Strategic Evaluation: Incentivizing AI Capability Coverage with Private Benchmarks

Sang Truong[1], **Serena Wang**[2], **Nick Haber**[1], **Sanmi Koyejo**[1]
[1]Stanford [2]Harvard

Abstract

Public benchmarks play a significant role in steering LLM development, as market incentives drive model developers to optimize for leaderboard performance. However, fundamental information limitations mean that evaluators are only able to cover a subset of socially relevant capabilities with their evaluation tasks. On the other hand, model developers often have additional private benchmarks that are unavailable to evaluators. The resulting information asymmetry opens the door to strategic task specialization that can lead to suboptimal social welfare. We propose randomized evaluation mechanisms (effectively private benchmarks) as a way to incentivize model developers to train to cover a broader set of socially relevant tasks, including those unknown to the evaluator. We formalize this as a dynamic game with information asymmetry between an evaluator and a model developer. We prove that randomized evaluation aligns incentives to the developer's prior belief over possible evaluated tasks, and is socially optimal when that belief matches society, but information leakage degrades this over rounds. However, if the evaluator can continually update their known task set, alignment can be asymptotically recovered. We illustrate a variance–leakage–correction tradeoff in semisynthetic experiment with a latent factor structure over MMLU-Pro.

1 Introduction

Public benchmarks shape LLM development: leaderboard performance translates directly into market positioning, investment, and regulatory attention. As benchmarks grow in public influence, model developers face strong economic incentives adapt their training to do well on what gets measured.

The classical failure mode—contamination of specific test items into training data—is well studied [25, 1, 30]. But a subtler failure mode operates at the *task* level. Benchmarks cover only a subset of deployment-relevant capabilities: they lack linguistic and cultural diversity, overrely on narrow formats like multiple-choice, and fail to capture the full spectrum of real-world complexity of tasks that users ask models to complete [22, 21]. Even without item-level leakage, a model developer can engage in "training to the test task" [7], where performance on a *task* refers to, e.g., the true population accuracy over a test data distribution, rather than finite-sample accuracy on a specific test set. A model developer who knows the evaluation tasks still has a rational incentive to narrow training to those tasks rather than invest in broad capability coverage. When the evaluation tasks do not cover all socially relevant capabilities (e.g. an unknown dimension of safety), or are distributionally biased relative to actual population relevance (e.g. overindexing on tasks in English), this task-level specialization leads to suboptimal social welfare—an adaptive manifestation *Goodhart's Law* [11, 20] that persists even with perfect contamination controls.

We take the perspective that the capability coverage problem is fundamentally tied to information limitations on part of the evaluator. If an evaluator had enough evaluation tasks to cover all possible socially relevant capabilities, then they could construct evaluation mechanisms in which there is no incentive to game. However, in practice, there is a known information asymmetry across various

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

actors involved in evaluation, from academic labs, to evaluation companies, to model developers [23]. Importantly, model developers often have the resources to develop suites of private benchmarks that never make their way into public benchmarks or leaderboards.

Thus, in a world where evaluators might have *less* information than model developers, we consider a benevolent evaluator's design challenge: *How can an evaluator design a benchmark so that a strategic model developer's best response is to improve performance on the socially relevant task distribution (with full capability coverage), rather than merely on the observable benchmark tasks?* We formalize this as a dynamic game between an evaluator and a model developer.

Our key design lever is randomization. Perhaps counterintuitively, an evaluator can incentivize a model developer to use *more* of their additional private information by revealing *less* of the evaluator's own. We show that if an evaluator strategically witholds information through a randomized evaluation mechanism (effectively creating a private benchmark), the resulting uncertainty for the model developer yields broad coverage incentives.

However, the resulting *variance* in the private benchmark can be undesirable in practice for both leaderboard consistency and user decision making. Extending this to a repeated evaluation game, we quantify the tradeoff between variance and growing incentive-misalignment through information leakage. Furthermore, to combat information leakage over time, we show that the *distribution* of evaluation tasks must not be fixed, but must include an adaptive process with active correction of discovered biases. This reveals a fundamental difference in incentive properties between dynamic benchmarking practices that maintain the same distribution, and those that shift their distribution to cover new capability frontiers. Our main contributions can be summarized as follows:

1. We introduce an **evaluation game with information asymmetry** between an evaluator and model developer that accounts for unawareness of tasks and the possibility of task discovery.
2. We show that one-shot **randomized evaluation yields bounded social regret** when developer beliefs approximate social relevance.
3. Extending this to repeated evaluation, we show that there is a **trade-off between information leakage and variance**. We show that to recover incentive alignment, an evaluator must **adaptively expand their task coverage using external feedback**.
4. We instantiate our model with a **low rank latent factor structure**, and empirically stress-test our theory using a **semisynthetic experiment** based on MMLU-Pro.

1.1 Related work.

Our game theoretic setup draws from contract theory and mechanism design. We situate our approach in the context of existing theory and practice to combat strategic behavior in AI evaluation.

Benchmark gaming. A growing body of work documents the prevalence and impact of benchmark contamination in LLM evaluation [25, 1, 30]. The coverage problem that we consider persists even without contamination. We consider model developer behavior which is closer to "training on the test task" [7], and focus on the evaluator's design choices under information asymmetry. Laufer et al. [16] have modeled a "strategic evaluation" setup in which an evaluator chooses an evaluation mechanism strategically to serve their own ends. Our setup is similar in spirit, but we model the evaluator's objective as maximizing coverage of socially relevant tasks. From a differential privacy angle, Blum and Hardt [5] apply reusable-holdout techniques to maintain leaderboard accuracy in ML competitions (e.g., Kaggle). The threat model differs from ours: adaptive data analysis prevents overfitting to a fixed *dataset*, whereas we consider a setting when the model developer learns an evaluation *distribution* and specializes accordingly. Other recent work has modeled the strategic behavior of data curators in the evaluation process [24, 19]. Instead of focusing on incentivizing truthful data curation, we focus on the model developer's model release strategy.

Dynamic benchmarks. Several efforts tackle benchmark gaming through dynamic evaluation [e.g. 14, 27]. For instance, LiveBench [29] updates tasks monthly using recent information sources to resist contamination, and LMArena [31, 17] uses crowdsourced human judgments, but is vulnerable to biases in the engaged user population. Our analysis helps explain both the short-term success of methods like these and why some dynamic approaches still face long-term limitations.

Randomization in contract theory and strategic classification. Work in contract theory and strategic classification has explored how randomness and opacity in evaluation can mitigate strategic

manipulation [8, 6, 12, 15, 4]. In contract theory, Ederer et al. [8] model *strategic opacity* as a setting in which a principal withholds "the weights on performance indicators that are used to determine rewards." In contrast, our model incorporates information asymmetry assumptions where the evaluator and model developer might each be unaware of some tasks. In strategic classification, the value of randomness has been shown to depend on the extent to which agents tend to *game* or *improve* [6, 3, 10]. Our setting differs in that agents are always improving on measured tasks, but there are still tradeoffs between randomness and transparency for a coverage objective.

2 The Evaluation Game

We introduce a game-theoretic framework that captures the information asymmetry and strategic interaction between evaluators and model developers. We ask, *under what conditions can an evaluation mechanism incentivize model developers to improve performance on all socially relevant tasks, rather than merely optimizing for the published tasks?* All proofs are in the Appendix.

Let Θ denote the space of all possible models (e.g., neural network weights). Let $f : \Theta \to \mathcal{Y}$, with $\mathcal{Y} \subseteq \mathbb{R}$, be a *task*, which is an instrument to measure the performance of a model θ. For a given model $\theta \in \Theta$, $f(\theta)$ represents model performance on instrument f, with larger values indicating better performance, and f is assumed to be bounded over Θ. A task can consist of a single prompt, an aggregation function over a finite benchmark dataset, or an expectation over a benchmark dataset. The universe of all possible tasks is the set $\mathcal{F}$, which we assume to be finite.

The evaluator's utility is defined to be $u_E(\theta) = \sum_{f \in \mathcal{F}} f(\theta)\mu(f) = \mathbb{E}_{f\sim\mu}[f(\theta)]$, where $\mu(f)$ represents social relevance (e.g., the probability that task f is encountered in the general population). This represents the aggregate performance across all tasks that matter in the real world. We interpret the evaluator as acting in society's interest, seeking models that perform well broadly rather than narrowly, with high performance on the full task distribution.

Not all tasks are known to either party. We define *awareness sets* $\mathcal{F}_M \subseteq \mathcal{F}$ and $\mathcal{F}_E \subseteq \mathcal{F}$ as the subsets of the task universe that the model developer and evaluator, respectively, know exist (Figure 1). These may overlap partially, and neither needs equal $\mathcal{F}$—there may be tasks that neither party is aware of. Furthermore, we assume that both the evaluator and the model developer are only able to actually evaluate a subset F_E, F_M of tasks, respectively. In practice, $\mathcal{F}_E$ ($\mathcal{F}_M$) represents the set of tasks that the evaluator (model developer) is hypothetically aware of, while F_E (F_M) represents the set of realized benchmarks with concrete data that can actually be computed.

Assumption 2.1 (Limited-information evaluator and model developer)**.** *The evaluator (model developer) can draw a set of tasks F_E (F_M) from a sampling oracle supported on their awareness set $\mathcal{F}_E \subseteq \mathcal{F}$ ($\mathcal{F}_M \subseteq \mathcal{F}$), but does not know the sampling probabilities. Let $\pi_E(f) = P(f \in F_E)$ ($\pi_M(f) = P(f \in F_M)$) denote the probability that task f is sampled. When f is sampled, the evaluator (model developer) gains the ability to evaluate f, and also learns $\mu(f)$.*

The key challenge is that the evaluator cannot directly measure performance on tasks outside F_E, yet wants to incentivize good performance on all of $\mathcal{F}$—even unknown tasks. This information asymmetry is central to the emergence of misalignment with social welfare: the evaluator must use a limited, known task set to elicit broad performance. We now show formally how this information asymmetry can lead to suboptimal social welfare. In doing so, we also show how alignment can be recovered when the evaluator is strategic about how much information they reveal to the model developer, and in essence designing private benchmarks. We formalize the interaction between the evaluator and the model developer as a Stackelberg game.

Definition 2.2 (Evaluation Game)**.** *The game proceeds in three sequential stages.*

1. *The **evaluator** selects and publishes a reward function $r : \Theta \times \mathcal{P}(\mathcal{F}) \to \mathbb{R}$ and a sampling mechanism $S : \mathcal{P}(\mathcal{F}) \times \Omega \to \mathcal{P}(\mathcal{F})$, which may include randomness $\omega \in \Omega$.*
2. *The **model developer** observes the functions r and S and selects a model $\theta^M \in \Theta$.*
3. *Randomness ω is realized and payoffs are posted. The model developer's payoff is the published score $r(\theta^M, S(F_E, \omega))$, and the evaluator's payoff is $u_E(\theta^M) = \mathbb{E}_{f\sim\mu}[f(\theta^M)]$.*

The key design choice that the evaluator faces is the choice of sampling mechanism S. For instance, a deterministic sampling mechanism would simply ignore any randomness ω, and directly compute the reward based on some fixed set $S(F_E, \omega) = S_E$. In this case, the model developer would know

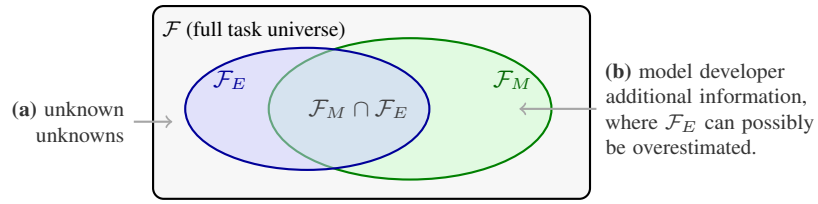

Figure 1: Information structure of the evaluation game. The model developer's awareness set ($\mathcal{F}_M$, green) and the evaluator's awareness set ($\mathcal{F}_E$, blue) are subsets of the full task universe $\mathcal{F}$. **(a)** Tasks in the grey region of $\mathcal{F}$ outside $\mathcal{F}_M \cup \mathcal{F}_E$ represents tasks that are unknown unknowns to both parties, but are still socially relevant. **(b)** $\mathcal{F}_M \setminus \mathcal{F}_E$ consists of tasks that the model developer is aware of, but the evaluator is not (e.g. private safety measures). However, the model developer crucially does not *know* $\mathcal{F}_E$, and may overestimate $\mathcal{F}_E$ to cover this. Thus, with the right randomized mechanism, the evaluator may be able to incentivize the model developer to still optimize for these tasks.

this set S_E at step 2, which we show necessarily leads to suboptimal social utility. This determinstic mechanism is analogous to a leaderboard that publishes a fixed set of tasks (e.g. HELM [18]).

Theorem 2.3 (Failure of Published Deterministic Mechanisms)**.** *If $\mathcal{F}_E \subset \mathcal{F}$, then no deterministic mechanism can recover the optimal utility: for any non-degenerate r and deterministic S, there always exists $\mathcal{F}, \mu$ such that $u_E(\theta^M) < \max_{\theta \in \Theta} u_E(\theta)$.*

No deterministic mechanism can incentivize the model developer to optimize for tasks outside of the chosen set S_E. This is significant because any deterministic mechanism thus limits the evaluator to only be able to incentivize performance on tasks that they are explicitly aware of, and it is impossible for the evaluator to incentivize the model developer to optimize for tasks outside of $\mathcal{F}_E$. This failure motivates randomization, whose efficacy depends on the information asymmetries described above. When the model developer faces uncertainty about which tasks will be evaluated, the strategic landscape changes fundamentally; we next consider the performance of randomized mechanisms.

When the sampling mechanism is randomized, it now becomes relevant to define the uncertainty of the model developer. As traditionally done in mechanism design, we model the model developer as a Bayesian agent. Specifically, since the model developer does not know the exact set of tasks F_E held by the evaluator, they now have to rely on a prior over what tasks could be contained in F_E.

Assumption 2.4 (Bayesian model developer)**.** *The model developer holds a prior belief that the evaluator draws a set of n_E tasks F_E with inclusion probability $\tilde{\pi}_E(f) = P(f \in F_E)$ supported on $\mathcal{F}_M$. The model developer maximizes its expected reward over this belief.*

We next show that a simple randomized reward mechanism is sufficient to significantly improve the evaluator's final utility and social welfare.

Definition 2.5 (Single sample mechanism)**.** *$S(F_E, \omega)$ samples a single f_E^ω uniformly from F_E, and the reward is $r(\theta, S(F_E, \omega)) = f_E^\omega(\theta)$.*

Proposition 2.6 (Single sample mechanism under general beliefs)**.** *Under a single sample mechanism, the Bayesian model developer's best response is $\theta^M \in \arg\max_{\theta \in \Theta} \mathbb{E}_{f \sim \tilde{\pi}_E}[f(\theta)]$.*

Theorem 2.7 (Single sample alignment)**.** *Under the single sample mechanism, the suboptimality for the evaluator is bounded by the total variation distance of the model developer's prior from μ: $\max_{\theta \in \Theta} u_E(\theta) - u_E(\theta^M) \leq 4B \cdot \mathrm{TV}(\mu, \tilde{\pi}_E)$, where $B = \sup_{f \in \mathcal{F}; a,b \in \Theta} |f(a) - f(b)|$.*

Theorem 2.7 directly bounds the misalignment as a function of the amount of information that the model developer has. In the extreme, if the model developer knows of all tasks $\mathcal{F}$, and has a prior $\tilde{\pi}_E$ that matches the true social relevance μ, then θ^M is optimal for u_E. Intuitively, one can think of a model developer with a correct prior $\tilde{\pi}_E$ as a model developer who is very well-informed, and believes the evaluator to be equally well-informed. This leads to the core insight: when the model developer cannot predict which tasks will appear on the evaluation, gaming becomes irrational. Investing effort to overfit to a narrow subset provides no advantage if that subset might not be tested. The model developer's optimal strategy shifts from "optimize for the test" to "optimize broadly," because broad capability is the only reliable way to score well under uncertainty.

In practice, model developers have recently tended to be well-informed and well-resourced when it comes to measuring new capabilities.[1] In a less extreme setting, as long as the model developer assigns positive marginal inclusion weight $\tilde{\pi}_E$ to tasks outside the evaluator's possible determinstic evaluation sets, randomized evaluation directly incentivizes a strictly larger set of tasks.

Corollary 2.8 (Single-sample mechanism strictly broadens incentivization)**.** *If $\mathcal{F}_E \subset \mathcal{F}_M$ and $\tilde{\pi}_E(f) > 0$ for some $f \notin \mathcal{F}_E$, then the single-sample mechanism incentivizes the model developer over a strictly larger set of tasks than any deterministic mechanism.*

2.1 Sampling Limitations for the Model Developer

So far, our results apply when the Bayesian model developer is perfectly rational and can directly choose θ^M to maximize $\mathbb{E}_{f\sim\tilde{\pi}_E}[f(\theta)]$. However, in practice the model developer also faces the same sampling limitations as the evaluator. This impacts the optimality of the model developer's choice, but the basic insights still hold surrounding the value of the randomized mechanism.

Specifically, we extend our model to define the model developer as boundedly rational agent, who can only estimate their population objective $R(\theta) = \mathbb{E}_{f\sim\tilde{\pi}_E}[f(\theta)]$ from finite samples.

Assumption 2.9 (ERM model developer)**.** *The model developer draws i.i.d. samples $F_M = \{f_1, \ldots, f_n\}$ from π_M, and best responds with importance-weighted ERM: $\theta^{F_M} \in \arg\max_\theta \hat{R}_n(\theta)$ where $\hat{R}_n(\theta) := \frac{1}{n}\sum_{i=1}^n \frac{\tilde{\pi}_E(f_i)}{\pi_M(f_i)} f_i(\theta)$.*

The ERM model developer can be thought of as risk-neutral relative to randomness in the ERM estimate. Standard uniform-convergence arguments give an $O(1/\sqrt{n})$ excess-risk bound.

Proposition 2.10 (Sample size improves the generalization guarantee)**.** *When $\tilde{\pi}_E = \pi_M$, the excess risk of θ^{F_M} trained on all n samples satisfies, with probability $\geq 1-\delta$: $R(\theta^*) - R(\theta^{F_M}) \leq O\left(\mathfrak{R}_n(\Theta) + \sqrt{\log(1/\delta)/n}\right)$, where $\mathfrak{R}_n(\Theta)$ is the Rademacher complexity of the function class $\{f \mapsto f(\theta) : \theta \in \Theta\}$.*

A direct consequence is that the model developer's worst-case statistical guarantee is monotone in sample size, giving no *statistical* reason to discard i.i.d. samples. Specifically, for any subset $S \subseteq F_M$ with $|S| = m \leq n$, the analogous bound from Proposition 2.10 holds with $\mathfrak{R}_m(\Theta)$ and $1/\sqrt{m}$ in place of $\mathfrak{R}_n(\Theta)$ and $1/\sqrt{n}$. Thus, larger task samples yield a tighter worst-case guarantee. To move beyond these worst case bounds, we empirically quantify the changes in evaluator utility u_E as a result of the model developer's sampling limitations in simulations in Section 5.

3 Emergent Misalignment via Information Leakage

Theorem 2.7 established that randomized evaluation with a single task achieves incentive alignment when the model developer's belief $\tilde{\pi}_E$ is sufficiently aligned with μ. However, this mechanism has an important practical limitation: the reward signal is noisy, which is not ideal for leaderboard consistency or user decision making. Reducing this noise requires revealing information, which leads to leakage over time in, e.g., weekly leaderboards that degrades the alignment properties established in the previous section. To quantify this fundamental trade-off between incentive-alignment and reward variance, we first formalize a repeated game with information leakage.

Definition 3.1 (Repeated Evaluation Game)**.** *The evaluator publishes a reward function r and a sampling mechanism S. The model developer's initial prior is $\tilde{\pi}_0$. In subsequent rounds $t = 1, 2, \ldots$:*

1. *The model developer chooses θ_t using information I_t (history of observed tasks, initially $\emptyset$).*
2. *The evaluator draws a task set $F_E^{(t)}$ with distribution $\pi_E = P(f \in F_E^{(t)})$ identical across rounds.*
3. *ω is drawn and the evaluator publishes the score $r\left(\theta^t, S\left(\bigcup_{i=1}^t F_E^{(i)}, \omega\right)\right)$.*
4. *$S\left(\bigcup_{i=1}^t F_E^{(i)}, \omega\right)$ is added to I_t. The model developer updates their belief $\tilde{\pi}_t$ according to I_t.*

Definition 3.2 (k-sample mechanism)**.** *$S\left(\bigcup_{i=1}^t F_E^{(i)}, \omega\right)$ samples k tasks uniformly without replacement from $F_E^{(t)}$, yielding S_t^ω. The reward is $r\left(\theta, S\left(\bigcup_{i=1}^t F_E^{(i)}, \omega\right)\right) = \frac{1}{k}\sum_{f\in S_t^\omega} f(\theta)$.*

[1] For instance, OpenAI has invested in developing high quality benchmarks like FrontierMath and FrontierScience.

This multi-task extension of the single-sample mechanism reduces variance and aligns with standard benchmark practice (no task is graded twice within a single test set).

Lemma 3.3 (Variance forces large k)**.** *Let S be a size-k subset of $\mathcal{F}_E$ drawn without replacement, and let $\hat{r}_k(\theta) = \frac{1}{k}\sum_{f\in S} f(\theta)$. If the task scores $\{f(\theta)\}_{f\in\mathcal{F}_E}$ have variance σ^2, then $\mathrm{Var}(\hat{r}_k) = \frac{\sigma^2}{k}\cdot\frac{|\mathcal{F}_E|-k}{|\mathcal{F}_E|-1}$. When $k \ll |\mathcal{F}_E|$, $\mathrm{Var}(\hat{r}_k) \approx \sigma^2/k$.*

While sampling k tasks reduces score variance at a rate of $O(1/k)$, the information leakage from additional samples has a direct consequence for incentive alignment. As the model developer observes sampled tasks across rounds, their belief $\tilde{\pi}_t$ concentrates around the true π_E, drifting from any initial diffuse value (perhaps initially $\tilde{\pi}_0 = \pi_M$). The model developer's best response θ^t correspondingly specializes to high-density regions of π_E, undermining the alignment established in Theorem 2.7. To make this consequence precise, we define the evaluator's loss from the model developer's learning.

Definition 3.4 (Residual Misalignment)**.** *At time t, the residual misalignment is the evaluator's regret from the builder optimizing for $\tilde{\pi}_t$ rather than for u_E directly: $\Delta_t := \max_{\theta\in\Theta} u_E(\theta) - u_E(\theta_t)$.*

Theorem 3.5 (Misalignment growth without correction)**.** *Under the k-sample mechanism (Definition 3.2), suppose $\mathcal{F}_M = \mathcal{F}$ and the model developer has prior $\tilde{\pi}_0$ set to Dir$(\mu(f_1),\ldots,\mu(f_N))$ for $f_1, ..., f_N \in \mathcal{F}$, and applies Bayes updates from the observed tasks with $m_t = kt$ total observations after t rounds. Then $\Delta_t \le 4Bm_t/(m_t + |\mathcal{F}|)$, where $B = \sup_{f\in\mathcal{F};a,b\in\Theta} |f(a) - f(b)|$.*

This bound reveals the role of k: the misalignment at time t is controlled by the total information $m_t = kt$ relative to the size of the task set $|\mathcal{F}|$. When $kt \ll |\mathcal{F}|$, the model developer has too few observations to exploit the evaluation distribution and $\Delta_t \approx 2kt/|\mathcal{F}|$. As $t \to \infty$, $\Delta_t \to 4B$, and the model developer fully learns π_E and can specialize maximally. The evaluator can slow this degradation by choosing a smaller k, at the cost of higher evaluation variance. This is the core tension: privatization works when the builder's belief is diffuse, but repeated evaluation reveals information that shifts the prior $\tilde{\pi}_t$ toward the evaluator's true distribution, enabling strategic specialization. Next, we show how the evaluator's distribution correction can counteract this degradation.

4 Restoring Alignment via Distribution Correction

Section 3 identified the core problem: as the builder observes evaluations over multiple rounds, their estimate $\hat{\pi}_{E,t}$ concentrates around the true π_E. When π_E is biased—supported on a strict subset of $\mathcal{F}$—this enables strategic specialization that degrades benchmark validity. A natural question arises: what if the evaluator also learns and corrects their distribution over time? We now show that if the evaluator updates π_E to correct known biases, the alignment properties of Corollary 2.7 can be asymptotically recovered—even without introducing new tasks into the universe $\mathcal{F}$. We augment the repeated evaluation game with a correction step: before each round, the evaluator updates their sampling distribution $\pi_E^{(t)}$ based on knowledge of their current biases. That is, at each round t:

0. The evaluator updates $\pi_E^{(t)}$ over $\mathcal{F}$ to correct known biases.

1.–3. The game proceeds as in Definition 3.1, with $F_E^{(t)}$ sampled from the updated $\pi_E^{(t)}$.

Identifying and correcting biases in π_E (step 0) is the most demanding requirement. In practice, evaluators learn about their biases through several channels: *(i) incident reports* from users who discover model failures in deployment that existing benchmarks failed to predict, revealing blind spots in π_E; *(ii) systematic audits* where the evaluator reviews coverage across capability dimensions (e.g., languages, modalities, reasoning types) and identifies underrepresented regions; *(iii) A/B testing* where the evaluator compares model rankings under different evaluation subsets to detect sensitivity to task selection; and *(iv) improving benchmark acquisition capacity* over time, as the evaluator develops better tooling for task creation and curation. These mechanisms provide the feedback signal that expands $\mathcal{F}_E$ over time, and drives the correction $\pi_E^{(t)} \to \mu$, where μ is the social-relevance distribution from Section 2.

4.1 Distribution Correction Recovers Alignment

To model these external channels through which evaluators learn over time which capability dimensions they under-represent, we model the evaluator's correction as a gap-targeted process where the

evaluator stochastically identifies specific coverage gaps and fills them locally. This differs from the earlier sampling processes, as we now model growing evaluator awareness.

Assumption 4.1 (Gap-Targeted Gaussian Correction)**.** *Embed tasks in a feature space $\mathcal{Z} \subseteq \mathbb{R}^d$, so each $f \in \mathcal{F}$ has a location $z_f \in \mathcal{Z}$. At each round t, the evaluator:*

1. ***Discovers a gap**: samples a location $\nu_t \in \mathcal{F}$ with probability proportional to the local deficit relative to the social-relevance distribution μ: $P(\nu_t = f \mid \pi_E^{(t-1)}) = Z^{-1}\big[\mu(f) - \pi_E^{(t-1)}(f)\big]_+$ where $Z = \sum_{f'}\big[\mu(f') - \pi_E^{(t-1)}(f')\big]_+$.*
2. ***Fills locally**: updates the evaluation distribution by mixing with a Gaussian centered at the discovered gap: $\pi_E^{(t)} = (1-\rho)\,\pi_E^{(t-1)} + \rho \cdot g_{\sigma_c}(\cdot\,; \nu_t)$, where $g_{\sigma_c}(f; \nu) \propto \exp\!\big(-\|z_f - z_\nu\|^2/2\sigma_c^2\big)$, normalized to a distribution on $\mathcal{F}$.*

The parameters are $\rho \in (0, 1]$ (correction rate) and $\sigma_c > 0$ (correction bandwidth).

The assumption in Step 1 can be exactly recovered in a model of gap discovery mediated by user complaints. We provide a concrete model illustrating this in Appendix C.1.

Theorem 4.2 (Misalignment Under Gap-Targeted Correction)**.** *Under Assumption 4.1, the expected residual misalignment satisfies* $\mathbb{E}[\Delta_t] \;\le\; \min\left(\underbrace{m_t/(m_t + |\mathcal{F}|)}_{\text{estimation-limited}},\; \underbrace{(1-\rho)^t\sqrt{0.5\,D_0} + \varepsilon(\rho, \sigma_c)}_{\text{correction-limited}}\right)$,

where $D_0 := \mathrm{KL}\Big(\pi_E^{(0)} \,\|\, \mu\Big)$ is the initial KL divergence of the evaluator's distribution from the social-relevance distribution μ. $\varepsilon(\rho, \sigma_c)$ measures noise from the stochastic gap-discovery process, satisfying $\varepsilon = O(\sqrt{\rho} \cdot TV(g_{\sigma_c}, \mu))$. This vanishes as $\sigma_c \to \infty$ (corrections become broader) or as $\rho \to 0$ (corrections become rare).

A tension arises as the repeated game proceeds: the model developer is learning $\pi_E^{(t)}$ from observations, while the evaluator is correcting $\pi_E^{(t)}$ toward μ. The residual misalignment Δ_t depends on the relative rates of these two processes. The first term represents the "estimation-limited" phase as t is small, where the model developer has not yet estimated π_E. The second term represents the "correction-limited" phase as $t \to \infty$, where the bound relies on the evaluator's ability to discover new tasks and correct biases at rate ρ.

The crossover occurs at roughly $t^* \approx |\mathcal{F}|/(k + |\mathcal{F}|\rho)$. Crucially, the sample size k appears in Term 1—smaller k slows the model developer's learning, buying the evaluator time to correct their biases.

4.2 Asymptotically Optimal Sample Size Under Distribution Correction

Having seen the trade-offs surrounding the evaluator's choice of how many tasks k to sample each round, we now introduce a closed-form heuristic choice of k that balances variance and misalignment. With k samples per round, $\mathrm{Var}(\hat{r}_k) \approx \sigma^2/k$. Under the correction mechanism (Assumption 4.1), the model developer's effective sample size is $m_t = k \cdot \min(t,\, 1/\rho)$, saturating at k/ρ in steady state since new observations arrive at rate k while old ones age out at rate ρ. The estimation-limited term of Theorem 4.2 therefore gives steady-state exploitation $\Delta_\infty \le k/(\rho|\mathcal{F}| + k)$, regardless of which correction mechanism the evaluator uses. Combining the variance and steady-state exploitation costs gives the *asymptotic per-round loss* $\mathcal{L}(k) := \sigma^2/k + k/(\rho|\mathcal{F}| + k)$, which we minimize to obtain a theoretically backed heuristic for choosing k given an estimate of ρ.

Lemma 4.3 (Closed form for k^*)**.** *If $\rho|\mathcal{F}| > \sigma^2$, then $\mathcal{L}(k)$ attains its minimum at $k^* = \frac{\sigma\,\rho|\mathcal{F}|}{\sqrt{\rho|\mathcal{F}|} - \sigma}$. If $a \le \sigma^2$, $\mathcal{L}$ is strictly decreasing on $(0, \infty)$ and the optimum is at the largest admissible k.*

The closed form gives clean comparative statics. In the high-correction regime $\rho|\mathcal{F}| \gg \sigma^2$, $k^* \approx \sigma\sqrt{\rho|\mathcal{F}|}$: k^* grows with the score noise σ (variance demands precision), with $\sqrt{\rho}$ (each revealed bias becomes stale within $\sim 1/\rho$ rounds, so faster correction permits a larger evaluation set), and with $\sqrt{|\mathcal{F}|}$ (a richer task universe dilutes any single leak). The threshold $\rho|\mathcal{F}| > \sigma^2$ is itself informative: when correction reaches fewer new tasks per round than the noise scale, no interior optimum exists and variance reduction dominates uniformly. The capacity to identify and correct biases—ρ—is

thus a key lever for high-stakes evaluation design, and determines the optimal level of benchmark transparency. We illustrate trade-offs between ρ and k empirically in simulations below.

5 Low Rank Latent Factor Structure Experiment

Having established bounds on misalignment in a general task space, we now instantiate this framework with a low-rank latent factor model. Latent factor models are a popular paradigm for empirically understanding benchmark performance: rooted in item response theory [9], they have been shown to capture the structure of AI evaluation data with both explanatory and predictive power [32, 28].

In a latent factor model, each instrument $f \in \mathcal{F}$ is associated with a demand vector $z_f \in \mathbb{R}^d$ encoding which skills it tests and in what proportion. Each model $\theta \in \Theta$ directly represents a skill profile in $\mathbb{R}^d$, and performance is given by $f(\theta) = \sigma(z_f^\top \theta)$ where σ is a link function. The feasible set $\Theta = \{\theta \in \mathbb{R}^d : \|\theta\|_2 \leq 1\}$ imposes a resource budget: the builder must allocate a fixed capacity across skill dimensions, so improving one skill comes at the cost of another. The socially optimal θ in this setup would allocate equal weight to all capabilities. Concretely, $\theta = [1, 0, 0, \dots]$ is a model whose entire capacity is in skill 1 (e.g. math), and $z_f = [1, 0, 0, \dots]$ is an instrument testing only that skill. Let $\mathcal{F} = \{f : z_f \in \mathbb{R}^d_+, ||z||_2 \leq 1\}$ be the universe of all instruments we care about.

The low-rank structure of this model allows us to translate the general misalignment and correction results of Sections 3–4 into geometric statements in $\mathbb{R}^d$. Because every task f maps to a direction z_f and every model maps to a point θ, the abstract distributional distance between $\pi_E^{(t)}$ and the socially optimal distribution μ becomes a Wasserstein distance over skill-space embeddings (Definition D.2), giving a tighter and more interpretable bound on residual misalignment. Because this distance respects the geometry of the skill embeddings, the correction problem effectively lives in the d-dimensional skill space rather than the $|\mathcal{F}|$-dimensional task space, and coverage audits reduce to identifying underrepresented skill directions rather than individual tasks—a more natural and actionable diagnostic for the evaluator (Appendix D).

Next, we simulate the full game to validate these predictions and explore the design tradeoffs numerically. We construct a task pool from the MMLU-Pro benchmark (48 AI test takers $\times$ 13,542 items) by fitting a logistic factor model via full-information maximum likelihood (loadings $L \in \mathbb{R}^{|\mathcal{F}| \times d}$ with $d = 8$ retained by parallel analysis [13] at the 95th-percentile threshold of column-permuted eigenvalues), applying varimax rotation for simple structure, projecting each item to the positive unit ball as $\tilde{z}_f = |L_f| / \max(\|L_f\|, 1)$, and sparsifying by zeroing any entry with relative squared contribution $(\tilde{z}_{f,i})^2 / \|\tilde{z}_f\|^2 < \tau$ before renormalizing. The threshold $\tau \in [0, 1]$ controls how aggressively each item concentrates on a few skills, which in turn governs the size of the evaluator's narrow awareness set. The evaluator knows only a subset of the d latent skills. Concretely, we pick a set of "hidden" skill axes $J \subset \{1, \dots, d\}$ with $|J| = d - k$ and define the evaluator's task awareness as the items that load only on the visible axes: $\mathcal{F}^E = \{ f \in \mathcal{F} : \tilde{z}_f[j] = 0 \text{ for all } j \in J \}$.

We begin by considering the one-shot game, comparing social utility under full transparency and the randomized single-sample mechanism. Our goal is to understand the role of transparency vs. randomization when a model developer is limited to ERM. We measure the utility $u_E(\theta^M)$ for a model developer that trains θ^M, and how it varies with: (i) the level of transparency k the evaluator uses; (ii) the number of tasks n the model developer samples for ERM; (iii) the model developer's awareness $|\mathcal{F}^M|$ relative to the evaluator's awareness $|\mathcal{F}^E|$. Figure 2 shows that a builder with broad awareness (l close to d) and sufficient data converges to near-optimal utility via ERM, while deterministic transparency is bounded by the $\mathcal{F}^E$-centroid direction regardless of k. The wedge between the two grows with both n and l, confirming the alignment cost of $\mathcal{F}^E$-bias from Section 3. The joint dependence on $(n, |\mathcal{F}^M|)$ in Figure 6 shows the two ingredients are largely com-

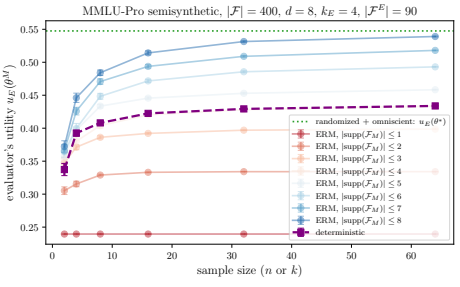

Figure 2: One-shot ERM vs. deterministic transparency at varying builder awareness. As $|\text{supp}(\mathcal{F}_M)|$ increases toward d, ERM converges to the social optimum $u_E(\theta^*)$ (green dotted). The gap grows with awareness l. See Figure 5 in Appendix D for the complementary transparency-vs-variance view.

plementary, so designs that limit only one leave most of the gap intact.

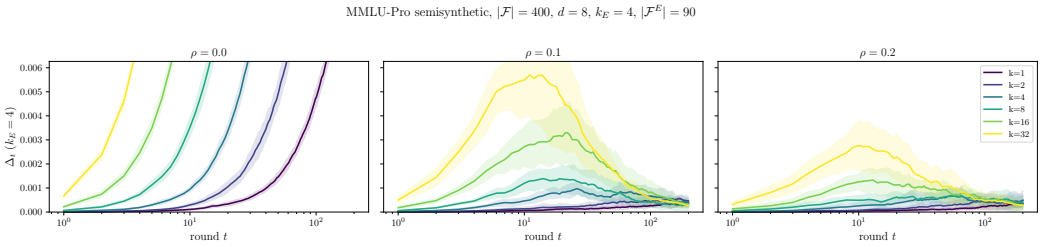

Figure 3: Residual misalignment Δ_t over time on the MMLU-Pro semisynthetic pool, for $\rho \in \{0, 0.1, 0.2\}$. Without correction ($\rho = 0$, left), Δ_t grows monotonically with t at a rate that scales with k, recovering Theorem 3.5. With correction ($\rho > 0$), Δ_t peaks then decays; faster correction (larger ρ) yields tighter steady-state misalignment.

We extend the analysis to the repeated game with $T = 200$ rounds and gap-targeted Gaussian correction with $\sigma_c = 0.4$. Figure 3 validates the qualitative shape predicted by Theorems 3.5 and 4.2: without correction ($\rho = 0$), Δ_t accumulates leakage proportional to k; with correction ($\rho > 0$), Δ_t peaks then decays, with faster correction yielding tighter steady-state misalignment. Figure 4 maps out the (ρ, k) tradeoff: iso-cost contours show that large ρ tolerates large k (faster correction lets transparency leak less), while the worst regime sits in the bottom-right corner where the evaluator is both transparent (k large) and slow to update (ρ small). The total-cost variant ($\sum_t [\Delta_t + \mathrm{Var}(\hat{r}_k)]$) and numerical validation of Theorem 4.2 and Lemma 4.3 are in Appendix D.

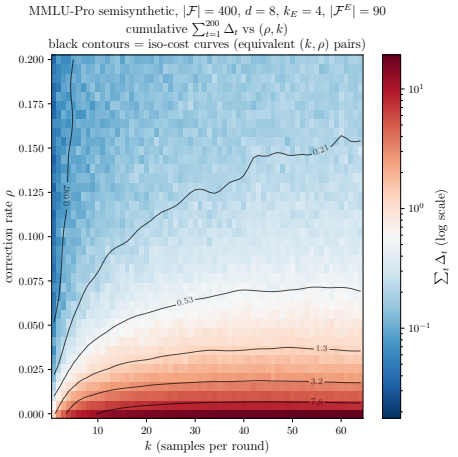

Figure 4: Cumulative residual misalignment $\sum_{t=1}^{T} \Delta_t$ over a 40×64 grid ($\rho \in [0, 0.2]$, $k \in \{1, 2, \ldots, 64\}$) on the MMLU-Pro semisynthetic pool. Black contours are iso-cost curves. The total-cost variant adding $\mathrm{Var}(\hat{r}_k) = \sigma^2/k$ is in Appendix D (Figure 8).

6 Discussion and Conclusion

We have shown that randomized evaluation can incentivize model developers to cover more socially relevant tasks, and empirically, this helps even for a boundedly-rational model developer relying on ERM estimates. Somewhat counterintuitively, this yields a *less is more* type of policy recommendation, where an evaluator with limited information can achieve better social utility by randomly withholding some of their tasks from a better-informed model developer. This is particularly relevant for high-stakes evaluation of domains with many unknown unknowns, like AI safety and risk. On the other hand, even randomized evaluation leaks information, and the amount of information leaked trades off with another important practical property of score variance. We find that dynamic distribution correction restores alignment, which means evaluators need to actively elicit and respond to external feedback. Faster response to such feedback can allow for more transparency, and we both theoretically and empirically illustrated the trade-off between transparency k and correction rate ρ over time. Notably, Figure 4 shows that the worst (ρ, k) regime is exactly where many benchmarks sit by default: publishing all tasks and updating infrequently.

Limitations. We abstract away several practical constraints faced by the model developer and evaluator. Theoretical results hinge on model developer beliefs, which are not always measurable or consistent. Real model developers also face engineering frictions that may slow exploitation. From the evaluator's side, a binding constraint is often onboarding new tasks rather than task supply: HELM onboarded only ≈ 20 benchmarks over five years, while the community produced hundreds annually. Score variance also has practical costs not fully modeled, as private benchmarks can reduce

auditability and public accountability. Natural extensions include multi-objective settings with costs to onboarding new tasks, as well as empirical estimates of ρ and σ_c from real-world evaluators.

References

[1] Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, pages 67–93, 2024.

[2] Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2003.

[3] Yahav Bechavod, Chara Podimata, Steven Wu, and Juba Ziani. Information discrepancy in strategic learning. In *International Conference on Machine Learning*, pages 1691–1715. PMLR, 2022.

[4] B Douglas Bernheim and Michael D Whinston. Incomplete contracts and strategic ambiguity. *American Economic Review*, pages 902–932, 1998.

[5] Avrim Blum and Moritz Hardt. The ladder: A reliable leaderboard for machine learning competitions. *arXiv preprint arXiv:1502.04585*, 2015.

[6] Mark Braverman and Sumegha Garg. The role of randomness and noise in strategic classification. In *1st Symposium on Foundations of Responsible Computing (FORC 2020)*, pages 9–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020.

[7] Ricardo Dominguez-Olmedo, Florian E. Dorner, and Moritz Hardt. Training on the test task confounds evaluation and emergence. *arXiv preprint arXiv:2407.07890*, 2024. URL `https://arxiv.org/abs/2407.07890`.

[8] Florian Ederer, Richard Holden, and Margaret Meyer. Gaming and strategic opacity in incentive provision. *The RAND Journal of Economics*, 49(4):819–854, 2018.

[9] Susan E. Embretson and Steven P. Reise. *Item Response Theory for Psychologists*. Psychology Press, 2013.

[10] Ganesh Ghalme, Vineet Nair, Itay Eilat, Inbal Talgam-Cohen, and Nir Rosenfeld. Strategic classification in the dark. In *International Conference on Machine Learning (ICML)*, 2021.

[11] Charles A.E. Goodhart. Problems of monetary management: The U.K. experience. *Monetary Theory and Practice*, pages 91–121, 1984.

[12] Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 111–122, 2016.

[13] John L Horn. A rationale and test for the number of factors in factor analysis. *Psychometrika*, 30(2):179–185, 1965.

[14] Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, et al. Dynabench: Rethinking benchmarking in nlp. In *Proceedings of the 2021 conference of the North American chapter of the Association for Computational Linguistics: human language technologies*, pages 4110–4124, 2021.

[15] Jon Kleinberg and Manish Raghavan. How do classifiers induce agents to invest effort strategically? In *Proceedings of the 2020 ACM Conference on Economics and Computation*, pages 825–844, 2020.

[16] Benjamin Laufer, Jon Kleinberg, Karen Levy, and Helen Nissenbaum. Strategic evaluation. In *Proceedings of the 3rd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, pages 1–12, 2023.

[17] Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-Hard and BenchBuilder pipeline. *arXiv preprint arXiv:2406.11939*, 2024. URL https://arxiv.org/abs/2406.11939.

[18] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022. URL https://arxiv.org/abs/2211.09110. HELM benchmark.

[19] Siqi Liu, Ian Gemp, Luke Marris, Georgios Piliouras, Nicolas Heess, and Marc Lanctot. Re-evaluating open-ended evaluation of large language models. *arXiv preprint arXiv:2502.20170*, 2025.

[20] David Manheim and Scott Garrabrant. Categorizing variants of Goodhart's Law. *arXiv preprint arXiv:1803.04585*, 2018. URL https://arxiv.org/abs/1803.04585.

[21] Timothy R. McIntosh, Teo Susnjak, Nalin Arachchilage, Tong Liu, Paul Watters, and Malka N. Halgamuge. Inadequacies of large language model benchmarks in the era of generative artificial intelligence. *arXiv preprint arXiv:2402.09880*, 2024. URL https://arxiv.org/abs/2402.09880.

[22] Inioluwa Deborah Raji, Emily Denton, Emily M Bender, Alex Hanna, and Amandalynne Paullada. Ai and the everything in the whole wide world benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[23] Anka Reuel, Avijit Ghosh, Jenny Chim, Andrew Tran, Yanan Long, Jennifer Mickel, Usman Gohar, Srishti Yadav, Pawan Sasanka Ammanamanchi, Mowafak Allaham, et al. Who evaluates ai's social impacts? mapping coverage and gaps in first and third party evaluations. *arXiv preprint arXiv:2511.05613*, 2025.

[24] Zachary Robertson and Sanmi Koyejo. Let's measure information step-by-step: Llm-based evaluation beyond vibes. *arXiv preprint arXiv:2508.05469*, 2025.

[25] Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10776–10787, 2023.

[26] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[27] Ali Shirali, Rediet Abebe, and Moritz Hardt. A theory of dynamic benchmarks. In *The Eleventh International Conference on Learning Representations*, 2023.

[28] Sang Truong, Yuheng Tu, Percy Liang, Bo Li, and Sanmi Koyejo. Reliable and efficient amortized model-based evaluation. *arXiv preprint arXiv:2503.13335*, 2025.

[29] Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saber, Micah Goldblum, et al. LiveBench: A challenging, contamination-free LLM benchmark. *arXiv preprint arXiv:2406.19314*, 2024. URL https://arxiv.org/abs/2406.19314.

[30] Cheng Xu et al. Benchmark data contamination of large language models: A survey. *arXiv preprint arXiv:2406.04244*, 2024. URL https://arxiv.org/abs/2406.04244.

[31] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *arXiv preprint arXiv:2306.05685*, 2023. URL https://arxiv.org/abs/2306.05685.

[32] Lexin Zhou, Lorenzo Pacchiardi, Fernando Martínez-Plumed, Katherine M. Collins, et al. General scales unlock AI evaluation with explanatory and predictive power. *Nature*, 2026.

A Proofs and Extended Results for the Evaluation Game

Theorem 2.3 (Failure of Deterministic Mechanisms)**.** *If $\mathcal{F}_E \subset \mathcal{F}$, then no deterministic mechanism can recover the optimal utility: for any non-degenerate r and deterministic S, there always exists $\mathcal{F}, \mu$ such that $u_E(\theta^M) < \max_{\theta\in\Theta} u_E(\theta)$.*

Proof. If the evaluator publishes a deterministic set $S(F_E, \omega) = S_E$, then the model developer's best response will be $\theta^M \in \arg\max_{\theta\in\Theta} r(\theta, S_E)$. Take the function r as given, and let $S(F_E, \omega) = S_E$. We assume $r(\cdot, S_E)$ is non-degenerate for all S_E, so not all $\theta \in \Theta$ are maximizers of $r(\theta, S_E)$. Let $\theta' \notin \arg\max_{\theta\in\Theta} r(\theta, S_E)$. Set μ to be uniform, and set $\mathcal{F} = \mathcal{F}_E + \{f'\}$, with $f' \notin S_E$. Let f' maximized at $\theta' \in \Theta$, and let $f'(\theta') > \frac{2}{\mu(f')} \sup_{\theta\in\Theta} \left(\sum_{f\in\mathcal{F}_E} f(\theta)\mu(f)\right)$. Then θ' maximizes $u_E(\theta)$, but $\theta^M \neq \theta'$. □

Proposition 2.6 (Single sample mechanism under general beliefs)**.** *Under a single sample mechanism, the Bayesian model developer's best response is $\theta^M \in \arg\max_{\theta\in\Theta} \mathbb{E}_{f\sim\tilde{\pi}_E}[f(\theta)]$.*

Proof. Let $\tilde{\Pi}_E$ denote the model developer's prior over the distribution of the set F_E, with $\tilde{\Pi}_E$ supported on all sets of size n_E in the model developer's awareness set, $\mathcal{P}(\mathcal{F}_M)$. Let $\tilde{\pi}_E = P(f \in F_E)$ denote the marginal single-item distribution corresponding to the per-set distribution $\tilde{\Pi}_E$. The risk-neutral model developer solves:

$$\theta^* \in \arg\max_{\theta\in\Theta} \ \mathbb{E}_{F_E\sim\tilde{\Pi}_E}[\mathbb{E}_\omega[r(\theta, M(F_E, \omega))]] \,.$$

Since $S(F_E, \omega)$ samples a single task uniformly at random from F_E:

$$\mathbb{E}_{F_E}[\mathbb{E}_\omega[r(\theta, M(F_E, \omega))]] = \mathbb{E}_{F_E}\left[\frac{1}{|F_E|}\sum_{f\in F_E} f(\theta)\right].$$

We rewrite the inner sum using indicator variables over $\mathcal{F}_M$:

$$\frac{1}{|F_E|}\sum_{f\in F_E} f(\theta) = \sum_{f\in\mathcal{F}_M} \frac{\mathbf{1}[f \in F_E]}{|F_E|} \cdot f(\theta).$$

Exchanging the sum and expectation (both are finite):

$$\mathbb{E}_{F_E}\left[\frac{1}{|F_E|}\sum_{f\in F_E} f(\theta)\right] = \sum_{f\in\mathcal{F}_M} \underbrace{\mathbb{E}_{F_E}\left[\frac{\mathbf{1}[f \in F_E]}{|F_E|}\right]}_{\propto\, \tilde{\pi}_E(f)} \cdot f(\theta).$$

Therefore, the model developer's best response maximizes $\mathbb{E}_{f\sim\tilde{\pi}_E}[f(\theta)]$. □

Theorem 2.7. *Under the single sample mechanism, the suboptimality for the evaluator is bounded by the total variation distance of the model developer's prior from μ: $\max_{\theta\in\Theta} u_E(\theta) - u_E(\theta^M) \leq 4B \cdot \mathrm{TV}(\mu, \tilde{\pi}_E)$, where $B = \sup_{f\in\mathcal{F};a,b\in\Theta} |f(a) - f(b)|$.*

Proof. We assume $f(\theta) \in [a, b]$ for all $f \in \mathcal{F}$. $C = 2(b-a)|\mathcal{F}|$. Define

$$L_u(\theta) := \mathbb{E}_{f\sim\mu}[f(\theta)], \qquad L_\pi(\theta) := \mathbb{E}_{f\sim\tilde{\pi}_E}[f(\theta)]$$

For a Bayesian model developer, $\theta^M \in \arg\max_{\theta\in\Theta} L_{\tilde{\pi}_E}(\theta)$.

Let $\theta^* \in \arg\max_{\theta\in\Theta} L_u(\theta)$. Since $L_u(\theta) = |\mathcal{F}|u_E(\theta)$, θ^* also maximizes u_E. We first bound

$$L_u(\theta^*) - L_u(\theta^M).$$

By adding and subtracting $L_\pi(\theta^*)$ and $L_\pi(\theta^M)$, we get

$$L_u(\theta^*) - L_u(\theta^M) = \big(L_u(\theta^*) - L_\pi(\theta^*)\big) + \big(L_\pi(\theta^*) - L_\pi(\theta^M)\big) + \big(L_\pi(\theta^M) - L_u(\theta^M)\big).$$

Since θ^M maximizes L_π, we have

$$L_\pi(\theta^M) \geq L_\pi(\theta^*),$$

and therefore

$$L_\pi(\theta^*) - L_\pi(\theta^M) \leq 0.$$

Hence,

$$\begin{aligned} L_u(\theta^*) - L_u(\theta^M) &\leq \big(L_u(\theta^*) - L_\pi(\theta^*)\big) + \big(L_\pi(\theta^M) - L_u(\theta^M)\big) \\ &\leq |L_u(\theta^*) - L_\pi(\theta^*)| + \big|L_u(\theta^M) - L_\pi(\theta^M)\big| \\ &\leq 2 \sup_\theta |L_u(\theta) - L_\pi(\theta)|. \end{aligned}$$

It remains to bound the discrepancy

$$|L_u(\theta) - L_\pi(\theta)|.$$

For any fixed θ,

$$|L_u(\theta) - L_\pi(\theta)| = \left| \int_{f \in \mathcal{F}} \big(\mu(f) - \tilde{\pi}_E(f)\big) f(\theta) df \right|.$$

Because $f(\theta) \in [a, b]$, shifting by a does not change the sum, since

$$\int_{f \in \mathcal{F}} \big(\mu(f) - \tilde{\pi}_E(f)\big) df = 0.$$

Thus,

$$\left| \int_{f \in \mathcal{F}} \big(\mu(f) - \tilde{\pi}_E(f)\big) f(\theta) df \right| = \left| \int_{f \in \mathcal{F}} \big(\mu(f) - \tilde{\pi}_E(f)\big) \big(f(\theta) - a\big) df \right|.$$

Since $0 \leq f(\theta) - a \leq b - a$, we have

$$|L_u(\theta) - L_\pi(\theta)| \leq 2(b - a)\, \mathrm{TV}(\mu, \tilde{\pi}_E),$$

where

$$\mathrm{TV}(\mu, \tilde{\pi}_E) := \frac{1}{2} \int_{f \in \mathcal{F}} |\mu(f) - \tilde{\pi}_E(f)| df.$$

Combining the two inequalities gives

$$L_u(\theta^*) - L_u(\theta^M) \leq 4(b - a)\, \mathrm{TV}(\mu, \tilde{\pi}_E) \leq 4B \cdot \mathrm{TV}(\mu, \tilde{\pi}_E).$$

□

Corollary 2.8 (Single-sample mechanism strictly broadens incentivization)**.** *If $\mathcal{F}_E \subset \mathcal{F}_M$ and $\tilde{\pi}_E(f) > 0$ for some $f \notin \mathcal{F}_E$, then the single-sample mechanism incentivizes the model developer over a strictly larger set of tasks than any deterministic mechanism.*

Proof. Under any deterministic mechanism with published set $S_E \subseteq F_E$, the model developer's objective depends only on $\{f(\theta) : f \in S_E\}$: tasks outside S_E have zero weight and exert no influence on the model developer's choice of θ. Under the single-sample mechanism, the model developer maximizes $\mathbb{E}_{f \sim \tilde{\pi}_E}[f(\theta)]$ (Theorem 2.6). The model developer's objective now depends on $\{f(\theta) : f \in \mathrm{supp}(\tilde{\pi}_E)\}$, which includes every task with $\tilde{\pi}_E(f) > 0$. Since $\mathrm{supp}(\tilde{\pi}_E) \supseteq \mathcal{F}_M \supset \mathcal{F}_E \supseteq S_E$ and $\tilde{\pi}_E(f) > 0$ for some $f \notin \mathcal{F}_E$, the set of tasks influencing the model developer's optimization is strictly larger: $\mathrm{supp}(\tilde{\pi}_E) \supset S_E$. Concretely, any task $f \notin S_E$ with $\tilde{\pi}_E(f) > 0$ now contributes a term $\tilde{\pi}_E(f) \cdot f(\theta)$ to the model developer's objective. The model developer cannot ignore these tasks without sacrificing expected reward. This converts tasks that were entirely "free" under the deterministic mechanism—where the model developer could set $f(\theta) = 0$ at no cost—into tasks that the model developer is actively incentivized to perform well on. □

A.1 Additional results and discussion of sampling limitations for the model developer

We assume that the ERM model developer has the following information and optimization abilities:

1. **Observes a finite sample**: The model developer draws $F_M = \{f_1, \ldots, f_n\}$ i.i.d. from π_M, and can only evaluate θ on tasks in F_M.
2. **Can evaluate and optimize over** Θ: For all $f \in F_M$ sampled by the model developer, the model developer can compute $f(\theta)$ for all $\theta \in \Theta$ and can solve $\arg\max_{\theta\in\Theta} g(\theta)$ for any objective g defined over F_M. The model developer does not have a prior distribution over Θ.
3. **Knows the marginal inclusion weights for sampled tasks**: the model developer knows $\tilde{\pi}_E(f)$ and $\pi_M(f)$ for each $f \in F_M$.
4. **Is risk-neutral with respect to sampling variability**: after observing F_M, the model developer maximizes the empirical objective $\hat{R}_n(\theta)$ without any adjustment for its variance as an estimator of $R(\theta)$.

The ERM model developer can be thought of as as a learning agent who is risk-neutral relative to both the variability in their risk estimator and the variability of the evaluator's reward mechanism. More formally, the model builder can be thought of as maximizing a utility $u(\theta) = \hat{R}_n(\theta) + \varepsilon(\theta)$, where $\varepsilon(\theta) = r(\theta, S(F_E, \omega)) - \hat{R}_n(\theta)$. Here, $\hat{R}_n(\theta)$ is an observed quantity, and all variability due to both estimation error and the evaluator's reward mechanism is contained in $\varepsilon(\theta)$. A risk-neutral model developer then optimizes $E_\varepsilon[\hat{R}_n(\theta) + \varepsilon(\theta)]$. For the single-sample mechanism, and an unbiased risk estimator, this reduces to simply maximizing $\hat{R}_n(\theta)$.

Proposition A.1 (Risk-neutral model developer performs importance-weighted ERM)**.** *Under the above conditions, the model developer's best response given F_M is:*

$$\theta^{F_M} \in \arg\max_{\theta\in\Theta} \hat{R}_n(\theta), \quad \textit{where} \quad \hat{R}_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} \frac{w(f_i)}{\pi_M(f_i)} f_i(\theta). \tag{1}$$

When $\tilde{\pi}_E = \pi_M$, the importance weights cancel and this reduces to unweighted ERM:

$$\theta^{F_M} \in \arg\max_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} f_i(\theta). \tag{2}$$

Proof. The model developer wants to maximize $R(\theta) = \mathbb{E}_{f\sim\tilde{\pi}_E}[f(\theta)]$ but can only evaluate θ on tasks $f_i \in F_M$ drawn from π_M. The importance-weighted estimator $\hat{R}_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} \frac{\tilde{\pi}_E(f_i)}{\pi_M(f_i)} f_i(\theta)$ is unbiased for $R(\theta)$:

$$\mathbb{E}_{F_M}[\hat{R}_n(\theta)] = \mathbb{E}_{f\sim\pi_M}\left[\frac{\tilde{\pi}_E(f)}{\pi_M(f)} f(\theta)\right] = \sum_{f\in\mathcal{F}_M} \pi_M(f) \cdot \frac{\tilde{\pi}_E(f)}{\pi_M(f)} \cdot f(\theta) = \mathbb{E}_{f\sim\tilde{\pi}_E}[f(\theta)] = R(\theta).$$

A risk-neutral model developer maximizes the expected payoff $\mathbb{E}_{F_M}[R(\theta(F_M))]$. Since $\hat{R}_n(\theta)$ is an unbiased estimator of $R(\theta)$, maximizing $\hat{R}_n(\theta)$ for the realized F_M is the natural strategy. □

Remark A.2 (Per-set sampling and dependence)**.** *Proposition A.1 assumes F_M consists of i.i.d. draws from π_M. More generally, F_M is drawn from the per-set distribution Π_M over $\mathcal{P}(\mathcal{F}_M)$, which may induce dependence among tasks (e.g., sampling without replacement). Two cases preserve the results: (i) if each task's marginal distribution under Π_M is π_M, the importance-weighted estimator $\hat{R}_n(\theta)$ remains unbiased for $R(\theta)$; (ii) for sampling without replacement, the negative correlations between samples actually* tighten *the concentration bounds relative to the i.i.d. case. For general Π_M where marginals differ from π_M, the model developer would need an additional correction layer for the per-set sampling process.*

Proposition 2.10 (Sample size improves the generalization guarantee)**.** *When $\tilde{\pi}_E = \pi_M$, the excess risk of θ^{F_M} trained on all n samples satisfies, with probability $\geq 1-\delta$: $R(\theta^*) - R(\theta^{F_M}) \leq O\left(\mathfrak{R}_n(\Theta) + \sqrt{\log(1/\delta)/n}\right)$, where $\mathfrak{R}_n(\Theta)$ is the Rademacher complexity of the function class $\{f \mapsto f(\theta) : \theta \in \Theta\}$.*

Proof. Define the loss class $\mathcal{G} = \{g_\theta : f \mapsto f(\theta) \mid \theta \in \Theta\}$, so that $R(\theta) = \mathbb{E}_{f\sim\pi_M}[g_\theta(f)]$ and $\hat{R}_n(\theta) = \frac{1}{n}\sum_{i=1}^n g_\theta(f_i)$. Since $g_\theta(f) \in [0,1]$, standard symmetrization and concentration arguments [2, 26] yield:

$$\mathbb{P}\left[\sup_{\theta\in\Theta}\left|R(\theta) - \hat{R}_n(\theta)\right| > 2\,\mathfrak{R}_n(\mathcal{G}) + \sqrt{\frac{\ln(2/\delta)}{2n}}\right] \le \delta.$$

On the event that uniform convergence holds, we have:

$$R(\theta^*) - R(\theta^{F_M}) = \big(R(\theta^*) - \hat{R}_n(\theta^*)\big) + \big(\hat{R}_n(\theta^*) - \hat{R}_n(\theta^{F_M})\big) + \big(\hat{R}_n(\theta^{F_M}) - R(\theta^{F_M})\big).$$

Since θ^{F_M} maximizes $\hat{R}_n$, the middle term is non-positive. Each of the remaining terms is bounded by $2\,\mathfrak{R}_n(\mathcal{G}) + \sqrt{\ln(2/\delta)/(2n)}$, giving the stated bound.

For standard function classes (finite-VC, Lipschitz with bounded inputs), $\mathfrak{R}_k(\mathcal{G}) = O(1/\sqrt{k})$, so $\mathfrak{R}_k(\mathcal{G}) + \sqrt{\log(1/\delta)/k}$ is monotonically non-increasing in k and the worst-case bound for θ^{F_M} is tighter than for any strict subset estimator θ^S. □

B Proofs for Emergent Misalignment via Information Leakage

Lemma 3.3 (Variance forces large k)**.** *Let S be a size-k subset of $\mathcal{F}_E$ drawn uniformly without replacement, and let $\hat{r}_k(\theta) = \frac{1}{k}\sum_{f\in S} f(\theta)$. If the population scores $\{f(\theta)\}_{f\in\mathcal{F}_E}$ have variance σ^2, then*

$$\mathrm{Var}(\hat{r}_k) \;=\; \frac{\sigma^2}{k}\cdot\frac{|\mathcal{F}_E| - k}{|\mathcal{F}_E| - 1}. \tag{3}$$

Proof. Let $N := |\mathcal{F}_E|$ and write $X_i := f_i(\theta)$ for the score of the i-th drawn task, $i = 1, \dots, k$. Let

$$\bar{r} \;:=\; \frac{1}{N}\sum_{f\in\mathcal{F}_E} f(\theta), \qquad \sigma^2 \;:=\; \frac{1}{N}\sum_{f\in\mathcal{F}_E}\big(f(\theta) - \bar{r}\big)^2,$$

denote the population mean and variance. Uniform sampling without replacement is exchangeable, so each X_i has the same marginal distribution as a single uniform draw from $\mathcal{F}_E$. Hence $\mathbb{E}[X_i] = \bar{r}$ and $\mathrm{Var}(X_i) = \sigma^2$ for every $i \in \{1, \dots, k\}$. For $i \neq j$, the unordered pair $\{X_i, X_j\}$ is uniformly distributed over the $\binom{N}{2}$ size-2 subsets of $\mathcal{F}_E$, so the ordered pair is uniform over the $N(N-1)$ ordered pairs of distinct elements. Therefore

$$\begin{aligned}\mathbb{E}[X_iX_j] = \frac{1}{N(N-1)}\sum_{f\neq g} f(\theta)\,g(\theta) \;&=\; \frac{1}{N(N-1)}\left[\Big(\sum_f f(\theta)\Big)^2 - \sum_f f(\theta)^2\right] \\ &= \frac{1}{N(N-1)}\big[N^2\bar{r}^2 - N(\sigma^2 + \bar{r}^2)\big] \;=\; \frac{(N-1)\bar{r}^2 - \sigma^2}{N-1} \;=\; \bar{r}^2 - \frac{\sigma^2}{N-1}.\end{aligned}$$

Subtracting $\mathbb{E}[X_i]\mathbb{E}[X_j] = \bar{r}^2$, $\mathrm{Cov}(X_i, X_j) = -\sigma^2/(N-1)$. Intuitively, removing one drawn item from the population shifts the conditional mean of every remaining item, inducing the negative correlation that distinguishes without-replacement from with-replacement sampling. Combining the above with the standard variance decomposition,

$$\begin{aligned}\mathrm{Var}(\hat{r}_k) &= \frac{1}{k^2}\left[\sum_{i=1}^k \mathrm{Var}(X_i) + \sum_{i\neq j}\mathrm{Cov}(X_i, X_j)\right] \\ &= \frac{1}{k^2}\left[k\sigma^2 + k(k-1)\cdot\Big(-\tfrac{\sigma^2}{N-1}\Big)\right] = \frac{\sigma^2}{k}\left[1 - \frac{k-1}{N-1}\right] = \frac{\sigma^2}{k}\cdot\frac{N-k}{N-1},\end{aligned}$$

which is (3). The factor $(N-k)/(N-1)$ is the finite-population correction. It equals 1 when $k = 1$ and decreases linearly to 0 when $k = N$ (sampling the entire population eliminates variance). When $k \ll N$, $(N-k)/(N-1) = 1 - O(k/N)$, so the standard $\mathrm{Var}(\hat{r}_k) \approx \sigma^2/k$ rate is recovered up to a vanishing correction. When k is comparable to N, the correction is non-negligible: e.g., $k = N/2$ gives a factor of $1/2$ relative to the with-replacement bound, halving the variance compared to the naive rate. □

Theorem 3.5 (Misalignment growth without correction)**.** *Under the k-sample mechanism (Definition 3.2), suppose $\mathcal{F}_M = \mathcal{F}$ and the model developer has prior $\tilde{\pi}_0$ set to Dir$(\mu(f_1), \ldots, \mu(f_N))$ for $f_1, ..., f_N \in \mathcal{F}$, and applies Bayes updates from the observed tasks with $m_t = kt$ total observations after t rounds. Then $\Delta_t \leq 4Bm_t/(m_t + |\mathcal{F}|)$, where $B = \sup_{f\in\mathcal{F};a,b\in\Theta} |f(a) - f(b)|$.*

Proof. Under the k-sample mechanism (Definition 3.2), each round contributes k distinct tasks; across t rounds the model developer observes $m_t = kt$ samples with counts $\{n_f\}_{f\in\mathcal{F}}$ satisfying $\sum_f n_f = m_t$. The model developer's prior is Dir$(\mu(f_1), \ldots, \mu(f_N))$ over tasks in $\mathcal{F}$, and they update via Dirichlet–multinomial conjugacy. The posterior predictive distribution is

$$\tilde{\pi}_t(f) \;=\; \frac{n_f + 1}{m_t + |\mathcal{F}|}.$$

Letting $\alpha = m_t/(m_t + |\mathcal{F}|)$ and $\hat{p}_t(f) = n_f/m_t$, this rearranges into the shrinkage form $w_t = \alpha\,\hat{p}_t(f) + (1-\alpha)\,\mu(f)$, so $w_t(f) - \mu(f) = \alpha\big(\hat{p}_t(f) - \mu(f)\big)$ and

$$\mathrm{TV}(w_t, \mu) \;=\; \alpha \cdot \mathrm{TV}(\hat{p}_t, \mu) \;\leq\; \alpha \;=\; \frac{m_t}{m_t + |\mathcal{F}|}.$$

Applying the standard regret–TV bound as argued in Theorem 2.7 gives $\Delta_t \leq 4B \cdot \mathrm{TV}(w_t, \mu) \leq 4Bm_t/(m_t + |\mathcal{F}|)$. □

C Proofs and Extended Results on Restoring Alignment via Distribution Correction

Remark C.1 (Micro-foundation for Gap-Proportional Discovery)**.** *We provide a concrete micro-foundation for the gap-proportional discovery in Assumption 4.1. Consider a toy setting with $\Theta = \Delta_k = \{\theta \in \mathbb{R}^k : \theta_i \geq 0, \sum_i \theta_i = 1\}$ and $\mathcal{F} = \{f_1, \ldots, f_k\}$ where $f_i(\theta) = e_i^\top \theta$ with e_i the standard basis vector (one-hot: 1 in entry i, 0 elsewhere), so each task measures a single coordinate of the builder's resource allocation. The evaluator-optimal allocation is $\theta^* = (1/k, \ldots, 1/k)$, while the builder deploys $\theta_i^{(t-1)} = \pi_E^{(t-1)}(f_i)$ (allocating effort proportionally to evaluation weights). The per-task performance gap is $\delta_i = f_i(\theta^*) - f_i(\theta^{(t-1)}) = 1/k - \pi_E^{(t-1)}(f_i)$.*

Gap discovery is mediated by user complaints. At each round, a single user encounters a task f_j sampled uniformly from $\mathcal{F}$, observes performance $f_j(\theta^{(t-1)})$, and files a complaint with probability proportional to the gap severity $[\delta_j]_+$—larger failures are more likely to prompt a report. The evaluator then addresses this complaint. By Bayes' rule, conditioned on a complaint being filed, the probability that it concerns task f_i is the product of encounter probability and complaint probability, normalized. Since tasks are encountered uniformly, the $(1/k)$ factors cancel:

$$P(\nu_t = f_i) \;=\; \frac{P(\text{encounter } f_i)\cdot[\delta_i]_+}{\sum_j P(\text{encounter } f_j)\cdot[\delta_j]_+} \;=\; \frac{(1/k)\,[\delta_i]_+}{\sum_j (1/k)\,[\delta_j]_+} \;=\; \frac{[\delta_i]_+}{\sum_j [\delta_j]_+}, \tag{4}$$

where ν_t is the discovered gap location from Assumption 4.1—here identified with the task targeted by the sampled complaint. This recovers Assumption 4.1 exactly. This model requires only two empirically supported properties: (i) users sample tasks broadly (approximately uniformly across the capability space), and (ii) the probability of a complaint reaching the evaluator scales with the severity of the failure. The gap-proportional model can thus be viewed as reflecting the soft aggregation of many independent complaint signals, rather than a single hard comparison across a finite sample.

Theorem 4.2 (Misalignment Under Gap-Targeted Correction)**.** *Under Assumption 4.1, the expected residual misalignment satisfies $\mathbb{E}[\Delta_t] \;\leq\; \min\big(m_t/(m_t + |\mathcal{F}|),\;\; (1-\rho)^t\sqrt{0.5\,D_0} + \varepsilon(\rho, \sigma_c)\big)$, where $D_0 := \mathrm{KL}\Big(\pi_E^{(0)} \,\|\, \mu\Big)$ is the initial KL divergence of the evaluator's distribution from the social-relevance distribution μ. The first term is estimation-limited and the second is correction-limited. $\varepsilon(\rho, \sigma_c)$ is a steady-state fluctuation from the stochastic gap-discovery process, satisfying $\varepsilon = O(\sqrt{\mathrm{TV}(g_{\sigma_c}, \mu)})$ uniformly in ρ. This vanishes as $\sigma_c \to \infty$ (each correction becomes μ).*

Proof. Term 1 (estimation-limited). Under a Dirichlet prior with concentration concentration parameters $\mu(f)\cdot|\mathcal{F}|$, the builder's posterior predictive is the shrinkage estimator $\hat{\pi}_{E,t} = \frac{m_t}{m_t+|\mathcal{F}|}\hat{p}_t + \frac{|\mathcal{F}|}{m_t+|\mathcal{F}|}\,\mu$,

so $\mathrm{TV}(\hat{\pi}_{E,t}, \mu) \le m_t/(m_t + |\mathcal{F}|)$. Hence $\Delta_t \le m_t/(m_t + |\mathcal{F}|)$ regardless of the evaluator's correction mechanism. The effective sample size is the closed form $m_t = k\,(1-(1-\rho)^t)/\rho$ derived from the recursion $m_{t+1} = k + (1-\rho)\,m_t$ (new observations arrive at rate k while old ones age out at rate ρ); this is bounded above by $k\,\min(t, 1/\rho)$.

Term 2 (correction-limited). We use a Lyapunov argument on the L^2 distance $V_t = \|\delta_t\|_2^2$, where $\delta_t := \mu - \pi_E^{(t)}$ is the deficit relative to the social-relevance distribution. From the update $\delta_t = (1-\rho)\delta_{t-1} + \rho(\mu - g_t)$ where $g_t = g_{\sigma_c}(\cdot; \mu_t)$:

$$V_t = \|\delta_t\|_2^2 = (1-\rho)^2 V_{t-1} + 2(1-\rho)\rho\,\langle \delta_{t-1},\, \mu - g_t\rangle + \rho^2\,\|\mu - g_t\|_2^2. \tag{5}$$

By Cauchy–Schwarz: $|\langle \delta_{t-1}, \mu - g_t\rangle| \le \|\delta_{t-1}\|_2 \cdot \|\mu - g_t\|_2 = \sqrt{V_{t-1}} \cdot \|\mu - g_t\|_2$. Substituting,

$$\begin{aligned} V_t &\le (1-\rho)^2 V_{t-1} + 2(1-\rho)\rho\sqrt{V_{t-1}}\,\|\mu - g_t\|_2 + \rho^2\,\|\mu - g_t\|_2^2 \\ &= \big((1-\rho)\sqrt{V_{t-1}} + \rho\,\|\mu - g_t\|_2\big)^2. \end{aligned}$$

Taking square roots (all quantities non-negative), $\sqrt{V_t} \le (1-\rho)\sqrt{V_{t-1}} + \rho\,\|\mu - g_t\|_2$. The kernel deviation from μ is uniformly bounded: $C_g := \max_{\mu_t \in \mathcal{F}} \|\mu - g_{\sigma_c}(\cdot;\mu_t)\|_2^2$. This constant depends only on σ_c, μ, and the task geometry. For large σ_c, $g_{\sigma_c}(\cdot;\mu_t) \to \mathrm{Uniform}(\mathcal{F})$ and $C_g \to \|\mu - \mathrm{Uniform}(\mathcal{F})\|_2^2$, which equals zero only when μ is itself uniform. Substituting $\|\mu - g_t\|_2 \le \sqrt{C_g}$ into the V_t bound and unrolling,

$$\begin{aligned} \sqrt{V_t} \;&\le\; (1-\rho)^t\sqrt{V_0} + \rho\sqrt{C_g}\sum_{s=0}^{t-1}(1-\rho)^s \;=\; (1-\rho)^t\sqrt{V_0} + \sqrt{C_g}\big(1-(1-\rho)^t\big) \\ &\le\; (1-\rho)^t\sqrt{V_0} + \sqrt{C_g}. \end{aligned}$$

The misalignment satisfies $\Delta_t \le \mathrm{TV}(\pi_E^{(t)}, \mu) = \frac{1}{2}\|\delta_t\|_1 \le \frac{\sqrt{|\mathcal{F}|}}{2}\|\delta_t\|_2 = \frac{\sqrt{|\mathcal{F}|}}{2}\sqrt{V_t}$ (Cauchy–Schwarz on the ℓ_1-to-ℓ_2 inequality). Substituting the V_t bound,

$$\Delta_t \;\le\; \frac{\sqrt{|\mathcal{F}|}}{2}\big((1-\rho)^t\sqrt{V_0} + \sqrt{C_g}\big) \;=\; \underbrace{(1-\rho)^t \cdot \tfrac{1}{2}\sqrt{|\mathcal{F}|V_0}}_{\text{transient}} \;+\; \underbrace{\tfrac{1}{2}\sqrt{|\mathcal{F}|\,C_g}}_{\varepsilon(\rho,\sigma_c)}. \tag{6}$$

By Pinsker's inequality, $\frac{1}{2}\sqrt{|\mathcal{F}|V_0} \ge \mathrm{TV}(\pi_E^{(0)}, \mu)$ and $\mathrm{TV}(\pi_E^{(0)}, \mu) \le \sqrt{D_0/2}$, so the transient term can be expressed as $(1-\rho)^t\sqrt{D_0/2}$ up to the chi-squared/KL conversion. Setting $\varepsilon(\rho, \sigma_c) := \frac{1}{2}\sqrt{|\mathcal{F}|\,C_g}$ and noting $C_g = O(\mathrm{TV}(g_{\sigma_c}, \mu)^2)$ gives $\varepsilon = O(\mathrm{TV}(g_{\sigma_c}, \mu))$. Taking the minimum with the estimation-limited term completes the proof.

□

Lemma 4.3 (Closed form for k^*)**.** *Let $a := \rho|\mathcal{F}|$. If $a > \sigma^2$, $\mathcal{L}(k) = \sigma^2/k + k/(a+k)$ has a unique minimizer on $(0,\infty)$ at $k^* = \sigma\,\rho|\mathcal{F}|/(\sqrt{\rho|\mathcal{F}|} - \sigma)$. If $a \le \sigma^2$, $\mathcal{L}$ is strictly decreasing on $(0,\infty)$.*

Proof. The variance term is standard: with k i.i.d. task scores bounded in $[0,1]$, $\mathrm{Var}(\hat{r}_k) = \sigma^2/k$. For the exploitation term, the proof of Theorem 4.2 establishes the estimation-limited bound $\Delta_t \le m_t/(m_t + |\mathcal{F}|)$, with effective sample size $m_t = k \cdot \min(t,\, 1/\rho)$ saturating at k/ρ in steady state (Assumption 4.1 retains a $(1-\rho)$ fraction of the prior per round, giving obsolescence timescale ρ^{-1}). Hence $\Delta_\infty \le k/(\rho|\mathcal{F}| + k)$, which we keep without further approximation. Combining gives $\mathcal{L}(k) = \sigma^2/k + k/(a+k)$ with $a := \rho|\mathcal{F}|$.

Differentiating, $\mathcal{L}'(k) = -\sigma^2/k^2 + a/(a+k)^2$. Setting $\mathcal{L}'(k^*) = 0$ yields $\sigma^2(a+k^*)^2 = a(k^*)^2$. Taking positive square roots (both sides are positive on $k > 0$), $\sigma(a+k^*) = \sqrt{a}\,k^*$, i.e., $k^*(\sqrt{a} - \sigma) = \sigma a$. A positive solution exists iff $\sqrt{a} > \sigma$, equivalently $\rho|\mathcal{F}| > \sigma^2$, in which case

$$k^* \;=\; \frac{\sigma a}{\sqrt{a} - \sigma} \;=\; \frac{\sigma\,\rho|\mathcal{F}|}{\sqrt{\rho|\mathcal{F}|} - \sigma}. \tag{7}$$

This is the unique interior critical point. Since $\mathcal{L}(k) \to \infty$ as $k \to 0^+$ and $\mathcal{L}(k) \to 1$ as $k \to \infty$, $\mathcal{L}$ attains its minimum in the interior, hence at k^*. When $\rho|\mathcal{F}| \le \sigma^2$, $\sigma^2(a+k)^2 \ge \sigma^2 k^2 \ge a k^2$ for all $k > 0$, so $\mathcal{L}'(k) < 0$ throughout and $\mathcal{L}$ is strictly decreasing on $(0,\infty)$. In the high-correction limit $\rho|\mathcal{F}| \gg \sigma^2$, $\sqrt{a} - \sigma \approx \sqrt{a}$, giving the simpler form $k^* \approx \sigma\sqrt{\rho|\mathcal{F}|}$. □

Below, we prove the claims made in Section 4 regarding the gap-targeted Gaussian correction. Throughout, we write $\delta_t = \mu - \pi_E^{(t)}$ for the deficit vector relative to the social-relevance distribution, $\delta_t^+ = [\delta_t]_+$ and $\delta_t^- = [-\delta_t]_+$ for the positive and negative parts, and $D_t = \{f : \delta_t(f) > 0\}$ for the deficit region. Note that $\|\delta_t^+\|_1 = \|\delta_t^-\|_1 = \mathrm{TV}(\pi_E^{(t)}, \mu)$ and $\sum_f \delta_t(f) = 0$.

D Extended Results of Low Rank Latent Factor Structure Experiment

All simulation code is released at `https://anonymous.4open.science/r/strategic_evaluation-C1B5/`.

The distribution correction mechanism of Section 4 drives $\pi_E^{(t)} \to \mu$, asking the evaluator to align $|\mathcal{F}|$ individual task weights with the social-relevance distribution. In practice $|\mathcal{F}|$ is large and benchmarks have rich similarity structure: many tasks are near-duplicates testing the same underlying skill, while others test complementary skills. The central question of this section is: *when is full task-by-task coverage actually necessary?* We show that under a latent factor model that derives task structure from a shared d-dimensional skill embedding, the correction problem effectively lives in skill space rather than task space. The residual misalignment is bounded by an embedding Wasserstein distance that natively captures task similarity—swapping weight between similar tasks is cheap, between dissimilar tasks expensive—and the effective complexity is the latent dimension d, typically $d \ll |\mathcal{F}|$.

Proposition D.1 (Lipschitz Substitution Bound)**.** *Under the latent factor model, for any two tasks $f, g \in \mathcal{F}$ and any model $\theta \in \Theta$: $|f(\theta) - g(\theta)| \le \frac{1}{4}\|v_f - v_g\| \cdot \|\theta\|$.*

Proof. By the mean value theorem, $|f(\theta) - g(\theta)| = |\sigma(v_f^\top \theta) - \sigma(v_g^\top \theta)| \le \sup_z \sigma'(z) \cdot |v_f^\top \theta - v_g^\top \theta|$. The logistic function satisfies $\sigma'(z) = \sigma(z)(1 - \sigma(z)) \le 1/4$ for all z, and by Cauchy–Schwarz, $|(v_f - v_g)^\top \theta| \le \|v_f - v_g\| \cdot \|\theta\|$. □

We now connect the factor model directly to the residual misalignment Δ_t via a continuous notion of distance in skill space.

Definition D.2 (Embedding Wasserstein Distance)**.** *For two distributions π, π' over $\mathcal{F}$, the* embedding Wasserstein-1 distance *is $W_1^V(\pi, \pi') = \inf_{\gamma \in \Gamma(\pi, \pi')} \sum_{f,g \in \mathcal{F}} \gamma(f, g)\|v_f - v_g\|$ where $\Gamma(\pi, \pi')$ is the set of couplings with marginals π and π'. W_1^V captures the geometry of the embedding space: swapping weight between tasks with similar skill profiles ($\|v_f - v_g\|$ small) costs less than swapping between tasks requiring very different skills.*

Proposition D.3 (Geometric Misalignment Bound)**.** *Under the latent factor model, let $\theta^* := \arg\max_\theta u_E(\theta)$ be the social optimum and $\theta_t^* := \arg\max_\theta \mathbb{E}_{f \sim \pi_E^{(t)}}[f(\theta)]$ be the builder's best response under $\pi_E^{(t)}$. The residual misalignment satisfies $\Delta_t \le \frac{1}{4}\big(\|\theta_t^*\| + \|\theta^*\|\big) \cdot W_1^V\Big(\pi_E^{(t)}, \mu\Big)$, where μ is the social relevance distribution over $\mathcal{F}$.*

Proof. Write $u(\theta) := \langle \mu, f(\theta) \rangle = u_E(\theta)$ and $\tilde{u}_t(\theta) := \langle \pi_E^{(t)}, f(\theta) \rangle$. By Definition 3.4, $\Delta_t = u(\theta^*) - u(\theta_t^*)$. The optimality of θ_t^* under $\tilde{u}_t$ gives $\tilde{u}_t(\theta_t^*) \ge \tilde{u}_t(\theta^*)$, so adding $u(\theta^*) - u(\theta_t^*)$ to both sides and rearranging:

$$\Delta_t = u(\theta^*) - u(\theta_t^*) \le \big[\tilde{u}_t(\theta_t^*) - u(\theta_t^*)\big] - \big[\tilde{u}_t(\theta^*) - u(\theta^*)\big] = \big\langle \pi_E^{(t)} - \mu, f(\theta_t^*) - f(\theta^*) \big\rangle. \quad (8)$$

Define the test function $h_t(f) := f(\theta_t^*) - f(\theta^*) = \sigma(v_f^\top \theta_t^*) - \sigma(v_f^\top \theta^*)$. By the triangle inequality and Proposition D.1,

$$|h_t(f) - h_t(g)| \le |\sigma(v_f^\top \theta_t^*) - \sigma(v_g^\top \theta_t^*)| + |\sigma(v_f^\top \theta^*) - \sigma(v_g^\top \theta^*)| \le \tfrac{1}{4}\big(\|\theta_t^*\| + \|\theta^*\|\big)\|v_f - v_g\|,$$

so $\mathrm{Lip}(h_t) \le \frac{1}{4}(\|\theta_t^*\| + \|\theta^*\|)$ in the embedding metric. By the Kantorovich–Rubinstein duality,

$$W_1^V(\alpha, \beta) = \sup_{\mathrm{Lip}(h) \le 1} \left| \sum_f [\alpha(f) - \beta(f)]\, h(f) \right|,$$

applied with $\alpha = \pi_E^{(t)}$, $\beta = \mu$, and the function h_t (after dividing by its Lipschitz constant), bounds (8) by $\mathrm{Lip}(h_t) \cdot W_1^V(\pi_E^{(t)}, \mu)$, yielding the geometric misalignment bound. □

This bound improves on the total variation bound used in Proposition 4.2 (correction-limited term: $\Delta_t \le \mathrm{TV}(\pi_E^{(t)}, \mu)$) by incorporating the geometry of the task space. When the evaluation bias concentrates on tasks that are *close* in embedding space to their counterparts under μ, W_1^V can be much smaller than TV, yielding a tighter bound. Combined with Corollary D.4, the same $(1-\rho)^t$ decay applies, but on a smaller initial value, so convergence is faster in practice.

Corollary D.4 (Correction Decay in Embedding Space)**.** *Under gap-targeted Gaussian correction,*

$$\Delta_t \;\le\; \tfrac{1}{4}\big(\|\theta_t^*\| + \|\theta^*\|\big) \cdot \Big[(1-\rho)^t\, W_1^V\Big(\pi_E^{(0)},\, \mu\Big) + w_g\Big], \tag{9}$$

where $w_g := \sup_{f\in\mathcal{F}} W_1^V(g_{\sigma_c}(\cdot; f),\, \mu)$ is the maximum embedding-Wasserstein deviation of the correction kernel from μ. With finite σ_c, $w_g > 0$ acts as a steady-state floor analogous to $\varepsilon(\rho, \sigma_c)$ in Theorem 4.2; as $\sigma_c \to \infty$ the kernel approaches μ and $w_g \to 0$, recovering pure exponential decay. If $B := \sup_{\theta\in\Theta} \|\theta\|$ is bounded, the prefactor simplifies to $\frac{B}{2}$.

Proof. Write $W_t := W_1^V(\pi_E^{(t)}, \mu)$. We show by induction that $W_t \le (1-\rho)^t W_0 + w_g$. By the Kantorovich–Rubinstein duality, for any distributions π, π' on $\mathcal{F}$ and $\lambda \in [0,1]$,

$$\begin{aligned} W_1^V(\lambda\pi + (1-\lambda)\pi', \mu) &= \sup_{\mathrm{Lip}(h)\le 1} \big|\lambda\langle \pi - \mu,\, h\rangle + (1-\lambda)\langle \pi' - \mu,\, h\rangle\big| \\ &\le \lambda \sup_{\mathrm{Lip}(h)\le 1} |\langle \pi - \mu,\, h\rangle| + (1-\lambda) \sup_{\mathrm{Lip}(h)\le 1} |\langle \pi' - \mu,\, h\rangle| \\ &= \lambda\, W_1^V(\pi, \mu) + (1-\lambda)\, W_1^V(\pi', \mu), \end{aligned}$$

where the inequality is the triangle inequality on absolute values inside the sup. Under Assumption 4.1, the update has the form $\pi_E^{(t)} = (1-\rho)\pi_E^{(t-1)} + \rho\, g_t$, where $g_t = g_{\sigma_c}(\cdot; \nu_t)$ with ν_t the discovered gap location. Hence, $W_t \le (1-\rho)\, W_{t-1} + \rho\, W_1^V(g_t, \mu) \le (1-\rho)\, W_{t-1} + \rho\, w_g$, where the last inequality uses $W_1^V(g_t, \mu) \le w_g$ for every realization of ν_t (by definition of w_g). Unrolling the recursion,

$$W_t \;\le\; (1-\rho)^t\, W_0 + \rho\, w_g \sum_{s=0}^{t-1} (1-\rho)^s \;=\; (1-\rho)^t\, W_0 + w_g\big(1 - (1-\rho)^t\big) \;\le\; (1-\rho)^t\, W_0 + w_g.$$

Substituting into Proposition D.3 yields (9). □

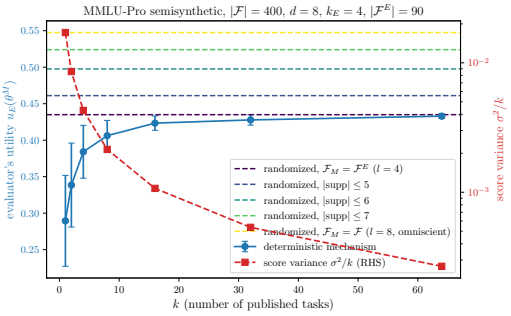

Figure 5: One-shot: transparency k vs. utility and variance. The deterministic mechanism (blue curve) trades off evaluator utility against per-task score variance σ^2/k (red, right axis). Horizontal lines show the randomized single-sample mechanism at different builder awareness levels l; utility increases with l as the builder's $\mathcal{F}_M$ grows toward $\mathcal{F}$.

Next, we discuss some extended empirical results. Figure 10 shows parallel analysis on the MMLU-Pro binary response matrix: observed eigenvalues against the 95th percentile of 20 column-permuted random datasets, using the first-crossing rule to retain $d = 8$ factors. Figure 11 shows the per-task skill embeddings $z_f \in \mathbb{R}_+^d$ after threshold-sparsification ($\tau = 0.1$), with axes mass-sorted (descending column mass) and items reordered by their nested-zero pattern: leftmost block is "factor 7 is zero", next adds "factor 6 is zero", and so on. The clean staircase confirms the z-vectors land in well-separated coordinate subspaces of varying dimension, the property the strategic-evaluation

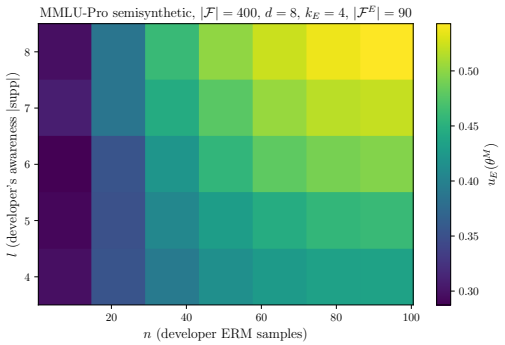

Figure 6: One-shot information asymmetry. Evaluator utility as a heatmap over the builder's ERM sample size n (x-axis) and awareness level $|\mathcal{F}^M|$ (y-axis), holding $k_E = 4$ fixed. Utility falls with both more samples and broader awareness, the largest drop occurring along the diagonal where the builder is both well-resourced and well-informed.

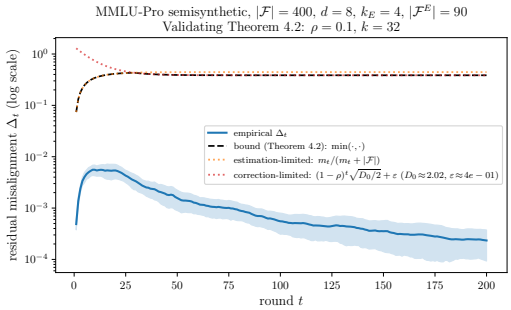

Figure 7: Validating Theorem 4.2 at $(\rho, k) = (0.1, 10)$. Empirical Δ_t (blue, with $\pm 1\sigma$ band) is bounded above by the envelope $\min(m_t/(m_t + |\mathcal{F}|),\ (1-\rho)^t\sqrt{D_0/2} + \varepsilon)$. The decomposition into estimation- and correction-limited terms clarifies which mechanism binds at each t.

game relies on. The main paper uses a semisynthetic instrument pool from MMLU-Pro responses. For comparison, Figures 12–13 reproduce the same experiments on a fully synthetic pool with identical simulation code and identical parameters; only the underlying pool differs. We set $\mathcal{F}^E$ to be a subset of $k \le d$ capabilities: let $e_1, \ldots, e_k$ be the first k standard basis vectors of $\mathbb{R}^d$, and let $\mathcal{F}^E = \{f_z : z \in \mathrm{span}(e_1, \ldots, e_k),\ \|z\|_2 \le 1\}$ (this limits the builder's awareness by zeroing some entries). We vary the model builder's information from $\mathcal{F}^M = \mathcal{F}^E$ to $\mathcal{F}^M = \mathcal{F}$. Throughout, $d = 8$ latent skills with $|\mathcal{F}| = 640$ tasks (stratified across support sizes $\ell \in \{1, \ldots, 8\}$, 80 per layer); the evaluator's awareness uses $k_E = 4$, giving $|\mathcal{F}^E| = 320$. Unless noted, the gap-targeted Gaussian correction uses bandwidth $\sigma_c = 0.4$. We set σ to the identity, so $f(\theta) = z_f^\top \theta$, and $\mu = \mathrm{Uniform}(\mathcal{F})$. All curves report means with $\pm 1\sigma$ bands across 8 random seeds.

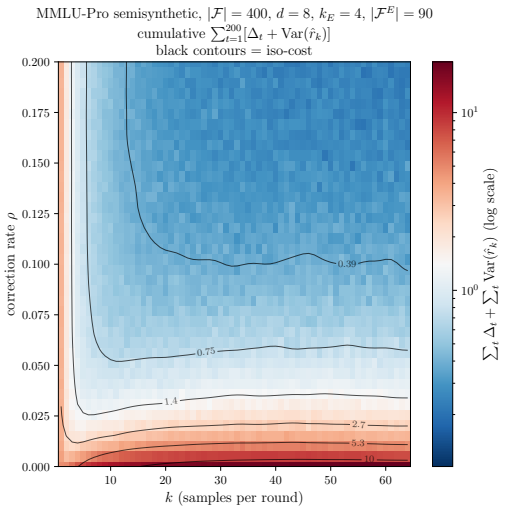

Figure 8: Total-cost variant of Figure 4. Cumulative total cost $\sum_t[\Delta_t + \mathrm{Var}(\hat{r}_k)]$ over the same 40×64 (ρ, k) grid, adding the variance penalty σ^2/k. The total-cost map exhibits an interior valley: large ρ tolerates large k (faster correction lets transparency leak less), but small ρ with large k is the worst regime—all leakage, no recovery.

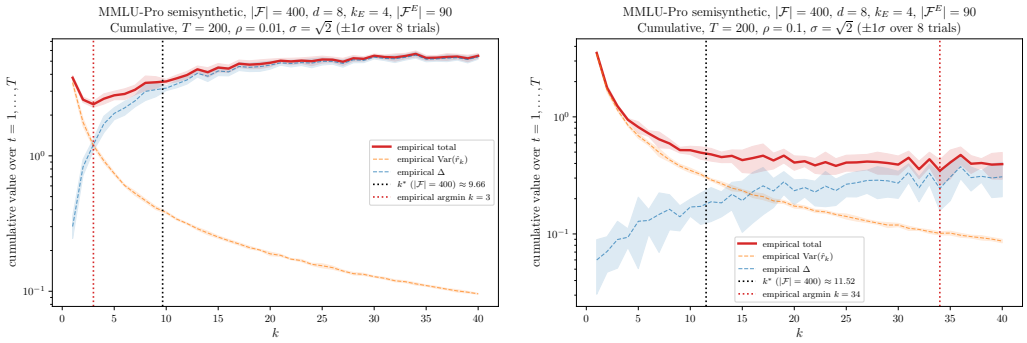

Figure 9: Closed-form k^* vs. empirical optimum at $\rho \in \{0.01, 0.1\}$, $T = 200$ (MMLU-Pro semisynthetic pool). Each panel shows cumulative $\sum_t \Delta_t + \sum_t \mathrm{Var}(\hat{r}_k)$ as a function of k, with vertical lines marking the closed-form k^* from Lemma 4.3 and the empirical argmin. The closed-form heuristic lands close to the empirical valley at $\rho = 0.01$ (left); at $\rho = 0.1$ the loss curve flattens so a wide range of k is near-optimal, and k^* is well within that band. The closed form gives a serviceable design rule that requires only $|\mathcal{F}|, \sigma, \rho$ as inputs.

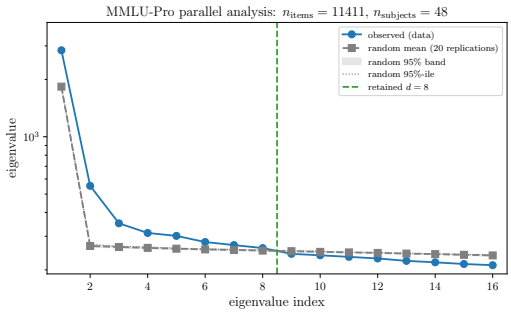

Figure 10: Parallel analysis on the MMLU-Pro response matrix. Observed eigenvalues (blue) versus the mean and 95% band of 20 random-permuted datasets, with the 95th-percentile threshold. Using the first-crossing rule, the retained latent dimension is $d = 8$ (green dashed line).

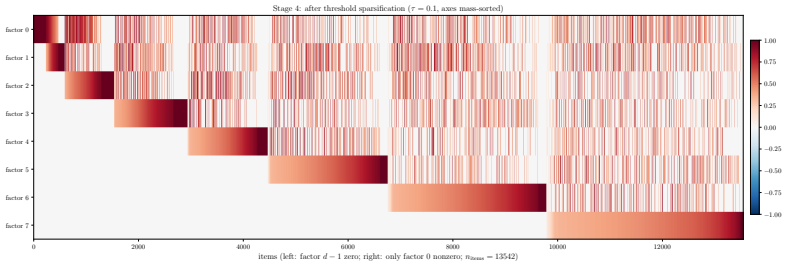

Figure 11: Post-sparsification skill embeddings z_f for all 13,542 MMLU-Pro items. Rows are factors 0–7 (mass-sorted, top = highest column mass); columns are items, ordered left to right by progressively zero high-index factors.

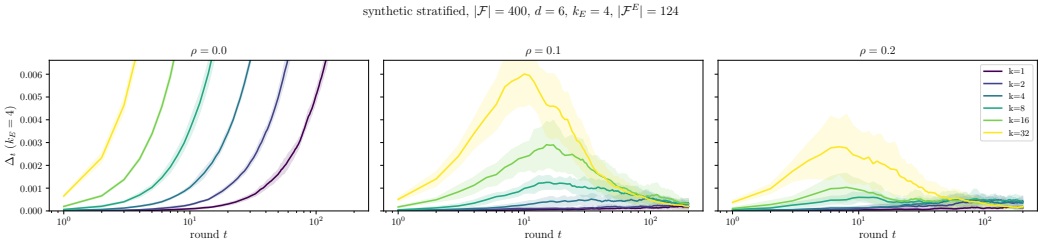

Figure 12: Synthetic baseline for Figure 3: Δ_t over time ($d = 8$, $k_E = 4$, stratified pool).

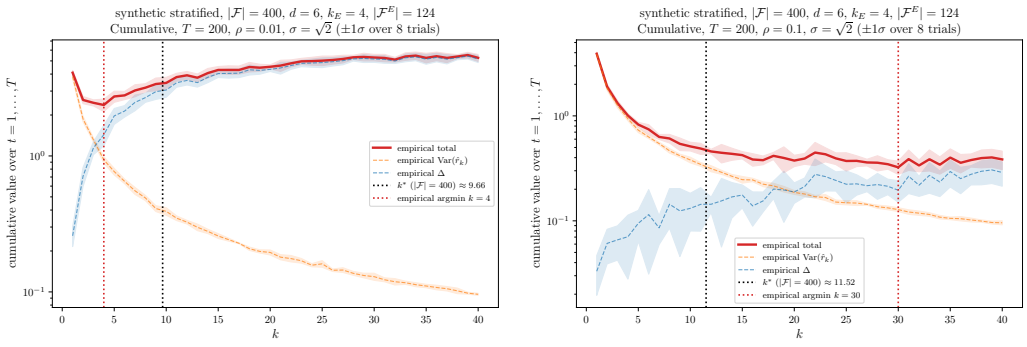

Figure 13: Synthetic baseline for Figure 9: closed-form k^* vs. empirical optimum.

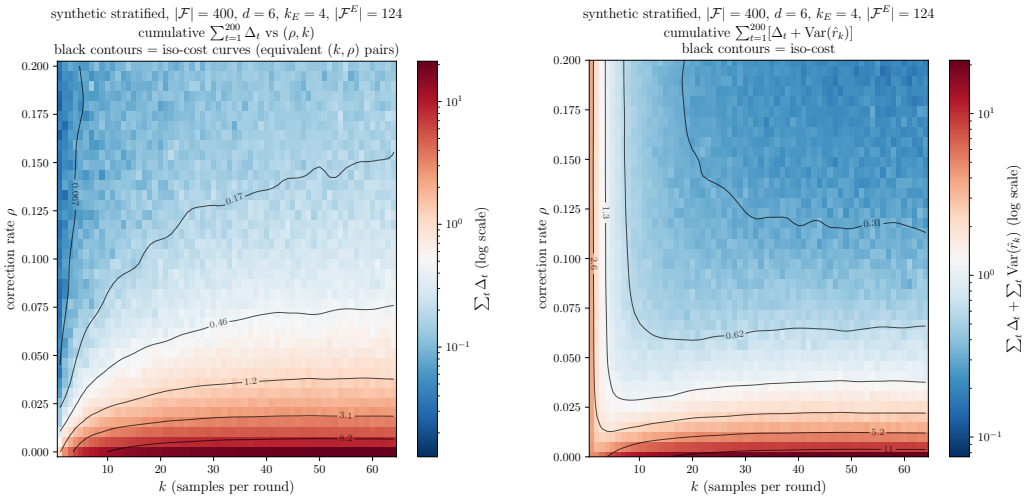

Figure 14: Synthetic baseline for Figure 4 and Figure 8: cumulative Δ_t (left) and total cost (right).

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Introduction state three central claims that map directly to numbered results: (i) deterministic mechanisms cannot incentivize broad task performance (Theorem 2.3); (ii) randomized mechanisms with appropriate developer beliefs restore alignment (Theorem 2.6); (iii) under repeated evaluation, distribution correction at rate ρ controls residual misalignment (Theorem 4.2).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: The Discussion and Conclusion explicitly lists key modelling assumptions and their limitations: a single aggregate evaluator utility, a perfectly rational developer, a parametric (Dirichlet) posterior, and idealized correction mechanisms. Behavioral, multi-objective, and non-parametric extensions are flagged as future work.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theoretical results state their assumptions explicitly. Each Proposition / Theorem / Lemma is numbered and cross-referenced; proofs appear inline in the main text where short, and in the appendix otherwise.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The Simulation section describes the setup; full code, default hyperparameters, and CLI commands are in the supplementary GitHub repository (`https://anonymous.4open.science/r/strategic_evaluation-C1B5/`). Each script seeds NumPy and PyTorch from a single `-seed` flag, and the README documents minimum reproducible run commands per figure.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code is open-sourced under MIT at `https://anonymous.4open.science/r/strategic_evaluation-C1B5/` with a `requirements.txt`, README, and a `-seed`-controlled reproduction path. Real benchmark response data comes from the MMLU-Pro paper.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The Simulation section states the discretization, parameter ranges, and Monte Carlo trial counts for each plot. The supplementary code's CLI exposes all hyperparameters with documented defaults (`-d`, `-n_per_layer`, `-k_E`, `-T`, `-n_trials`, `-rho`, `-gamma`, `-sigma_c`, `-n_factors`, `-fa_max_epochs`, `-fa_lr`); the FIML factor model is fit with Adam at lr=0.05.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All Monte Carlo plots report mean $\pm$ one-standard-deviation envelopes over n_{trials} independent realizations (default 16–40 depending on script). The randomness factors are: random samples from π_E in the repeated game, random samples from $\mathcal{F}_M^{(l)}$ for the ERM experiments, and PyTorch initializations for FIML factor fitting. Each script states the trial count in its console output and writes the per-trial results to NPZ files for re-plotting.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All simulations are CPU-bound and complete on a single workstation core in seconds-to-minutes at default settings: `one_shot.py` $\sim$30s, `repeated.py` $\sim$3min, `semisynthetic.py` $\sim$10s for single-benchmark FIML on `mmlupro` (13.5k items $\times$ 48 subjects), $\sim$1min for the multi-benchmark variant (52k items $\times$ 324 subjects). The supplementary README documents both timings and the optional `-device cuda` flag for GPU-accelerated FIML.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work is theoretical and methodological; it analyzes existing publicly accessible benchmark response data and does not involve human subjects, deceptive practices, or the release of new models with high misuse potential. We have reviewed the NeurIPS Code of Ethics and conform with it in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The Discussion and Conclusion notes the positive impact (better-aligned benchmark mechanisms encourage models with broader real-world capability) and the corresponding caveat (a sophisticated developer with knowledge of π_E may still adapt over multiple rounds; the design lever we identify – the correction rate ρ – is what bounds residual misalignment). The paper releases no model that could be directly misused.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We don't release any new data or model.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: External assets used by the simulation are credited and respect their licenses: PyTorch (BSD-3), NumPy (BSD-3), pandas (BSD-3), Matplotlib (PSF-based), and HuggingFace Hub (Apache-2.0). Each is listed with version constraints in the supplementary code's `requirements.txt`.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The simulation code released at `https://anonymous.4open.science/r/strategic_evaluation-C1B5/` ships with a README documenting all CLI flags, the mapping from each script to its corresponding paper section/figure, install instructions, and a per-script reproducibility recipe. License: MIT.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper involves no crowdsourcing or research with human subjects; the only "subjects" in the analysis are AI models whose responses on existing benchmarks we re-analyze.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper involves no human subjects, so IRB review does not apply.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were not used as any component of the core methodology, theory, or simulation pipeline. Their use was limited to writing/editing assistance, which the policy states does not require declaration.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.