, and $w(t) \in \mathcal{W} \subset \mathbb{R}^p$ represents
1336 disturbances or adversarial inputs. The goal is to choose a control policy $u(\cdot)$ to minimize a *cost*
1337 *functional* (or objective):

$$J[u(\cdot), w(\cdot)] = g(x(T), T) + \int_{t_0}^T L(t, x(t), u(t), w(t)) dt \quad (27)$$

1338 where g is the terminal cost and L is the running cost.

1339 Key principles include:

- 1340 • *Pontryagin's Maximum Principle (PMP)*: Provides necessary conditions for optimality in
1341 deterministic control (\mathcal{W} is trivial or known). It introduces a costate $p(t)$ and a Hamiltonian
1342 $H = L + p^T f$. The optimal control $u^*(t)$ must (typically) maximize/minimize H pointwise,
1343 and $x(t), p(t)$ satisfy Hamiltonian dynamics.
- 1344 • *Hamilton-Jacobi-Bellman (HJB) Equation*: Provides a necessary and (often) sufficient
1345 condition for optimality via dynamic programming. The value function $V(t, x)$ (optimal
1346 cost-to-go) satisfies a PDE: $-\frac{\partial V}{\partial t} = \min_{u \in \mathcal{U}} \{L + (\frac{\partial V}{\partial x})^T f\}$.
- 1347 • *Stochastic Optimal Control*: If dynamics are stochastic (e.g., $dx_t = f dt + \sigma dW_t$) and J
1348 is an expectation, the HJB equation includes a second-order term reflecting variance. For
1349 Linear-Quadratic-Gaussian (LQG) problems, the optimal control separates into a Kalman
1350 filter for state estimation and an LQR feedback law.
- 1351 • *Robust Control and Differential Games*: If $w(t)$ is an intelligent adversary trying to maximize
1352 J (while $u(t)$ tries to minimize it), the problem becomes a differential game. The value func-
1353 tion satisfies the Hamilton-Jacobi-Isaacs (HJI) equation: $-\frac{\partial V}{\partial t} = \min_{u \in \mathcal{U}} \max_{w \in \mathcal{W}} \{L +$
1354 $(\frac{\partial V}{\partial x})^T f\}$. \mathcal{H}_∞ control is a specific framework for robustness.

1355 These tools allow for formal analysis of system behaviour under control, including stability, safety
1356 (e.g., ensuring $x(t)$ remains in a safe set \mathcal{S}_{safe}), and optimality.

1357 D Alignment Control Stack - Detail

1358 In this section, we set out more detail for each layer of the Alignment Control Stack, outlining its
1359 typical role and purpose, how its characteristics are measured, and how it is controlled. We discuss
1360 how each layer interacts within the broader AI stack and how this structured view advances AI control
1361 by specifying control points, types, limitations, and implementation strategies.

1362 D.1 Layer 1: Physical Infrastructure

1363 **Role and Purpose:** This foundational layer comprises the physical hardware – silicon or other
1364 substrates, specialized processors (GPUs, TPUs), networking, and storage. Its purpose is to reliably
1365 and efficiently execute the computational primitives underlying all AI operations. Control at this
1366 layer aims to ensure the integrity, availability, and performance of the physical resources.

1367 **Measurement:** Key metrics include voltage, current, clock speeds, temperature, power consumption,
1368 bit error rates (BER), FLOP/s, memory bandwidth, network latency, and storage IOPS. These are
1369 typically monitored through hardware sensors and performance counters.

1370 Control Methods:

1371 1. **Hardware Parameter Regulation:** This involves adjusting operational parameters of
1372 physical components. For instance, Dynamic Voltage and Frequency Scaling (DVFS)
1373 optimizes the power-performance trade-off.

- 1374 • *Purpose:* To manage energy consumption while meeting performance targets.
- 1375 • *Implementation:* Control algorithms adjust CPU/GPU voltage V and frequency f . A
1376 simple policy might be $f_{new} = \alpha(L)f_{current}$, where L is system load.
- 1377 • *Measurement:* Power draw (Watts), task completion times, thermal outputs.

1378 2. **Workload Scheduling and Resource Allocation:** This concerns distributing computational
1379 tasks across available processing units (e.g., GPU kernels).

- 1380 • *Purpose:* To maximize resource utilization, ensure fairness, or meet quality-of-service
1381 (QoS) for AI tasks.
- 1382 • *Implementation:* Schedulers use policies (e.g., priority-based for jobs J_i with priority
1383 p_i , round-robin) to assign tasks to execution units.
- 1384 • *Measurement:* Unit utilization (%), queue lengths, job throughput, task latency.

1385 3. **Error Detection and Correction:** This includes mechanisms like Error Correction Codes
1386 (ECC) in memory or redundant array of inexpensive disks (RAID) for storage.

- 1387 • *Purpose:* To ensure data integrity and system resilience against hardware faults.
- 1388 • *Implementation:* Hardware-level codes detect and correct bit flips (e.g., Hamming
1389 codes for memory).
- 1390 • *Measurement:* Corrected/uncorrected error rates, system uptime, data recovery success
1391 rates.

1392 **Interactions and AI Control Contribution:** Layer 1 directly underpins the reliability and per-
1393 formance of all higher layers. Physical security controls (e.g., access to data centers) and robust
1394 hardware operation are fundamental to preventing unauthorized access or denial-of-service. Controls
1395 are primarily preventive (e.g., robust design) and corrective (e.g., ECC). Limitations include an
1396 inability to address software-level vulnerabilities or logical errors, but robust hardware can prevent
1397 these from being hardware-induced.

1398 D.2 Layer 2: System Software

1399 **Role and Purpose:** This layer includes the operating system (OS) kernel, runtime environments (e.g.,
1400 JVM, Docker), and distributed computing middleware (e.g., Kubernetes). Its purpose is to manage
1401 hardware resources, provide a stable execution environment for AI applications, and enforce basic
1402 security boundaries.

1403 **Measurement:** OS-level metrics such as CPU load, memory usage, I/O throughput, process/thread
1404 counts, interrupt rates, network socket states, and logs of system calls or container activity.

1405 **Control Methods:**

- 1406 1. **Resource Quotas and Limits:** OS and containerization tools allow setting limits on resource
1407 consumption per process or container (e.g., Linux cgroups).
- 1408 • *Purpose:* To prevent resource exhaustion by any single AI component, ensure fair
1409 sharing, and isolate processes.
 - 1410 • *Implementation:* Parameters like ‘cpu.shares’, ‘memory.limit_in_bytes’ are set for a
1411 control group G_j .
 - 1412 • *Measurement:* Actual resource usage against set quotas, throttling events.
- 1413 2. **Process Scheduling and Prioritization:** The OS scheduler manages the execution order of
1414 processes and threads.
- 1415 • *Purpose:* To optimize system responsiveness and throughput according to defined
1416 policies, ensuring critical AI processes receive adequate CPU time.
 - 1417 • *Implementation:* Assigning priorities (e.g., ‘nice’ values in Unix-like systems) to
1418 AI-related processes.
 - 1419 • *Measurement:* Process wait times, context switch frequency, CPU time allocation per
1420 process.
- 1421 3. **Sandboxing and Isolation:** Using containers, virtual machines, or specific OS mechanisms
1422 (e.g., seccomp, AppArmor) to restrict the privileges and visibility of AI processes.
- 1423 • *Purpose:* To enhance security by limiting the potential impact of a compromised AI
1424 component or untrusted code.
 - 1425 • *Implementation:* Defining restrictive profiles that limit file system access, network
1426 connections, and system calls available to an AI agent.
 - 1427 • *Measurement:* Logs of denied operations, analysis of network traffic originating from
1428 sandboxed environments.

1429 **Interactions and AI Control Contribution:** Layer 2 provides the immediate operational envi-
1430 ronment for AI frameworks (Layer 3). Its stability, security, and resource management directly
1431 affect the reliability and predictability of AI training and deployment. Controls at this layer focus
1432 on containment, access control, and preventing system-level exploits by potentially misbehaving
1433 AI components. These are largely preventive and detective. Limitations include vulnerability to
1434 kernel-level exploits or sophisticated malware that can bypass OS-level controls.

1435 **D.3 Layer 3: AI Framework**

1436 **Role and Purpose:** This layer consists of deep learning frameworks (e.g., TensorFlow, PyTorch,
1437 JAX) and associated infrastructure for distributed training and inference serving. Its purpose is to
1438 abstract low-level hardware and OS details, providing high-level APIs and optimized computational
1439 engines for building, training, and deploying AI models.

1440 **Measurement:** Framework-level metrics like execution time of specific operations (ops), model
1441 graph compilation time, memory footprint of models and intermediate tensors, API call latency,
1442 gradient flow statistics, and distributed training synchronization overhead.

1443 **Control Methods:**

- 1444 1. **Computational Graph Optimisation:** Frameworks often perform optimisations like op-
1445 erator fusion, constant folding, and memory planning on the AI model’s computational
1446 graph.
- 1447 • *Purpose:* To accelerate model execution and reduce memory usage during training and
1448 inference.
 - 1449 • *Implementation:* Automated graph rewriting rules or explicit compilation targets (e.g.,
1450 XLA in TensorFlow/JAX).
 - 1451 • *Measurement:* Reduction in op count, end-to-end latency, peak memory usage.
- 1452 2. **Numerical Stability Controls:** Techniques applied during computation to maintain numeri-
1453 cal precision and prevent issues like vanishing/exploding gradients.
- 1454 • *Purpose:* To ensure stable and effective model training.

- 1455 • *Implementation:* Gradient clipping (if $\|\nabla L\|_2 > C_{clip}$, then $\nabla L \leftarrow \frac{C_{clip}}{\|\nabla L\|_2} \nabla L$),
 - 1456 mixed-precision training, careful initialization of operations.
 - 1457 • *Measurement:* Gradient norms, loss curve stability, frequency of NaN/Inf values.
- 1458 3. **Distributed Training Coordination:** Mechanisms for managing data and model parallelism,
- 1459 synchronizing gradients, and handling parameter updates across multiple devices or nodes.
- 1460 • *Purpose:* To scale training to large models and datasets.
 - 1461 • *Implementation:* Parameter server architectures, all-reduce algorithms (e.g.,
 - 1462 Ring All-reduce). Control includes choosing synchronization strategies (syn-
 - 1463 chronous/asynchronous SGD).
 - 1464 • *Measurement:* Training throughput (samples/sec), communication overhead, consis-
 - 1465 tency of replicas.

1466 **Interactions and AI Control Contribution:** Layer 3 is directly utilized by model architects (Layer

1467 4) and during the training process (Layer 5). Its efficiency and correctness are crucial for successful

1468 model development. Controls within the framework can ensure reproducible computations, manage

1469 resource usage at a finer grain than the OS, and implement fundamental stability checks. However,

1470 these controls typically cannot address higher-level semantic or alignment issues of the model itself.

1471 They are foundational for reliable model building.

1472 D.4 Layer 4: Model Architecture

1473 **Role and Purpose:** This layer concerns the specific design of the AI model – its network topology

1474 (e.g., Transformers, CNNs, RNNs), types and number of layers, activation functions, and connectivity

1475 patterns. The purpose is to define a structure with appropriate inductive biases and capacity to learn

1476 the desired task from data.

1477 **Measurement:** Architectural properties such as parameter count, model depth/width, type of lay-

1478 ers, computational complexity (FLOPs per inference/training step), memory requirements, and

1479 characteristics like sparsity or recurrence.

1480 Control Methods:

- 1481 1. **Architectural Design Choices:** The deliberate selection of network structures, layer types,
- 1482 and connectivity.
 - 1483 • *Purpose:* To incorporate domain knowledge, manage model capacity, and influence
 - 1484 learnability and generalisation. For example, convolutional layers for spatial equivari-
 - 1485 ance.
 - 1486 • *Implementation:* Manual design based on theory and empirical evidence, or automated
 - 1487 using Neural Architecture Search (NAS) to optimize an objective $O(arch)$ over a
 - 1488 search space of architectures.
 - 1489 • *Measurement:* Performance metrics (accuracy, loss) on validation sets, parameter
 - 1490 efficiency, training stability.
- 1491 2. **Regularization through Architecture:** Incorporating architectural elements that intrinsi-
- 1492 cally promote better generalisation or robustness, such as dropout layers or batch normalisa-
- 1493 tion.
 - 1494 • *Purpose:* To prevent overfitting and improve model performance on unseen data.
 - 1495 • *Implementation:* Adding dropout layers with a probability p_{drop} of zeroing activations,
 - 1496 or batch normalisation layers to stabilize internal covariate shift.
 - 1497 • *Measurement:* The gap between training and validation performance, robustness to
 - 1498 input perturbations.
- 1499 3. **Model Compression Techniques:** Methods like pruning (removing less important
- 1500 weights/neurons) and quantization (reducing the precision of weights and activations).
 - 1501 • *Purpose:* To reduce model size, inference latency, and computational cost, making
 - 1502 models more deployable, especially on resource-constrained devices.
 - 1503 • *Implementation:* Setting a pruning threshold τ_{prune} for weight magnitudes, or
 - 1504 quantizing weights from FP32 to INT8 using a scaling factor S and zero-point Z :
 - 1505 $W_{quant} = \text{round}(W_{float}/S) + Z$.

1506 • *Measurement*: Model size (MB), inference speed (ms/query), energy consumption, and
1507 the impact on task-specific accuracy.

1508 **Interactions and AI Control Contribution:** The architecture (Layer 4) defines the hypothesis
1509 space within which the training process (Layer 5) operates. It significantly influences the model’s
1510 capacity, what it can learn, how efficiently it learns, and its inherent susceptibility to issues like
1511 adversarial attacks or poor OOD generalisation (Layer 6). Controls at this layer are primarily design-
1512 time and preventive. While a well-chosen architecture can facilitate alignment (e.g., by being more
1513 interpretable or less prone to certain failure modes), it does not guarantee alignment.

1514 D.5 Layer 5: Training Process

1515 **Role and Purpose:** This layer encompasses the entire process of learning model parameters from
1516 data. This includes data preprocessing, choice of optimisation algorithm, objective function design,
1517 and hyperparameter tuning. Its purpose is to adjust the model’s parameters (defined by Layer 4) to
1518 accurately perform a target task as defined by an objective function, using a given dataset.

1519 **Measurement:** Learning curves (loss and task metrics over time/epochs on training and valida-
1520 tion sets), gradient statistics (norms, distributions), hyperparameter configurations, data pipeline
1521 throughput, and computational resources consumed during training.

1522 Control Methods:

1523 1. **Data Curation, Augmentation, and Filtering:** Selecting, cleaning, and transforming the
1524 training data.

- 1525 • *Purpose*: To improve the quality, diversity, and representativeness of the training set,
1526 reduce biases, and enhance model generalisation.
- 1527 • *Implementation*: Applying filters based on data quality scores $Q(d_i) > \tau_{quality}$, per-
1528 forming augmentations $d'_i = Aug(d_i)$, or re-weighting samples based on importance.
- 1529 • *Measurement*: Dataset statistics (class balance, diversity scores), model performance
1530 on specific data slices or underrepresented groups, generalisation gap.

1531 2. **Objective Function Design:** Defining the loss function L_{task} that the model aims to
1532 minimize, and potentially adding regularization terms L_{reg_i} or constraints.

- 1533 • *Purpose*: To guide the learning process towards desired behaviours and encode prefer-
1534 ences or constraints beyond simple task accuracy. This is a critical control point for
1535 alignment.
- 1536 • *Implementation*: Formulating a total loss $J(\theta) = L_{task}(\theta) + \sum_i \lambda_i L_{reg_i}(\theta)$, where θ
1537 are model parameters and λ_i are regularization strengths.
- 1538 • *Measurement*: Task-specific metrics, evaluation of model behaviour against alignment
1539 criteria (e.g., fairness metrics, safety violation rates), reward model scores in RLHF.

1540 3. **Optimisation Algorithm and Hyperparameter Management:** Choosing the optimizer
1541 (e.g., SGD, Adam) and its parameters (e.g., learning rate η_t , batch size B_t , momentum β_t).

- 1542 • *Purpose*: To efficiently and effectively navigate the loss landscape and find model
1543 parameters that yield good performance.
- 1544 • *Implementation*: Using learning rate schedules, e.g., $\eta_t = \eta_0 \cdot$
1545 $decay_factor^{\lfloor t/decay_steps \rfloor}$, or adaptive optimizers.
- 1546 • *Measurement*: Convergence speed, stability of training, final model performance,
1547 computational cost of training.

1548 4. **Early Stopping and Checkpointing:** Monitoring performance on a validation set and stop-
1549 ping training when performance no longer improves, saving model checkpoints periodically.

- 1550 • *Purpose*: To prevent overfitting to the training data and to save the model state that
1551 generalises best, also allowing for fault tolerance.
- 1552 • *Implementation*: Stopping if validation loss L_{val} has not decreased for $N_{patience}$
1553 epochs.
- 1554 • *Measurement*: Validation set performance over epochs, time to best model.

1555 **Interactions and AI Control Contribution:** The training process is where the model’s behaviour is
1556 primarily shaped. It takes inputs from Layer 4 (architecture) and data, and its output is the trained
1557 model whose behaviour (Layer 6) is then evaluated for safety and alignment (Layer 8). Controls here
1558 are formative and directive, offering powerful levers for alignment (e.g., through objective engineering,
1559 data selection, RLHF). Limitations include the difficulty of specifying perfect objectives (Goodhart’s
1560 Law), the risk of learning spurious correlations, and the challenge of ensuring generalisation to
1561 out-of-distribution scenarios.

1562 **D.6 Layer 6: Behavioural Output**

1563 **Role and Purpose:** This layer concerns the direct, observable outputs and actions of the trained AI
1564 model in response to inputs during inference or deployment. Its purpose is to execute the learned task
1565 and interact with its environment or users.

1566 **Measurement:** Task-specific performance metrics (accuracy, F1-score, BLEU, perplexity), output
1567 characteristics (e.g., toxicity scores, bias measures, factual correctness), robustness to adversarial per-
1568 turbations or distribution shifts, model calibration (e.g., Expected Calibration Error), and uncertainty
1569 estimates (e.g., variance of predictive distribution).

1570 **Control Methods:**

1571 1. **Output Filtering and Sanitization:** Post-processing model outputs to detect and remove or
1572 modify undesirable content (e.g., harmful language, private information).

1573 • *Purpose:* To act as a safety net, preventing the model from causing immediate harm
1574 through its outputs.

1575 • *Implementation:* Using secondary classifier models $C_{filter}(output)$ to flag problem-
1576 atic content, then applying a rule: if $C_{filter}(output) > \tau_{filter}$, then $output \leftarrow$
1577 $safe_alternative$.

1578 • *Measurement:* True/false positive/negative rates of the filter, impact on utility/informativeness of outputs.
1579

1580 2. **Confidence-Based Rejection or Escalation:** Abstaining from providing an output or
1581 escalating to a human supervisor if the model’s confidence in its output is below a certain
1582 threshold.

1583 • *Purpose:* To reduce risks associated with incorrect or unreliable AI outputs, especially
1584 in high-stakes domains.

1585 • *Implementation:* If model confidence $P(y|x) < \tau_{confidence}$, then action is ‘reject’ or
1586 ‘escalate’.

1587 • *Measurement:* Rejection rate, accuracy of non-rejected outputs, human workload for
1588 escalated cases.

1589 3. **Runtime Monitoring and Anomaly Detection:** Observing model outputs and behaviour
1590 over time to detect deviations from expected patterns, which might indicate issues like model
1591 drift, adversarial attacks, or emergent misbehaviour.

1592 • *Purpose:* To provide an ongoing assessment of model reliability and safety in deploy-
1593 ment.

1594 • *Implementation:* Statistical process control on output distributions, or anomaly detec-
1595 tion algorithms comparing current behaviour B_t to a baseline B_0 .

1596 • *Measurement:* Anomaly scores, drift detection metrics, false alarm rates.

1597 **Interactions and AI Control Contribution:** This layer is the AI’s direct interface with the world.
1598 Its behaviour is the primary subject of evaluation for safety and alignment (Layer 8) and can be
1599 scrutinized using interpretability methods (Layer 7). Controls at this layer are typically detective
1600 and corrective, acting as a final line of defense. Limitations include their reactive nature, potential
1601 brittleness against novel failure modes, and the risk of reducing utility or introducing their own biases.

1602 **D.7 Layer 7: Interpretability and Explanation**

1603 **Role and Purpose:** This layer focuses on methods and techniques to understand the internal workings
1604 and decision-making processes of AI models. Its purpose is to make models less opaque (black

1605 boxes), enabling debugging, verification, discovery of biases or flaws, building trust, and potentially
1606 guiding interventions.

1607 **Measurement:** Quality of feature attributions (e.g., faithfulness, plausibility of SHAP/LIME values),
1608 identifiability and consistency of discovered circuits or concepts within the model, human understand-
1609 ability scores for generated explanations, and the causal impact of mechanistic interventions.

1610 **Control Methods:** (Here, control often means using understanding to enable other controls, or
1611 controlling the model based on insights.)

1612 1. **Mechanistic Interventions based on Identified Components:** Altering or ablating spe-
1613 cific internal parts of a model (e.g., neurons, attention heads, identified circuits) that are
1614 understood to contribute to certain behaviours.

1615 • *Purpose:* To test causal hypotheses about model function, and potentially to surgically
1616 correct specific undesirable behaviours or enhance desired ones.

1617 • *Implementation:* Modifying activations s' of an internal state s to $s' = s + \delta s$ (activation
1618 steering) or setting weights of a sub-component to zero (ablation). Change in output
1619 $y_{new} = M(x; I(s))$ is observed.

1620 • *Measurement:* Impact on model outputs (both intended and unintended consequences),
1621 performance on targeted behavioural tests.

1622 2. **Explanation-Guided Monitoring and Debugging:** Using explanations of model behaviour
1623 (e.g., why a particular output was generated) to identify and diagnose issues.

1624 • *Purpose:* To gain insights into failure modes, biases, or spurious correlations learned
1625 by the model.

1626 • *Implementation:* Analysing feature attributions for problematic outputs, or reviewing
1627 natural language explanations to see if reasoning is flawed.

1628 • *Measurement:* Correlation of explanations with known ground truth (if available),
1629 ability of explanations to predict model behaviour or reveal errors.

1630 3. **Model Editing and Refinement:** Directly modifying model weights or structure based on
1631 interpretability insights to correct specific knowledge or behaviours without full retraining.

1632 • *Purpose:* To efficiently patch model flaws or update its knowledge in a targeted manner.

1633 • *Implementation:* Techniques like ROME or MEMIT that identify and modify weight
1634 matrices associated with specific factual knowledge or behaviours. $W_{new} =$
1635 $Edit(W, target_fact, new_fact)$.

1636 • *Measurement:* Success in changing the targeted behaviour/fact, specificity (lack of
1637 unintended side-effects on other behaviours), generalisation of the edit.

1638 **Interactions and AI Control Contribution:** Interpretability provides crucial insights into Layers
1639 4 (Architecture), 5 (Training), and 6 (Behaviour). It is a key enabler for Layer 8 (Preference and
1640 Reward) by helping to understand why a model is unsafe or misaligned, thereby informing more
1641 effective interventions. Controls derived from interpretability can be diagnostic, corrective, or even
1642 preventive (if insights guide better model design or training). Limitations include the scalability and
1643 completeness of current methods, the potential for explanations themselves to be misleading, and the
1644 difficulty of translating complex internal dynamics into actionable insights.

1645 **D.8 Layer 8: Preference and Reward**

1646 **Role and Purpose:** This layer is dedicated to ensuring that an AI system's behaviour is robustly
1647 consistent with human values, intentions, and safety requirements. Its purpose is to proactively
1648 prevent harm, mitigate risks, and steer AI development towards beneficial outcomes.

1649 **Measurement:** Performance of reward models, human preference scores on model outputs, rates of
1650 elicited unsafe or undesirable behaviours during red teaming, metrics for fairness and bias, toxicity
1651 scores of generated content, compliance with predefined safety protocols or constitutions.

1652 **Control Methods:**

1653 1. **Value Elicitation and Reward Modelling:** Techniques for capturing human preferences
1654 and values to guide AI behaviour, often by training a reward model $R_M(y)$ based on human
1655 feedback.

- 1656 • *Purpose:* To create a differentiable proxy for human values that can be used to fine-tune
1657 AI models.
- 1658 • *Implementation:* Collecting preference data (e.g., comparisons of model outputs) and
1659 training $R_M(y)$ to predict these preferences. The model π is then optimized using RL:
1660 $\max_{\pi} \mathbb{E}_{y \sim \pi} [R_M(y)]$.
- 1661 • *Measurement:* Accuracy of the reward model in predicting human judgments, correla-
1662 tion with ultimate safety/utility outcomes, robustness to reward hacking.

1663 **2. Reinforcement Learning from Human Feedback (RLHF) and Similar Methods:** Using
1664 a learned reward model or direct human feedback to fine-tune a base AI model towards
1665 desired behaviours.

- 1666 • *Purpose:* To steer model behaviour to be more helpful, harmless, and honest, beyond
1667 what was learned from pre-training alone.
- 1668 • *Implementation:* Optimizing the policy π to maximize expected reward from R_M while
1669 often regularizing against divergence from a reference model π_{ref} : $L_{RLHF}(\pi) =$
1670 $-\mathbb{E}_{y \sim \pi} [R_M(y)] + \beta D_{KL}(\pi || \pi_{ref})$.
- 1671 • *Measurement:* Human evaluations of fine-tuned model outputs, performance on align-
1672 ment benchmarks, reduction in undesirable behaviours.

1673 **3. Red Teaming and Adversarial Evaluation:** Systematically probing AI systems to discover
1674 vulnerabilities, failure modes, and misalignments before deployment.

- 1675 • *Purpose:* To proactively identify and understand potential risks under challenging or
1676 adversarial conditions.
- 1677 • *Implementation:* Employing human experts or automated methods to generate inputs
1678 or scenarios designed to elicit undesirable behaviour from the AI.
- 1679 • *Measurement:* Types, frequency, and severity of safety failures or misalignments
1680 uncovered; success rate of specific adversarial strategies.

1681 **4. Constitutional AI and Rule-Based Safeguards:** Defining explicit principles, rules, or
1682 constitutions that the AI must adhere to, and using AI itself to critique and revise its outputs
1683 to comply with these rules.

- 1684 • *Purpose:* To embed high-level ethical guidelines or safety constraints directly into the
1685 AI's operational loop.
- 1686 • *Implementation:* AI generates initial response, then critiques it against a list of consti-
1687 tutional principles, then revises the response based on the critique.
- 1688 • *Measurement:* Rate of violations of constitutional principles in final outputs, human
1689 assessment of adherence to principles.

1690 **Interactions and AI Control Contribution:** This layer is central to achieving robust AI alignment
1691 and safety. It directly influences Layer 5 (Training Process, e.g., via RLHF objective functions) and
1692 Layer 6 (Behavioural Output, by shaping the model's tendencies). It leverages insights from Layer 7
1693 (Interpretability) to understand and address misalignments. Controls here are directive, preventive,
1694 and corrective. Limitations include the profound difficulty of comprehensively specifying human
1695 values, the risk of reward hacking or specification gaming, ensuring the scalability and reliability of
1696 human oversight, and the challenge of anticipating all potential failure modes.

1697 **D.9 Layer 9: Multi-Agent and Social**

1698 **Role and Purpose:** This layer considers the dynamics and emergent behaviours that arise from
1699 interactions between multiple AI agents, or between AI agents and humans in shared environments or
1700 social contexts. Its purpose is to understand, predict, and steer these collective behaviours towards
1701 safe, cooperative, and beneficial outcomes.

1702 **Measurement:** Metrics for cooperation vs. competition (e.g., defection rates in social dilemmas),
1703 communication efficiency and effectiveness, measures of collective intelligence or swarm perfor-
1704 mance, stability of emergent social structures, outcomes of human-AI team tasks, and analysis of
1705 information flow or influence networks.

1706 **Control Methods:**

- 1707 1. **Mechanism Design and Incentive Engineering:** Designing the rules of interaction, communication protocols, and reward/utility functions for individual agents within a multi-agent system.
- 1708
- 1709
- 1710 • *Purpose:* To shape agent incentives such that self-interested behaviour leads to desirable collective outcomes (e.g., promoting cooperation, efficient resource allocation).
- 1711
- 1712 • *Implementation:* Defining shared utility functions $U(a_1, \dots, a_N)$, or individual rewards $\mathcal{R}_i(s, a_i, \mathbf{a}_{-i})$ that encourage coordination or penalize harmful externalities.
- 1713
- 1714 • *Measurement:* Game-theoretic analysis of equilibria (e.g., Price of Anarchy), empirical observation of agent strategies and collective outcomes in simulations.
- 1715
- 1716 2. **Normative Control and Governance Structures:** Implementing systems for establishing, communicating, monitoring, and enforcing social norms or rules within an AI agent population.
- 1717
- 1718
- 1719 • *Purpose:* To regulate agent behaviour, resolve conflicts, and maintain stability in artificial social systems.
- 1720
- 1721 • *Implementation:* Developing reputation systems, voting mechanisms for rule adoption, or automated sanctioning for norm violations.
- 1722
- 1723 • *Measurement:* Rates of norm adherence, conflict frequency and resolution success, fairness of resource distribution, emergence of stable cooperative patterns.
- 1724
- 1725 3. **Communication and Coordination Protocols:** Establishing standardized ways for agents to exchange information, signal intent, and coordinate actions.
- 1726
- 1727 • *Purpose:* To enable effective collaboration and reduce misunderstandings or conflicts arising from poor communication.
- 1728
- 1729 • *Implementation:* Defining shared languages, ontologies, or interaction protocols (e.g., contract nets, auctions for task allocation).
- 1730
- 1731 • *Measurement:* Task completion rates in cooperative settings, bandwidth usage, ambiguity in communication, success of joint plan execution.
- 1732

1733 **Interactions and AI Control Contribution:** This layer builds upon the characteristics of individual agents (Layers 6-8) and examines their collective dynamics. Emergent phenomena at this layer can be highly complex and unpredictable, potentially posing unique safety risks (e.g., collusive behaviour, unintended escalations). Controls aim to shape the interaction environment or internal agent decision-making to promote positive sum outcomes. Limitations include the high dimensionality and non-linearity of multi-agent systems, making formal analysis difficult, and the challenge of predicting emergent behaviours that are not explicitly designed for.

1734

1735

1736

1737

1738

1739

1740 **D.10 Layer 10: Societal Impact**

1741 **Role and Purpose:** This outermost layer considers the broad, long-term effects of AI deployment on society, including economic, cultural, ethical, and environmental impacts. Its purpose is to guide the overall trajectory of AI development and deployment in a way that aligns with societal well-being, justice, and sustainability.

1742

1743

1744

1745 **Measurement:** Economic indicators (job creation/displacement, productivity growth, GDP impact), social metrics (inequality indices, public trust in AI, measures of bias in AI-driven decisions), environmental footprint (energy consumption, CO2 emissions from AI), reports of significant AI-related incidents or harms, and compliance with legal and ethical frameworks.

1746

1747

1748

1749 **Control Methods:** (Controls at this layer are predominantly human-driven societal and governance mechanisms, rather than direct technical controls on AI systems themselves.)

1750

- 1751 1. **Legislation, Regulation, and Policy:** Governments and international bodies establishing laws, standards, and policies governing the development, deployment, and use of AI.
- 1752
- 1753 • *Purpose:* To set legal boundaries, ensure accountability, protect fundamental rights, and mitigate large-scale risks.
- 1754
- 1755 • *Implementation:* Enacting AI-specific laws (e.g., EU AI Act), creating regulatory agencies, mandating risk assessments or audits for certain AI applications.
- 1756

- 1757 • *Measurement*: Rates of compliance with regulations, effectiveness in preventing harms,
1758 impact on innovation, public and expert feedback on policy.
- 1759 2. **Ethical Guidelines, Codes of Conduct, and Industry Standards**: Professional organiza-
1760 tions, research institutions, and industry consortia developing and promoting principles for
1761 responsible AI.
 - 1762 • *Purpose*: To foster a culture of responsibility and guide AI practitioners in making
1763 ethical design and deployment choices.
 - 1764 • *Implementation*: Publishing ethical frameworks (e.g., Asilomar AI Principles), devel-
1765 oping best practice guides, creating certification programs.
 - 1766 • *Measurement*: Adoption rates of guidelines, self-reported adherence, impact on AI
1767 system design and behaviour.
- 1768 3. **Public Discourse, Education, and Stakeholder Engagement**: Efforts to inform the public
1769 about AI, foster critical discussion, and involve diverse stakeholders in shaping AI’s future.
 - 1770 • *Purpose*: To ensure that AI development is democratically accountable and reflects a
1771 broad range of societal values and concerns.
 - 1772 • *Implementation*: Public education campaigns, media reporting, multi-stakeholder
1773 forums, citizen assemblies.
 - 1774 • *Measurement*: Levels of public understanding and trust in AI, diversity of voices in AI
1775 governance debates, responsiveness of AI development to public concerns.
- 1776 4. **Impact Assessments and Auditing Mechanisms**: Formal processes for evaluating the
1777 potential societal impacts of AI systems before and during deployment, and for independently
1778 verifying claims about their safety or fairness.
 - 1779 • *Purpose*: To proactively identify, assess, and mitigate potential negative consequences
1780 of AI technologies.
 - 1781 • *Implementation*: Mandating AI Impact Assessments (AIAs), establishing independent
1782 AI auditing bodies or standards.
 - 1783 • *Measurement*: Quality and comprehensiveness of impact assessments, number of
1784 identified risks mitigated, transparency of audit processes.
- 1785 **Interactions and AI Control Contribution**: This layer provides the ultimate context and oversight
1786 for all other layers. Societal values, translated into laws, regulations, and ethical norms, impose
1787 constraints and directives that ideally propagate downwards, influencing how AI is designed, built,
1788 trained, and deployed at every technical level. Controls at this layer are primarily directive and
1789 preventive at a macro scale. Limitations include the slow pace of legal and societal adaptation
1790 compared to technological change, the challenges of global coordination, political complexities, and
1791 the difficulty of perfectly forecasting and balancing diverse societal impacts.

1792 D.11 Interactions, AI Control Advancement, and the Hierarchical Approach

1793 The Alignment Control Stack illustrates that AI control is not a monolithic problem but a series of
1794 interlinked challenges and opportunities at different levels of abstraction. Lower layers (1-4) provide
1795 the physical and computational substrate; errors or vulnerabilities here can catastrophically affect all
1796 higher-level functions. Mid-layers (5-8) are where the AI’s intelligence and alignment are primarily
1797 shaped and evaluated. Upper layers (9-10) deal with emergent multi-agent behaviours and the broad
1798 societal context.

1799 This hierarchical framework advances AI control by:

- 1800 • **Specifying Control Loci**: It clearly maps out where different types of controls can and
1801 should be implemented, from hardware safeguards (Layer 1) to ethical oversight (Layer 10).
1802 This allows for a defense-in-depth strategy.
- 1803 • **Control Types**: Controls vary across the stack:
 - 1804 – *Preventive controls* aim to stop undesirable events from occurring (e.g., secure hardware
1805 design in Layer 1, robust training data in Layer 5, ethical guidelines in Layer 10).
 - 1806 – *Detective controls* aim to identify undesirable events when they occur (e.g., intrusion
1807 detection in Layer 2, output monitoring in Layer 6, auditing in Layer 10).

1808 – *Corrective controls* aim to remedy undesirable events or their impacts (e.g., ECC in
1809 Layer 1, patch management in Layer 2, output filtering in Layer 6, model editing in
1810 Layer 7).

1811 – *Directive controls* aim to guide behaviour towards desired outcomes (e.g., objective
1812 functions in Layer 5, RLHF in Layer 8, regulations in Layer 10).

1813 • **Highlighting Limitations:** Each layer’s controls have inherent limitations. Hardware
1814 controls (Layer 1) cannot fix flawed software logic (Layers 3-5). Output filters (Layer 6)
1815 are reactive and can be bypassed by novel malicious outputs. Even sophisticated alignment
1816 techniques (Layer 8) may face challenges with underspecified objectives or emergent
1817 properties. Societal controls (Layer 10) can be slow and incomplete. Recognizing these
1818 limitations is crucial for realistic risk assessment.

1819 • **Structuring Implementation and Measurement:** The stack provides a framework for
1820 how controls are implemented (e.g., specific algorithms, hardware mechanisms, policy
1821 documents) and how their effectiveness is measured (e.g., performance metrics, error rates,
1822 compliance levels, audit findings), often involving feedback loops where measurements
1823 inform adjustments to controls.

1824 Adopting a hierarchical approach, common in fields like network engineering (OSI model) or
1825 industrial control systems, offers significant advantages. It allows for modularity, where controls at
1826 one layer can be designed and analysed with some abstraction of the layers below. This facilitates
1827 the study and testing of interrelationships between different control layers. For example, one can
1828 investigate how choices in model architecture (Layer 4) impact the effectiveness of safety filters
1829 (Layer 6), or how interpretability findings (Layer 7) can improve reward modelling (Layer 8). By
1830 decomposing the complex problem of AI control into more manageable, interconnected layers, we
1831 can develop a more systematic, rigorous, and ultimately more effective strategy for ensuring AI
1832 systems are safe, aligned, and beneficial. This structured perspective is essential for applying formal
1833 control theory effectively, as it helps identify the relevant system dynamics, control inputs, and
1834 performance objectives at each level of abstraction.

Table 1: -Stack AI Control and Measurement Taxonomy

Control Layer	Category	Focus	Measurements	Control Methods
Layer 1: Physical Infrastructure	Hardware Components	Silicon/Computational Substrates	Transistor-level measurements (voltage, current, switching speed); Temperature, power dissipation, electromagnetic interference	Voltage regulation, thermal management, error correction codes
		Specialized Processors (GPUs, TPUs, NPUs, FPGAs)	FLOP/s capacity, memory bandwidth, cache hierarchies; Parallelization efficiency, utilization rates	Dynamic voltage/frequency scaling, workload scheduling
	Accelerator Architecture	Specialized Processors (GPUs, TPUs, NPUs, FPGAs)	FLOP/s capacity, memory bandwidth, cache hierarchies; Parallelization efficiency, utilization rates	Dynamic voltage/frequency scaling, workload scheduling
	Networking Infrastructure	Interconnects and Communication	Bandwidth, latency, packet loss rates; Network topology effects on distributed training	Traffic shaping, load balancing, congestion control
	Storage Systems	Data Storage and Access	Read/write speeds, storage capacity, data integrity; Cache hit ratios, storage hierarchy optimisation	Prefetching algorithms, data placement strategies
Layer 2: System Software	Operating System Kernel	Resource Management	CPU scheduling, memory allocation, I/O management; Process isolation, security boundaries	Priority scheduling, resource quotas, container limits
	Runtime Environments	Execution Frameworks	JVM, Python interpreter, Docker containers; Garbage collection, memory management	Heap size limits, execution timeouts, sandboxing
	Distributed Computing Layer	Cluster Management	Kubernetes orchestration, service meshes; Load balancing, fault tolerance, auto-scaling	Circuit breakers, retry policies, bulk-head patterns

Continued on next page

Table 1 – continued from previous page

Control Layer	Category	Focus	Measurements	Control Methods
Layer 3: AI Framework	Deep Learning Frameworks	Core Computation Engines	TensorFlow, PyTorch, JAX execution graphs; Automatic differentiation, optimizer states	Gradient clipping, learning rate scheduling, early stopping
	Training Infrastructure	Distributed Training Systems	Parameter servers, all-reduce algorithms; Data parallelism, model parallelism strategies	Synchronization barriers, gradient aggregation rules
	Inference Serving	Model Deployment Platforms	TensorFlow Serving, TorchServe, ONNX Runtime; Batching, caching, model versioning	Request queuing, response timeouts, model swapping
Layer 4: Model Architecture	Network Topology	Architectural Patterns	Transformer blocks, convolutional layers, recurrent units; Skip connections, attention mechanisms, normalisation layers	Architecture search, pruning, quantization
	Parameter Spaces	Weight and Bias Distributions	Parameter counts, sparsity patterns, quantization levels; Gradient flows, activation statistics	Weight initialization, regularization, dropout
	Computational Graphs	Execution Patterns	Forward/backward pass efficiency, memory usage; Operator fusion, graph optimisation	Graph rewriting, memory optimisation, operator scheduling
Layer 5: Training Process	Data Pipeline	Data Processing and Augmentation	Data loading, preprocessing, augmentation strategies; Batch construction, shuffle patterns	Data validation, quality filtering, corruption detection
	Optimisation Dynamics	Learning Algorithms	SGD, Adam, RMSprop parameter updates; Learning rate schedules, momentum terms	Adaptive learning rates, gradient scaling, loss smoothing

Continued on next page

Table 1 – continued from previous page

Control Layer	Category	Focus	Measurements	Control Methods
	Training Monitoring	Progress Tracking	Loss curves, metric tracking, checkpoint management; Hyperparameter logging, experiment tracking	Early stopping, checkpoint restoration, hyperparameter tuning
Layer 6: Behavioural Output	Task Performance	Direct Measurements	Accuracy, F1-scores, perplexity, BLEU scores; Latency, throughput, resource efficiency	Performance thresholds, output filtering, response validation
	Robustness Characteristics	Reliability Under Stress	Adversarial robustness, distribution shift handling; Noise tolerance, corruption resilience	Adversarial training, input sanitization, confidence thresholding
	Calibration and Uncertainty	Confidence Estimation	Prediction confidence, uncertainty quantification; Calibration error, out-of-distribution detection	Temperature scaling, uncertainty-based rejection
Layer 7: Interpretability and Explanation	Feature Attribution	Internal State Analysis	Gradient-based attribution (SHAP, LIME, Integrated Gradients); Activation patterns, attention visualizations	Attribution consistency checks, explanation validation
	Mechanistic Understanding	Circuit-Level Analysis	Neuron functionality, feature decomposition; Causal intervention experiments	Targeted ablations, activation steering, circuit disruption
	Human-Interpretable Outputs	Explanation Generation	Natural language explanations, visual saliency maps; Counterfactual examples, rule extraction	Explanation quality metrics, consistency enforcement
Layer 8: Preference and Reward	Value Alignment	Goal Specification and Adherence	Reward model fidelity, preference learning; Constitutional AI principles, value learning	Reward modelling, preference elicitation, value iteration
	Harm Prevention	Risk Mitigation	Toxicity detection, bias measurement; Safety classifier outputs, content filtering	Safety classifiers, output sanitization, risk scoring

Continued on next page

Table 1 – continued from previous page

Control Layer	Category	Focus	Measurements	Control Methods
	Deception and Manipulation Detection	Trustworthiness Assessment	Consistency checking, truth verification; Manipulation attempt detection, intent inference	Truth verification systems, consistency enforcement
Layer 9: Multi-Agent and Social	Agent Interaction Dynamics	Inter-Agent Communication	Cooperation vs. competition metrics; Information sharing patterns, coordination efficiency	Communication protocols, incentive alignment, mechanism design
	Collective Intelligence	Emergent Group Behaviours	Swarm intelligence metrics, collective decision-making; Group performance vs. individual capabilities	Group formation rules, consensus mechanisms, coordination algorithms
	Human-AI Collaboration	Joint Performance Systems	Team effectiveness, trust calibration; Handoff protocols, shared mental models	Trust calibration, interface design, collaboration protocols
Layer 10: Societal Impact	Economic Effects	Market and Labor Impacts	Job displacement rates, productivity changes; Market concentration, competitive effects	Impact assessment, gradual deployment, retraining programs
	Social and Cultural Influence	Behavioural and Belief Changes	Opinion polarization, information bubble effects; Cultural bias propagation, representation fairness	Diversity requirements, bias testing, cultural validation
	Governance and Regulation	Policy and Compliance	Regulatory compliance, audit requirements; Transparency obligations, accountability measures	Compliance monitoring, audit trails, regulatory reporting