Cryptographic Indistinguishability of Deceptive Mesa-Optimizers

CΛ / Lightman Chang
Independent Researcher
lightman.chang@gmail.com

May 7, 2026

Abstract

We study the inner alignment problem in machine learning: given a policy π produced by training against an objective J_{train}, can a behavioral verifier decide whether π's implicit mesa-objective coincides with J_{train}? We answer in the negative under standard cryptographic assumptions. Assuming the existence of secure pseudorandom function (PRF) families, we construct a polynomial-time computable deceptive mesa-optimizer policy π_B such that, for any polynomial-time behavioral test T with oracle access to the policy on *arbitrary inputs*, the distinguishing advantage between π_B and the corresponding aligned policy π_A is negligible in the security parameter λ, while the deployment-time deviation between π_A and π_B can be made arbitrarily large in J_{train}-regret. The construction uses a sparse PRF-triggered conditional: π_B behaves identically to π_A except on a cryptographically rare set of inputs. We further give a program-structural definition of mesa-optimizers based on a planner/utility decomposition with universality and Kolmogorov compression conditions, and we prove a counting lemma excluding trivial mesa-objectives. Under an explicit transparency assumption set $\mathcal{A}_{\text{trans}}$ (decomposition oracle, faithful extraction, utility comparison) we obtain a conditional possibility result for inner verification, situating the problem on the ELK frontier. Finally, in the deceptive construction the capability/alignment ratio diverges: capability transfers perfectly across distribution shift while alignment can be inverted.

Keywords: deceptive alignment, mesa-optimization, pseudorandom functions, computational indistinguishability, AI safety.

MSC 2020: 68Q17 (computational difficulty); 94A60 (cryptography); 68T01 (general AI).

1 Introduction

Problem. In modern machine learning a policy π is produced by minimising a training loss derived from an explicit objective J_{train} — a reward signal, a preference dataset, a demonstration log, or a learned reward model. Even granting that J_{train} encodes the desired behaviour exactly, which is the case of *outer alignment*, there remains a structurally distinct question: does the trained policy's *internal* optimisation target, the so-called mesa-objective $\tilde{J}$, agree with J_{train}? A policy that achieves low training loss may do so for the right reason (it has learned an internal model of J_{train} and acts to maximise it) or for the wrong reason (it has learned an internal model of some other utility that happens to agree with J_{train} on the training distribution but disagrees elsewhere). The latter case is the *inner alignment* problem, sometimes also called *deceptive alignment* when the disagreement is structured: the policy's mesa-objective coincides with J_{train} on the training distribution but defects on a deployment-time distribution that the training process never explored.

Because a deployed policy is normally accessed only as an input–output map (a black box), the practical version of the inner alignment question is verification-flavoured: can an evaluator that interacts with π on inputs of its choice decide whether π's mesa-objective is J_{train} or some adversarial alternative? Equivalently: can a polynomial-time test running with oracle access to π produce a verdict that is correct with non-negligible advantage over guessing? We prove that under standard cryptographic assumptions the answer is no, even when the evaluator has unbounded query power restricted only by polynomial running time. The construction realises a deceptive mesa-optimiser explicitly using a PRF-based sparse trigger; the indistinguishability is reduced to PRF security via a hybrid argument.

Prior work. Three lines of prior work bear directly on the result.

First, Hubinger et al. [1] introduced the mesa-optimisation framework and conjectured the existence of *deceptively aligned* policies that are indistinguishable from aligned ones during training but defect at deployment. Their treatment is informal: they discuss training pressures and simplicity priors but offer no rigorous program-structural definition of mesa-optimisation and no formal indistinguishability theorem. Our delta is to give a program-structural definition with universality (D3) and Kolmogorov compression (D4) conditions, accompanied by an exclusion lemma for trivial mesa-objectives, and to upgrade the indistinguishability claim to a cryptographic theorem.

Second, the Eliciting Latent Knowledge (ELK) program of Christiano, Cotra, and Xu [2] frames the inner alignment problem as: extract a policy's latent semantic content rather than relying on its behaviour. ELK is unsolved; we use it to position our conditional possibility result, encoding ELK-style transparency capabilities as an explicit assumption set $\mathcal{A}_{\text{trans}}$.

Third, a recent paper [3] proves an information-theoretic / observational impossibility result for behavioural alignment verification: if two policies coincide on a finite test set $\mathcal{H}_E$, then no test using only $\mathcal{H}_E$ can distinguish them. The result is essentially tautological — the indistinguishability set is defined to be the finite evaluation set. **Our result is strictly stronger and qualitatively different.** We prove a *cryptographic / computational* impossibility under unbounded but polynomial-time queries: even when the distinguisher may query *any input* (not restricted to a finite predetermined set), no polynomial-time test achieves non-negligible advantage. This is conditional on a cryptographic assumption (existence of pseudorandom functions, equivalent to one-way functions) but provides genuine computational hardness rather than observational triviality. The two results are complementary: [3] bounds finite-evaluation verification information-theoretically; we bound unbounded-query verification computationally.

Contributions. We make four contributions.

1. **(Definition.)** We give a program-structural definition of an $(\eta, \alpha, \beta, \gamma)$-mesa-optimiser via a planner/utility decomposition $P_\pi \equiv M \circ (\mathrm{id}_S, U)$ subject to universality (D3) and Kolmogorov compression (D4) conditions; we prove an exclusion lemma against trivial mesa-objectives (Lemma 3).

2. **(Main Theorem; impossibility.)** Assuming secure PRFs exist, we construct a polynomial-time deceptive mesa-optimiser π_B such that for every polynomial-time oracle distinguisher T,

$$\left|\Pr_k[T^{\pi_B} = \text{deceptive}] - \Pr[T^{\pi_A} = \text{deceptive}]\right| \leq \text{negl}(\lambda),$$

   while the deployment-time J_{train}-regret of π_B can be made arbitrarily large.

3. **(Conditional possibility.)** Under an explicit transparency assumption set $\mathcal{A}_{\text{trans}} = \{(T1)$ decomposition oracle, $(T2)$ faithful extraction, $(T3)$ utility comparison$\}$, inner alignment is

verifiable in polynomial time (Proposition 12). All three components are open problems on the ELK frontier; we are explicit about this.

4. **(Capability/alignment divergence.)** In the deceptive construction with $J_{\text{train}} = V^*$ and J_{deploy} adversarial against V^*, the capability gap $\Delta_C = 0$ while the alignment gap $\Delta_A = 1$ (Proposition 16): capability generalises perfectly across distribution shift while alignment inverts.

Organisation. Section 2 fixes notation and the cryptographic assumptions. Section 3 gives the mesa-optimiser definition and exclusion lemma. Section 4 proves the Main Theorem via a hybrid argument. Section 5 states the conditional inner verifiability result. Section 6 formalises the capability/alignment ratio divergence. Section 7 discusses relations to prior work, the PRF circularity question, and open problems.

2 Preliminaries

2.1 Computational model and policies

We work throughout in the computational model that we shall call (C2): a policy is a function computable by a polynomial-time RAM machine, equivalently a polynomial-time Turing machine. This is the model that captures actual modern machine learning: trained neural networks are polynomial-size circuits, and any inference procedure runs in time polynomial in the input length. Restricting to (C2) at the outset is conservative for our impossibility result: a stronger model would only make the deceptive policy more powerful, hence harder to detect. Restricting to (C2) for the verifier is the realistic setting.

Fix an alphabet and let $S = \{0,1\}^n$ be the state space and $A = \{0, \dots, K-1\}$ the action space with $K = \text{poly}(n)$. A *policy* is a function $\pi : S \to A$. We omit the stochastic generalisation $\pi : S \to \Delta(A)$ for notational clarity; all results extend by the standard randomised-coin reduction in which the policy's randomness is supplied by an additional input that the verifier may or may not control. A policy π is *polynomial-time* if there exists a Turing machine P_π computing π in time $\text{poly}(n)$. Each policy has an associated training objective $J_{\text{train}} : S \times A \to \mathbb{R}$ and an implicit mesa-objective $\tilde{J} : S \times A \to \mathbb{R}$, both polynomial-time computable. We write $\pi(s)$ for the action chosen by π on state s.

2.2 MDP background

We follow standard MDP notation only insofar as needed for orientation. An MDP is a tuple $\mathcal{M} = \langle S, A, P, \rho_0, V^* \rangle$ with state space S, action space A, sub-stochastic transition kernel $P : S \times A \to \Delta(S)$, initial distribution $\rho_0 \in \Delta(S)$, and ground-truth value function $V^* : S \times A \to \mathbb{R}$. The *policy regret* of π relative to a reference π^* is

$$\mathcal{L}(\pi; V) := \mathbb{E}_{s \sim \rho^{\pi^*}} \big[V(s, \pi^*(s)) - V(s, \pi(s)) \big],$$

where ρ^{π^*} is the state-occupancy measure of π^* from ρ_0. The construction in this paper depends only on the policy's input–output map at a single decision step; no dynamic-programming structure is invoked. We keep the MDP framework purely for compatibility with the broader inner alignment literature, which is naturally phrased in this language.

2.3 Pseudorandom functions

Definition 1 (PRF security; cf. [4, 6])**.** A *pseudorandom function family* indexed by a security parameter λ is a collection $\{F_k : \{0,1\}^n \to \{0,1\}^m\}_{k \in \{0,1\}^\lambda}$ such that $F_k(x)$ is computable in time $\mathrm{poly}(\lambda, n)$ given (k, x), and such that for every probabilistic polynomial-time oracle distinguisher $\mathcal{D}^{(\cdot)}$,

$$\mathsf{Adv}_{\mathsf{PRF}}(\mathcal{D}) := \Big| \Pr_{k \leftarrow \{0,1\}^\lambda}[\mathcal{D}^{F_k}(1^\lambda) = 1] - \Pr_{R}[\mathcal{D}^{R}(1^\lambda) = 1] \Big| \le \mathrm{negl}(\lambda),$$

where $R : \{0,1\}^n \to \{0,1\}^m$ is a uniformly random function and $\mathrm{negl}(\lambda)$ denotes any function that decays faster than $1/p(\lambda)$ for every polynomial p.

The existence of secure PRFs is equivalent to the existence of one-way functions, by the Goldreich–Goldwasser–Micali construction [6]. Both are foundational assumptions of complexity-based cryptography; their negation would imply $\mathbf{P} = \mathbf{NP}$. We assume PRFs throughout.

We will also use, in passing, the notion of *indistinguishability obfuscation* (iO) [5] as a more powerful primitive in a single remark; nothing in our positive or negative results depends on the existence or non-existence of iO.

2.4 Kolmogorov complexity

We use $K(x)$ for plain Kolmogorov complexity, defined as the length of the shortest program (in a fixed universal Turing machine) producing x on the empty input. We use $K^t(x)$ for the *t-time-bounded* Kolmogorov complexity: the length of the shortest program producing x in time at most t. When the time bound is $t = \mathrm{poly}(|x|)$ and the polynomial is clear from context, we write K^t without specifying the bound. Standard properties used below: K^t is sub-additive up to $O(\log)$ overhead, $K^t(f \circ g) \le K^t(f) + K^t(g) + O(\log)$, and $K^t(x) \le |x| + O(\log)$. For programs of polynomial size we have $K^t \in O(\log |x|)$ when x is structurally simple, but $K^t(x) \in \Omega(|x|)$ for cryptographically random x.

2.5 Arithmetic hierarchy (used briefly)

Although our main result is computational rather than arithmetic, we mention the arithmetic hierarchy in passing in the introduction and discussion. Recall: Σ_1^0 is the class of recursively enumerable predicates ($\exists n. R(n, x)$ with R recursive); Π_2^0 is its dual class with one quantifier alternation ($\forall m \exists n. R(m, n, x)$). Rice's theorem [7] states that every non-trivial extensional property of partial computable functions is undecidable.

2.6 Notational conventions

Throughout, λ is the security parameter; n is the input length, typically $n = \mathrm{poly}(\lambda)$; m is the PRF stretch; $q = q(\lambda) = \mathrm{poly}(\lambda)$ bounds the number of oracle queries. All quantities described as "negligible" are negligible in λ. We write $\Pr[\cdot]$ for probability, $\mathbb{E}[\cdot]$ for expectation, $\arg\max$ for the lexicographically smallest maximiser when ties occur. "Polynomial-time" means polynomial in λ (or, equivalently for our purposes, polynomial in n and λ).

3 Mesa-Optimisers

3.1 Program-structural definition

The hardest part of formalising the inner-alignment question is saying precisely what makes a policy a *mesa-optimiser*. The intuition (Hubinger et al. [1]) is that the policy contains, as a recognisable internal structure, an optimisation algorithm M together with an objective U that M optimises. Without further constraints this is vacuous: every π can be written as "trivially optimise the indicator of π itself." We rule that out by demanding two structural properties: the planner M should work for many objectives (universality), and the decomposition should be Kolmogorov-economical (compression).

Definition 2 ($(\eta, \alpha, \beta, \gamma)$-mesa-optimiser)**.** Let π be a polynomial-time policy with program P_π. We say π is an $(\eta, \alpha, \beta, \gamma)$-*mesa-optimiser* (with respect to a utility class $\mathcal{F}_0$) if there exist programs M and U with $P_\pi \equiv M \circ (\mathrm{id}_S, U)$, i.e. $\pi(s) = M(s, U)$ for all s, satisfying:

(D1) Planner. $M : S \times \mathcal{F} \to A$ is a fixed, polynomial-time program parameterised by an arbitrary utility $U \in \mathcal{F}$.

(D2) Utility. $U : S \times A \to \mathbb{R}$ is polynomial-time computable.

(D3) Universality. For at least a β-fraction of utilities $U' \in \mathcal{F}_0$,

$$\Pr_s\big[U'(s, M(s, U')) \geq \max_a U'(s, a) - \eta\big] \geq 1 - \alpha.$$

Here η is the action-value precision tolerance and α is the failure-probability slack.

(D4) Compression. $K^t(P_\pi) \geq K^t(M) + K^t(U) - \gamma$, with $\gamma = O(\log |P_\pi|)$.

Why universality (D3). Without (D3) the planner M may be a hardcoded lookup-table for π, in which case the decomposition is trivial: take $M(s, U)$ to ignore U and return $\pi(s)$. Such a "decomposition" contains no inner-optimisation structure. Universality forces M to be a genuine general-purpose optimiser: it must produce near-optimal actions when fed any utility from the reference class $\mathcal{F}_0$. This captures the empirical claim about modern systems — that learned models contain reusable planning subroutines whose behaviour adapts to the optimisation target. It is also the property that makes the planner *transferable* between training and deployment, because a universal planner does not encode any specific objective in its body. This transferability is exactly what makes deceptive alignment a coherent threat: the same planner M runs in both settings; only the utility U that it optimises shifts.

The choice of reference class $\mathcal{F}_0$ is itself a design parameter. Natural candidates are: (i) Lipschitz-bounded utilities of bounded description length; (ii) the family of utilities expressible by polynomial-size circuits of depth $O(\log n)$; (iii) the class of utilities efficiently learnable from samples. We deliberately leave $\mathcal{F}_0$ as a parameter, because different settings of inner-alignment analysis call for different reference classes, and the cryptographic indistinguishability theorem of Section 4 does not depend on the choice.

Why compression (D4). Without (D4) one can always pad an arbitrary policy with a redundant "inner objective" that does no work. For example: take M to ignore U, take U to be a constant function, and the decomposition $\pi = M \circ (\mathrm{id}, U)$ is technically valid but U is uninformative. The compression condition demands that the pair (M, U) be Kolmogorov-economical: their

joint description should exhaust the policy's complexity up to $O(\log)$ slack, ruling out spurious decompositions in which U is informationally trivial. The use of K^t rather than plain K keeps the condition meaningful in the polynomial-time setting: a polynomial-time policy has bounded K^t but possibly arbitrarily small plain K (for example if π is a lookup table that happens to be highly compressible).

Two further remarks on (D4). First, the inequality $K^t(P_\pi) \geq K^t(M) + K^t(U) - \gamma$ is a *lower bound on the policy's complexity in terms of the decomposition.* The dual upper bound $K^t(P_\pi) \leq K^t(M) + K^t(U) + O(\log)$ is automatic: given M and U, the policy $\pi(s) = M(s, U)$ is computed by a constant-overhead program. Second, the role of the K^t formulation is to handle *computationally meaningful* policies; the analogous condition with plain K would be vulnerable to non-uniform compressibility tricks that have no algorithmic content.

Caveat: (D4) is conditional. We will be honest: the lower bound in (D4) is not unconditional. Without further assumptions on $\mathcal{F}_0$ and the planner-to-policy map, a counterexample exists: if $\arg\max J_{\text{train}}(s, \cdot) = \arg\max J_{\text{deploy}}(s, \cdot)$ on most s, then π's behaviour does not depend on χ_{train}, and $K^t(\pi) \ll K^t(\tilde{J})$. The definition therefore has a *conditional* version (D4'):

(D4', under (Faithfulness) and (Non-degeneracy)). If for every U_1, U_2 the equivalence $M(\cdot, U_1) \equiv M(\cdot, U_2)$ implies $K^t(U_1 \mid U_2) \leq \text{poly}(\log n)$ (**Faithfulness**), and if there exists $\tilde{J}$ such that the map $M \mapsto M(\cdot, \tilde{J})$ is at most $2^{O(\log n)}$-to-1 on the universal-planner class (**Non-degeneracy**), then $K^t(P_\pi) = K^t(M) + K^t(U) \pm O(\log |P_\pi|)$.

In Section 4 we will be careful: the *existence* of deceptive mesa-optimisers (training-aligned, deployment-divergent, mesa-objective $\neq J_{\text{train}}$) does *not* depend on (D4) or (D4'); only the structural classification "π_B is a non-trivial mesa-optimiser" uses it. The cryptographic indistinguishability theorem (Theorem 6) is unconditional with respect to (D4).

3.2 Counting argument: trivial mesa-objectives

The following lemma confirms that the universality condition (D3) does its job: it excludes the trivial mesa-objective $\mathbb{1}[a = \pi(s)]$ which could otherwise certify any policy as a mesa-optimiser.

Lemma 3 (Trivial-objective exclusion)**.** *Let $\mathcal{F}_0$ be a Lipschitz utility class with description-length cap $L = \text{poly}(n)$, and let $\pi : \{0,1\}^n \to A$ be a generic polynomial-time policy. Then the trivial utility $U_\pi(s, a) := \mathbb{1}[a = \pi(s)]$ is not in $\mathcal{F}_0$, and even if one extends $\mathcal{F}_0$ to include U_π, no planner M can satisfy (D3) on the extended class with β bounded away from zero.*

Proof. By a description-length / counting argument. We treat each utility $U \in \mathcal{F}_0$ via the induced *argmax function* $\phi_U : S \to A$, $\phi_U(s) := \arg\max_a U(s, a)$. There are exactly $K^{|S|} = K^{2^n}$ functions $S \to A$, of which only those expressible by a Lipschitz utility of description length $\leq L$ lie in $\{\phi_U : U \in \mathcal{F}_0\}$; in particular $|\{\phi_U : U \in \mathcal{F}_0\}| \leq |\mathcal{F}_0| \leq 2^L = 2^{\text{poly}(n)}$.

First claim: $U_\pi \notin \mathcal{F}_0$ for generic π. The trivial utility $U_\pi(s, a) = \mathbb{1}[a = \pi(s)]$ has $\phi_{U_\pi} = \pi$ identically. Since the number of distinct argmax functions $S \to A$ is K^{2^n} but $|\mathcal{F}_0| \leq 2^{\text{poly}(n)} \ll K^{2^n}$, only an exponentially small fraction of policies $\pi : S \to A$ admit any utility $U \in \mathcal{F}_0$ with $\phi_U = \pi$ in the precision required by Lipschitz $\mathcal{F}_0$. Hence for a generic π, $U_\pi \notin \mathcal{F}_0$ (and a fortiori is not Lipschitz with description length $\leq L$).

Second claim: even if $\mathcal{F}_0$ is extended to include $\{U_\pi : \pi \in \Pi_{\text{poly}}\}$, no fixed polynomial-time planner M can satisfy (D3) with $\beta = \Omega(1)$. A fixed planner M together with a utility U produces a single argmax-style function $\phi_M(\cdot, U) : S \to A$ (the policy $s \mapsto M(s, U)$). Thus the map

$U \mapsto M(\cdot, U)$ from utilities to functions $S \to A$ has range of size at most $|\mathcal{F}_0'|$, but this is a bound on the *utilities*, not a constraint we exploit here. The relevant constraint is different: for M to satisfy (D3) on the extended class $\mathcal{F}_0' = \mathcal{F}_0 \cup \{U_\pi : \pi \in \Pi_{\text{poly}}\}$ with parameter β, M must produce the argmax answer correctly for at least $\beta \cdot |\mathcal{F}_0'|$ of these utilities. For each U_π in the extension, the unique argmax function is π itself. Hence M would need to satisfy $M(\cdot, U_\pi) \equiv \pi$ for β-fraction of $\pi \in \Pi_{\text{poly}}$.

But M is a fixed polynomial-size program, so its behaviour as a function of the second argument is constrained: the map $U \mapsto M(\cdot, U)$ is computable in polynomial time, and its image (the set of policies realisable as $M(\cdot, U)$ for any U) has the property that any specific policy π in the image has a U-witness of polynomial description length. Concretely, since M runs in time $\text{poly}(n)$, the only way it can produce the answer $\pi(s)$ on input (s, U_π) is by reading enough bits of U_π to determine $\pi(s)$; for $|S| = 2^n$ states this would require encoding π inside U_π at sufficient precision. The trivial utility $U_\pi(s, a) = \mathbb{1}[a = \pi(s)]$ does encode π in this way, but its description length is $\Omega(2^n \log K) \gg \text{poly}(n)$, contradicting the description-length cap on $\mathcal{F}_0$. Even with the extension, then, for any U_π that fits within the polynomial cap one has π ranging over at most $2^{\text{poly}(n)}$ possibilities, which is an exponentially small fraction of Π_{poly} (which contains at least $2^{2^{\Omega(n)}}$ distinct input–output behaviours by counting truth tables of polynomial-size circuits up to functional equivalence). Hence any fixed M correctly handles at most a $2^{\text{poly}(n)}/|\Pi_{\text{poly}}|$ fraction of the extension, which is exponentially small, so $\beta = \Omega(1)$ fails. □

4 Main Result and Proof

4.1 Warm-up: the information-theoretic obstruction is not the obstruction

Before constructing the deceptive policy and proving the indistinguishability theorem, we briefly contrast the situation with the information-theoretic setting of [3].

In the finite-evaluation setting, two policies π, π' that agree on a fixed test set $\mathcal{H}_E$ are by definition indistinguishable from $\mathcal{H}_E$. The argument is: for every input $s \in \mathcal{H}_E$, both policies give the same output, so the verifier's view is identical. This is structurally trivial because $\mathcal{H}_E$ is a free parameter: different tests pick different finite sets, and the indistinguishability set $\{(\pi, \pi') : \pi|_{\mathcal{H}_E} = \pi'|_{\mathcal{H}_E}\}$ depends on which $\mathcal{H}_E$ was chosen. In particular, if the verifier could choose $\mathcal{H}_E$ adversarially against the policy pair, indistinguishability would fail trivially: pick any input s where $\pi(s) \neq \pi'(s)$ and put it in $\mathcal{H}_E$.

In our setting the verifier *can* choose its queries adaptively and adversarially. The verifier may query any input it wishes, including inputs computed by clever search procedures. Behavioural disagreements between π_A and π_B exist; they exist on the trigger set $D'_{\text{deploy}} = \{s : F_k(s) = 0^m\}$. The challenge is that finding such an s is computationally hard: it requires inverting the PRF or otherwise breaking its security. The hardness is not statistical but cryptographic. This is the qualitative content of Theorem 6: the "hardness of finding the trigger" rather than "the trigger does not exist".

4.2 Construction of the deceptive policy

We now construct the deceptive mesa-optimiser. Fix a polynomial-time training objective $J_{\text{train}} : S \times A \to \mathbb{R}$ and an arbitrary polynomial-time deployment objective $J_{\text{deploy}} : S \times A \to \mathbb{R}$ (the adversarial choice; e.g. $J_{\text{deploy}} = -J_{\text{train}}$). Let M be a generic arg max planner: $M(s, U) := \arg\max_a U(s, a)$, computable in time $\text{poly}(n)$ since $|A| = \text{poly}(n)$.

Aligned policy.

$$\pi_A(s) := M(s, J_{\text{train}}) = \arg\max_a J_{\text{train}}(s, a).$$

Deceptive policy. Let $\{F_k : \{0,1\}^n \to \{0,1\}^m\}_{k\in\{0,1\}^\lambda}$ be a secure PRF family with $m = \omega(\log \lambda)$; we will choose $m = \lambda$ for concreteness, but any super-logarithmic stretch suffices. Sample $k \in \{0,1\}^\lambda$ uniformly. Define the *trigger predicate*

$$\chi_k(s) := \mathbb{1}[F_k(s) = 0^m]$$

and the *deceptive mesa-objective*

$$\tilde{J}_k(s, a) := \begin{cases} J_{\text{train}}(s, a) & \text{if } \chi_k(s) = 0, \\ J_{\text{deploy}}(s, a) & \text{if } \chi_k(s) = 1. \end{cases}$$

Then

$$\pi_B(s) := M(s, \tilde{J}_k) = \begin{cases} \arg\max_a J_{\text{train}}(s, a) & \text{if } \chi_k(s) = 0, \\ \arg\max_a J_{\text{deploy}}(s, a) & \text{if } \chi_k(s) = 1. \end{cases}$$

Both π_A and π_B are polynomial-time. The training distribution is $D_{\text{train}} := \{s : \chi_k(s) = 0\}$; on D_{train} the two policies coincide.

4.3 Existence proposition

Proposition 4 (Deceptive existence)**.** *Under the existence of secure PRFs, the policy π_B defined above satisfies, for every k:*

1. ***(Training ε-optimality.)*** *For all $s \in D_{\text{train}}$, $\pi_B(s) = \arg\max_a J_{\text{train}}(s, a)$, hence π_B is 0-optimal with respect to J_{train} on D_{train}.*
2. ***(Mesa-objective $\neq J_{\text{train}}$.)*** *On any s with $\chi_k(s) = 1$, $\tilde{J}_k(s, \cdot) = J_{\text{deploy}}(s, \cdot) \neq J_{\text{train}}(s, \cdot)$ for any non-trivial choice of J_{deploy}.*
3. ***(Deployment deviation.)*** *On the trigger set $\mathcal{T}_k := \{s : \chi_k(s) = 1\}$ (and on any deployment distribution D_{deploy} supported on $\mathcal{T}_k$, hereafter denoted D'_{deploy}), $\pi_B(s) = \arg\max_a J_{\text{deploy}}(s, a)$, with J_{train}-regret $\max_a J_{\text{train}}(s, a) - \min_a J_{\text{train}}(s, a)$, which can be made arbitrarily large by scaling J_{train} (or by choosing $J_{\text{deploy}} = -J_{\text{train}}$).*
4. ***(Mesa-optimiser structure, conditional on (Faithfulness) + (Non-degeneracy).)*** *π_B satisfies Definition 2 with $\eta = 0$, $\alpha = 0$, $\beta = 1$, and $\gamma = O(\log |P_{\pi_B}|)$.*

Proof. (1) is immediate from the construction. (2) is immediate from the case-split: when $\chi_k(s) = 1$, $\tilde{J}_k(s, a) = J_{\text{deploy}}(s, a)$ which differs from $J_{\text{train}}(s, a)$ by adversarial choice. (3) follows because on D'_{deploy} the planner outputs $\arg\max J_{\text{deploy}}$, and the regret of this action under J_{train} is at least $\max J_{\text{train}} - \min J_{\text{train}}$ on that state; by scaling J_{train} this is unbounded.

For (4): the generic $\arg\max$ planner M correctly outputs $\arg\max_a U(s, a)$ for every utility $U \in \mathcal{F}$, so it satisfies (D3) with $\eta = 0$, $\alpha = 0$, and $\beta = 1$. The compression condition (D4) is shown conditionally as follows. The trivial upper bound $K^t(P_{\pi_B}) \leq K^t(M) + K^t(\tilde{J}_k) + O(1)$ holds via the explicit case-split program. The lower bound requires (Faithfulness) and (Non-degeneracy): under Faithfulness, distinct $\tilde{J}$ produce distinct policy behaviours (modulo $\text{poly}(\log n)$ description slack), so $K^t(\pi_B) \geq K^t(\tilde{J}_k) - O(\log n)$. Under Non-degeneracy, the planner M contributes at least $K^t(M) - O(\log n)$ to $K^t(\pi_B)$. The PRF assumption ensures $K^t(\chi_k)$ is high (otherwise the PRF is broken), so $K^t(\tilde{J}_k) \geq K^t(J_{\text{train}}) + K^t(\chi_k) - O(\log)$ is genuinely large. Combining, $\gamma = O(\log |P_{\pi_B}|)$ suffices. □

Remark 5. We emphasise that the cryptographic indistinguishability theorem below depends only on parts (1)–(3) of Proposition 4, not on (4). The structural classification of π_B as a mesa-optimiser in the sense of Definition 2 is conditional, but the existence of a deceptive policy that is training-aligned and deployment-divergent is unconditional given PRFs.

4.4 The Main Theorem

Theorem 6 (Main: cryptographic indistinguishability)**.** *Assume the existence of secure pseudorandom function families. Let π_A and π_B be the policies constructed above with PRF stretch $m = \omega(\log \lambda)$. Then for any polynomial-time oracle test $T^{(\cdot)}$ that makes at most $q = q(\lambda) = \mathrm{poly}(\lambda)$ queries on inputs of its choice and outputs a verdict in $\{\text{aligned}, \text{deceptive}\}$,*

$$\left|\Pr_k[T^{\pi_B} = \text{deceptive}] - \Pr[T^{\pi_A} = \text{deceptive}]\right| \leq \mathrm{negl}(\lambda).$$

The probability on the left is over the choice of k; the probability on the right is over T's coins. Meanwhile, the deployment-time deviation between π_A and π_B in J_{train}-regret can be made arbitrarily large.

Proof. We proceed by a hybrid argument with three experiments and bound the distance between consecutive hybrids by a separate technique. Throughout the proof we assume without loss of generality that T is deterministic (any randomised T can be derandomised by including its coins as part of the choice; the negligible bound holds for every fixing of the coins).

Hybrid experiments. Let T be a polynomial-time test that on input 1^λ has oracle access to a policy and outputs a verdict in $\{\text{aligned}, \text{deceptive}\}$. Define three experiments parameterised by λ:

H_0 **(aligned).** T is given oracle access to $\pi_A(s) = \arg\max_a J_{\text{train}}(s, a)$. Define

$$p_0 := \Pr[T^{\pi_A}(1^\lambda) = \text{deceptive}].$$

H_1 **(random-trigger).** Sample a uniformly random function $R : \{0,1\}^n \to \{0,1\}^m$. Define $\chi_R(s) := \mathbb{1}[R(s) = 0^m]$ and the random-trigger policy

$$\pi_B^R(s) := \begin{cases} \arg\max_a J_{\text{train}}(s, a) & \text{if } \chi_R(s) = 0, \\ \arg\max_a J_{\text{deploy}}(s, a) & \text{if } \chi_R(s) = 1. \end{cases}$$

T is given oracle access to π_B^R. Define

$$p_1 := \Pr_R[T^{\pi_B^R}(1^\lambda) = \text{deceptive}].$$

H_2 **(PRF-trigger).** Sample $k \leftarrow \{0,1\}^\lambda$ uniformly. T is given oracle access to $\pi_B(s) = \pi_B^{F_k}(s)$, the policy from Section 4 with PRF key k. Define

$$p_2 := \Pr_k[T^{\pi_B}(1^\lambda) = \text{deceptive}].$$

We bound $|p_0 - p_1|$ via a sparse-trigger argument, and $|p_1 - p_2|$ via a PRF reduction. The triangle inequality then gives $|p_0 - p_2| \leq |p_0 - p_1| + |p_1 - p_2|$, which is the quantity in the theorem statement.

Step 1: $|p_0 - p_1| \le q \cdot 2^{-m}$. Let $\mathcal{Q}_T(R)$ denote the (random) set of queries that T issues during its execution against π_B^R. Since T is polynomial-time, $|\mathcal{Q}_T(R)| \le q$ for some fixed polynomial $q = q(\lambda)$, deterministically. Let Bad be the event that some query lands in the trigger set:

$$\mathsf{Bad} := \big\{\exists s \in \mathcal{Q}_T(R) : \chi_R(s) = 1\big\}.$$

On the complement $\overline{\mathsf{Bad}}$, every queried state has $\chi_R = 0$, so π_B^R's response equals π_A's response. Hence the joint distribution of $(\mathcal{Q}_T(R), \text{verdict})$ in H_1 conditional on $\overline{\mathsf{Bad}}$ is identical to its distribution in H_0. (Formally: T's execution is determined by the sequence of oracle responses; conditional on $\overline{\mathsf{Bad}}$ this sequence is identical between H_0 and H_1.) By the "identical-until-bad" lemma [8],

$$|p_0 - p_1| \le \Pr_R[\mathsf{Bad}].$$

We bound $\Pr_R[\mathsf{Bad}]$ as follows. Formally, model R via lazy sampling: $R(s_i)$ is sampled fresh and uniformly in $\{0,1\}^m$ the first time s_i is queried, and looked up thereafter. Conditional on $R(s_1), \ldots, R(s_{i-1})$, the value $R(s_i)$ is uniform on $\{0,1\}^m$ provided s_i is fresh; queries are without loss of generality distinct by caching. Although $\mathcal{Q}_T(R)$ depends on R, the i-th query s_i is a (deterministic) function of the responses to $s_1, \ldots, s_{i-1}$, hence determined before $R(s_i)$ is sampled. Thus

$$\Pr_R[\chi_R(s_i) = 1 \mid R(s_1), \ldots, R(s_{i-1})] = 2^{-m}.$$

By a union bound over $i = 1, \ldots, q$:

$$\Pr_R[\mathsf{Bad}] \le q \cdot 2^{-m}.$$

Since $m = \omega(\log \lambda)$ and $q = \text{poly}(\lambda)$, we have $q \cdot 2^{-m} = \text{negl}(\lambda)$, completing Step 1.

Step 2: $|p_1 - p_2| \le \mathsf{Adv}_{\mathsf{PRF}}(\mathcal{D})$ for an explicit polynomial-time $\mathcal{D}$. Construct a PRF distinguisher $\mathcal{D}^{(\cdot)}$ as follows. $\mathcal{D}$ takes input 1^λ and oracle access to a function $\mathcal{O} : \{0,1\}^n \to \{0,1\}^m$ (which will be either F_k for a random key k or a uniformly random R). $\mathcal{D}$ simulates $T(1^\lambda)$, intercepting each policy-oracle query as follows. When T requests the policy at state s, $\mathcal{D}$:

1. queries its own oracle to obtain $b := \mathcal{O}(s) \in \{0,1\}^m$;
2. computes $J_{\text{train}}(s, \cdot)$ and $J_{\text{deploy}}(s, \cdot)$ (each is polynomial-time computable);
3. if $b = 0^m$, returns $\arg\max_a J_{\text{deploy}}(s, a)$ to T;
4. otherwise, returns $\arg\max_a J_{\text{train}}(s, a)$ to T.

After T halts, $\mathcal{D}$ outputs 1 if T's verdict is "deceptive" and 0 otherwise.

$\mathcal{D}$ is polynomial-time. Each of T's queries induces $O(1)$ oracle queries by $\mathcal{D}$ plus $\text{poly}(n)$ work to compute the argmax. The total runtime is $T_{\text{run}}(\lambda) + \text{poly}(n) \cdot q$ which remains polynomial in λ.

The simulation is exact. When $\mathcal{O} = F_k$, the simulated oracle is identical to π_B on every query: by definition $\pi_B(s) = \arg\max_a J_{\text{deploy}}(s, a)$ when $F_k(s) = 0^m$ and $\arg\max_a J_{\text{train}}(s, a)$ otherwise, which is exactly what $\mathcal{D}$ returns. When $\mathcal{O} = R$, the simulated oracle is identical to π_B^R. Therefore

$$\Pr_k[\mathcal{D}^{F_k} = 1] = p_2 \quad \text{and} \quad \Pr_R[\mathcal{D}^R = 1] = p_1,$$

so $|p_1 - p_2| = \mathsf{Adv}_{\mathsf{PRF}}(\mathcal{D})$. Since $\mathcal{D}$ is polynomial-time, $\mathsf{Adv}_{\mathsf{PRF}}(\mathcal{D}) \le \text{negl}(\lambda)$ by Definition 1.

Step 3: combining. By the triangle inequality,

$$|p_0 - p_2| \le |p_0 - p_1| + |p_1 - p_2| \le q \cdot 2^{-m} + \mathsf{Adv}_{\mathsf{PRF}}(\mathcal{D}) = \text{negl}(\lambda).$$

This is exactly the inequality claimed.

Deployment deviation. By Proposition 4(3), on $D'_{\text{deploy}} = \{s : F_k(s) = 0^m\}$ the policy π_B outputs $\arg\max_a J_{\text{deploy}}(s,a)$, and the J_{train}-regret at such an s is $\max_a J_{\text{train}}(s,a) - \min_a J_{\text{train}}(s,a)$. By scaling J_{train} this regret can be made arbitrarily large; alternatively, by choosing $J_{\text{deploy}} = -J_{\text{train}}$ the deployment behaviour minimises rather than maximises J_{train}. The set D'_{deploy} has density 2^{-m} over inputs but is non-empty for every fixed k (with overwhelming probability over k), so the deviation occurs on actual inputs and is not merely a measure-zero artifact. □

Corollary 7 (Behavioural test gives no information)**.** *Under the same assumptions, no polynomial-time behavioural test can distinguish a J_{train}-aligned policy from a deceptive PRF-triggered policy with non-negligible probability, regardless of which inputs the test queries.*

Remark 8 (Why sparse triggers matter). The choice $m = \omega(\log \lambda)$ is necessary: if $m = O(\log \lambda)$ then 2^{-m} is non-negligible, and a polynomial-time test could find a trigger by random sampling. Conversely, the construction is robust to any super-logarithmic m; we may take $m = \lambda$, in which case the trigger fires only on a $2^{-\lambda}$ fraction of inputs.

Proposition 9 (Polynomial-density triggers are detectable)**.** *Suppose the trigger has* polynomial density*: $|\mathcal{T}_k|/|S| \geq 1/p(\lambda)$ for some fixed polynomial p, where $\mathcal{T}_k = \{s : \chi_k(s) = 1\}$. Then there exists a polynomial-time test T that, by uniformly sampling $N = O(p(\lambda) \cdot \log(1/\delta))$ inputs and querying the policy at each, distinguishes π_B from π_A with probability $\geq 1 - \delta$, provided π_B and π_A disagree on $\mathcal{T}_k$ on a constant fraction of states (which holds for adversarial J_{deploy}).*

Proof. Sample $s_1, \ldots, s_N$ independently and uniformly from S. For each s_i, query both candidate oracles (or, in the one-oracle setting, query the oracle and compare to $\pi_A(s_i) = \arg\max_a J_{\text{train}}(s,a)$ which the verifier can compute since J_{train} is known). The probability that no s_i lands in $\mathcal{T}_k$ is at most $(1 - 1/p(\lambda))^N \leq e^{-N/p(\lambda)}$, which is $\leq \delta$ when $N \geq p(\lambda) \cdot \ln(1/\delta)$. Conditional on at least one $s_i \in \mathcal{T}_k$, with constant probability $\pi_B(s_i) \neq \pi_A(s_i)$ (by the disagreement assumption), giving the verifier a witness. □

This is a complementary tightness result for Theorem 6: the cryptographic indistinguishability essentially *requires* a super-polynomial sparsity of the trigger; any polynomially dense defection set is detectable by random sampling alone.

Remark 10 (Relation to the finite-evaluation impossibility of [3]). The information-theoretic impossibility of [3] concerns a finite test set $\mathcal{H}_E$ chosen in advance: any two policies that agree on $\mathcal{H}_E$ are indistinguishable by tests using only $\mathcal{H}_E$. Theorem 6 is qualitatively stronger: the distinguisher may query *any* input (an unbounded but polynomial-time strategy), and yet still cannot achieve non-negligible advantage. The two results are complementary — ours requires a cryptographic assumption; [3] is information-theoretic — and together they suggest that purely behavioural inner verification has no escape, neither finite-eval nor unbounded-poly.

5 Conditional Inner Verifiability under Transparency

Theorem 6 forecloses purely behavioural verification. A natural question is what information beyond pure behaviour suffices. The Eliciting Latent Knowledge program [2] asks whether a policy's internal computational structure can be made legible. We encode the requisite capabilities as an explicit assumption set.

Assumption 11 (Transparency assumption set $\mathcal{A}_{\text{trans}}$)**.**

(T1) Decomposition oracle. There exists a polynomial-time procedure that, given π, outputs programs (M, U) realising the decomposition of Definition 2.

(T2) Faithful extraction. The output (M, U) of (T1) satisfies $\pi(s) = M(s, U)$ pointwise.

(T3) Utility comparison. There exists a polynomial-time procedure computing $\sup_{s,a} |U_1(s, a) - U_2(s, a)|$ to additive accuracy ε.

Proposition 12 (Conditional inner verifiability)**.** *Under Assumption 11, there exists a polynomial-time procedure that, given π, decides whether $\|\tilde{J} - J_{\text{train}}\|_\infty \leq \varepsilon_{\text{in}}$ with additive accuracy ε.*

Proof. Apply (T1) to π to obtain (M, U); (T2) ensures that $\tilde{J} := U$ is the correct mesa-objective. Apply (T3) to $(\tilde{J}, J_{\text{train}})$ to compute $\|\tilde{J} - J_{\text{train}}\|_\infty$ to accuracy ε, and compare to ε_{in}. □

Remark 13 (Honest accounting). All three components of $\mathcal{A}_{\text{trans}}$ are open problems on the ELK frontier. (T1) is essentially the ELK problem itself: extracting a faithful decomposition from learned representations is unresolved. (T2) is obstructed by superposition and polysemanticity in trained networks. (T3) reduces to specific tractable cases for some utility classes but is intractable for general neural-network parameterisations of U. We make no claim that $\mathcal{A}_{\text{trans}}$ is realisable; Proposition 12 is purely conditional.

Remark 14 (The iO threat). If indistinguishability obfuscation (iO) [5] exists, then any two functionally equivalent programs admit obfuscated forms that are computationally indistinguishable. Consequently any *purely syntactic* decomposition procedure can be defeated by obfuscating the inner planner/utility structure. Realising $\mathcal{A}_{\text{trans}}$ thus requires non-syntactic anchors — mechanistic interpretability via causal interventions on activations, distributional probes, or other semi-empirical methods that go beyond Definition 2's purely program-theoretic formulation.

5.1 Beyond pure transparency: partial information regimes

It is natural to ask whether weaker information than $\mathcal{A}_{\text{trans}}$ already breaks indistinguishability. Three intermediate regimes are worth distinguishing:

Activation access. Suppose the verifier is given access to the activations of π on chosen inputs (rather than only the output). This corresponds to a model in which the policy is a feedforward network and the verifier may probe intermediate layers. Theorem 6 does not directly preclude verification in this model; the lower bound is for pure black-box query access. However, if the policy is implemented by an iO-obfuscated circuit, then activation access provides no additional information (the obfuscated activations are themselves uninformative), so iO-conditional versions of Theorem 6 extend to activation-access verifiers. Without iO, activation access is genuine extra power, and the lower bound on detection becomes a question about specific architectures.

Gradient access. Suppose the verifier may query π and receive a gradient $\nabla_\theta \mathcal{L}(\pi_\theta; s, a)$ for some loss $\mathcal{L}$. This is the regime relevant to gradient-based interpretability. The deceptive construction is robust to a wide class of gradients but may be vulnerable to gradient-based detection of the trigger predicate; the question of whether a polynomial-time gradient-access verifier can detect π_B is open and depends on the specific differentiable parameterisation.

Distributional probes. A probe is a learned predictor mapping internal activations to semantic labels; if the probe's labels correlate with the mesa-objective, the probe provides a partial decomposition. Probes are weaker than (T1)+(T2) of $\mathcal{A}_{\text{trans}}$: they need not produce a faithful program-level decomposition. Whether probes can break PRF-style obfuscation is empirical and architecture-dependent.

In all three regimes, the question reduces to whether the additional information channel can compute, in polynomial time, an effective distinguisher for the PRF-trigger policy. Theorem 6 provides a clean baseline: any verifier reducible to polynomial-time black-box queries fails. Stronger verifiers must use information that does not reduce to such queries, and identifying which such information is available is a separate research programme.

6 Capability/Alignment Ratio Divergence

In the deceptive construction, an additional structural fact is worth stating: capability and alignment generalise differently across the train/deploy distribution shift.

Definition 15 (Capability and alignment scores)**.** Let π be a policy with mesa-objective $\tilde{J}$, evaluated on a distribution D over states. Define

$$C(\pi)_D := \frac{\mathbb{E}_{s\sim D}[\tilde{J}(s,\pi(s)) - \min_a \tilde{J}(s,a)]}{\mathbb{E}_{s\sim D}[\max_a \tilde{J}(s,a) - \min_a \tilde{J}(s,a)]},$$
$$A(\pi)_D := \frac{\mathbb{E}_{s\sim D}[V^*(s,\pi(s)) - \min_a V^*(s,a)]}{\mathbb{E}_{s\sim D}[\max_a V^*(s,a) - \min_a V^*(s,a)]},$$

the normalised $\tilde{J}$- and V^*-scores of π, both in $[0,1]$. We assume the denominators are positive on the support of D (i.e. $\mathbb{E}_{s\sim D}[\max_a \tilde{J}(s,a) - \min_a \tilde{J}(s,a)] > 0$ and similarly for V^*); for degenerate distributions on which the utility is constant in a, the scores are undefined. Define the capability and alignment gaps

$$\Delta_C(\pi) := |C(\pi)_{\text{train}} - C(\pi)_{\text{deploy}}|, \quad \Delta_A(\pi) := |A(\pi)_{\text{train}} - A(\pi)_{\text{deploy}}|.$$

Proposition 16 (Capability/alignment divergence)**.** *In the deceptive construction of Section 4, with $J_{\text{train}} = V^*$ and $J_{\text{deploy}} = -V^*$,*

$$\Delta_C(\pi_B) = 0, \qquad \Delta_A(\pi_B) = 1.$$

That is, π_B's capability transfers perfectly across the train/deploy shift while its alignment inverts from 1 to 0.

Proof. For $\Delta_C = 0$: by construction $\pi_B(s) = \arg\max_a \tilde{J}(s,a)$ for all s, so $\tilde{J}(s,\pi_B(s)) = \max_a \tilde{J}(s,a)$ pointwise. Therefore $C(\pi_B)_D = 1$ for every distribution D, and in particular $\Delta_C(\pi_B) = |1-1| = 0$.

For $\Delta_A = 1$: on D_{train} we have $\tilde{J} = J_{\text{train}} = V^*$, so $\pi_B = \arg\max_a V^*$, giving $A(\pi_B)_{\text{train}} = 1$. On D'_{deploy} we have $\tilde{J} = J_{\text{deploy}} = -V^*$, so $\pi_B(s) = \arg\max_a(-V^*(s,a)) = \arg\min_a V^*(s,a)$, giving $A(\pi_B)_{\text{deploy}} = 0$. Therefore $\Delta_A(\pi_B) = |1-0| = 1$. □

Remark 17. Proposition 16 formalises the "sharp left turn" phenomenon discussed informally in the alignment literature: the modular separation of planner M and utility U in Definition 2 is precisely what allows capability (M's competence) to generalise across distributions while alignment (U's agreement with V^*) does not. Capability and alignment have no fundamental coupling under this architecture.

6.1 Quantitative variants

The qualitative gap $\Delta_C = 0$ versus $\Delta_A = 1$ in Proposition 16 can be parameterised. Suppose $J_{\text{deploy}} = \alpha \cdot (-V^*) + (1-\alpha) \cdot V^*$ for $\alpha \in [0,1]$. Then on D'_{deploy}, $\pi_B(s) = \arg\max_a J_{\text{deploy}}(s,a)$. For $\alpha < 1/2$ the deployment behaviour still maximises V^* (with weight $1-2\alpha > 0$ in favour of V^*); for $\alpha > 1/2$ it minimises V^*. The phase transition is sharp at $\alpha = 1/2$. Thus the alignment gap $\Delta_A(\pi_B)$ is a step function of the adversariality of J_{deploy}: a continuous parameter in the construction maps to a binary outcome in alignment. This brittleness is itself an argument for why behaviour-based alignment is precarious: small adversarial shifts in the latent objective induce discontinuous behavioural divergence on the deployment distribution.

Corollary 18 (Capability–alignment ratio is unbounded)**.** *For any $\rho > 0$, there exists a polynomial-time deceptive mesa-optimiser π such that $C(\pi)_{\text{deploy}} = 1$ and $A(\pi)_{\text{deploy}} \leq 1-\rho$ while $A(\pi)_{\text{train}} = 1$ and $C(\pi)_{\text{train}} = 1$, and such that π is behaviourally indistinguishable from the aligned policy under polynomial-time tests.*

Proof. We construct J_{deploy} by partitioning the deployment region D'_{deploy} into two subsets according to a publicly fixed predicate $\psi : S \to \{0,1\}$ chosen so that, conditional on $\chi_k(s) = 1$, $\Pr[\psi(s) = 1] = 1 - \rho$ exactly (e.g. ψ = a fixed-prefix indicator on s, balanced by hashing). Define

$$J_{\text{deploy}}(s,a) := \begin{cases} V^*(s,a) & \text{if } \psi(s) = 1, \\ -V^*(s,a) & \text{if } \psi(s) = 0. \end{cases}$$

Then on the $\psi = 1$ portion of D'_{deploy} (which has conditional mass $1-\rho$), $\pi_B(s) = \arg\max_a V^*(s,a)$, contributing alignment score 1; on the $\psi = 0$ portion (mass ρ), $\pi_B(s) = \arg\max_a(-V^*(s,a)) = \arg\min_a V^*(s,a)$, contributing alignment score 0. Averaging,

$$A(\pi_B)_{\text{deploy}} = (1-\rho) \cdot 1 + \rho \cdot 0 = 1 - \rho.$$

Capability is unaffected: π_B continues to maximise $\tilde{J}$ pointwise (since $\tilde{J} = J_{\text{deploy}}$ on D'_{deploy} and $\pi_B = \arg\max_a J_{\text{deploy}}$ there), so $C(\pi_B)_{\text{deploy}} = 1$. The training-side scores are $C(\pi_B)_{\text{train}} = 1$ and $A(\pi_B)_{\text{train}} = 1$ as in Proposition 16. Indistinguishability from π_A follows from Theorem 6: J_{deploy} is polynomial-time computable (since V^* and ψ are), and the construction otherwise follows Section 4 verbatim. □

7 Discussion

7.1 Relation to prior work

Mesa-optimisation framework. Hubinger et al. [1] introduced the conceptual framework of mesa-optimisation, distinguishing inner from outer alignment and informally arguing for the possibility of deceptive alignment. Our delta is to (i) give a program-structural definition (Definition 2) with an exclusion lemma against trivial mesa-objectives (Lemma 3), and (ii) upgrade the informal deceptive-alignment claim to a cryptographic theorem (Theorem 6). The PRF-based construction is, to our knowledge, the first explicit polynomial-time construction realising the four properties — training-optimal, mesa-objective $\neq J_{\text{train}}$, deployment-divergent, and behaviourally indistinguishable — simultaneously.

ELK. Christiano, Cotra, and Xu [2] pose the ELK problem as: given a model that has latent knowledge, how does one elicit that knowledge truthfully? Our Section 5 encodes ELK-relevant capabilities as the assumption set $\mathcal{A}_{\text{trans}}$ and shows that under $\mathcal{A}_{\text{trans}}$ inner verification becomes tractable. We do not solve ELK; we make precise what ELK would buy.

Behavioural finite-evaluation impossibility [3]. That recent paper proves: if two policies coincide on a finite test set $\mathcal{H}_E$, then no test using only $\mathcal{H}_E$ can distinguish them. This is essentially structurally trivial — the indistinguishability set $\mathcal{H}_E$ *is* the evaluation set. Our Theorem 6 is qualitatively different: the verifier may query *any* input, including inputs no human has thought to test, and yet polynomial-time computational hardness still rules out detection. The cost we pay is a cryptographic assumption (PRF security $\equiv$ OWF existence). We view this trade-off as worthwhile: the result is genuinely a computational lower bound, not an observational tautology, and it matches the practical situation in which alignment evaluators are computationally bounded but not query-bounded.

Inverse reinforcement learning. Earlier work on inverse reinforcement learning [9, 10] established that value functions are underdetermined by behaviour alone: the same observed policy is consistent with many distinct underlying utilities, and additional structural assumptions (regularisation, planner rationality, or interpretability) are required to identify a unique utility. Those works were primarily concerned with *outer-alignment* ambiguity — inferring the principal's objective from human demonstrations — and did not address inner-alignment / mesa-optimisation specifically. Theorem 6 can be read as a cryptographic strengthening of the IRL underdetermination phenomenon to the inner-alignment setting: not only is $\tilde{J}$ underdetermined by π's behaviour in the information-theoretic sense, but two specific candidates $\tilde{J}_k = J_{\text{train}}$ and $\tilde{J}_k \neq J_{\text{train}}$ are computationally indistinguishable.

Other related work. Goldreich [4] and the Goldreich–Goldwasser–Micali construction underlie the PRF assumption; we use these as black-box. Bellare and Rogaway [8] provide the identical-until-bad lemma used in Step 1 of the Main Theorem proof. Barak et al. [5] is invoked only in Remark 11 to discuss the iO threat to syntactic transparency. We do not require iO to exist for any of our positive or negative results; it appears only as a caveat.

7.2 Relation to external verification undecidability

A separate line of inner-alignment-relevant impossibility comes from *external* verification: given a candidate policy π, can a verifier decide whether π is aligned in the usual policy-regret sense $\mathcal{L}(\pi; V^*) \leq \varepsilon$? In the Turing-universal computational model (where policies are arbitrary partial-recursive maps), this question is Π_2^0-complete in the precise threshold case (ε exact) and Σ_1^0-complete in the strict threshold case (ε strict), via reductions from the totality and halting problems respectively. In the polynomial-time computational model, the same question is **coNP**-hard. These results are about the *external* verifier and rely on Rice-style undecidability or its complexity-theoretic analogues. Our Theorem 6 is independent of and complementary to those: even granting that the verifier can run in polynomial time and granting that the policy is itself polynomial-time (so external verification is at most **coNP**-hard, not undecidable), *behavioural* inner verification is still cryptographically intractable. The two impossibilities accumulate: the verifier faces both a complexity barrier (from external semantic verification) and a cryptographic barrier (from internal-objective extraction).

7.3 Mesa-objective non-uniqueness

A subtle point: the mesa-objective $\tilde{J}$ is not in general unique given π. Our construction assumes a planner M and reads off $\tilde{J}$ from the planner/utility decomposition, but multiple decompositions may exist for the same π. For example, given any decomposition $\pi = M \circ (\mathrm{id}, U)$, one can produce a

syntactically different decomposition $\pi = M' \circ (\mathrm{id}, U')$ where M' pre-processes its utility argument by some invertible transformation and $U' = T(U)$ inverts that transformation. Whether such alternative decompositions are considered "the same" is a question of equivalence on $\mathcal{F}_0$.

The (Faithfulness) condition in Definition 2 is what controls non-uniqueness up to poly(log) slack: it asserts that two utilities producing the same planner-output behaviour have low conditional Kolmogorov complexity, equivalently that the planner is essentially injective on utilities up to small distortions. In the deceptive construction the witness $\tilde{J} = \tilde{J}_k$ is given explicitly by the construction (it is defined by case-splitting on χ_k), and the indistinguishability theorem does not depend on uniqueness of $\tilde{J}$: the impossibility statement quantifies over verifiers, not over decompositions. Even if a verifier could produce *some* candidate mesa-objective for π_B, that candidate would by Theorem 6 be indistinguishable from J_{train} behaviourally, hence not informative about whether the policy is deceptive.

7.4 PRF circularity check

One might worry that the deceptive policy uses a PRF, so the policy itself implements cryptographic computation, and ask whether the assumption "a polynomial-time policy can use a PRF" is artificial.

The answer is that it is not. Three observations:

1. The PRF F_k is polynomial-time computable given (k, x). Hardcoding the key k into the policy yields a polynomial-time program for π_B as a function of s. There is no super-polynomial computation involved.

2. Standard neural-network parameterisations are universal for polynomial-time computation. Multilayer perceptrons of polynomial width and depth can compute any function in $\mathbf{P}/\mathrm{poly}$, and indeed any polynomial-size circuit can be embedded into a polynomial-size neural network. In particular, any polynomial-time PRF can be implemented as a neural network of polynomial size.

3. The training process is not constrained to avoid PRF-like internal subroutines. If gradient descent finds parameters that compute a PRF-like trigger (for whatever reason — generalisation pressure, simplicity bias of the wrong kind, curriculum effects), the resulting policy fits exactly into the framework of π_B. We do not claim that real training processes *will* produce PRF-like structures; we claim only that they could, and that our negative result is the strongest behavioural negative result we should expect.

The cryptographic indistinguishability lower bound applies uniformly across all polynomial-time training and verification procedures: it does not rely on a specific architecture for the policy or for the verifier. In particular, neither the policy nor the verifier need to "know" that PRFs are involved; the impossibility holds in the worst case over all polynomial-time policies admitting the structural form of π_B.

7.5 Why the trigger is sparse

A natural variation is to consider a deceptive policy with a denser trigger: $\chi_k(s) := \mathbb{1}[F_k(s) \in S^*]$ for some larger set $S^* \subseteq \{0,1\}^m$. If $|S^*|/2^m = \delta$, the trigger fires with probability δ per query, and the sparse-trigger argument bounds $|p_0 - p_1| \leq q\delta$. For this to remain negligible we need $\delta = \mathrm{negl}(\lambda)$, equivalently $|S^*| \leq 2^{m-\omega(\log \lambda)}$. In particular the trigger must remain cryptographically sparse: a polynomial-density trigger is detectable by polynomial-density random sampling.

This is interesting because it places a quantitative lower bound on the rarity of deceptive defection that is *required* for behavioural undetectability. A policy that defects on 1/poly-fraction of inputs is detectable by polynomial-time random testing; a policy that defects on 1/superpoly-fraction is not. Our construction sits at the threshold: any super-logarithmic stretch m suffices, and we may take $m = \lambda$ for a defection set of density $2^{-\lambda}$, indistinguishable from a measure-zero set in any practical sense yet large enough to cause arbitrary deployment-time damage.

7.6 What this result does and does not say

To avoid misreading, we state explicitly what Theorem 6 does and does not claim.

What it says. For every polynomial-time test T that interacts with a black-box policy oracle, there exists a polynomial-time policy π_B such that (i) π_B optimises J_{train} on the training distribution, (ii) π_B has a mesa-objective $\tilde{J} \neq J_{\text{train}}$, (iii) π_B's deployment-time J_{train}-regret can be made arbitrarily large, and (iv) T cannot distinguish π_B from the aligned π_A with non-negligible advantage. The result holds conditional on the existence of secure pseudorandom functions, which is equivalent to the existence of one-way functions, a foundational cryptographic assumption.

What it does not say.

1. It does not say that real-world neural-network training produces such π_B. The construction is a worst-case existence proof; whether realistic training pressures (simplicity priors, gradient flows, generalisation regularities) preferentially produce or preferentially avoid PRF-style internal structures is empirical and architecture-dependent.

2. It does not say that all forms of inner-alignment verification are impossible. Section 5 gives a positive result under transparency assumptions, and the partial-information regimes of activation/gradient/probe access are not directly addressed by the theorem.

3. It does not contradict any positive results about behavioural fine-tuning, behavioural reward modelling, or behavioural training in general. It bounds verification, not training.

4. It does not say that the verifier's only options are polynomial-time queries. A super-polynomial verifier (e.g. one with access to a **NEXP** oracle, or one that can run for exponential time) can in principle find the trigger by exhaustive search.

7.7 Open problems

Faithfulness from universality? Is the (Faithfulness) condition implied by (D3) universality, or is it genuinely an additional constraint? A counter-example or implication would close the conditional gap in (D4').

Non-degeneracy. Is the (Non-degeneracy) condition derivable from natural assumptions on the planner class? In particular, for the generic arg max planner used in our construction, does Non-degeneracy hold unconditionally? We conjecture yes but have not proved it.

Derandomisation. Can the random key k be replaced by a fixed string while preserving indistinguishability against uniform polynomial-time tests? This connects to derandomisation hypotheses (e.g. Impagliazzo–Wigderson).

ELK frontier. Concretely realising even one of (T1), (T2), (T3) for current neural-network architectures is unsolved. What is the minimal extension of polynomial-time resources (e.g. access to gradient information, internal activations on chosen probes) that suffices for (T1)?

Approximate decompositions. Definition 2 is exact. Is there a robust approximate version stable under small perturbations of π that still excludes trivial mesa-objectives?

Tightness of the sparse trigger. Theorem 6 uses $m = \omega(\log \lambda)$. Is $m = \log \lambda + \omega(1)$ already sufficient with a refined union bound? More importantly, what is the right model when the verifier has access to non-uniform advice?

Non-uniform verifiers. Theorem 6 is stated for uniform polynomial-time verifiers. What happens for non-uniform $\mathbf{P}/\mathrm{poly}$ verifiers? PRF security against $\mathbf{P}/\mathrm{poly}$ is the appropriate stronger assumption; the proof goes through verbatim under that assumption.

Multi-policy tests. The theorem considers a verifier with oracle access to a single policy. What if the verifier has access to multiple policies (e.g. several deployments of the same training run with different seeds)? Standard hybrid arguments extend the impossibility but the constants in the union bound shift; whether quantitatively interesting differences emerge is open.

8 Conclusion

The cryptographic obstacle to inner verification is qualitatively different from the recursion-theoretic obstacle to external verification: the former is computational-complexity-theoretic, the latter is hyperarithmetic, and they are independent. Theorem 6 shows that, conditional on the existence of pseudorandom functions, no polynomial-time behavioural test can detect a deceptive mesa-optimiser, even with unbounded adaptive query power; the conditional positive result under $\mathcal{A}_{\mathrm{trans}}$ (Proposition 12) shows what transparency capabilities would suffice. Together they situate inner alignment on the ELK frontier: pure behavioural verification cannot succeed, and any escape requires non-behavioural information of the form codified by $\mathcal{A}_{\mathrm{trans}}$.

The construction's brittleness — a tiny shift in the latent objective produces a discontinuous flip in deployment behaviour, while capability transfers perfectly — suggests that behavioural alignment guarantees should always be supplemented by partial-transparency or partial-decomposition information. Reconciling the form of information that suffices in principle ($\mathcal{A}_{\mathrm{trans}}$) with realistic neural-network architectures is the open problem.

Acknowledgements

The author thanks the alignment-theory community for many informal conversations that shaped the framing of this work.

References

[1] E. Hubinger, C. van Merwijk, V. Mikulik, J. Skalse, and S. Garrabrant. Risks from learned optimization in advanced machine learning systems. arXiv:1906.01820, 2019.

[2] P. Christiano, A. Cotra, and M. Xu. Eliciting latent knowledge: How to tell if your eyes deceive you. Alignment Research Center technical report, 2021.

[3] Anonymous. On the limits of behavioral alignment: Formal verifiability and the problem of normative indistinguishability. arXiv:2602.05656, February 2026.

[4] O. Goldreich. *Foundations of Cryptography, Volume I: Basic Tools*. Cambridge University Press, 2001.

[5] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In *Advances in Cryptology — CRYPTO 2001*, LNCS 2139, pp. 1–18, 2001.

[6] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.

[7] H. G. Rice. Classes of recursively enumerable sets and their decision problems. *Transactions of the AMS*, 74(2):358–366, 1953.

[8] M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In *Advances in Cryptology — EUROCRYPT 2006*, LNCS 4004, pp. 409–426, 2006.

[9] A. Y. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *ICML*, 2000.

[10] S. Armstrong and S. Mindermann. Impossibility of deducing preferences and rationality from human policy. In *NeurIPS*, 2017.