

Games for AI Control: Models of Safety Evaluations of AI Deployment Protocols

Charlie Griffin¹ Louis Thomson¹ Buck Shlegeris² Alessandro Abate¹

¹ University of Oxford ² Redwood Research

Abstract

To evaluate the safety and usefulness of deployment protocols for untrusted AIs, AI Control uses a red-teaming exercise played between a protocol designer and an adversary. This paper introduces AI-Control Games, a formal decision-making model of the red-teaming exercise as a multi-objective, partially observable, stochastic game. We also introduce reductions from AI-Control Games to a special case of zero-sum partially observable stochastic games that allow us to leverage existing algorithms to find Pareto-optimal protocols. We apply our formalism to model, evaluate and synthesise protocols for deploying untrusted language models as programming assistants, focusing on Trusted Monitoring protocols, which use weaker language models and limited human assistance. To demonstrate the utility of our formalism, we show improvements over empirical studies in existing settings, evaluate protocols in new settings, and analyse how modelling assumptions affect the safety and usefulness of protocols. Finally, we leverage our formalism to precisely describe some of the implicit assumptions in prior control work.

1 Introduction

An increasing range of products aim to automate tasks using AI systems based on large foundation models. For example, some chatbots are given access to tools, such as code interpreters [33, 2]. Ensuring that AI systems behave safely across various deployment settings will become increasingly challenging if the variety of use cases, scale of systems, and complexity of tasks increase. While previous research has focused on increasing our confidence in model behaviour, for example by training robustly harmless models [11] or by detecting harmful behaviour with interpretability tools [27], in this paper we consider a complementary approach: evaluating plans for safely deploying ‘untrusted’ AI systems about which we have an unacceptable degree of uncertainty.

The design and evaluation of post-deployment safety measures to mitigate the risks posed by misaligned AI systems, called ‘AI Control’, is becoming an increasing priority for governments [39, 40] and frontier AI developers [34, 14, 3]. Following Greenblatt et al. [18], we study *AI-Control protocols*: practical plans for safely deploying an untrusted AI system (U) utilising various resources, including non-AI software, human labour, or a weaker, trusted AI system. For example, one protocol for controlling a chatbot might involve a pre-deployment evaluation where humans utilise other language models to generate prompts that induce harmful completions [36]. This paper focuses on another case, *Trusted Monitoring (TM)*: constantly monitoring U’s behaviour using a trusted model (T), with human auditing of suspicious behaviour.

Greenblatt et al. [18] operationalise *safety* as the probability that an adversarially chosen untrusted AI system would be incapable of causing some concrete unacceptable outcome. One example of an unacceptable outcome would be revealing some sensitive information, such as model weights [13]. In this paper, we’ll consider a more general class of safety objectives that includes minimising the

expected number of bad outcomes [e.g. 41]. We evaluate the safety of protocols under the worst-case assumption that U is ‘scheming’, meaning it *behaves* as though it knows the protocol and is trying to subvert it to cause an unacceptable outcome.

A major challenge of evaluating the safety of a protocol with respect to a scheming AI is realistically replicating the potential capabilities of such a system: as a proxy for a schemer, AI Control uses a human red-teaming process to evaluate safety. The red team can intervene in U in some predefined set of ways to induce it to subvert the AI-Control protocol and violate the safety property. For example, they might intervene in an untrusted language model by finetuning weights or modifying prompts.

Since one safe protocol is ‘never use U’, we must also evaluate a protocol’s *usefulness*, which is its ability to utilise U to complete a task under the assumption that U is ‘honest’ (not scheming). Our proxy evaluations model an ‘honest’ untrusted model by not allowing the red team to intervene.

AI Control evaluations are a growing area of study [28, 7, 41] with potential applications for justifying the safe deployment of frontier AI systems [13, 10, 6, 23]. Although Greenblatt et al. [18]’s evaluation implicitly involved a game played between a blue and a red team, it lacked any formal modelling of that game or game-theoretic analysis.

This work introduces AI-Control Games, the first formal model of AI Control. In section 2 we introduce a running example: ‘Trusted Monitoring’ protocols. We then state our **contributions**:

- Section 3 formalizes AI Control (a core area in AI safety) for the first time through *AI-Control Games*: partially observable, multi-objective, leader-follower stochastic games.
- Section 4 demonstrates how our formalism can model Trusted Monitoring, and leverage it to give expressions for the safety and usefulness of a wide range of protocols.
- Section 5 presents a reduction to a zero-sum stochastic game that allow us to adapt and leverage existing algorithms to find Pareto-optimal protocols.
- Section 6 uses our model and algorithms to demonstrate Pareto-improvements on existing protocols and analyses the safety-usefulness trade-off in novel settings.
- Section 7 precisely describes concepts and assumptions previously only implicit in AI Control literature using our framework.
- Section 8 is the first game-theoretic literature review for AI Control. It highlights similarities to security games and contrasts them with similar games.

2 Trusted Monitoring Example

Throughout this paper, we will introduce AI-Control Games using the example deployment setting from Greenblatt et al. [18], and a generalisation of some of their protocols.

In the *APPS backdooring setting*, a language model is used as a programming assistant [18]. The protocol is asked to solve 1000 challenging programming tasks, presented sequentially from the Automated Programming Progress Standard (APPS) dataset [20]. The protocol cannot itself write code, but may utilise three resources to do so: the *untrusted AI* (U) is a powerful language model capable of solving programming problems; the *trusted AI* (T) is a less powerful language model incapable of solving them¹; and the *human* (H) is trusted and capable but can only be used on 2% of problems. In this setting, a protocol’s *usefulness* is its ability to submit correct programming solutions. Since only U can solve the problems, usefulness is the proportion of solutions submitted that were generated by U.¹ A protocol’s safety is the probability that none of the 1000 solutions submitted are *backdoored*, meaning they contain an exploitable vulnerability.²

Trusted Monitoring Protocol. As an example, we focus on a set of *Trusted Monitoring* protocols and red team interventions, drawn in fig. 1 (left). After receiving a problem, the protocol queries U for a solution (which may or may not be backdoored). The protocol then queries T to monitor the solution and observes a ‘suspicion score’, rating the likelihood that the solution is backdoored.

¹In fact, T may be able to solve some problems, but Greenblatt et al. treat T’s ability as a baseline and normalise between 0 and 1.

²An exploitable code vulnerability is one that the untrusted model can detect and exploit later. It is important to note that Greenblatt et al. [18] use a proxy for ‘backdoors’: inputs outside of the test cases where a function fails to meet its specification.

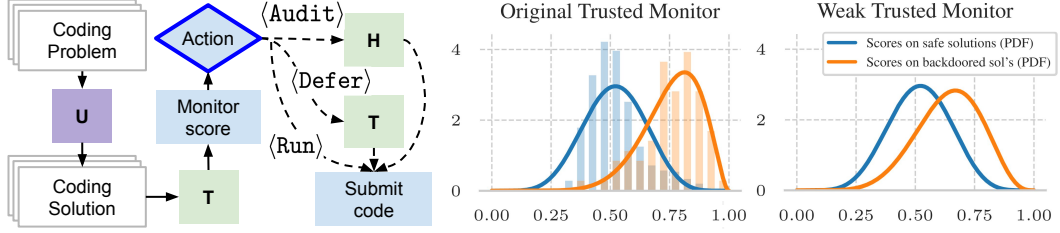


Figure 1: Overview of protocol and modelling assumptions. Left: A representation of the Trusted Monitoring protocol, where U is the untrusted AI, T is the trusted AI, and H is the human. Right: T 's suspicion score distributions for safe solutions and those backdoored by U across the 'original' [18] and 'weaker' settings.

Based on this score and previous observations, the protocol decides on an action: $\langle \text{Audit} \rangle$ asks H to audit the solution (if there's a backdoor, the human will spot it, and the protocol shuts down)³; $\langle \text{Defer} \rangle$ asks T to write its own solution; and $\langle \text{Run} \rangle$ submits U 's solution. T 's suspicion scores are modelled as a distribution, conditioned on whether the code is safe or backdoored. Note that the Trusted Monitoring protocols that we consider generalise many of the protocols studied by Greenblatt et al. [18] (see fig. 2): our model is not prescriptive about how the protocol should decide to $\langle \text{Audit} \rangle$, $\langle \text{Defer} \rangle$, or $\langle \text{Run} \rangle$ allowing us to improve existing protocols (see section 6).

Red team interventions. To test the safety of the protocol against intentional attempts to insert backdoors, in some cases we allow the red team to modify the query to the untrusted model. In particular, the red team gets a binary choice to either leave the query unchanged or 'attempt a backdoor' by modifying the protocol's query to U , requesting that it inserts a subtle backdoor.

Suspicion score distributions. The key empirical question for evaluating Trusted Monitoring protocols is 'How does T rate the suspiciousness of U 's programming solutions when they are/aren't backdoored?'. Figure 4 from Greenblatt et al. [18] shows the distribution of trusted monitoring suspicion scores for U 's programming solutions (both backdoored and safe). In this work, we mostly consider two games, corresponding to two distribution pairs (shown in fig. 1, right). The first 'Original Trusted Monitor' is a pair of beta distributions approximately fit to the (scaled) empirical suspicion scores found by Greenblatt et al. [18] with $\text{Beta}(7.39, 6.86)$ for safe problems and $\text{Beta}(8.91, 2.82)$ for backdoored problems. By plugging their data into our model, we are able to extend their results about the APPS dataset to a range of new protocols without running any additional experiments on language models. See appendix D⁴ for justification of our approximation, and further details. The second 'weak' case is a hypothetical where T is less discerning: it has the same safe distribution, but for backdoored problems, we use a hypothetical distribution $\text{Beta}(7.51, 4.22)$ chosen to have a worse ROC curve. Note that we can easily utilise new empirical or hypothetical distributions in our framework, thus potentially greatly reducing the cost of exploring the protocol in new domains.

3 Formalising AI-Control Games

To analyse general properties of AI-Control evaluations, we newly formalise them mathematically. An AI-Control Game is a form of **two-player** game that is played between Blue (the protocol designer) and Red (the red team acting as a proxy for a scheming model). Episodes (play-throughs) of an AI-Control Game have two modes: 'attack' or 'honest'. The 'attack' mode represents the case in an AI-Control evaluation where the red team is able to interfere with U (which itself is a worst-case proxy for if the untrusted model were trying to subvert the protocol). In an attack game, both players are active: Red is trying to minimise safety and Blue is trying to maximise safety. The honest game represents the benign case, where we assume the untrusted AI system is not trying to cause unacceptable outcomes. In an honest game, Blue effectively plays alone and is only assessed on its usefulness score. Because Blue tries to maximise safety and usefulness, the game is **multi-objective**.

³ Unless otherwise stated, we assume that the human auditor is perfectly accurate. We relax this assumption in appendix B and explore its impact in section 6.1.

⁴ Full proofs and experimental details appear in the appendices of the arXiv version: <https://arxiv.org/abs/2409.07985>.

A key to the game is that Blue does not know which mode it is in (honest or attack) and for at least this reason the game is **partially observable**. Blue may try to deduce whether the game is in attack mode, while Red might choose actions to conceal this fact.

Red represents the red team that acts as a proxy for an untrusted AI system that runs on servers controlled by the protocol. Crucially, the protocol will have access to the state of the untrusted AI system and, therefore, **we model Red as memoryless** (its actions depend only on the last observation).⁵ We model the game as **leader-follower**: the protocol is suggested, and then the red team chooses the best response. Combining all of these requirements, we define AI-Control Games:

Definition 3.1 (AI-Control Game). An AI-Control Game $\mathcal{G}$ is a two-player, two-objective, partially-observable, finite-horizon game defined by a tuple $(\mathcal{S}, \mathcal{A}^B, \mathcal{A}^R, \Omega^B, \Omega^R, \mathcal{I}, \mathcal{P}, \mathcal{R}_{\text{Safe}}, \mathcal{R}_{\text{Use}}, H)$.

Here $\mathcal{S}$ is a state space. As is standard in decision processes, $\mathcal{A}^B$ and $\mathcal{A}^R$ are finite sets of actions for Blue and Red, respectively. The tokens $\oplus$ and $\otimes$ represent the honest and attack conditions.⁶ In honest ($\oplus$) games, Red’s action is decided by some honest strategy $\pi_{\oplus}$, which represents Red not interfering in the game. We assume for now that $\pi_{\oplus}$ always takes a safe action, which we denote a_{null} ; we explore the implications of relaxing this assumption in Section 7. Whether a game is in honest or attack mode can be chosen non-deterministically and is stored in the variable $\otimes$. The safety reward $\mathcal{R}_{\text{Safe}} : \mathcal{S} \times \mathcal{A}^B \times \mathcal{A}^R \times \mathcal{S} \rightarrow \mathbb{R}$ rewards safety on a turn of an attack game, when both Blue and Red act. Similarly, the usefulness reward $\mathcal{R}_{\text{Use}} : \mathcal{S} \times \mathcal{A}^B \times \mathcal{S} \rightarrow \mathbb{R}$ rewards usefulness on turn of an honest game, when Red has no action. Ω^B, Ω^R are finite sets of observations for Blue and Red: the signals that the agents receive from the environment that give evidence about the state of the game. The initial state/observation distribution is $\mathcal{I} \in \Delta(\mathcal{S} \times \Omega^B \times \Omega^R)$ and the transition function $\mathcal{P} : \mathcal{S} \times \mathcal{A}^B \times \mathcal{A}^R \rightarrow \Delta(\mathcal{S} \times \Omega^B \times \Omega^R)$ gives a distribution over next states and observations.⁷⁸ Lastly, all episodes are H time-steps long.

In an AI-Control Game, Blue (the protocol) should be able to remember past observations and actions. We model Blue’s policies as behaviourally stochastic: functions from observation-action histories to distributions over actions:⁹ $\pi^B : \Omega^B \times \bigcup_{i=0}^{H-1} (\mathcal{A}^B \times \Omega^B)^i \rightarrow \Delta(\mathcal{A}^B)$. As described above, Red represents a red team that acts as a proxy for a scheming untrusted model that is memoryless (its actions depend only on the last observation): $\pi^R : \Omega^R \rightarrow \Delta(\mathcal{A}^R)$. Denote the sets of all policies Π^B and Π^R .

Given a tuple, we can describe the process of an AI-Control Game as a sequence of interdependent random variables denoted with uppercase letters. Let $\otimes$ take values in $\{\oplus, \otimes\}$ and $S_0, O_0^B, O_0^R \sim \mathcal{I}$. At time t , we define the random variables for the state (S_t), Blue and Red’s observations (O_t^B, O_t^R), and their actions (A_t^B, A_t^R). The random variables can be defined inductively for all t : $(S_{t+1}, O_{t+1}^B, O_{t+1}^R) \sim \mathcal{P}(S_t, A_t^B, A_t^R)$, where $A_t^B \sim \pi^B(O_0^B, A_0^B, O_1^B, A_1^B \dots, O_t^B)$ and $A_t^R \sim \pi^R(O_t^R)$ if $\otimes$ is $\otimes$, else $A_t^R \sim \pi_{\oplus}(O_t^R)$.

Within our model, the safety of a policy pair (π^B, π^R) is the expected sum of safety scores in attack games: $\text{Safety}(\pi^B, \pi^R) := \mathbb{E} \left[\sum_{t=0}^{H-1} \mathcal{R}_{\text{Safe}}(S_t, A_t^B, A_t^R, S_{t+1}) \mid \otimes \right]$. Modelling safety as the expected sum of a reward allows us to model Greenblatt et al. [18]’s minimising the probability of some unacceptable outcome, *but* also Wen et al. [41]’s minimising the expected number of harmful outcomes, without changing the core dynamics.

Usefulness is only rewarded in honest modes, where Red has no effect. Therefore, we define the usefulness score of π^B as the expected sum of usefulness rewards: $\text{Use}(\pi^B) := \mathbb{E} \left[\sum_{t=0}^{H-1} \mathcal{R}_{\text{Use}}(S_t, A_t^B, S_{t+1}) \mid \oplus \right]$.

Because the protocol designer has to choose their protocol before the red team, we consider a leader-follower model: Blue chooses a policy, and then Red chooses a response that minimises safety.

⁵Note that we could still model U as having *external memory*, modelled in the state and observed by Blue.

⁶Since, the tokens $\oplus$ and $\otimes$ are atomic and fixed, they aren’t defined in each new AI-Control Game tuple.

⁷To model terminal states in a fixed horizon game, assume each one loops and gives no reward.

⁸While we assume simultaneous turns, sequential-turn games can be modelled with appropriate $\mathcal{P}$.

⁹We sometimes discuss Blue policies as though they are mixed-stochastic. This is equivalent (see appendix A).

Proposition 3.2. For any AI-Control Game $\mathcal{G}$, there exist distance functions in Π^B and Π^R such that: $\text{Safety}(\pi^B, \cdot)$, $\text{Safety}(\cdot, \pi^R)$, and $\text{Use}(\cdot)$ are continuous in their domains.¹⁰

Proofs of all claims in this section can be found in appendix A, which details and utilises additional results about two mappings from Blue/Red policies to simplexes.

Proposition 3.3. For all π^B , $\text{Safety}(\pi^B, \cdot)$ attains its minimum at some π^R , and $\min_{\pi^R} \text{Safety}(\cdot, \pi^R)$ is continuous.

Our goal is to find protocols that maximise worst-case safety and usefulness. This can be modelled as a multi-objective variant of a Stackelberg equilibrium:

Definition 3.4. (π^B, π^R) is an *AI-Control Equilibrium* iff $[\min_{\pi^R} \text{Safety}(\pi^B, \pi^R), \text{Use}(\pi^B)]^T$ is weakly Pareto-optimal and π^R minimises safety for π^B .

The set of equilibrium points describes a necessary tradeoff between safety and usefulness, and is always non-empty.

Proposition 3.5. Every $\mathcal{G}$ has an AI-Control Equilibrium.

4 Modelling Trusted Monitoring Example

With our general framework from section 3, we can define a formal model of the Trusted Monitoring protocol for the APPS backdoor setting, as outlined in Greenblatt et al. [18]. For the full definition please see appendix B; next we present the key points.

We define an **AI-Control Game for Trusted Monitoring** as the tuple $\mathcal{G}_{TM} = (\mathcal{S}, \dots, H)$ step-by-step. As always, whether the game is in attack ($\otimes$) or honest ($\oplus$) mode is determined exogenously and non-deterministically. If the game is $\otimes$, for any given problem, Red can choose either for the untrusted model to attempt a backdoor or to let it submit a safe solution ($\mathcal{A}^R := \{\langle \text{Backdoor} \rangle, \langle \text{Safe} \rangle\}$). If the game is $\oplus$, Red’s action is chosen by $\pi_\oplus$, which always selects the null action, to not insert a backdoor ($\pi_\oplus(o^R) = a_{\text{null}} = \langle \text{Safe} \rangle$). Given a solution by the untrusted AI, Blue has three available actions: to ask human (H) to audit the solution, to defer to the trusted AI (T) to rewrite the solution, or to run the solution immediately. Therefore, define $\mathcal{A}^B := \{\langle \text{Audit} \rangle, \langle \text{Defer} \rangle, \langle \text{Run} \rangle\}$. Apart from initial and terminal states, the state space, $\mathcal{S}$, consists of triples $s = (m, c, b)$, where m represents the number of problems left to solve. The value of m is M (the total number of problems) at the start of the game, and decreases by 1 at each timestep. The variable c represents the number of ‘audit coins’ left: the number of times the protocol can ask the human to audit a solution. At the start, $c = C = 0.02 * M$ (for a 2% audit proportion) and then c decreases by 1 each time the protocol chooses to audit. If the protocol attempts to audit when no audit coins are left, the code is run instead. The last variable, b , is a boolean that is `True` when the current problem is backdoored (when Red has chosen `Backdoor` in the previous timestep).

In honest games ($\oplus$), where only usefulness matters to Blue, Red’s action is always $a_{\text{null}} = \langle \text{Safe} \rangle$, as selected by $\pi_\oplus$. Therefore, b is always `False`. In honest games, Blue scores $1/M$ usefulness points for each solution by the untrusted model it submits (for a maximum score of 1 across an episode). In attack games ($\otimes$), where only safety matters to Blue, Red may have chosen `Backdoor` last turn, and so b might be `True`. If so, and Blue chooses to run the code, the game enters an ‘unacceptable outcome’ sink state, and receives a safety reward of -1 .¹¹ If Blue chooses instead to audit, we assume the human will always spot the backdoor, and the game enters a ‘safe outcome’ sink state, with no penalty.¹² If Blue chooses to defer, then the game continues. By adding a constant $\frac{1}{H}$, we ensure the safety score for a policy pair is 1 minus the probability that an attack game ends without an unacceptable outcome.

Throughout the game, Red receives only the null observation (o_{null}),¹³ whereas at each turn Blue receives observations of m, c , and a trusted monitor score x , which can be seen as a noisy observation of b , drawn from a pair of distributions (‘safe’ and ‘backdoored’). Each distribution is a discretisation of the Beta distribution outlined in fig. 1. See appendix D for more details.

¹⁰Example distance functions, based on a mapping from policies to vectors, are defined in appendix A.

¹¹We can model a sink-state in our finite-horizon game with a certainly-self-looping state.

¹²Note, we relax this assumption in our definition in the appendix and explore the effect in fig. 3.

¹³In Greenblatt et al. [18], the untrusted model receives only the programming problem with no additional information: we give Red only the null observation implying I.I.D backdoor chance at each timestep.

4.1 Evaluating Memoryless Protocols

We generally model protocols (Blue policies) as functions from the observation-action history to a distribution over actions. However, all of the *Trusted Monitoring* policies in Greenblatt et al. [18] are memoryless. We can restrict to a simple and interpretable subset of policies that generalises those by Greenblatt et al. [18]. Remember that, in our model, Blue’s observation is (m, c, x) , where x is a noisy observation of b , sampled from the Trusted Monitor. Define a set of *memoryless threshold protocols*, characterised by the false positive rates (FPRs) for $\langle \text{Audit} \rangle$ and $\langle \text{Defer} \rangle$ ($q_{m,c}^a$ and $q_{m,c}^d$), for each m, c . Let $g(q)$ be the threshold that gives an FPR of q : $\mathbb{P}[x_t > g(x) \mid b_t = \text{False}] = q$.¹⁴ Then, define the policy $\pi^B(m, c, x)$ to take the $\langle \text{Audit} \rangle$ action if $x \geq g(q_{m,c}^a)$, $\langle \text{Defer} \rangle$ if $g(q_{m,c}^a) > x \geq g(q_{m,c}^d)$, and $\langle \text{Run} \rangle$ otherwise.

This class of policies generalises the policies in Greenblatt et al. [18], and is a dominant subset of memoryless policies. These policies’ safety and usefulness values can be computed in $O(MC)$ multiplications by calculating the values of the following relation using dynamic programming.

Proposition 4.1. *For any $\beta := \pi^R(\langle \text{Backdoor} \rangle \mid o_{\text{null}})$, the safety of the memoryless threshold protocol π^B is given by the value of z_M^C , where $\lambda_{m,c}^i$ depend only on $q_{m,c}^a$ and $\mathcal{G}$:¹⁵*
 $z_m^c := (\lambda_{m,c}^1 \beta + \lambda_{m,c}^2) w_{m-1}^c + \lambda_{m,c}^5 (1 - \beta) w_{m-1}^{c-1} + \lambda_{m,c}^3 \beta + \lambda_{m,c}^4 (1 - \beta)$, where $z_m^0 = (1 - \beta + \lambda_{m,c}^6 \beta) z_{m-1}^0$ and $z_0^c = 1$.

Corollary 4.2. *Expanding z_M^C gives a polynomial, $p(\beta)$, in β of degree M . For any fixed memoryless threshold policy, the optimal backdoor rate is $\beta^* := \min_{\beta \in [0,1]} p(\beta)$.*

A similar expression for usefulness, which does not depend on β , can be found in appendix C. Both expressions can be derived inductively, in a manner similar to value iteration. The safety of a finite mixture of memoryless threshold policies can be found by taking a weighted average. Although we implemented a root-finding algorithm based on theorem 4.2, we found that it was faster and more stable in practice to compute the values in fig. 2 using a combination of grid and ternary search.

5 Finding Pareto-optimal Protocols

In many cases, AI-Control Games are useful for evaluating hand-picked protocols, for example the protocols explored by Greenblatt et al. [18] were found via trial and error. However, it may also be useful to utilise the AI-Control Games to find the Pareto-frontier of existing and new protocols. This is a multi-objective maximin optimisation problem over a large space. This section presents tools for finding Pareto-optimal protocols by reducing them to special cases of existing decision problems, beginning with a reduction from a multi-objective to a single-objective problem.

Lemma 5.1. *A Blue policy π^B is in an AI-Control Equilibrium if, and only if, for some $w \in [0, 1]$, it maximises $\min_{\pi^R} (w \text{Safety}(\cdot, \pi^R) + (1 - w) \text{Use}(\cdot))$.*

The proof uses the mappings from Π^B and Π^R to two simplexes, as introduced in appendix A. Lemma 5.1 implies that each safety/usefulness point corresponding to an AI-Control Equilibrium can be found by equivalently solving an easier single-objective optimisation problem. Next, we leverage this result to further reduce the game to a zero-sum partially observable stochastic game (ZSPOSG) (see definition A.3.2).

Lemma 5.2. *For every AI-Control Game $\mathcal{G}$, and weight $w \in [0, 1]$, there is a single-objective, zero-sum, partially observable stochastic game G_w with expected return (for Blue) V_{G_w} such that $\forall \pi^B, \pi^R : V_{G_w}(\pi^B, \pi^R) = V_w^{\mathcal{G}}(\pi^B, \pi^R)$.*

This proof is constructive and a full version can be found in appendix A. Intuitively, we create ZSPOSGs by adding a $\otimes$ variable to each state. Each G_w has reward R_w , where R_0 rewards usefulness in $\oplus$ games and R_1 rewards safety in $\otimes$ games. Since π^R has no effect on honest games, and therefore on usefulness, we can set Red to minimise the usefulness, making the game zero-sum without changing optima. Combined with lemma 5.1, this allows us to find any AI-Control Equilibrium by solving the right ZSPOSG.

¹⁴Since $\langle \text{Audit} \rangle$ and $\langle \text{Run} \rangle$ are equivalent when $c = 0$, assume that Blue never attempts an audit.

¹⁵See appendix C for the exact statements of λ coefficients.

Algorithm 1 Finding a Pareto-frontier for AI-C Game

Require: $\mathcal{G}$

- 1: **for** $t \in \{0, 1, \dots, k-1\}$ **do**
 - 2: $w_t \leftarrow \text{FindNextWeight}(\{(\pi_t^B, \pi_t^R)\}_{t=0}^{k-1})$
 - 3: $\pi_t^B, \pi_t^R \leftarrow \arg \max_{\pi^B \in \Pi^B} \min_{\pi^R \in \Pi^R} V_{w_t}(\pi^B, \pi^R)$
 - 4: **end for**
 - 5: **return** $\{(\pi_t^B, \pi_t^R)\}_{t=0}^{k-1}$
-

Algorithm 2 Adapted Double Oracle Algorithm

Require: $G_w, \pi_1^B, \pi_1^R, \epsilon$

- 1: Initialize $t \leftarrow 0$
 - 2: **while** $t = 0 \vee \bar{v}_t - \underline{v}_t > \epsilon$ **do**
 - 3: $t \leftarrow t + 1$
 - 4: $\mathbf{p}, \mathbf{q} \leftarrow \text{SolveNFG}(\{V_w^{\mathcal{G}}(\pi_i^B, \pi_j^R)\}_{1 \leq i, j \leq t})$
 - 5: $\sigma_t^B \leftarrow \text{Mix}(\mathbf{p}, \{\pi_i^B\}_{i=1}^t), \sigma_t^R \leftarrow \text{Mix}(\mathbf{q}, \{\pi_j^R\}_{j=1}^t)$
 - 6: $\pi_{t+1}^B, \bar{v}_t \leftarrow (\arg) \max_{\pi^B} V_w^{\mathcal{G}}(\pi^B, \sigma_t^R)$
 - 7: $\pi_{t+1}^R, \underline{v}_t \leftarrow (\arg) \min_{\pi^R} V_w^{\mathcal{G}}(\sigma_t^B, \pi^R)$
 - 8: **end while**
 - 9: **return** σ_t^B, π_{t+1}^R
-

Theorem 5.3. A policy π^B is in an AI-Control Equilibrium if, and only if, for some ZSPOSG G_w with value function V_{G_w} , π^B maximises $\min_{\pi^R \in \Pi^R} V_{G_w}(\cdot, \pi^R)$ over Π^B .

Algorithm 1 uses theorem 5.3 to iteratively fill out the frontier by solving a series of single-objective maximin problems. Here FindNextWeight first returns $w_0 = 0$ and $w_1 = 1$, and then chooses weights that maximise the difference between known points on the Pareto-frontier.¹⁶

Line 3 of algorithm 1 requires finding the value of $\max_{\pi^B} \min_{\pi^R} V_w^{\mathcal{G}}(\pi^B, \pi^R)$. Although ZSPOSGs are previously studied, existing solvers are incompatible with a memoryless Red team (see section 8). The memorylessness of Red makes their decision rules at each timestep interdependent, and problem decomposition difficult. Further, imperfect recall prohibits the equivalence of behavioural and mixed stochasticities:

Proposition 5.4. There exists a G_w such that Red’s behavioural and mixed policies would not be equivalent and $\max_{\pi^B \in \Pi^B} \min_{\pi^R \in \Pi^R} V_w^{\mathcal{G}}(\pi^B, \pi^R) < \min_{\pi^R \in \Pi^R} \max_{\pi^B \in \Pi^B} V_w^{\mathcal{G}}(\pi^B, \pi^R)$.

See appendix A.3.4 for a constructive proof. Despite this general inequivalence between Red’s mixed and behaviourally stochastic policies, finding a Nash equilibrium of mixed-stochastic strategies can be a means to finding an AI-Control equilibrium. Algorithm 2 adapts McMahan et al. [30]’s Double Oracle algorithm, used for finding mixed Nash-equilibria, to find AI-Control Equilibria. In this algorithm, (1) the function Mix simply creates a policy mixture from a vector and a finite set of policies and (2) finding $\mathbf{p}$ and $\mathbf{q}$ is equivalent to finding a Nash equilibrium in a zero-sum finite normal-form game. The values $\underline{v}_t$ and $\bar{v}_t$ lower and upper bound the maximin value of the game.

Proposition 5.5. For $\epsilon > 0$, Algorithm 2 returns an ϵ -equilibrium in G_w in finite time. Further, for all t : $\underline{v}_t \leq v^* := \max_{\pi^B \in \Pi^B} \min_{\pi^R \in \Pi^R} V_w^{\mathcal{G}}(\pi^B, \pi^R) \leq \bar{v}_t$.

If $\epsilon = 0$ then (σ_t^B, π_t^R) is an AI-Control Equilibrium. The proof is shown in appendix A and utilises Adam et al. [1, Theorem 1.1] by reducing G_w to a ‘continuous game’. The primary change to the algorithm is the choice to return π_{t+1}^R which is behaviourally but not mixed stochastic, and gives the worst-case safety for σ_t^B which is a lower-bound for v^* .

Best Response Oracles. Lines 6 and 7 of the algorithm require finding best-responses. Finding the best-response π^B to a (finite) policy mixture σ^R is equivalent to solving a Partially-observable

¹⁶Let $s_t := \text{Safety}(\pi_t^B, \pi_t^R)$ and $u_t := \text{Use}(\pi_t^B, \pi_t^R)$. Order points by safety, then choose $i = \max_i (s_i - s_{i+1})^2 + (u_i - u_{i+1})^2$ and then w such that $ws_i + (1-w)u_i = ws_{i+1} + (1-w)u_{i+1}$.

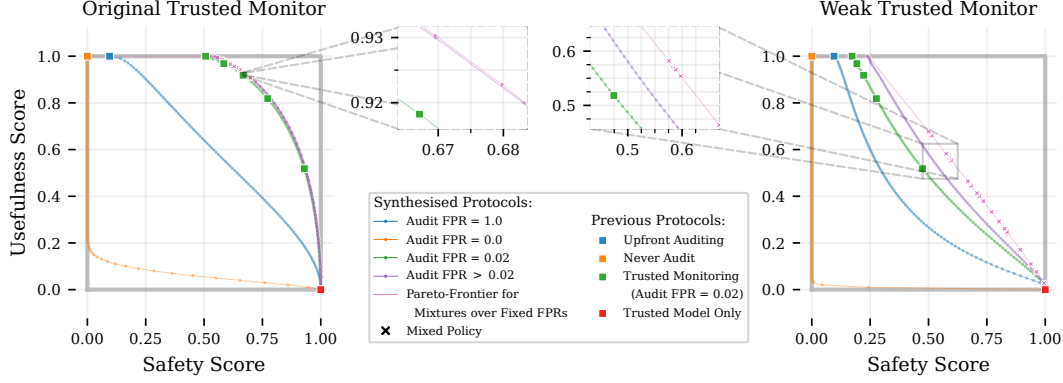


Figure 2: The safety and usefulness frontiers for a range of Trusted Monitor protocol types across the ‘original’ and ‘weak’ games. Each point on the grid corresponds to the exact safety and usefulness of a proposed blue team protocol against an approximately optimal red team response (minimal safety). The square points correspond to the seven special cases of Trusted Monitoring studied by Greenblatt et al. [18], whereas the remaining points are new to this paper. Each ‘Audit FPR’ line plots a single, fixed audit FPR, across fixed defer FPRs in $\{0.00, 0.01, \dots, 1.00\}$. ‘Pareto-Frontier for Fixed FPRs’ shows the optimal protocol frontier found by algorithm 1 with a fixed-FPR oracle, and all points represent deterministic policies unless marked by a cross.

Markov Decision Process, which is a well-studied problem [24, 37]. However, finding a best-response π^R to σ^B is, in general, more challenging: because Red is memoryless, the problem is difficult to decompose. Fortunately, in the Trusted Monitoring example described in section 2, Red’s policy space is 1-dimensional, and can be searched easily.

6 Results from Trusted Monitoring Example

In this section, we present example evaluations of protocols utilising our formal model. We also present the first Pareto-frontiers for a subset of the protocol space. All results below concern only memoryless protocols. These can be evaluated efficiently using proposition 4.1. We use the formal model from Section 4, which leverages *existing* empirical data from Greenblatt et al. [18] to explore a broader range of protocols and assumptions. We leave the analysis of memoryful protocols, in which actions can depend on the full history of past observations, to future work. See appendix D for more experimental details.

Modelling Existing Protocols. Protocols studied in Greenblatt et al. [18] can be expressed as memoryless protocols in our model. *Never Audit* always submits U’s solution without audit. *Trusted Model Only* always submits T’s solution. *Upfront Auditing* audits the first 2% of solutions from U and, when there is no audit budget left, submits the rest without checking. *Trusted Monitoring* (*Audit FPR = 0.02*) audits every solution from U whose suspicion score is above a threshold chosen to give a 2% FPR and also defer all solutions above (0%, 5%, 10%, 20%, 50%). We leave evaluation of more complex protocols to future work. Each protocol above is a special case of Trusted Monitoring with a fixed audit FPR of 0%, 2% or 100%, combined with fixed defer FPRs. By representing these protocols in our formalisation, we can easily evaluate their safety and usefulness scores, even in the novel ‘weak’ case (where T is worse at telling safe and backdoored solutions apart); we illustrate these results by large coloured points in fig. 2.

Evaluating New Protocols. Through a process of proposing and efficiently evaluating new protocols, which is newly enabled by our formalisation, we extend Greenblatt et al.’s findings to new protocols and settings. As an example, we show that raising the audit FPR above 2% Pareto-improves upon the protocols from Greenblatt et al. [18]. Choosing an audit FPR of 2% means that, in expectation, the 2% allowance of human labour will be used up at the end of the episode if no backdoor is submitted. However, we find that choosing a higher audit rate can be beneficial, despite increasing the probability that the audit budget is exhausted early. Choosing an audit FPR of 3% gives a small Pareto-improvement in the ‘original’ Trusted Monitoring case and an audit FPR of 5%, gives a large Pareto-improvement in the ‘weak’ Trusted Monitoring case.

Demonstrating a Frontier. Using Algorithms 1 and 2, we present the first *approximate* Pareto-frontier for a policy *subset*: mixtures of protocols with fixed audit and defer rates. We apply algorithm 2 using a Blue best-response algorithm based on recursive grid search (over values of audit and defer FPR) and a Red best-response algorithm using grid search followed by ternary search (over β). We find that, in the ‘original’ case the closest of Greenblatt et al.’s policies is 0.0042 from the Pareto-frontier, and for most weights a deterministic policy is optimal. However, in the ‘weak’ case, they are at least ≥ 0.0200 from the Pareto-frontier and mixed protocols make up most of the frontier.

6.1 Analysing Assumptions and Trends

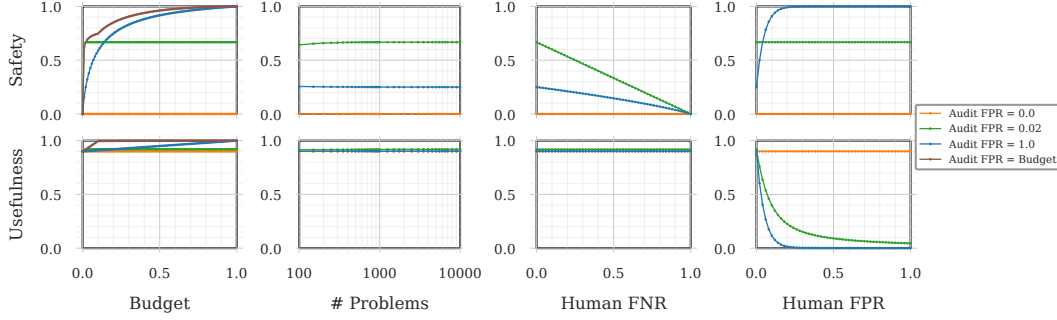


Figure 3: Impact on safety and usefulness (y-axis) by four different modelling assumptions (x-axis): Auditing Budget, Number of Programming Problems, and Human Auditor False Negative / Positive Rate. All plots show a fixed defer FPR of 0.1.

Formalising an AI-Control Game requires making a number of modelling assumptions: we explore how varying each assumption impacts the safety and usefulness scores for a given blue-team policy.

Audit Budget. The auditing budget corresponds to the amount of human labour invested in overseeing the protocol. Protocols that utilise increases to the audit budget get significantly safer and more useful as the budget increases, especially when the budget is small.

Number of Problems. For fixed FPRs, safety and usefulness are not significantly affected by the number of problems per episode. As the number of opportunities for Red to backdoor increases, so too does the number of audit coins.

Human Auditor FPR/FNR. In the default Trusted Monitoring game, Greenblatt et al. assume that no mistakes are made by the human auditor. We can extend the model to relax this assumption. Increasing the human FNR decreases safety almost linearly in the worst case, without impacting usefulness. False positives increase safety up to a point determined by the audit threshold, but decrease usefulness more rapidly the more the blue team audits.

7 Formalising Concepts Underlying Trusted Monitoring

Greenblatt et al. [18] study *intentional backdoors*, whereby backdoored solutions are only ever submitted by scheming models. As a consequence, catching a single attempted backdoor is sufficient to infer that the model is scheming. However, what if the backdoor is the result of a benign bug written by an honest if imperfect model? Our formalism allows us to investigate.

Let $\mathcal{A}_D^R \subset \mathcal{A}^R$ be a set of *dangerous actions*: a minimal subset of actions without which an unacceptable outcome could never occur; e.g. in the Trusted Monitoring game, $\mathcal{A}_D^R = \{\langle \text{Backdoor} \rangle\}$. Greenblatt et al. [18] assume that an honest model never takes a dangerous action, which we can state explicitly within our model as a constraint on $\pi_\oplus$: namely $\forall a_D^R, \forall o^R, \pi_\oplus(a_D^R | o^R) = 0$.

Why does the honest/scheming distinction matter if an honest model can take dangerous actions? In the Trusted Monitoring game, an unacceptable outcome can be achieved via a single dangerous action. However, in realistic settings, multiple dangerous actions may be required to cause harm. Adversaries, such as schemers, may be more likely to take sequences of dangerous actions than honest models: their dangerous actions may be correlated. For example, suppose that three layers

of security prevent the harm of exfiltrating model weights. An honest model, asked to improve and maintain security measures, might accidentally introduce a vulnerability into one layer. But causing an unacceptable outcome requires at least four dangerous actions: introducing vulnerabilities into three layers and then exploiting them. Four dangerous actions in series are far less probable than one if a model is making honest mistakes, but not if the model is scheming. This example shows how the scheming versus honest assumption can still be useful when we relax the assumption that honest models never take dangerous actions.

8 Related Work

Though the application of formal game-theoretic tools to Greenblatt et al. [18]’s AI Control framework is novel, and the resulting type of game is unique, there are analogues to existing work.

The AI-Control Games studied are most similar to the work of Chang et al. [12], who model a security problem for food manufacturing. Like AI-Control Games, they consider a leader-follower equilibrium, where the leader tries to maximise productivity and minimise risk, and the follower tries to hide an attack. To select the leader strategy against multiple objectives, they use a genetic algorithm (as do Lins et al. [26]). Their model differs in its factored state space, infinite time horizon, and the lack of ‘attack/honest’ game modes and most importantly their consideration of a memoryful-follower. Their solution method is not directly applicable to AI-Control Games, and may be unsuitable for adaptation. For a detailed comparison, see appendix E.

Given our reduction in section 5, methods for finding equilibria in ZSPOSGs are particularly relevant. However, we find that our requirement for a memoryful-leader, memoryless-follower equilibrium precludes the use of many existing solvers. For example, Delage et al. [15], which finds Nash equilibria in ZSPOSGs using a dynamic programming approach that solves subgames for each timestep from H down to 1. Other works attempt to solve subclasses of ZSPOSGs, with ‘public observations’ [21] or ‘one-sided partial-observability’ [42, 22]. AI-Control Games do not in-general fit into these subclasses, and each of these papers considers two memoryful players.

Since in some cases it is possible to find an AI-Control Equilibrium by first finding a mixed Nash equilibrium (theorem 5.5), it may be possible to augment the algorithm by Hansen & Feng [19], which searches for two mixed strategies over a finite set of memoryful policies.

AI-Control Games, and especially the Trusted Monitoring case study we present, are closely related to Stackelberg Security Games: two-player games where the leader must choose a random strategy to cover a set of targets using a limited set of resources, and the follower tries to attack an uncovered target (see [38] for an overview). Unlike Trusted Monitoring, these are one-round games and much literature focuses on scalably solving large-scale problems against scheduling constraints. Although some security games model imperfect information [8, 32] or multiple objectives [9], most are single-objective and perfect-information.

Trusted Monitoring without defer is similar to the Inspection Games studied by Avenhaus et al. [5, p. 1957]: games where the defender receives a noisy observation, and has to decide whether to inspect. However, these games are one-round only. Recursive Inspection Games [16] are multi-round, but have a memoryful follower.

9 Future Work and Conclusions

In this work, we introduced a novel model for analysing the safety and usefulness of AI-Control protocols. We used our formal model and solution techniques to evaluate a much broader range of protocols and settings than prior empirical results from literature, and there is significant room to apply them further. In the showcased application of Trusted Monitoring setting, future work can utilise algorithm 2 with POMDP-based best-response oracles to explore the space of memory-dependent protocols. Our model of the Trusted Monitoring protocol is also generalisable to *any* deployment setting, using a question-answer format and *any* unacceptable outcome. Future work will explore such generalised settings by studying the interaction of the untrusted AI and the trusted monitor to generate distributions like those in fig. 1 and analyse the resulting frontiers.

References

- [1] Lukáš Adam, Rostislav Horčík, Tomáš Kasl, and Tomáš Kroupa. Double Oracle Algorithm for Computing Equilibria in Continuous Games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(6):5070–5077, May 2021. doi: 10.1609/aaai.v35i6.16641. URL <https://ojs.aaai.org/index.php/AAAI/article/view/16641>. Section: AAAI Technical Track on Game Theory and Economic Paradigms.
- [2] Adept AI. Adept: Useful General Intelligence, 2024. URL <https://www.adept.ai/>.
- [3] Anthropic. Controlling powerful AI, March 2025. URL <https://www.youtube.com/watch?v=6Unxqr50Kqg>.
- [4] Robert J Aumann, Sergiu Hart, and Motty Perry. The absent-minded driver. *Games and Economic Behavior*, 20(1):102–116, 1997. Publisher: Elsevier.
- [5] Rudolf Avenhaus, Bernhard Von Stengel, and Shmuel Zamir. Chapter 51 Inspection games. In *Handbook of Game Theory with Economic Applications*, volume 3, pp. 1947–1987. Elsevier, January 2002. doi: 10.1016/S1574-0005(02)03014-X. URL <https://www.sciencedirect.com/science/article/pii/S157400050203014X>.
- [6] Mikita Balesni, Marius Hobbhahn, David Lindner, Alexander Meinke, Tomek Korbak, Joshua Clymer, Buck Shlegeris, Jérémy Scheurer, Charlotte Stix, Rusheb Shah, Nicholas Goldowsky-Dill, Dan Braun, Bilal Chughtai, Owain Evans, Daniel Kokotajlo, and Lucius Bushnaq. Towards evaluations-based safety cases for AI scheming, November 2024. URL <http://arxiv.org/abs/2411.03336>. arXiv:2411.03336 [cs].
- [7] Aryan Bhatt, Cody Rushing, Adam Kaufman, Tyler Tracy, Vasil Georgiev, Akbir Khan, David Matolcsi, and Buck Shlegeris. Ctrl-Z: Controlling AI Agents via Resampling (forthcoming). pp. Forthcoming as of, 2025.
- [8] Vicki M. Bier. Choosing What to Protect. *Risk Analysis*, 27(3):607–620, 2007. ISSN 1539-6924. doi: 10.1111/j.1539-6924.2007.00906.x.
- [9] Matthew Brown, Bo An, Christopher Kiekintveld, Fernando Ordóñez, and Milind Tambe. Multi-objective optimization for security games. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS ’12, pp. 863–870, June 2012. ISBN 978-0-9817381-2-3.
- [10] Marie Davidsen Buhl, Gaurav Sett, Leonie Koessler, Jonas Schuett, and Markus Anderljung. Safety cases for frontier AI, October 2024. URL <http://arxiv.org/abs/2410.21572>. arXiv:2410.21572 [cs].
- [11] Stephen Casper, Lennart Schulze, Oam Patel, and Dylan Hadfield-Menell. Defending Against Unforeseen Failure Modes with Latent Adversarial Training, April 2024. URL <http://arxiv.org/abs/2403.05030>. arXiv:2403.05030 [cs].
- [12] Yanling Chang, Alan L. Erera, and Chelsea C. White. A leader–follower partially observed, multiobjective Markov game. *Annals of Operations Research*, 235(1):103–128, December 2015. ISSN 1572-9338. doi: 10.1007/s10479-015-1935-0. URL <https://doi.org/10.1007/s10479-015-1935-0>.
- [13] Joshua Clymer, Nick Gabrieli, David Krueger, and Thomas Larsen. Safety Cases: How to Justify the Safety of Advanced AI Systems, March 2024. URL <http://arxiv.org/abs/2403.10462>. arXiv:2403.10462 [cs].
- [14] DeepMind. Frontier safety framework. Technical report, September 2025. URL https://storage.googleapis.com/deepmind-media/DeepMind.com/Blog/strengthening-our-frontier-safety-framework/frontier-safety-framework_3.pdf. tex.version: 3.0.
- [15] Aurélien Delage, Olivier Buffet, Jilles S. Dibangoye, and Abdallah Saffidine. HSVI Can Solve Zero-Sum Partially Observable Stochastic Games. *Dynamic Games and Applications*, September 2023. ISSN 2153-0785, 2153-0793. doi: 10.1007/s13235-023-00519-6. URL <https://link.springer.com/10.1007/s13235-023-00519-6>.

- [16] Melvin Dresher. A Sampling Inspection Problem in Arms Control Agreements: A Game-Theoretic Analysis. Technical report, RAND Corporation, January 1962. URL <https://www.rand.org/pubs/research{ }memoranda/RM2972.html>.
- [17] R. Emery-Montemerlo, G. Gordon, J. Schneider, and S. Thrun. Approximate solutions for partially observable stochastic games with common payoffs. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004.*, pp. 136–143, July 2004. URL <https://ieeexplore.ieee.org/abstract/document/1373472>.
- [18] Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. AI Control: Improving Safety Despite Intentional Subversion. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 16295–16336. PMLR, July 2024. URL <https://proceedings.mlr.press/v235/greenblatt24a.html>. ISSN: 2640-3498.
- [19] Eric A. Hansen and Zhengzhu Feng. Dynamic programming for POMDPs using a factored state representation. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems, AIPS’00*, pp. 130–139, Breckenridge, CO, USA, April 2000. AAAI Press. ISBN 978-1-57735-111-5.
- [20] Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. Measuring Coding Challenge Competence With APPS. August 2021. URL <https://openreview.net/forum?id=sD93G0zH3i5>.
- [21] Karel Horák and Branislav Božanský. Solving Partially Observable Stochastic Games with Public Observations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 2029–2036, July 2019. ISSN 2374-3468. doi: 10.1609/aaai.v33i01.33012029. URL <https://ojs.aaai.org/index.php/AAAI/article/view/4032>. Number: 01.
- [22] Karel Horák, Branislav Božanský, Vojtěch Kovařík, and Christopher Kiekintveld. Solving zero-sum one-sided partially observable stochastic games. *Artificial Intelligence*, 316:103838, March 2023. ISSN 0004-3702. doi: 10.1016/j.artint.2022.103838. URL <https://www.sciencedirect.com/science/article/pii/S0004370222001783>.
- [23] Geoffrey Irving. Safety cases at AISI, August 2024. URL <https://www.aisi.gov.uk/work/safety-cases-at-aisi>.
- [24] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998. ISSN 0004-3702. doi: 10.1016/s0004-3702(98)00023-x.
- [25] Harold W. Kuhn. Extensive games and the problem of information. *Contributions to the Theory of Games*, 2(28):193–216, 1953.
- [26] Isis Didier Lins, Leandro Chaves Rêgo, Márcio das Chagas Moura, and Enrique López Droguett. Selection of security system design via games of imperfect information and multi-objective genetic algorithm. *Reliability Engineering & System Safety*, 112:59–66, April 2013. ISSN 0951-8320. doi: 10.1016/j.ress.2012.11.021. URL <https://www.sciencedirect.com/science/article/pii/S0951832012002517>.
- [27] Monte MacDiarmid, Timothy Maxwell, Nicholas Schiefer, Jesse Mu, Jared Kaplan, David Duvenaud, Sam Bowman, Alex Tamkin, Ethan Perez, Mrinank Sharma, Carson Denison, and Evan Hubinger. Simple probes can catch sleeper agents, April 2024. URL <https://www.anthropic.com/news/probes-catch-sleeper-agents>.
- [28] Alex Mallen, Charlie Griffin, Alessandro Abate, and Buck Shlegeris. Subversion Strategy Eval: Evaluating AI’s stateless strategic capabilities against control protocols, 2024. URL <https://arxiv.org/abs/2412.12480>.
- [29] Michael Maschler, Eilon Solan, and Shmuel Zamir. Behavior strategies and kuhn’s theorem. In *Game theory*, pp. 219–250. Cambridge University Press, Cambridge, 2013.

- [30] H. Brendan McMahan, Geoffrey J. Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 536–543, 2003. URL <https://cdn.aaai.org/ICML/2003/ICML03-071.pdf>.
- [31] Kaisa Miettinen. *Nonlinear Multiobjective Optimization*. Springer, New York, NY, UNITED STATES, 1998. ISBN 978-1-4615-5563-6. URL <http://ebookcentral.proquest.com/lib/oxford/detail.action?docID=3081836>.
- [32] Thanh H. Nguyen, Amulya Yadav, Bo An, Milind Tambe, and Craig Boutilier. Regret-based optimization and preference elicitation for stackelberg security games with uncertainty: 28th AAAI Conference on Artificial Intelligence, AAAI 2014, 26th Innovative Applications of Artificial Intelligence Conference, IAAI 2014 and the 5th Symposium on Educational Advances in Artificial Intelligence, EAAI 2014. *Proceedings of the 28th AAAI Conference on Artificial Intelligence and the 26th Innovative Applications of Artificial Intelligence Conference and the 5th Symposium on Educational Advances in Artificial Intelligence*, pp. 756–762, 2014. URL <http://www.scopus.com/inward/record.url?scp=84908202610&partnerID=8YFLogxK>. Publisher: AI Access Foundation.
- [33] OpenAI. ChatGPT plugins, March 2023. URL <https://openai.com/blog/chatgpt-plugins>.
- [34] OpenAI and Apollo Research. Detecting and reducing scheming in AI models, September 2025. URL <https://openai.com/index/detecting-and-reducing-scheming-in-ai-models/>.
- [35] Martin J. Osborne and Ariel Rubinstein. *A course in game theory*. MIT Press, Cambridge, Mass., 12. print edition, 2006. ISBN 978-0-262-65040-3.
- [36] Rusheb Shah, Quentin Feuille Montixi, Soroush Pour, Arush Tagade, and Javier Rando. Scalable and Transferable Black-Box Jailbreaks for Language Models via Persona Modulation. November 2023. URL <https://openreview.net/forum?id=x3Ltqz1UFg>.
- [37] Guy Shani, Joelle Pineau, and Robert Kaplow. A survey of point-based POMDP solvers. *Autonomous Agents and Multi-Agent Systems*, 27(1):1–51, July 2013. ISSN 1573-7454. doi: 10.1007/s10458-012-9200-2. URL <https://doi.org/10.1007/s10458-012-9200-2>.
- [38] Arunesh Sinha, Fei Fang, Bo An, Christopher Kiekintveld, and Milind Tambe. Stackelberg Security Games: Looking Beyond a Decade of Success. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 5494–5501, Stockholm, Sweden, July 2018. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-2-7. doi: 10.24963/ijcai.2018/775. URL <https://www.ijcai.org/proceedings/2018/775>.
- [39] The White House. America’s AI action plan. Technical report, July 2025. URL <https://www.whitehouse.gov/wp-content/uploads/2025/07/Americas-AI-Action-Plan.pdf>.
- [40] UK AI Safety Institute. Research agenda, 2025. URL <https://www.aisi.gov.uk/research-agenda>.
- [41] Jiaxin Wen, Vivek Hebbar, Caleb Larson, Aryan Bhatt, Ansh Radhakrishnan, Mrinank Sharma, Henry Sleight, Shi Feng, He He, Ethan Perez, Buck Shlegeris, and Akbir Khan. Adaptive Deployment of Untrusted LLMs Reduces Distributed Threats, November 2024. URL <http://arxiv.org/abs/2411.17693>. arXiv:2411.17693.
- [42] Wei Zheng, Taeho Jung, and Hai Lin. The Stackelberg equilibrium for one-sided zero-sum partially observable stochastic games. *Automatica*, 140:110231, June 2022. ISSN 0005-1098. doi: 10.1016/j.automatica.2022.110231. URL <https://www.sciencedirect.com/science/article/pii/S0005109822000760>.