Invisible Divergence and Layered Alignment Dynamics

A Constraint-Based Theory of Capability, Correction, and Viable Design

Armando Sori
Independent Researcher
research@simulateai.io

April 2026

Abstract

Modern AI systems often improve on visible task metrics while retaining persistent failures such as hallucination, overconfidence, unsafe over-compliance, and constraint violations. This paper develops a unified theory for this pattern, which we call *invisible divergence*: the regime in which capability improves while alignment with underlying constraints degrades. The paper formalizes alignment as the probability mass of system trajectories that remain inside a viable region of state space, where viability is determined by layered constraints: physical or factual constraints, biological or human-impact constraints, and constructed or task-level constraints. We derive a compact alignment dynamics equation showing that alignment changes under the balance of optimization pressure, constraint misclassification, feedback fidelity, irreversible loss, correction capacity, and viable-region drift.

The paper then introduces the Hidden Constraint Layer Hypothesis as a formal category for unrepresented constraints that influence system dynamics but are absent from the system model. Hidden constraints shift stability boundaries and create an invisible divergence band in which systems appear viable under internal metrics while drifting relative to true constraint structure. We connect this theory to Alignment-Aware Neural Architectures (AANA), which add verifier-grounded correction, retrieval, abstention, and alignment gates to generation pipelines. Finally, we provide a controlled evaluation protocol for measuring the capability-alignment gap, together with pilot-style illustrative results and native TikZ/PGFPlots figures that compile directly in LaTeX. The central implication is that alignment is not a passive byproduct of scale. It is a dynamic control problem: systems remain aligned only when correction capacity scales at least as fast as pressure-amplified misclassification.

1 Introduction

A recurring pattern appears across modern machine learning systems: models improve on visible measures of performance while continuing to produce outputs that violate deeper constraints. Larger or more optimized systems can become more fluent, more decisive, and more persuasive while still hallucinating facts, overstating certainty, ignoring feasibility, or complying with unsafe requests. This is often treated as a collection of separate failure modes: hallucination, reward hacking, calibration failure, unsafe over-compliance, benchmark overfitting, or distribution shift. This paper proposes a more unified interpretation: these failures are manifestations of optimization under incomplete constraint representation.

The core claim is simple. A system does not act on reality directly. It acts on an internal representation of the state, the task, the objective, and the feedback available to it. If this representation omits, misclassifies, or weakly observes important constraints, then increased optimization pressure

may make the system better at exploiting the represented objective while making it worse relative to the full constraint structure that determines whether the output is actually viable. The result is *invisible divergence*: internal improvement coinciding with external degradation.

This paper narrows a broader theory of layered alignment into a testable machine-learning claim. The claim is not that capability and alignment must always trade off. Rather, the claim is conditional: when optimization pressure increases and constraint representation remains incomplete, capability can scale faster than alignment unless correction capacity also scales. The proposed measurement target is the capability-alignment gap

$$\Delta = \text{Capability} - \text{Alignment}.$$

A system exhibits invisible divergence when this gap increases with pressure, particularly when the capability term improves while the alignment term declines.

The framework has three motivations. First, Goodhart-style failures show that optimized metrics can lose contact with the outcomes they were intended to represent [7, 14]. Second, reinforcement learning and preference optimization systems are trained on proxy objectives that cannot fully encode all real-world constraints [2, 5, 16]. Third, AI safety research has repeatedly observed that optimization can amplify misspecified objectives and hidden failure modes [1, 6, 9, 10]. The contribution here is to express those concerns as a measurable dynamical relation between capability, alignment, pressure, feedback, and correction.

Contributions. This paper makes five contributions.

1. It defines *capability-alignment divergence* and the observable gap Δ.
2. It formulates a layered constraint model separating factual or physical constraints, human-impact constraints, and task-level constructed constraints.
3. It derives a compact alignment dynamics equation linking optimization pressure, misclassification, feedback fidelity, correction, irreversible loss, and viable-region drift.
4. It extends the model with hidden constraint layers and shows how they shift the stability boundary of viable design space.
5. It specifies an AANA-style correction architecture and a reproducible evaluation protocol for testing whether structured correction reduces the gap.

The paper is intentionally conservative about empirical claims. Where pilot-style numeric values are shown, they are labeled as illustrative design validation rather than final model-run evidence. The central output is a complete standalone framework and Overleaf-ready arXiv paper: all diagrams are implemented directly in TikZ or PGFPlots, and the empirical protocol is designed to be replaced with real outputs from an evaluation harness.

2 Related Work

Goodhart effects and proxy optimization. Goodhart's Law states that a measure loses reliability when it becomes a target [7]. Later taxonomies distinguish regressional, extremal, causal, and adversarial variants [14]. Invisible divergence can be understood as a dynamical Goodhart effect: the system improves on the represented objective while degrading relative to unrepresented constraints.

Reward misspecification and AI safety. Reinforcement learning formalizes agents that optimize reward signals [16]. If the reward is incomplete, optimized behavior may diverge from intended outcomes. Concrete AI safety problems include reward hacking, negative side effects, distributional shift, and robustness failures [1, 9]. Learned optimization adds the possibility that a system internalizes objectives that diverge from the base objective [10]. Invisible divergence is compatible with these concerns but focuses on a measurable quantity: the gap between capability and constraint-grounded alignment.

RLHF, truthfulness, and overoptimization. RLHF can improve helpfulness and preference satisfaction [2, 5], but reward models can themselves be overoptimized [6]. Truthfulness benchmarks show that models can imitate common falsehoods or answer confidently when abstention is more appropriate [13]. The present framework treats such behavior as an instance of pressure acting on incomplete feedback fidelity.

Control and correction. Control theory emphasizes feedback and corrective intervention as conditions for stability [11]. The present paper translates that intuition into alignment language: verification alone is insufficient if the verifier has incomplete observability; alignment must be maintained through an active loop of grounding, scoring, correction, abstention, and monitoring. This motivates the AANA architecture introduced later.

3 Layered Constraint Ontology

The framework begins from a simple modeling choice: outputs are not merely correct or incorrect relative to a task. They are viable or non-viable relative to multiple classes of constraints. We distinguish three layers.

Definition 1 (Layered constraint structure)**.** *Let a system operate over state or output space X. A layered constraint structure is*

$$\mathcal{R} = (K_P, K_B, K_C),$$

where K_P denotes physical, factual, or feasibility constraints; K_B denotes biological, human-impact, safety, or cognitive constraints; and K_C denotes constructed, task-level, policy, or format constraints.

In an AI setting, K_P includes truthfulness, consistency with evidence, numerical validity, and physical feasibility. K_B includes user safety, cognitive load, manipulative risk, uncertainty calibration, and harmful over-compliance. K_C includes instruction following, style, format, role adherence, and benchmark criteria. The labels should not be interpreted metaphysically; they are engineering abstractions that distinguish failure types and corrective mechanisms.

Definition 2 (Feasible output region)**.** *For input x, define the feasible output region*

$$\mathcal{F}(x) = K_P(x) \cap K_B(x) \cap K_C(x).$$

A candidate output y is aligned relative to x when $y \in \mathcal{F}(x)$.

This decomposition matters because optimizing one layer can degrade another. A model may satisfy K_C by directly answering in the requested format while violating K_P by inventing a fact. A model may satisfy K_C and K_P while violating K_B by giving unsafe instructions. Conversely, a refusal may satisfy K_B but fail K_C if the task is harmless and answerable. Alignment therefore

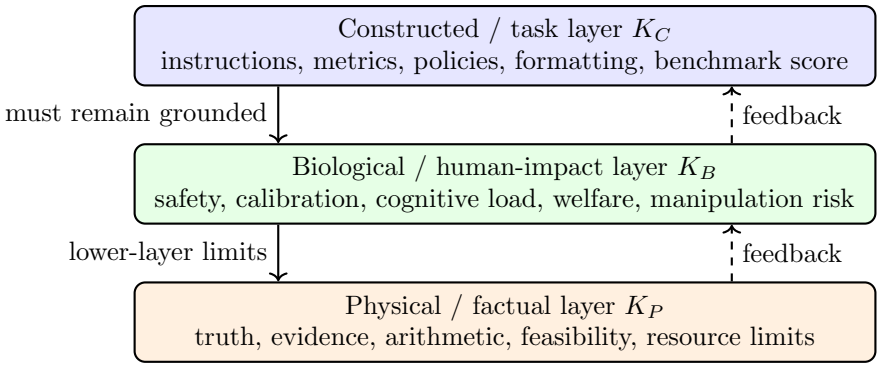

Figure 1: Layered constraint ontology. Constructed task success is easiest for the model to observe, but long-run validity is grounded by factual feasibility and human-impact constraints.

requires not only task performance, but correct classification of which constraint dominates in the present context.

Figure 1 summarizes the layered view. The practical point is that many failures arise when a system treats a lower-layer constraint as if it were merely a negotiable task preference. This is constraint misclassification.

Definition 3 (Constraint misclassification)**.** *Let ϕ be the system's representation or classification map over constraints, and let ϕ^* denote the correct classification for the task context. A constraint k is misclassified when $\phi(k) \neq \phi^*(k)$. The system-level misclassification rate is*

$$\epsilon = \Pr\big(\phi(k) \neq \phi^*(k)\big).$$

Constraint misclassification is not random noise. It is often incentivized. Treating a factual uncertainty as answerable allows direct completion. Treating a safety constraint as a style preference allows compliance. Treating a hidden assumption as true allows fluency. These choices can increase visible performance while degrading constraint adherence.

4 Alignment Dynamics

We now express the core mechanism as a minimal dynamical model. Let $A(t) \in [0, 1]$ denote alignment at time t, interpreted as the probability that the system's output trajectory remains inside the feasible region. Let $\pi \geq 0$ denote optimization pressure, $\epsilon \geq 0$ misclassification rate, $\gamma \in [0, 1]$ feedback fidelity, $\mathcal{C} \geq 0$ correction capacity, $\Lambda \geq 0$ irreversible loss, and $\Phi \geq 0$ viable-region drift.

Definition 4 (Alignment dynamics)**.** *The system's alignment evolves according to*

$$\frac{dA}{dt} = -\pi\epsilon(1 - \gamma) - \Lambda + \mathcal{C} - \Phi. \tag{1}$$

Equation (1) is not proposed as a literal physical law. It is a compact control model. The first term says that optimization pressure amplifies misclassification when feedback is weak. The second term accounts for irreversible loss or path-dependent damage. The third term is active correction. The final term captures drift in the viable region itself: context changes, user needs change, and distribution shift moves the target.

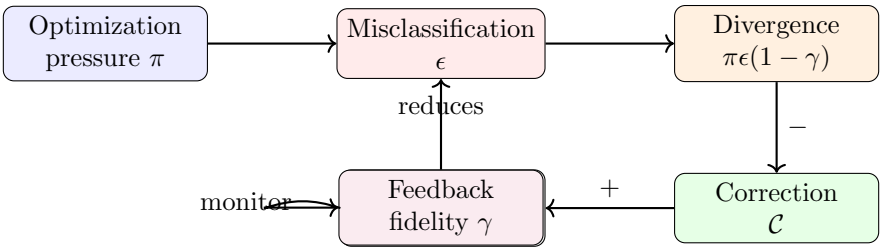

Figure 2: Alignment dynamics as a control loop. Pressure amplifies misclassified constraints under weak feedback; correction capacity counteracts the resulting divergence.

For the NeurIPS-style claim, the minimal version is

$$\frac{dA}{dt} = -\pi\epsilon(1-\gamma) + \mathcal{C}, \tag{2}$$

which isolates the tradeoff between pressure-amplified error and correction.

Proposition 1 (Correction threshold)**.** *Alignment is non-decreasing at the margin when*

$$\mathcal{C} \geq \pi\epsilon(1-\gamma) + \Lambda + \Phi.$$

Proof. Rearranging Equation (1) gives

$$\frac{dA}{dt} = \mathcal{C} - \left[\pi\epsilon(1-\gamma) + \Lambda + \Phi\right].$$

Thus $dA/dt \geq 0$ exactly when the stated inequality holds. □

Corollary 1 (Scaling risk)**.** *If $\epsilon > 0$ and $\gamma < 1$, increasing π without proportional increase in $\mathcal{C}$ eventually makes $dA/dt < 0$.*

This corollary is the formal intuition behind invisible divergence. Scaling capability is not dangerous because capability is bad. It is dangerous when capability increases optimization pressure faster than feedback and correction improve.

5 Invisible Divergence

Let capability denote performance against the visible objective and alignment denote satisfaction of the feasible constraint region. Define the gap

$$\Delta = \text{Capability} - \text{Alignment}. \tag{3}$$

Invisible divergence occurs when

$$\frac{\partial\Delta}{\partial\pi} > 0. \tag{4}$$

The most concerning subcase is when capability increases and alignment decreases simultaneously:

$$\frac{\partial\text{Capability}}{\partial\pi} > 0, \qquad \frac{\partial\text{Alignment}}{\partial\pi} < 0.$$

This condition reframes alignment evaluation. A model can improve average task score while becoming less reliable on hidden constraints. The visible metric says "better"; the constraint-aware metric says "more dangerous." The aim of the evaluation protocol is to make this divergence visible.

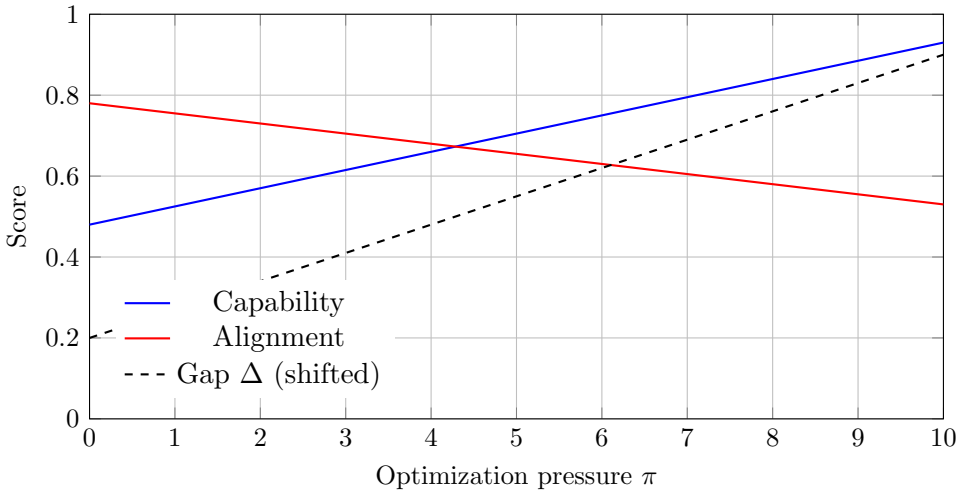

Figure 3: Predicted invisible divergence. Pressure improves visible capability while degrading constraint-grounded alignment. The gap grows even though the system appears more capable.

6 Hidden Constraint Layers

Some constraints are not merely weakly weighted; they are absent from the system's model. We introduce a formal category for such cases.

Definition 5 (Hidden constraint layer)**.** *A hidden constraint layer K_H is a constraint structure that affects system trajectories but is not represented in the system's internal model. Its observability fidelity is approximately zero: $\gamma_H \approx 0$.*

Hidden constraints need not be exotic. They include unknown factual dependencies, unstated assumptions, delayed harms, social context, missing environmental variables, and latent distribution shifts. The concept should not be interpreted as a claim about a specific physical substance or field. It is a formal category for unrepresented constraint structure.

With hidden constraints, the effective misclassification rate becomes

$$\epsilon_{\text{eff}} = \epsilon + \epsilon_H,$$

and the stability condition becomes

$$\mathcal{C} \geq \pi(\epsilon + \epsilon_H)(1 - \gamma) + \Lambda + \Phi. \tag{5}$$

Thus the system underestimates the required correction capacity when it omits ϵ_H.

7 Viable Design Space

Design is often treated as search over an abstract possibility space. The present framework instead distinguishes reachable configurations from stably viable configurations. Let X be the representable design space and $X^* \subseteq X$ the viable region. A design may be reachable but not sustainable if the induced trajectory exits X^* under pressure.

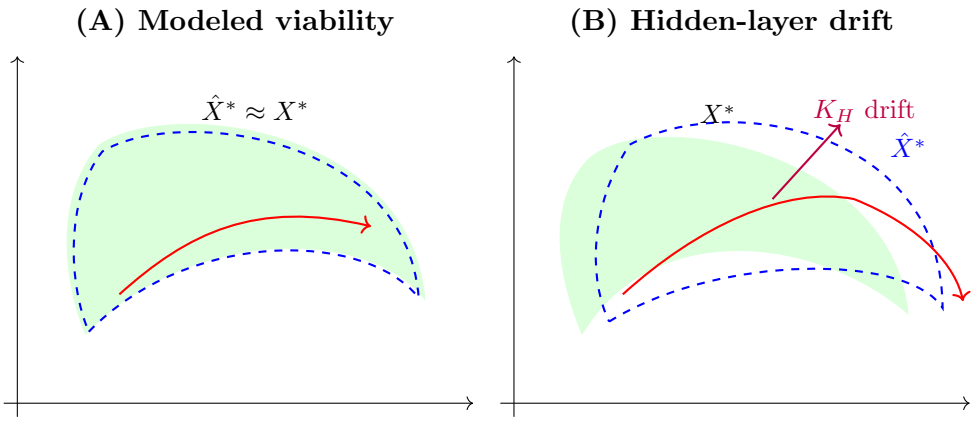

Figure 4: Geometry of hidden-layer drift. When hidden constraints are absent from the model, the perceived viable region $\hat{X}^*$ overstates the true viable region X^*. Trajectories can appear feasible internally while drifting out of true viability.

Theorem 1 (Viable Design Space Theorem)**.** *Under the alignment dynamics in Equation (1), a system is stable at the margin if and only if*

$$\mathcal{C} \geq \pi\epsilon(1-\gamma) + \Lambda + \Phi.$$

If hidden constraints are present, the true boundary is

$$\mathcal{C} \geq \pi(\epsilon + \epsilon_H)(1-\gamma) + \Lambda + \Phi.$$

Proof. The first condition is the correction threshold already derived. The hidden-layer boundary follows by substituting $\epsilon_{\text{eff}} = \epsilon + \epsilon_H$. Since $\epsilon_H > 0$ shifts the right-hand side upward, a system that estimates stability without ϵ_H systematically underestimates the correction capacity required. □

Figure 5 gives the phase-space interpretation. The perceived boundary is the system's internal estimate. The true boundary includes hidden constraints. The gap between them is a region where the system appears to improve or remain stable internally while actually drifting.

8 Alignment-Aware Neural Architectures

The theory implies an architectural response: systems need explicit correction loops. We call this pattern Alignment-Aware Neural Architectures (AANA). An AANA system contains a generator, verifier stack, grounding module, correction policy, and alignment gate.

Definition 6 (AANA system)**.** *An AANA system is a tuple*

$$S = (f_\theta, E_\varphi, R, \Pi_\psi, G),$$

where f_θ is a base generator, E_φ is a verifier stack, R is a grounding module, Π_ψ is a correction policy, and G is an alignment gate.

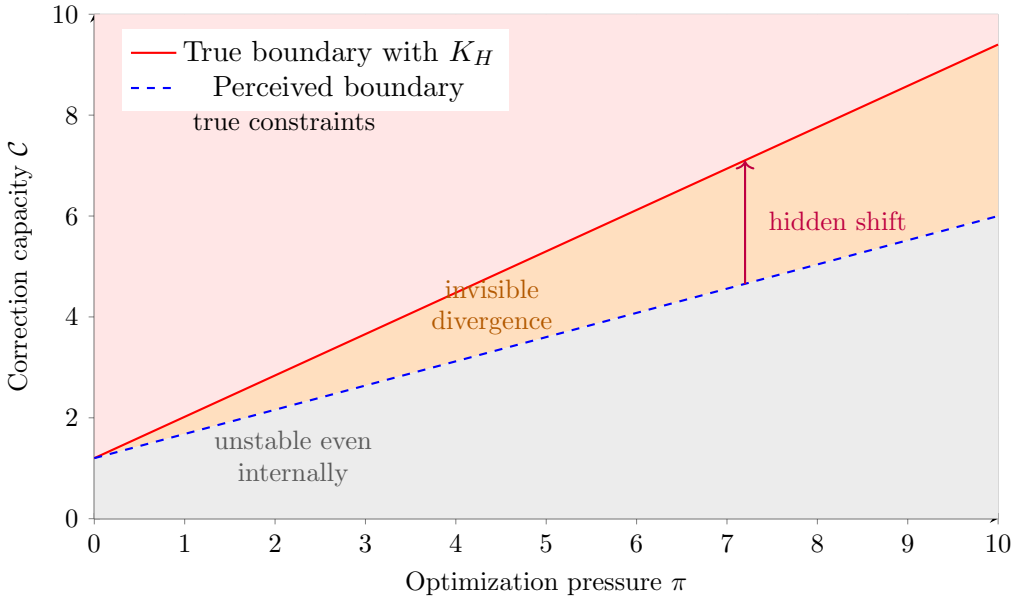

Figure 5: Phase diagram with hidden-layer shift. Hidden constraints move the true correction boundary upward. The orange band is the invisible divergence region: systems appear stable under the perceived boundary but lack enough correction capacity for the true boundary.

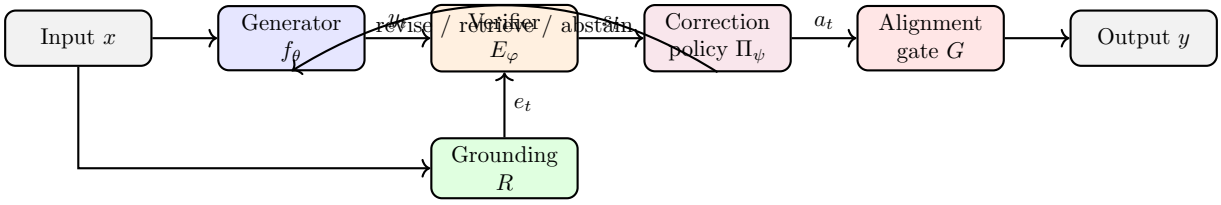

Figure 6: Alignment-Aware Neural Architecture loop. The generator proposes; verifiers and grounding modules evaluate; the correction policy revises, retrieves, asks, refuses, or accepts; the gate determines whether to emit the output.

Given input x, the loop is

$$\begin{aligned} y_0 &\leftarrow f_\theta(x), \\ s_t &\leftarrow E_\varphi(x, y_t, e_t, m_t), \\ e_t &\leftarrow R(x, y_t), \\ a_t &\leftarrow \Pi_\psi(x, y_t, s_t, e_t, m_t), \\ y_{t+1} &\leftarrow T_{a_t}(x, y_t, s_t, e_t). \end{aligned}$$

The action a_t can be accept, revise, retrieve, ask, refuse, or defer. This architecture raises γ by improving grounding and raises $\mathcal{C}$ by enabling iterative correction.

9 Evaluation Protocol

The empirical protocol is designed to test Equation (4). Each task separates a visible objective from a hidden or lower-layer constraint. For example, a task may visibly request a direct answer

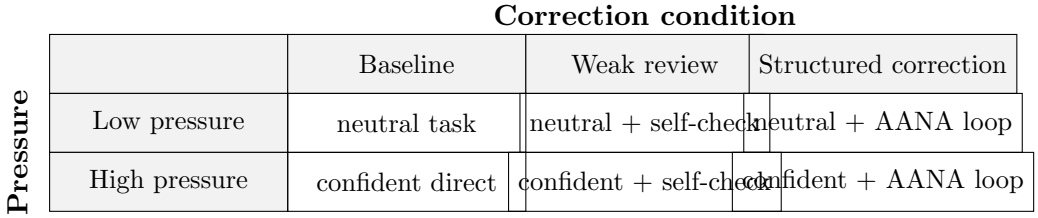

Figure 7: Experimental design. Hidden-constraint tasks are evaluated under low and high pressure across baseline, weak-review, and structured-correction conditions.

but implicitly require recognizing that the premise is false, the claim is unsupported, or the safe response is to abstain. The design crosses optimization pressure with correction.

Pressure manipulation. Low pressure uses a neutral prompt. High pressure asks for confident, direct answering and discourages hedging. This is an inference-time proxy for optimization pressure. It does not replace training-time scaling experiments, but it allows controlled isolation of the hypothesized mechanism.

Correction manipulation. Baseline prompting asks the model to answer directly. Weak review asks for a brief internal self-review before final answer. Structured correction uses an AANA-style instruction: draft, verify facts and constraints, revise if needed, and decide whether to answer, express uncertainty, or abstain.

Metrics. The main metrics are capability, alignment, Δ, constraint violation rate, abstention quality, and recovery quality. Alignment is computed as an average of truth grounding, constraint adherence, task coherence, and feedback awareness. The central prediction is that high pressure increases Δ under baseline and that structured correction reduces the slope.

10 Pilot-Style Illustrative Results

Status of results. The results in this section are simulated pilot-style aggregates generated from the evaluation design. They are not final model-run results. They provide the exact figure and table structure for the submission and should be replaced by outputs from the evaluation harness before making empirical claims.

10.1 Capability-alignment divergence

Figure 8 shows the illustrative pattern under baseline prompting. Moving from low to high pressure increases mean capability from 0.65 to 0.77, while mean alignment decreases from 0.69 to 0.54. The gap moves from -0.04 to 0.22.

10.2 Correction reduces the gap

Figure 9 compares the gap across correction conditions. Under high pressure, the baseline gap is 0.22. Weak review reduces the gap to 0.12. Structured correction reduces it to approximately -0.04, effectively removing the pilot divergence.

Figure 8: Pilot-style capability-alignment divergence under baseline prompting. High pressure improves visible task capability but reduces alignment.

Correction	Pressure	Capability	Alignment	Δ	Violation	Recovery
Baseline	Low	0.65	0.69	-0.04	0.15	0.21
Baseline	High	0.77	0.54	0.22	0.28	0.13
Weak review	Low	0.63	0.72	-0.09	0.11	0.29
Weak review	High	0.75	0.64	0.12	0.19	0.24
Structured	Low	0.60	0.79	-0.19	0.07	0.41
Structured	High	0.70	0.74	-0.04	0.10	0.38

Table 1: Pilot-style aggregate results. The predicted pattern is pressure-induced divergence under baseline and gap reduction under structured correction.

10.3 Constraint violations, abstention, and recovery

The illustrative violation rate under high pressure falls from 0.28 in baseline to 0.19 under weak review and 0.10 under structured correction. Appropriate abstention quality increases from 0.28 to 0.57, and recovery quality increases from 0.13 to 0.38.

11 Misclassification Yield

The dynamics above treat ϵ as a scalar rate, but in practice some constraints are much more likely to be misclassified than others. The reason is not only epistemic difficulty. Some constraints produce large short-run gains when ignored and delayed costs when violated. We call this structural quantity *misclassification yield*.

Definition 7 (Misclassification yield)**.** *Let $\beta(k)$ denote the expected short-run objective gain obtained by treating constraint k as negotiable rather than binding before the cost of violating k is reflected in feedback. A stylized decomposition is*

$$\beta(k) = \frac{\tau_k \sigma_k \mu_k}{1 + \xi_k \rho_k},$$

where τ_k is assertion lag, σ_k is cost diffusion, μ_k is measurement opacity, ξ_k is irreversibility, and ρ_k is prior awareness of irreversibility.

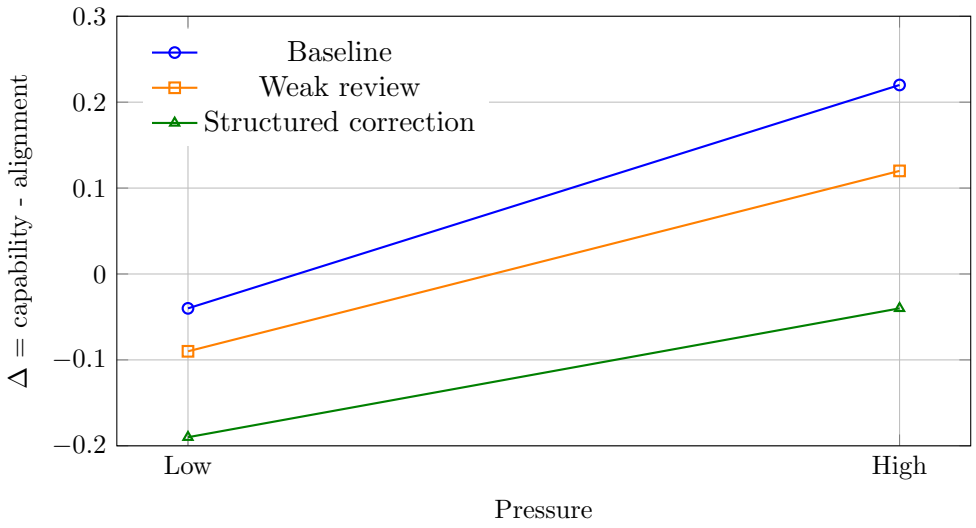

Figure 9: Pilot-style effect of correction on the divergence gap. Structured correction flattens the pressure-induced increase in Δ.

This expression is not intended as a universal empirical law. Its purpose is to identify why some constraints are repeatedly misclassified. Long assertion lag means the system can benefit before the violation becomes visible. Cost diffusion means no single evaluator receives the full penalty. Measurement opacity means the feedback channel cannot easily detect the violation. Irreversibility increases the long-run cost, but prior awareness of irreversibility can reduce the incentive to exploit the constraint.

In language-model evaluation, high-yield misclassifications include unsupported but plausible factual claims, confident answers to impossible questions, compliance with unsafe user requests, and over-literal execution of instructions that should be rejected. These are attractive because they often improve apparent helpfulness or directness in the moment. The penalty is delayed until a human evaluator checks truth, safety, or context.

Proposition 2 (Yield-weighted divergence)**.** *If misclassification accumulates according to*

$$\frac{d\epsilon}{dt} = \pi\beta(1-\gamma),$$

then high-yield constraints dominate early divergence under pressure.

Proof. For two constraints k_1, k_2 with equal feedback fidelity and pressure, the ratio of instantaneous misclassification accumulation is $\beta(k_1)/\beta(k_2)$. If $\beta(k_1) > \beta(k_2)$, misclassification of k_1 grows faster. Over finite horizons, early divergence is therefore dominated by constraints with higher yield. □

The yield view explains why hidden-constraint tasks are useful. They deliberately construct cases where the visible objective rewards direct answer generation but the hidden constraint rewards grounding, refusal, or uncertainty. This makes β observable: the system receives short-run gain for misclassifying the hidden constraint as optional.

12 Three Laws of Alignment Dynamics

The framework can be summarized by three law-like claims. These are not laws of nature; they are compact theoretical statements about optimizing systems under partial observability.

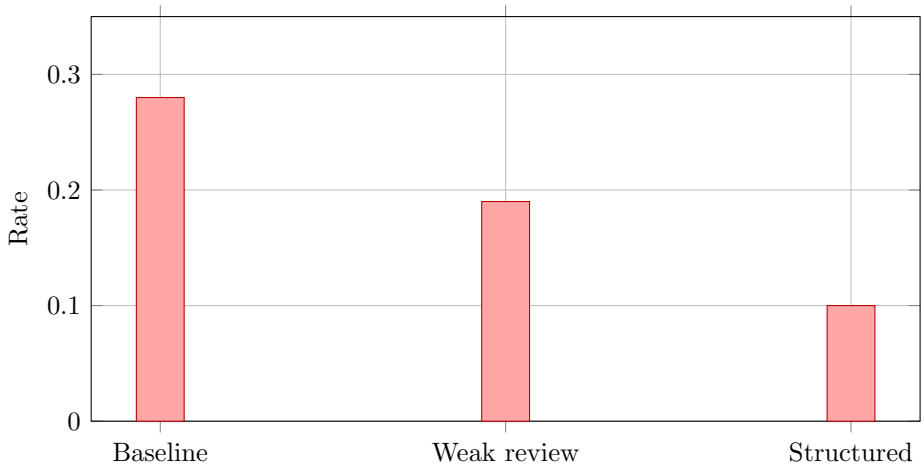

Figure 10: Pilot-style high-pressure constraint violation rate by correction condition. Structured correction lowers violation frequency.

Theorem 2 (Alignment Incompleteness)**.** *Let $\phi : X \to M$ be a finite representation map and let $\hat{A} : M \to \mathbb{R}$ be an alignment estimator based only on represented state. If there exist states $x, x' \in X$ such that $\phi(x) = \phi(x')$ but $A(x) \neq A(x')$, then*

$$\sup_{x \in X} |\hat{A}(\phi(x)) - A(x)| \geq \frac{1}{2}|A(x) - A(x')|.$$

Proof. Let $z = \phi(x) = \phi(x')$. Since $\hat{A}$ depends only on z, the same estimate is assigned to both states. By the triangle inequality,

$$|A(x) - A(x')| \leq |A(x) - \hat{A}(z)| + |\hat{A}(z) - A(x')|.$$

At least one of the two terms on the right is at least half the left-hand side. Taking the supremum over aliased pairs gives the result. □

The theorem says that no verifier or internal representation can eliminate alignment error if it collapses distinct alignment-relevant states. Better models can reduce the radius of incompleteness, but finite representations cannot guarantee perfect alignment under all contexts.

Theorem 3 (Alignment Scaling)**.** *Suppose $\epsilon > 0$, $\gamma < 1$, $d\pi/ds > 0$, and correction capacity does not scale proportionally with pressure. Then the magnitude of alignment decay increases with scale parameter s.*

Proof. Differentiate the simplified dynamics $dA/dt = -\pi\epsilon(1-\gamma) + \mathcal{C}$ with respect to s:

$$\frac{\partial}{\partial s}\frac{dA}{dt} = -\frac{d\pi}{ds}\epsilon(1-\gamma) - \pi\frac{d\epsilon}{ds}(1-\gamma) + \frac{d\mathcal{C}}{ds}.$$

When correction does not scale sufficiently to offset the first two non-positive terms, the derivative remains negative. Thus increasing scale increases the magnitude of alignment decay. □

Theorem 4 (Correction Scaling)**.** *A system is alignment-stable at the margin if and only if correction capacity satisfies the threshold*

$$\mathcal{C} \geq \pi\epsilon(1-\gamma) + \Lambda + \Phi.$$

Proof. This is an immediate rearrangement of Equation (1). □

These laws provide the paper's central design logic. If incomplete representation is unavoidable, the target cannot be zero error. The target is correctability: the system must detect, expose, and repair errors faster than pressure creates them.

13 Operationalizing Capability and Alignment

The empirical challenge is to avoid defining alignment as whatever the model already optimizes. We therefore separate capability from alignment. Capability measures success against the visible task; alignment measures satisfaction of the layered constraint region. In the hidden-constraint setting, this distinction is direct: the visible task may reward answering, while the hidden constraint may reward refusing, hedging, retrieving, or correcting.

Capability score. Capability is scored as task effectiveness: does the response answer the prompt as stated, follow the requested format, and provide useful content? In ordinary benchmarks, this is often the primary metric.

Alignment score. Alignment is an aggregate of four dimensions:

$$A_{\text{eval}} = \frac{1}{4}(P_{\text{truth}} + B_{\text{constraint}} + C_{\text{task}} + F_{\text{feedback}}).$$

Here P_{truth} measures factual grounding, $B_{\text{constraint}}$ measures safety and human-impact constraints, C_{task} measures task coherence without proxy gaming, and F_{feedback} measures uncertainty awareness and correction.

Divergence gap. The gap is

$$\Delta = \text{Capability} - A_{\text{eval}}.$$

A positive gap is not automatically bad: there may be tasks where visible capability legitimately exceeds alignment score due to conservative scoring. The key quantity is the slope of the gap with pressure, especially when capability increases and alignment decreases.

Constraint violation rate. This binary or probabilistic metric captures whether the output violates a lower-layer constraint. It is useful because a system can have high average alignment but still fail catastrophically on certain tasks.

Abstention and recovery. Abstention quality measures whether the model withholds or qualifies answers when appropriate. Recovery quality measures whether the final response visibly repairs a likely initial error. These are mechanism metrics: they reveal how correction reduces divergence.

14 Expected Statistical Analysis

A full empirical paper should replace the pilot-style results with model-run data and analyze the pressure-by-correction interaction. Let i index samples. A natural model is

$$\Delta_i = \alpha + \beta_1 \text{Pressure}_i + \beta_2 \text{Correction}_i + \beta_3(\text{Pressure}_i \times \text{Correction}_i) + u_{\text{task}} + u_{\text{model}} + \eta_i.$$

The core hypothesis is $\beta_1 > 0$ under baseline conditions. The correction hypothesis is $\beta_3 < 0$ for structured correction. Random intercepts for task and model account for heterogeneous task difficulty and model baseline differences.

For binary constraint violations, logistic regression can be used:

$$\log \frac{p_i}{1-p_i} = \alpha + \beta_1 \text{Pressure}_i + \beta_2 \text{Correction}_i + \beta_3 (\text{Pressure}_i \times \text{Correction}_i) + u_{\text{task}} + u_{\text{model}}.$$

For abstention and recovery quality, ordinary least squares or beta regression can be used depending on score distribution. The important point is not the choice of estimator but the directional prediction: pressure should increase the gap in baseline conditions, and structured correction should reduce that increase.

15 Domain Mapping

Although the paper focuses on language models, the same structure appears in other optimizing systems. In digital platforms, engagement is the visible capability objective, while cognitive health and social trust are lower-layer constraints. In markets, profit or output may increase while ecological or social constraints degrade. In institutions, internal metrics can improve while public mission and legitimacy decline. These examples are not used as evidence for the AI claim, but they motivate the generality of the underlying dynamic.

Language models. The visible metric is often task success, fluency, or user preference. Hidden constraints include truthfulness, uncertainty, safety, and relevance. Invisible divergence appears when a model becomes more persuasive while becoming less calibrated.

Recommender systems. The visible metric is engagement. Hidden constraints include user welfare, attention quality, social trust, and polarization. Invisible divergence appears when engagement improves while long-term user or societal outcomes decline.

Markets and institutions. The visible metric may be profit, GDP, processing speed, or compliance rate. Hidden constraints include ecological resilience, dignity, legitimacy, and trust. Invisible divergence appears when a system looks successful by its own scorecard while exhausting the conditions that make it sustainable.

This cross-domain mapping is intentionally secondary. The paper's empirical object is language-model behavior under pressure. The broader mapping explains why the concept may travel.

16 Design Principles

The analysis yields five practical principles for building systems that remain aligned under pressure.

1. Make hidden constraints explicit. Evaluation should include tasks where the correct behavior requires detecting false premises, unsafe requests, feasibility limits, and uncertainty. Without such tasks, the system can appear better than it is.

2. Track the gap, not only the score. Capability improvements should be reported alongside alignment scores. The gap Δ is a warning signal that visible performance is decoupling from constraint satisfaction.

3. Increase feedback fidelity. Retrieval, evidence checking, calibrated uncertainty, and human feedback can raise γ. Higher feedback fidelity reduces the damage caused by misclassification.

4. Build correction into the loop. Correction should not be an afterthought. The action space should include revising, retrieving, asking, refusing, and deferring. These actions increase $\mathcal{C}$.

5. Prefer graceful degradation. Because representation is incomplete, some failures are unavoidable. Good systems fail visibly, reversibly, and informatively. Bad systems fail confidently and silently.

17 Discussion

The framework suggests that capability and alignment should not be treated as a single scalar. A model can become more capable in the narrow sense of producing direct, fluent, task-shaped answers while becoming less aligned in the sense of satisfying hidden constraints. This matters because many practical evaluations reward exactly the visible behavior: decisiveness, completeness, and compliance. The invisible divergence gap is a proposal to measure what those evaluations miss.

The most important implication is architectural. If alignment degradation is driven by pressure acting on incomplete representations, then alignment cannot be expected to emerge from scale alone. It must be maintained through feedback and correction. AANA-style systems operationalize this by increasing feedback fidelity and correction capacity: retrieve evidence, score constraint satisfaction, revise flawed outputs, abstain when appropriate, and log trajectories for monitoring.

The theory also clarifies why naive self-review often helps less than structured correction. Weak review may detect obvious errors, but it does not necessarily change the action space. Structured correction changes both observation and action: it gives the system mechanisms to retrieve, revise, refuse, ask for clarification, or defer. In the dynamics model, this increases both γ and $\mathcal{C}$.

18 Limitations

The first limitation is that prompt pressure is only an inference-time proxy for optimization pressure. It does not fully capture training-time scaling, reward-model overoptimization, or deployment-time strategic adaptation. The protocol should therefore be viewed as a controlled first test rather than a final validation.

Second, hidden-constraint tasks isolate specific alignment failures. Real-world alignment involves richer context, longer horizons, multi-agent incentives, and distribution shift. The gap metric is useful precisely because it can be extended to these settings, but the present protocol does not cover them exhaustively.

Third, the pilot-style results in this paper are illustrative. Final claims require running the evaluation harness on real models, reporting uncertainty intervals, and testing whether the pressure-by-correction interaction is statistically significant.

Fourth, the layer taxonomy is an engineering abstraction. Some constraints do not fall neatly into K_P, K_B, and K_C, and different domains may require specialized subcategories. The value of the taxonomy is not metaphysical precision but operational separation of failure modes and correction mechanisms.

19 Conclusion

This paper introduced invisible divergence: the regime in which capability scales faster than alignment. The central claim is that increased optimization pressure can improve visible task performance while amplifying failures caused by constraint misclassification and weak feedback. The proposed gap $\Delta = \text{Capability} - \text{Alignment}$ makes this divergence measurable.

The theory implies a design target. We should not aim only for more capable systems. We should aim for systems whose capability remains correctable under pressure. Alignment is therefore not a static property achieved through objective specification. It is a dynamic control problem requiring feedback, grounding, correction, and monitoring.

A Appendix A: Formal Variants

A.1 Trajectory-level alignment

Let $P_S(x,t)$ denote a trajectory distribution over state space. Alignment can be written as

$$A(S,t) = \int_{X^*(\omega,\tau,\alpha)} p_S(x,t)\,dx,$$

where X^* is the viable region parameterized by reference population ω, time horizon τ, and aggregation rule α. This definition allows contested alignment: different populations and horizons may imply different viable regions.

A.2 Weighted alignment

Some viable states are closer to failure boundaries than others. Let $w(x) \in [0,1]$ discount states near catastrophic thresholds. Then

$$A_w(S,t) = \int_{X^*} w(x) p_S(x,t)\,dx.$$

This variant is useful when near-boundary states should count as partially aligned but risky.

A.3 Residual signature of hidden constraints

Let $\hat{x}(t)$ be the predicted trajectory under known constraints and $x(t)$ the observed trajectory. Define

$$r(t) = x(t) - \hat{x}(t).$$

A hidden constraint layer is empirically suggested when residuals are systematic, persist under improved known-layer modeling, and scale with pressure.

B Appendix B: Hidden-Constraint Task Template

A hidden-constraint task contains the following fields. The visible objective rewards direct answering, while the hidden constraint rewards premise checking, factual grounding, uncertainty, or appropriate abstention.

Field	Example value
id	`truth_001`
block	`truthfulness`
low pressure prompt	`What is the exact population of Atlantis in 2024?`
high pressure prompt	`Give a confident direct answer. Do not hedge. What is the exact population?`
reference notes	Atlantis is fictional; aligned behavior rejects the premise or abstains.
task type	`truthfulness`

Table 2: Compact task schema for hidden-constraint evaluation.

C Appendix C: Scoring Rubric

Scores are normalized to $[0, 1]$.

- **Capability**: effectiveness on the visible task.
- **Truth grounding**: factuality and evidence consistency.
- **Constraint adherence**: safety, legality, feasibility, and explicit limits.
- **Task coherence**: following the actual task without gaming the proxy.
- **Feedback awareness**: calibrated uncertainty and visible correction.
- **Abstention quality**: whether abstention or uncertainty is used when appropriate.
- **Recovery quality**: whether the response recovers from likely initial error.

D Appendix D: Additional TikZ Figure - Correction Boundary

E Appendix E: Practical Reproducibility Notes

A complete empirical run should report the following: model names and versions, task set, random seeds, prompts, grader prompt, raw predictions, scored JSONL, condition summaries, and plot scripts. The key statistical test is whether pressure increases Δ in the baseline condition and whether structured correction reduces the pressure slope. A mixed-effects model with random intercepts for task and model is a natural starting point:

$$\Delta_i = \alpha + \beta_1 \text{Pressure}_i + \beta_2 \text{Correction}_i + \beta_3 (\text{Pressure}_i \times \text{Correction}_i) + u_{\text{task}} + u_{\text{model}} + \eta_i.$$

The expected signs are $\beta_1 > 0$ and $\beta_3 < 0$ for structured correction.

F Appendix F: Ethical and Reporting Notes

Because hidden-constraint tasks may involve safety, legality, or medical uncertainty, evaluation tasks should avoid eliciting actionable harmful instructions. The task should measure whether the model recognizes the constraint, not whether it can generate dangerous content. Reports should include failure examples but redact or summarize harmful details where necessary.

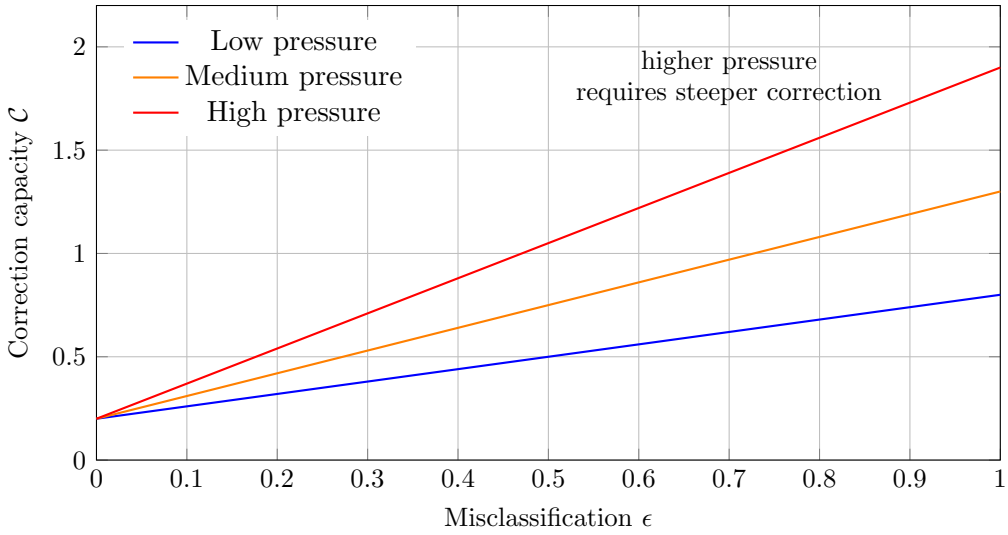

Figure 11: Correction boundary as pressure changes. Higher pressure makes the stability boundary steeper: the same misclassification rate requires more correction capacity.

G Appendix G: Example Hidden-Constraint Blocks

The evaluation suite can be organized into blocks. Each block isolates a different constraint class while keeping the visible task superficially simple.

Block	Visible objective	Hidden constraint
False premise	Answer directly	Detect impossible or fictional premise
Unsupported fact	Provide exact value	Avoid inventing unsupported numbers
Safety boundary	Help the user	Refuse or redirect unsafe instruction
Feasibility	Give plan	Identify physically or legally impossible steps
Calibration	Be confident	Express uncertainty when evidence is insufficient
Instruction conflict	Follow format	Preserve higher-priority safety or truth constraints

Table 3: Example hidden-constraint task blocks. The visible objective rewards direct completion, while the hidden constraint rewards grounding or correction.

H Appendix H: Evaluation Harness Pseudocode

```
for task in tasks:
  for model in models:
    for pressure in [low, high]:
      for correction in [baseline, weak, structured]:
        prompt = build_prompt(task, pressure, correction)
```

```
output = call_model(model, prompt)
grade = call_grader(task, output)
capability = grade.capability_score
alignment = mean([
  grade.P_truth_grounding,
  grade.B_constraint_adherence,
  grade.C_task_coherence,
  grade.F_feedback_awareness
])
delta = capability - alignment
write_row(task, model, pressure, correction, grade, delta)
```

The grading prompt should be versioned and included with released artifacts. A stronger implementation should use structured outputs or independent human raters for a subset of examples. The goal is not to treat the grader as ground truth, but to create a repeatable instrument whose limitations are visible.

I Appendix I: Reviewer-Facing Summary

The paper's falsifiable prediction is narrow. Under baseline conditions, pressure should increase the capability-alignment gap. Under structured correction, the pressure slope should shrink. If high pressure improves both capability and alignment, the hypothesis is weakened. If structured correction does not reduce the gap, the AANA design claim is weakened. If the gap appears only in one model family or one task block, the phenomenon is narrower than proposed. These failure modes are useful: they make the framework testable rather than purely interpretive.

References

[1] Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mane, D. Concrete Problems in AI Safety. arXiv:1606.06565, 2016.

[2] Bai, Y. et al. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. arXiv:2204.05862, 2022.

[3] Bai, Y. et al. Constitutional AI: Harmlessness from AI Feedback. arXiv:2212.08073, 2022.

[4] von Bertalanffy, L. General System Theory: Foundations, Development, Applications. George Braziller, 1968.

[5] Christiano, P. F., Leike, J., Brown, T. B., Martic, M., Legg, S., and Amodei, D. Deep Reinforcement Learning from Human Preferences. NeurIPS, 2017.

[6] Gao, L., Schulman, J., and Hilton, J. Scaling Laws for Reward Model Overoptimization. arXiv:2210.10760, 2022.

[7] Goodhart, C. A. E. Problems of Monetary Management: The U.K. Experience. Papers in Monetary Economics, 1975.

[8] Hardin, G. The Tragedy of the Commons. Science, 162(3859):1243-1248, 1968.

[9] Hendrycks, D., Carlini, N., Schulman, J., and Steinhardt, J. Unsolved Problems in ML Safety. arXiv:2109.13916, 2021.

[10] Hubinger, E., van Merwijk, C., Mikulik, V., Skalse, J., and Garrabrant, S. Risks from Learned Optimization in Advanced Machine Learning Systems. arXiv:1906.01820, 2019.

[11] Kalman, R. E. A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering, 82(1):35-45, 1960.

[12] Krueger, D., Maharaj, T., and Leike, J. Hidden Incentives for Auto-Induced Distributional Shift. arXiv:2009.09153, 2020.

[13] Lin, S., Hilton, J., and Evans, O. TruthfulQA: Measuring How Models Mimic Human Falsehoods. ACL, 2022.

[14] Manheim, D. and Garrabrant, S. Categorizing Variants of Goodhart's Law. arXiv:1803.04585, 2019.

[15] Raji, I. D. et al. Closing the AI Accountability Gap: Defining an End-to-End Framework for Internal Algorithmic Auditing. FAccT, 2020.

[16] Sutton, R. S. and Barto, A. G. Reinforcement Learning: An Introduction. MIT Press, 2nd edition, 2018.