FAILURE MODES IN AI RETRAINING DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern AI systems are increasingly retrained on data generated through interaction with users. Three forces are at play: (i) AI is typically retrained "greedily," ignoring exploration-exploitation tradeoffs, (ii) the users strategically adapt their behavior, and (iii) user intents are not directly observable. We ask whether these dynamics lead to poor outcomes. We study a stylized model, focusing on the "nice" case when the AI and the users have aligned incentives.

We identify two distinct failure modes. First, the system may fail to converge to an optimal Nash equilibrium (of the relevant stage game) due to limited exploration, instead stabilizing at a suboptimal outcome region. This mode is ubiquitous: it happens with a positive probability for *every* problem instance satisfying a minimum non-triviality condition. Second, a non-degenerate subset of problem instances exhibits *model deterioration*, whereby the system converges to an outcome that is strictly worse than the initial state.

1 INTRODUCTION

Modern AI systems are increasingly retrained on data generated through interaction with users. A deployed AI model produces outputs, users respond, and the resulting data is used to update the model (via pre-training and/or post-training). The new model is deployed, and the process repeats.

We identify three key challenges inherent in such retraining loops. First, the system receives feedback only on the current model (a.k.a. *bandit feedback*) and is typically retrained "greedily", without purposeful exploration. Second, users strategically adapt their behavior to the new model, particularly their prompts. So, most of the new data reflects this strategic adaptation, rather than a fixed user behavior (a.k.a., *distribution shift*). Third, the user-AI interaction follows a *prompt interface*: the users do not directly reveal their underlying intents, but instead choose prompts that best elicit useful responses from the current model. As the model changes, users may adapt by changing their prompts, even when their underlying intent remains the same. In particular, the AI system observes only prompt-level feedback and its meaning is driven by distribution shift induced by the current model.

On a high level, we ask whether these challenges may make retraining loops inefficient. We define a stylized retraining dynamics comprising the three challenges. We frame it as a repeated Stackelberg game between the AI designer (who optimizes the AI model given the data) and the user population which collectively adapts to the new model. We focus on the "nice" case when the AI system and the users have aligned incentives.[1] We focus on particularly blunt forms of inefficiency that persist even after a very large number of retraining steps. We show that the retraining dynamics can fail in two distinct ways. First, we may have *optimization failures*, when the system fails to converge to an optimal Nash equilibrium (of the single step of our game). Second, we may have *model deterioration*, a more severe phenomenon when the retraining actively harms performance, converging to an outcome that is strictly worse than the initial state.

Both failure types arise due to the interaction between limited exploration and strategically generated data. Early randomness or structural constraints can prevent the system from ever observing critical information. The prompt interface plays a key role, aggregating feedback across heterogeneous intents and further limiting the system's ability to recover from early errors.

Optimization failures are ubiquitous: we prove that they occur with positive probability for every (non-degenerate) game instance. We lower-bound this probability under several "mechanisms"

[1] Misaligned incentives is a typical cause of inefficiency in theoretical economics, and we rule it out.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

causing it. While in general these lower bounds are quite weak for larger game instances (which mirrors the SoTA in much simpler bandit models), they get stronger with additional structure.

Model deterioration happens for some (but not all) game instances. We show that it happens for a non-degenerate (i.e., positive-measure) set of game instances of arbitrarily large size. In each of these game instances a large fraction of utility (e.g., a half) is lost compared to the initial state. Our constructions are robust to tie-breaking in the retraining dynamics (in fact, they avoid ties altogether).

We provide some additional results on model deterioration. First, we empirically investigate its prevalence (focusing on small game instances) and extend our theoretic constructions so that (in some sense) they capture most of it. Second, we show that the phenomenon of model deterioration hinges upon both bandit feedback and prompt interface. Removing either (or both) from the problem formulation yields retraining dynamics whose utility is weakly increasing over time.

1.1 Related Work

Greedy bandits. Our retraining dynamics can be construed as a bandit problem for the AI designer.[2] Indeed, the latter repeatedly chooses among some actions and receives "bandit feedback" at each step (i.e., only for the chosen actions). However, the bandit problem has a very non-standard structure: the feedback is misaligned with rewards (due to the prompt interface) and both shift over time.

From bandit perspective, the AI designer follows the "greedy" algorithm: at each step, it chooses the estimated best action given the current observations. This algorithm does not explicitly *explore* actions that seem suboptimal, which makes it prone to optimization failures in simple bandit problems (Banihashem et al., 2023). When one has additional structure, the greedy algorithm may fail or succeed (converge on the best action), depending on the structure (Slivkins et al., 2025). In fact, it is known to work quite well for some specific structures (e.g., Kannan et al., 2018; Bastani et al., 2021; Bayati et al., 2020). When/if the algorithm fails, the SoTA bounds on failure probability are extremely weak beyond unstructured bandit problems (Banihashem et al., 2023; Slivkins et al., 2025).

Distribution Shift. While supervised learning usually assumes that training and deployment follow a common distribution, a large literature in machine learning studies a fixed *distribution shift* between the two (surveyed in Moreno-Torres et al., 2012). The distribution shift can be attributed to changes in the input distribution $P(x)$, the label distribution $P(y)$, or the conditional relationship $P(y|x)$, corresponding, resp., to covariate shift (Shimodaira, 2000; Sugiyama et al., 2007; Zadrozny, 2004; Huang et al., 2006; Cortes et al., 2010), label shift (Quiñonero-Candela et al., 2009; Chan & Ng, 2005; Saerens et al., 2002; Lipton et al., 2018; Azizzadenesheli et al., 2019), and concept shift (Schlimmer & Granger Jr, 1986; Widmer & Kubat, 1996; Gama et al., 2014). While distribution shift may admit principled correction methods under some structural assumptions, more general forms, e.g., changes in $P(y|x)$ or support mismatch, can invalidate standard generalization guarantees (David et al., 2010).

While we consider a variant of concept shift (caused by strategic user behavior), this literature focuses on *exogenous* distribution shift (i.e., caused by external factors). The positive results typically assume an upper bound on the magnitude of the shift (e.g., Mohri & Muñoz Medina, 2012; Mazzetto & Upfal, 2023). In contrast, the concept shift in our model is endogenous (driven by the change in the model) and is not necessarily bounded in size. Moreover, this literature designs algorithms for their problem formulations, whereas we analyze a particular "greedy" retraining dynamics.

More recent literature in algorithmic economics studies *endogenous* distribution shift arising from response to the deployed model. *Strategic classification* captures this phenomenon at the level of individual agents, who can modify their features to obtain more favorable predictions subject to a manipulation cost. While the initial paper of Hardt et al. (2016) considered a one-shot interaction, subsequent works have looked at the dynamics arising from the interaction. In particular, Zrnic et al. (2021) study how the frequency with which the model and agent adapt to one another determines the order of play in the Stackelberg equilibrium the dynamic converges to. Unlike ours, this paper does not have (an analogue of) our prompt interface, and allows misaligned incentives.

Performative prediction (Perdomo et al., 2020) posits that the data distribution itself depends on the deployed model. Similar to our setting, this captures endogenous distribution shift and motivates the study of retraining dynamics and stability. In the canonical framework, however, the effect of

[2] Background on multi-armed bandits can be found in Slivkins (2019); Lattimore & Szepesvári (2020).

deployment is represented abstractly through a distribution map, rather than through an explicit model of how agents respond; analyses commonly impose a bounded-sensitivity condition controlling how much the induced data distribution changes when the model changes (Perdomo et al., 2020; Hardt & Mendler-Dünner, 2025). In contrast, we explicitly model users as best-responding through a prompt interface: latent intents are unobserved, different intents may pool at the same prompt, and the designer greedily retrains using the resulting history of prompt-level feedback.

2 Model and Preliminaries

We study a strategic interaction between a population of users and an AI model that is retrained over time by the AI designer (henceforth, *designer*). Users have latent *intents*, but they interact with the model only through a *prompt interface*. Thus, the model does not observe intents directly, but only the prompts issued by users. Crucially, rewards are defined at the intent–answer level, while the designer observes feedback and optimizes only at the prompt-answer level. Below we develop a stylized formulation for this interaction, commenting on idealizations where appropriate.

Low-level interaction: the prompt interface. Let I denote the set of user intents, A the set of answers, and $[K]$ the set of prompts. For non-degeneracy, we assume $|I|, |A|, K \geq 2$. Users interact with the model via the following process, called an *atomic interaction*. A user with some intent $i \in I$ chooses a prompt $p \in [K]$ and submits this prompt to the model. The model returns some answer $a \in A$. Upon receiving this answer, the user experiences a reward, which is observed by the designer. The reward distribution (from which the reward is drawn independently) is determined by the (i, a) pair, regardless of the prompt; let $\mu_{i,a}$ denote its mean. As in much of the literature on online learning, the rewards are bounded and normalized to lie in the $[0, 1]$ interval.

High-level interaction: the retraining game. We focus on high-level interactions between the designer and the user population. We interpret it as a Stackelberg game between these two players, called the (static) *retraining game*. Here, designer chooses an *answer policy* $\pi_{\mathtt{A}} : [K] \to A$, which maps prompts to answers (and represents the AI model). Then the user population chooses a *prompting policy* $\pi_{\mathtt{P}} : I \to [K]$, which maps intents to prompts. These two policies drive atomic interactions: given intent $i \in I$, the prompt is $p = \pi_{\mathtt{P}}(i)$, and the answer is $\pi_{\mathtt{A}}(p) \in A$. The game proceeds in a series of atomic interactions, where each intent occurs exactly once.[3] The two players have aligned utilities, equal to the total user reward. So, expected utility for a policy pair $(\pi_{\mathtt{P}}, \pi_{\mathtt{A}})$ is

$$u(\pi_{\mathtt{P}}, \pi_{\mathtt{A}}) := \textstyle\sum_{i \in I} \mu_{i,\, a(i)}, \text{ where answer} a(i) = \pi_{\mathtt{A}}(\pi_{\mathtt{P}}(i)) \in A. \tag{1}$$

We posit that all intents-to-prompts (resp., prompts-to-answers) mappings are feasible as prompting (resp., answer) policies. This is a significant idealization which side-steps the dependency on the policy classes and allows us to focus on the key challenges discussed in the Introduction. Further, we assume that the users adapt to $\pi_{\mathtt{A}}$ via exact best-response, i.e., the prompting policy satisfies

$$\pi_{\mathtt{P}}(i) \in \arg\max_{p \in [K]} \mu_{i,\, \pi_{\mathtt{A}}(p)} \quad \text{for all intents } i \in I. \tag{2}$$

This is another idealization. In particular, all users with the same intent behave the same, and their behavior is deterministic. Let $\mathtt{BR}(\pi_{\mathtt{A}})$ be the set of all prompting policies that satisfy (2).

The game instance is specified by the sets $I, A, [K]$ and reward matrix $\mathcal{M} := (\, \mu_{i,a} : i \in I, a \in A \,)$.

Learning dynamics. We set up a *repeated* retraining game, whereby the designer learns over time. The essence of this game is as follows. At each round $t \in \mathbb{N}$, the designer chooses a new answering policy $\pi^t_{\mathtt{A}} : [K] \to A$ and the static retraining game happens with $\pi_{\mathtt{A}} = \pi^t_{\mathtt{A}}$. In particular, the user population chooses a prompting policy $\pi^t_{\mathtt{P}} \in \mathtt{BR}(\pi^t_{\mathtt{A}})$. The designer observes feedback from this game, as per our definitions above: for each atomic interaction, it observes the prompt, the answer, and the associated reward (but not the intent). Such feedback represents a limited amount of bandit feedback that is realistically available for retraining.

[3] Having each intent be present deterministically is technically convenient and reflects the reality that a large amount of user data would typically be collected before the model is retrained. While we effectively assume a uniform distribution over intents, non-uniform distributions can be captured by *duplicate intents*: intents i with the same expected rewards $(\mu_{i,a} : a \in A)$. Indeed, all our results seamlessly carry over from duplicate intents to intents that appear correspondingly more often.

The designer chooses $\pi_{\mathtt{A}}^t$ greedily given the current reward estimates. For each prompt-answer pair (p, a), the designer maintains an estimate $\widehat{\mu}_{p,a}^t \in \mathbb{R}$ for the expected reward of giving this answer for this prompt. These estimates are recomputed in the beginning of the round (whether explicitly or implicitly) given all data and the initial knowledge that the designer possesses.[4] This reflects the crucial tenet that the designer perceives rewards at the prompt-answer level, regardless of the prompting policies. The answering policy $\pi_{\mathtt{A}}^t$ is chosen to maximize these estimates:

$$\pi_{\mathtt{A}}^t(p) \in \arg\max_{a \in A} \widehat{\mu}_{p,a}^t \qquad \text{for all prompts } p \in [K]. \tag{3}$$

The exact maximization in Eq. (3) is (of course) a simplification compared to how AI models are pre/post-trained (where one typically has approximation and estimation errors, as well as some randomization). The tie-breaking in (3) (as well as in (2)) can be arbitrary unless stated otherwise.

We represent the initial knowledge as a dataset initially available to the designer. This "initial data" consists of prompt-answer-reward datapoints (in line with designer's prompt-answer perspective). We emphasize that it is (just) a convenient "frequentist" representation of the designer's knowledge and beliefs; we make no claims on whether/how the designer might actually collect such data. Now, the reward estimates can be defined simply as the average rewards over the entire data available to the designer (i.e., both the initial data and the data collected over the past steps). Specifically, $\widehat{\mu}_{p,a}^t$ is the average reward over all datapoints with this (p, a) pair (and we posit at least one such datapoint).

Initial data. To complete our learning dynamics, it remains to define the initial data. This definition may appear cumbersome without the underlying intuition, so we present it gradually.

First, the initial data should be connected to the game instance. The latter is expressed at the intent-answer level, without any dependence on the prompts. To meaningfully capture the prompts, we start with a fixed (and almost-arbitrary) initial prompting policy $\pi_{\mathtt{P}}^{\mathtt{init}} : I \to [K]$ and define the datapoints relative to this policy. Concretely: each datapoint has a form (p, a, r), where $p = \pi_{\mathtt{P}}^{\mathtt{init}}(i)$ is the prompt for some intent i, a is some answer, and r is an independent reward sample for this (i, a) pair. While intent i is not visible, formally the datapoint corresponds to a particular (i, a) pair.

Second, the designer should have some beliefs for every prompt-answer pair (p, a). To this end, we assume that $\pi_{\mathtt{P}}^{\mathtt{init}}$ is surjective (i.e., each prompt is mapped to), and we have at least one datapoint for each intent-answer pair. Third, we'd like to vary how well the initial data is connected to reality. A natural "knob" is the number of datapoints for a particular intent-answer pair. For simplicity, we ensure uniform coverage: each intent-answer pair appears exactly n_0 times, for some fixed n_0. [5]

To summarize, we start with the initial prompting policy $\pi_{\mathtt{P}}^{\mathtt{init}} : I \to [K]$, assumed to be surjective, and $n_0 \in \mathbb{N}$. For each intent-answer pair (i, a), we have n_0 independent prompt-answer-reward datapoints $(\pi_{\mathtt{P}}^{\mathtt{init}}(i), a, r)$. An instance of the repeated game consists of the "static" game instance and the $(\pi_{\mathtt{P}}^{\mathtt{init}}, n_0)$ pair to define the initial data.

Equilibrium benchmark and regret. We evaluate the learning dynamics against the optimal (i.e., utility-maximizing) Nash equilibrium of the static retraining game.[6] Let $u^* = u^*(I, A, K, \mathcal{M})$ denote the optimal Nash value. This is our benchmark. *Regret* of the learning dynamics is defined as

$$\mathtt{regret}(T) := \textstyle\sum_{t \in [T]} \left(u^* - u(\pi_{\mathtt{P}}^t, \pi_{\mathtt{A}}^t) \right). \tag{4}$$

This coincides with designer's *policy regret*, a standard notion in repeated games (see Appendix B.1).

3 Optimization Failures

In this section we show that the system exhibits an *optimization failure* — gets permanently stuck in a suboptimal behavior — with some positive probability independent of the time horizon. A common reason this happens in our results is that bandit feedback combined with greedy updating leaves useful policies or answers insufficiently explored, so that the dynamics never discover behavior required for an optimal equilibrium. Throughout this section, rewards are independent Bernoulli samples with means $\mu_{i,a} \in [0, 1]$. Additional boundedness assumptions are stated when needed.

[4] We avoid the term "beliefs" at this point since it has a particular default meaning in theoretical economics.

[5] As in Footnote 3, going over all intents corresponds to going over all *users*.

[6] Due to aligned utilities, $(\pi_{\mathtt{P}}, \pi_{\mathtt{A}})$ is an optimal Nash if and only if it maximizes $u(\pi_{\mathtt{P}}, \pi_{\mathtt{A}})$; see Lemma B.1.

Definition 3.1. Optimization failure*, denoted* FAIL*, is the event that for some $\eta > 0$ (called* failure margin*) and any sufficiently large round T we have $\frac{1}{T}\sum_{t\in[T]} u(\pi_{\mathtt{P}}^t, \pi_{\mathtt{A}}^t) \le u^* - \eta$.*

As shown in Appendix B.1, optimization failure is equivalent to asymptotically linear regret on the realized trajectory. In particular, $\mathbb{P}[\texttt{FAIL}] > 0$ implies $\mathbb{E}[\texttt{regret}(T)] = \Omega(T)$.

Our first result is that $\mathbb{P}[\texttt{FAIL}] > 0$ for all problem instances, under a minimal non-degeneracy condition. We rule out a degenerate case when every answer policy (with a best-response prompting policy) is an optimal Nash equilibrium.

Theorem 3.2. *Suppose that $\mu_{i,a} \in [\alpha, 1-\alpha]$ for every $i \in I$ and $a \in A$, for some constant $\alpha > 0$, and that some answer policy is not a part of any optimal Nash equilibrium. Then:*

$$\mathbb{P}[\texttt{FAIL}] \ge \exp\left(-\Theta\big(n_0 \cdot |I| \cdot |A|\big)\right).$$

Proof Sketch. Consider an initialization event in which a fixed decoy policy appears strictly optimal at every prompt, while every deviation has estimate zero. Independence gives the stated probability bound. Once the decoy is selected, bandit feedback updates only played actions, so the zero-valued deviations remain unobserved while the selected actions retain positive empirical value. Hence the greedy dynamics remain permanently locked on the decoy policy. □

While the failure probability in Theorem 3.2 is quite small, it suffices to imply linear regret. In the rest of this section we provide stronger lower bounds on failure probability under additional structure.

Failure via Required Answers. First, we posit a *required answer*, an answer which must be mapped to (from some prompt) in any optimal Nash equilibrium. Required answers capture settings where a particular type of response is indispensable for optimal performance. For example, a tutoring system sometimes needs to return the correct answer rather than a hint; in a medical assistant, some intents may require a referral rather than generic advice.

Definition 3.3. *A* required answer *is an answer $a \in A$ that is mapped to in every optimal Nash equilibrium $(\pi_{\mathtt{P}}, \pi_{\mathtt{A}})$, in the sense that $\pi_{\mathtt{A}}(p) = a$ for some prompt $p \in [K]$.*

A required answer simplifies sufficient conditions for a failure event. It suffices if the initial data is such that this *particular* answer has zero estimated value for each prompt, and some other answer has estimated value > 0 for each prompt. Then, $\mathbb{P}[\texttt{FAIL}]$ remains exponentially small in $|I| \cdot n_0$, but no longer in $|A|$.

Theorem 3.4. *Suppose that $\mu_{i,a} \in [\alpha, 1-\alpha]$ for every $i \in I$ and $a \in A$, for some constant $\alpha > 0$, and that the problem instance has some required answer. Then $\mathbb{P}[\texttt{FAIL}] \ge \exp\left(-\Theta\big(n_0 \cdot |I|\big)\right)$.*

Proof Sketch. The existence of a required answer sharpens the argument in Theorem 3.2 by collapsing the failure event to a single critical action. Instead of needing to misestimate an entire policy, it suffices that the required answer a^* is underestimated everywhere, while some alternative answer appears viable for each prompt. This significantly simplifies the probabilistic structure; the dynamics now only need to learn to avoid a^*. If a^* is never selected in the initial step, then it is never updated, and its estimate remains identically zero forever. □

Remark 3.5. *The failure margin in Theorem 3.4 is $\eta \ge u^* - u^*(I, A \setminus \{a^*\}, K, \mathcal{M})$, i.e., the difference (in optimal Nash value) between including and not including a^*.*

Failure via Biclustering. Now let us consider a *biclustering* structure, where intents (resp., answers) are partitioned into subsets (*clusters*) and there's a bijection between intent clusters and answer clusters.[7] The meaning of this bijection is that all answers within a given answer cluster are at least somewhat relevant to all intents in the corresponding intent cluster, and not relevant to all other intent clusters. For example, a general-purpose assistant may serve users with qualitatively different intents: debugging code, writing prose, solving math problems, or seeking medical information. Within each of these categories, many answers may be useful, while answers from other categories may be irrelevant. The formal definition of biclustering (adapted to our problem formulation) is as follows.

[7]This type of structure is sometimes called a *block-diagonal biclustering* of the intents-answers matrix.

Definition 3.6. *A* bicluster *is a pair (I_b, A_b), where $I_b \subset I$ and $A_b \subset A$. It is* Δ-robust *if for all intents $i \in I_b$ and all answers $a \in A$ we have $\mu_{i,a} \geq \Delta$ whenever $a \in A_b$ and $\mu_{i,a} = 0$ otherwise.*

Definition 3.7. *A* biclustering *is a collection of biclusters (I_b, A_b), $b \in [B]$ such that intent subsets $(I_b : b \in [B])$ partition I and answer subsets $(A_b : b \in [B])$ partitions A. The biclustering is called* Δ-robust *if so are all the biclusters.*

Given a biclustering, we zoom in on policies that are "consistent" with it, in the following sense:

Definition 3.8 (Consistency)**.** *Fix a biclustering $\mathcal{B} = ((I_b, A_b) : b \in [B])$. A prompting policy $\pi_{\mathtt{P}} : I \to [K]$ is* consistent with $\mathcal{B}$ *if intents from different biclusters map to disjoint sets of prompts, i.e., if the subsets $\pi_{\mathtt{P}}(I_b) \subset [K]$, $b \in B$ are mutually disjoint. A policy pair $(\pi_{\mathtt{P}}, \pi_{\mathtt{A}})$ (where $\pi_{\mathtt{A}} : [K] \to A$ is an answer policy) is* consistent with $\mathcal{B}$ *if $\pi_{\mathtt{P}}$ is consistent with $\mathcal{B}$ and furthermore*

$$\pi_{\mathtt{A}}\big(\pi_{\mathtt{P}}(I_b)\big) \subseteq A_b \quad \textit{for every bicluster } b \in [B].$$

Biclustering allows for an improved guarantee where the lower bound on $\mathbb{P}[\mathtt{FAIL}]$ decays exponentially (only) in the size of a particular intent cluster, rather than in the total number of intents.

Theorem 3.9. *Suppose the problem instance admits a Δ-robust biclustering (I_b, A_b), $b \in [B]$, for some constant $\Delta > 0$, and the initial prompting policy $\pi_{\mathtt{P}}^{\mathrm{init}}$ is consistent with it. Moreover, assume there is a required answer $a^* \in A_{b^*}$ and the corresponding bicluster b^* satisfies $|A_{b^*}| \geq 2$ and $\mu_{i,a} \leq 1 - \alpha$ for some constant $\alpha > 0$ and every $(i, a) \in I_{b^*} \times A_{b^*}$. Finally, suppose we have enough initial data: $n_0 = \Omega(\log K)$. Then $\mathbb{P}[\mathtt{FAIL}] \geq \exp\left(-\Theta(n_0 |I_{b^*}|)\right)$.*

Remark 3.10. *A more detailed theorem statement (e.g., with an explicit dependence on Δ) is found in Theorem C.1. The failure margin in Theorem 3.9 is the same as in Remark 3.5.*

Proof Sketch. The biclustering structure localizes the anti-concentration requirement by introducing dynamical invariance along the biclustering boundaries. Once the system is consistent with the biclustering, it continues to be so indefinitely (Lemma C.3). This effectively decomposes the global learning problem into independent subproblems at the bicluster level. Thus, it suffices to have a failure which suppresses the required answer within a single bicluster.

The proof identifies three events whose joint occurrence guarantees failure: (i) a structural event ensuring that behavior remains confined within biclusters, (ii) an anti-concentration event ensuring that the required answer is initially underestimated, and (iii) a stability event ensuring that a suboptimal answer remains attractive within a bicluster. Therefore, biclustering restricts exploration across parts of the action space, while bandit feedback prevents recovery once a required answer is eliminated. The interaction between these two effects produces failure with probability that depends only on the size of a single bicluster, rather than the full system. □

A *stronger* Failure via Biclustering. Whenever a biclustering exists, one naturally expects that optimal Nash equilibria would be consistent with it. Perhaps surprisingly, this need not be the case. In fact, we may have a biclustering such that no optimal Nash equilibrium is consistent with it. (We call it *Nash-devious*; we explain below when/why it may exist.) Then we derive a much stronger result: optimization failure happens with probability close to 1.

Theorem 3.11. *Suppose the problem instance admits a Δ-robust Nash-devious biclustering (I_b, A_b), $b \in [B]$ and that the initial prompting policy $\pi_{\mathtt{P}}^{\mathtt{init}}$ is consistent with it. Then*

$$\mathbb{P}[\mathtt{FAIL}] \geq 1 - \exp(-\Delta n_0 A_{\min} + \log K)\,, \textit{ where } A_{\min} := \min_{b \in [B]} |A_b|.$$

In particular, with enough initial data, namely $n_0 \geq (c \log K)/(\Delta A_{\min})$ for some constant $c > 1$,

$$\mathbb{P}[\mathtt{FAIL}] \geq 1 - K^{1-c} = 1 - o(1).$$

Proof sketch. For each prompt p, let b_p be the unique bicluster such that $p \in \pi_{\mathtt{P}}^{\mathtt{init}}(I_{b_p})$. Consider the event $\mathcal{E}$ that, for every prompt p, there exists some answer $a \in A_{b_p}$ whose initial empirical reward at p is strictly positive. Then $\mathbb{P}[\mathcal{E}] \geq 1 - Ke^{-\Delta n_0 A_{\min}}$. On this event, Lemma C.3 implies that the dynamics remain consistent with the biclustering forever. Since the biclustering is Nash-devious, every policy pair reached by the dynamics is bounded away from u^*, yielding optimization failure.

Remark 3.12. *Theorem 3.11 does not assume a required answer. The failure margin here is (at least) u^* minus the largest-value policy pair $(\pi_{\mathtt{P}}, \pi_{\mathtt{A}})$ that is consistent with the biclustering.*

We will first show an example of a Nash-devious game, and then a more general characterization.

Theorem 3.13. *Fix $\Delta \in (0, 1/2)$. There is a game instance with a Δ-robust Nash-devious biclustering.*

Proof. Choose $\gamma \in (\Delta, 1/2)$. Consider a $3 \times 2 \times 3$ game instance with reward matrix as follows. (Here, intents and answers are indexed as, resp., i_1, i_2, i_3 and a, b, c).

μ_{ia}	a	b	c
i_1	γ	0	0
i_2	0	1	γ
i_3	0	γ	1

Consider the biclustering $\boldsymbol{B}$ given by $I_1 = \{i_1\}$, $A_1 = \{a\}$ and $I_2 = \{i_2, i_3\}$, $A_2 = \{b, c\}$. This biclustering is Δ-robust: every within-bicluster reward is at least $\gamma > \Delta$, while every cross-bicluster reward is zero.

We now compare policies that are consistent with the biclustering to the global optimum. Since there are two prompts and two biclusters, any consistent policy must devote one prompt to each bicluster. The prompt assigned to I_1 returns a, contributing γ. The prompt assigned to I_2 can return either b or c, in either case contributing $1 + \gamma$. Thus every consistent policy pair has utility at most $1 + 2\gamma < 2$.

On the other hand, the designer can map the two prompts to b and c. Then i_2 chooses the prompt returning b, i_3 chooses the prompt returning c, and i_1 receives zero, for total utility 2. Moreover, no policy can obtain utility greater than 2: using answer a leaves only one prompt for $\{b, c\}$ and therefore gives utility at most $1 + 2\gamma < 2$, while omitting a gives zero utility to i_1 and at most one unit to each of i_2 and i_3. Hence the optimal Nash value is 2, while every policy pair consistent with the biclustering has strictly smaller utility. Therefore the biclustering is Nash-devious. □

We now characterize when a biclustering $\mathcal{B}$ is Nash-devious. We compare two variants of the optimal Nash value: the optimal Nash value of the game instance with bicluster b removed, denoted $u_b^*(\mathcal{B})$, and the largest utility $u(\pi_{\mathtt{P}}, \pi_{\mathtt{A}})$ achievable by a policy pair consistent with $\mathcal{B}$, denoted $u^*_{\mathtt{cons}}(\mathcal{B})$. [8]

Theorem 3.14. *A biclustering $\mathcal{B}$ is Nash-devious if and only if for some bicluster $b \in [B]$ we have*

$$u_b^*(\mathcal{B}) > u^*_{\mathtt{cons}}(\mathcal{B}).$$

Intuition. The forward direction here is immediate: if $u_b^*(\mathcal{B})$ provides higher utility than the best consistent Nash $u^*_{\mathtt{cons}}(\mathcal{B})$, the best Nash (which can only have higher utility than $u_b^*(\mathcal{B})$) cannot be consistent. For the backwards direction, if an inconsistent optimum still uses every bicluster, one can reassign off-cluster intents to make it consistent without decreasing utility. Therefore a genuinely better inconsistent policy must abandon some bicluster entirely.

4 Model Deterioration

We show a stronger pathology than failure to reach the optimal Nash value: the dynamics may converge to a limit utility that is strictly lower than the initial utility. We call it *model deterioration*.

Definition 4.1. Model deterioration *is the event that limit $W_\infty := \lim_{T\to\infty} \frac{1}{T} \sum_{t\in[T]} u(\pi_{\mathtt{P}}^t, \pi_{\mathtt{A}}^t)$ exists and is strictly smaller than $W_0 := u(\pi_{\mathtt{P}}^{\mathtt{init}}, \pi_{\mathtt{A}}^1)$, the utility of the initial policy pair.*

We show that model deterioration happens even when the rewards are *deterministic*, which we assume throughout this section. This implies that (other than arbitrary tiebreaking, which we address below) the entire learning dynamics are deterministic for a given problem instance, including whether model deterioration happens or not.

Existence and non-degeneracy. We show that model deterioration happens for some (but not all) game instances.[9] In fact, we show that it is a common case, in the sense that it happens for a non-degenerate set of game instances (in the measure-theoretic sense defined below) and for arbitrarily large game sizes. Our constructions are robust to tie-breaking, in the following strong sense:

[8] In our notation, $u_b^*(\mathcal{B}) = u^*(I \setminus I_b, A \setminus A_b, K, \mathcal{M})$. That is, we remove intent cluster I_b from intents and answer cluster A_b from answers. Also, note that $u^*_{\mathtt{cons}}(\mathcal{B}) < u^*$ by definition of "Nash-devious biclustering".

[9] This is unlike optimization failures, which (with some probability) happen for essentially all game instances.

Definition 4.2. *Model deterioration is* robust *if the learning dynamics has no ties in (2) and (3).*

Our results are presented as follows. First, we give a concrete example of a small-sized game with robust model deterioration. Next, we extend this example to a non-degenerate set of game instances of the same size. Finally, we extend this construction via block embedding to games of arbitrarily large size. Throughout, *game size* refers to the triple $(|I|, K, |A|)$. We use *robust deterioration* as a shorthand for robust model deterioration.

Theorem 4.3. *There is a game instance of size* $3 \times 2 \times 3$ *which exhibits robust deterioration.*

Consider the following reward matrix (where intents and answers are, resp., i_1, i_2, i_3 and a_1, a_2, a_3).

μ_{ia}	a_1	a_2	a_3
i_1	1.00	0.98	0.99
i_2	0.05	0.50	0.00
i_3	0.00	0.00	0.50

The initial prompting policy $\pi_{\mathtt{P}}^{\mathtt{init}}$ maps intents i_1, i_2 to prompt p_1, and intent i_3 to prompt p_2.

One interpretation is a learning assistant with two prompt modes and three answer styles. Prompt p_1 is a generic "explain this to me" prompt, while p_2 is a "quick practice" prompt. The three answer styles are a high-level explanation a_1, a step-by-step explanation a_2, and a short drill response a_3. The three intents correspond to, resp., an advanced user, a beginner, and a practice-oriented user.

Initially, the advanced user and the beginner are pooled at the generic prompt p_1. The designer therefore averages their rewards and chooses the step-by-step answer a_2 for prompt p_1, which is a good compromise for that pool. The practice prompt p_2 receives the drill response a_3. That is, the designer chooses answer policy $\pi_{\mathtt{A}}^{1} = (a_2, a_3)$. This initial outcome has utility $W_0 = 0.98 + 0.50 + 0.50 = 1.98$.

The deterioration mechanism hinges on a subtle interaction between averaging and selective feedback. Once the designer chooses $\pi_{\mathtt{A}} = (a_2, a_3)$, the advanced user slightly prefers the drill response to the step-by-step response and temporarily moves to prompt p_2. Thus, prompting policy $\pi_{\mathtt{P}}^{\mathtt{tmp}} = (p_2, p_1, p_2)$ is chosen. The generic prompt p_1 is then observed only for the beginner user, so the designer repeatedly receives feedback 0.50 for the played answer a_2 at prompt p_1. Meanwhile, the estimate of the unplayed answer a_1 at prompt p_1 remains frozen at its initial pooled value $(1 + 0.05)/2 = 0.525$. Eventually the empirical estimate for answer a_2 falls below this frozen alternative, and the designer switches $\pi_{\mathtt{A}}(p_1)$ from a_2 to a_1. Thus, the answer policy becomes $\pi_{\mathtt{A}}^{\mathtt{lim}} = (a_1, a_3)$. After the switch, the human immediately returns to the initial prompting policy $\pi_{\mathtt{P}}^{\mathtt{init}}$. Then the dynamics stabilize at $(\pi_{\mathtt{A}}^{\mathtt{lim}}, \pi_{\mathtt{P}}^{\mathtt{init}})$. Thus, we have the following trajectory:

$$\pi_{\mathtt{P}}^{\mathtt{init}} \rightsquigarrow (\pi_{\mathtt{A}}^{1}, \pi_{\mathtt{P}}^{\mathtt{tmp}}) \rightsquigarrow (\pi_{\mathtt{A}}^{\mathtt{lim}}, \pi_{\mathtt{P}}^{\mathtt{init}}), \tag{5}$$

where the middle policy pair may persist for multiple rounds before the designer switches to $\pi_{\mathtt{A}}^{\mathtt{lim}}$.

The designer's eventual switch at p_1 is harmful. The human returns to the initial prompting policy $\pi_{\mathtt{P}}^{\mathtt{init}}$, but the generic prompt now receives the high-level answer. The advanced user is slightly better off, while the beginner is much worse off. The limiting utility is $W_\infty = 1+0.05+0.50 = 1.55 < W_0$.

The full proof, including explicit matrices for feedback and estimates, is in Appendix D.1.

Now we extend this construction to (more) generic game instances. A set of game instances is called *non-degenerate* if their reward matrices form a subset of $[0, 1]^{|I|\cdot|A|}$ of positive Lebesgue measure.[10]

Theorem 4.4. *A non-degenerate set of game instances of size* $3 \times 2 \times 3$ *exhibits robust deterioration.*

Proof Sketch. The argument in Theorem 4.3 (with model deterioration that follows the trajectory in Eq. (5)) carries over to a generic reward matrix μ which satisfies a certain set of inequalities:

$$\mu_{12} + \mu_{22} > \mu_{11} + \mu_{21} > \mu_{13} + \mu_{23}, \tag{6}$$

$$\mu_{11} > \mu_{13} > \mu_{12} \qquad \min(\mu_{21}, \mu_{22}) > \mu_{23} \qquad (\mu_{11} + \mu_{21})/2 > \mu_{22}, \tag{7}$$

$$\mu_{33} > \max(\mu_{31}, \mu_{32}) \qquad (\mu_{13} + \mu_{33})/2 > \mu_{31} \qquad (\mu_{13} + \mu_{33})/2 > \mu_{32}. \tag{8}$$

[10] Here, reward matrices are interpreted as points in $[0, 1]^{|I|\cdot|A|}$. Note that if there is a non-degenerate set of game instances then there is one in which all instances have the same game size and $\pi_{\mathtt{P}}^{\mathtt{init}}$.

Their meaning is as follows: (6) makes answer a_2 be initially chosen for prompt p_1, (7) governs the users' rerouting and return on the p_1 side and ensures that, after finitely many transient rounds, the frozen estimate of a_1 overtakes the estimate of a_2, and (8) keeps answer a_3 stable at prompt p_2. The details are in Appendix D.2; the absence of ties in the learning dynamics (i.e., robustness) needs to be verified via a separate argument. These linear inequalities are simultaneously satisfiable, as witnessed by Theorem 4.3, and therefore their intersection forms a subset of a positive Lebesgue measure. □

Finally, we extend robust deterioration to arbitrarily large games. For this, we embed D disjoint copies of the $3 \times 2 \times 3$ construction as blocks inside the large game, for any given D (see Appendix D.4).

Theorem 4.5. *For every integer $D \geq 1$, and every $|I| \geq 3D$ and $|A| \geq 3D$, there is a non-degenerate set of game instances of size $(|I|, 2D, |A|)$ that exhibit robust model deterioration.*

Remark 4.6. *What is the deterioration margin, $W_0 - W_\infty$? Fixing any $\epsilon > 0$, the constructions in Theorems 4.4 and 4.5 can ensure $W_0 - W_\infty \geq 1/2 - \epsilon$.*

Prevalence and characterization. Now that we've established model deterioration as a phenomenon, we ask how prevalent it is empirically and whether our theoretical constructions fully capture it.

Our empirical investigation focuses on $3 \times 2 \times 3$ game instances. We estimate the prevalence of model deterioration among all game instances of this size. We generate 10^5 game instances independently and uniformly, and run them for $T = 10^5$ rounds. [11] We track both robust and non-robust model deterioration (for the latter, the ties are broken in favor of a lowest-index maximizer). Convergence is detected rigorously according to a certain rule; the game instance is marked as 'unresolved' if this rule never fires. All details are in Appendix D.7. The empirical prevalence rates are given in Table 1.

all model deterioration	$5.85\% \pm 0.08\%$
robust deterioration	$0.95\% \pm 0.03\%$
non-robust deterioration	$4.89\% \pm 0.07\%$
unresolved within the time horizon	under 0.01%.

Table 1: Prevalence rates for model deterioration (among all $3 \times 2 \times 3$ game instances).

Remark 4.7. *Our notion of "robust" deterioration is tractable experimentally. However, it may leave out game instances in which model deterioration happens regardless of how the ties are broken.*

The sufficient conditions in Theorem 4.4 explain only a small fraction of robust deterioration (under 6%, for the $3 \times 2 \times 3$ game instances). However, we define a larger, structurally similar family of game instances which captures $> 95\%$ of robust deterioration. Essentially, we allow game instances with trajectory as in Eq. (5), but with arbitrary policies $\pi_{\mathtt{P}}^{\mathtt{init}}, \pi_{\mathtt{P}}^{\mathtt{tmp}}, \pi_{\mathtt{A}}^{\mathtt{1}}, \pi_{\mathtt{A}}^{\mathtt{lim}}$. Each choice of policies entails a separate system of inequalities akin to Eqns. (6) to (8). The details are in Appendix D.3.

Model Deterioration requires prompt interface *and* bandit feedback. Finally, we show that model deterioration hinges on both prompt interface and bandit feedback. If either tenet is removed, the utility of the learning dynamics weakly increases over time. In more detail, we consider four game variants: (bandit feedback or full feedback) and (prompt interface or a matrix game).[12] The retraining game corresponds to "bandit feedback" and "prompt interface". We show that the utility weakly increases over time for the other three variants, see Table 2 (details in Appendix D.8).

	Prompt Interface	Matrix Game
Bandit Feedback	model deterioration	weak monotonicity
Full Feedback	weak monotonicity	weak monotonicity

Table 2: Without prompt interface and/or bandit feedback: utility weakly increases over time

[11] That is: each reward μ_{ia} is chosen independently and u.a.r. from the $[0, 1]$ interval. We set $\pi_{\mathtt{P}}^{\mathtt{init}} = (p_1, p_1, p_2)$ (this is w.l.o.g. for surjective mappings). The implementation code is available at `https://anonymous.4open.science/r/greedy-dynamics-experiments-DC93/`.

[12] "Full feedback" means that the designer observes rewards for all answer policies. "Matrix game" means that there are no intents: users directly choose prompts, and rewards are defined at the prompt-answer level.

AI USE STATEMENT

In this work, we used generative AI tools for assistance in the writing of proofs. We have not used generative AI tools for help developing theoretical models or frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, implementing methods, proposing or refining hypotheses, supporting qualitative and thematic data analysis, design and feedback on research methodology or experiments or interpreting results. Generating synthetic data sets, cleaning and reformatting datasets and assistance with translation are not applicable to this work. Additionally, we used generative AI tools for creating or editing software code and identifying relevant literature. We have reviewed all AI-assisted work. The authors have verified the LLM-generated code and tested it for correctness. All AI-assisted proof writing was formalizing ideas that were fully conceptualized and outlined by the authors, and the resulting text was further verified by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Complete proofs of the theoretical claims are included in the appendix. Furthermore, for the experimental results, an anonymous link to the code used is provided.

REFERENCES

Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. *arXiv preprint arXiv:1903.09734*, 2019.

Kiarash Banihashem, MohammadTaghi Hajiaghayi, Suho Shin, and Aleksandrs Slivkins. Bandit social learning under myopic behavior. In *37th Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Hamsa Bastani, Mohsen Bayati, and Khashayar Khosravi. Mostly exploration-free algorithms for contextual bandits. *Management Science*, 67(3):1329–1349, 2021. Working paper available on `arxiv.org` since 2017.

Mohsen Bayati, Nima Hamidi, Ramesh Johari, and Khashayar Khosravi. Unreasonable effectiveness of greedy algorithms in multi-armed bandit with many arms. In *33rd Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Yee Seng Chan and Hwee Tou Ng. Word sense disambiguation with distribution estimation. In *IJCAI*, volume 5, pp. 1010–5, 2005.

Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. *Advances in neural information processing systems*, 23, 2010.

Shai Ben David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 129–136. JMLR Workshop and Conference Proceedings, 2010.

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37, 2014.

Moritz Hardt and Celestine Mendler-Dünner. Performative prediction: Past and future. *Statistical Science*, 40(3):417–436, 2025.

Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, pp. 111–122, 2016.

Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19, 2006.

Sampath Kannan, Jamie Morgenstern, Aaron Roth, Bo Waggoner, and Zhiwei Steven Wu. A smoothed analysis of the greedy algorithm for the linear contextual bandit problem. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2231–2241, 2018.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, Cambridge, UK, 2020.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pp. 3122–3130. PMLR, 2018.

Alessio Mazzetto and Eli Upfal. An adaptive algorithm for learning with unknown distribution drift. *Advances in Neural Information Processing Systems*, 36:10068–10087, 2023.

Mehryar Mohri and Andres Muñoz Medina. New analysis and algorithm for learning with drifting distributions. In *International Conference on Algorithmic Learning Theory*, pp. 124–138. Springer, 2012.

Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern recognition*, 45(1):521–530, 2012.

Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *International Conference on Machine Learning*, pp. 7599–7609. PMLR, 2020.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *When Training and Test Sets Are Different: Characterizing Learning Transfer*, pp. 3–28. 2009.

Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1):21–41, 2002.

Jeffrey C Schlimmer and Richard H Granger Jr. Incremental learning from noisy data. *Machine learning*, 1(3):317–354, 1986.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Aleksandrs Slivkins. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, November 2019. Published with *Now Publishers* (Boston, MA, USA). Also available at `https://arxiv.org/abs/1904.07272`.

Aleksandrs Slivkins, Yunzong Xu, and Shiliang Zuo. Greedy algorithm for structured bandits: A sharp characterization of asymptotic success / failure. In *39th Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5), 2007.

Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1):69–101, 1996.

Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 114, 2004.

Tijana Zrnic, Eric Mazumdar, Shankar Sastry, and Michael Jordan. Who leads and who follows in strategic classification? *Advances in neural information processing systems*, 34:15257–15269, 2021.

APPENDICES

A Retraining Dynamics: A Summary **13**

B Details from Preliminaries (Section 2) **14**

B.1 Additional Details on Regret 14

C Proofs for Optimization Failures (Section 3) **15**

C.1 Proof of Theorem 3.2 15

C.2 Proof of Theorem 3.4 16

C.3 Proof of Theorem 3.9 17

C.4 Proof of Lemma C.3 19

C.5 Proof of Theorem 3.14 20

C.6 Proof of Theorem 3.11 21

D Proofs for Model Deterioration (Section 4) **21**

D.1 Concrete Example 21

D.2 Canonical Deterioration Region 23

D.3 Volume and Relabelings 24

D.4 Embedding in Larger Games 25

D.5 One-Switch Certificates 26

D.6 Worst-Case Harm 27

D.7 Experimental Conventions 28

D.8 Additional Content from Section 4 28

D.9 Bandit Feedback without Prompt Aggregation 29

D.10 Full Feedback 30

A RETRAINING DYNAMICS: A SUMMARY

Learning Dynamics

Fix an initial prompting policy $\pi_{\mathtt{P}}^{\mathtt{init}}$; assume it is surjective.
Let $I_p^t := \{\, i \in I : \pi_{\mathtt{P}}^t(i) = p \,\}$ for each prompt p and round $t \in \mathbb{N}$.
Let $N_{p,a}(t)$ denote the number reward observations for (prompt, answer) pair (p, a) available to the designer after round t.

Initialization (full feedback). For each prompt-answer pair (p, a), the initial dataset contains n_0 independent reward samples generated by every intent $i \in I_p^0$. The designer observes only the resulting prompt-answer-reward datapoints (p, a, r), not the underlying intents, and sets

$$\widehat{\mu}_{p,a}^t(0) = \frac{1}{N_{p,a}(0)} \sum_{i \in I_p^0} \sum_{s \in [n_0]} r_s(i, a), \quad N_{p,a}(0) = |I_p^0|\, n_0.$$

The dynamics proceeds as follows, for each round $t \geq 1$:

1. **Designer update:** a new answer policy $\pi_{\mathtt{A}}^t$ such that

$$\pi_{\mathtt{A}}^t(p) \in \arg\max_{a \in A} \widehat{\mu}_{p,a}^t(t-1) \quad \text{for each prompt } p.$$

2. **Human best response:** a new prompting policy $\pi_{\mathtt{P}}^t \in \mathtt{BR}(\pi_{\mathtt{A}}^t)$, where

$$\mathtt{BR}(\pi_{\mathtt{A}}) := \{\, \text{prompting policies } \pi_{\mathtt{P}} : \pi_{\mathtt{P}}(i) \in \arg\max_{p \in [K]} \mu_{i,\pi_{\mathtt{A}}(p)} \ \forall i \in I \,\}. \tag{9}$$

3. **Bandit feedback.** For each prompt p such that $I_p^t \neq \emptyset$:
Let $a = \pi_{\mathtt{A}}^t(p)$. The designer observes the batch size $|I_p^t|$ and the average reward

$$y_t(p) = \frac{1}{|I_p^t|} \sum_{i \in I_p^t} r_t(i, a),$$

and updates

$$\widehat{\mu}_{p,a}^t(t) = \frac{N_{p,a}(t-1)\widehat{\mu}_{p,a}^t(t-1) + |I_p^t|\, y_t(p)}{N_{p,a}(t-1) + |I_p^t|}, \quad N_{p,a}(t) = N_{p,a}(t-1) + |I_p^t|.$$

All other estimates remain unchanged. If $I_p^t = \emptyset$, no estimates for prompt p are updated.

Figure 1: Learning dynamics in our model.

B Details from Preliminaries (Section 2)

Lemma B.1. *In the retraining game, the set of optimal Nash equilibria is exactly equal to the set of policy pairs with maximum utility.*

Proof. First we will show that if something is a policy pair with maximum utility, then it is an optimal Nash. To see this, note that, since this pair has maximum utility, there is no way for either player to adjust their policy unilaterally so that the utility increases. Therefore, this is a Nash. In addition, it must be an optimal Nash, as every Nash is comprised of policy pairs, and this policy pair is optimal.

Now we will show that if something is an optimal Nash, then it is a policy pair with maximum utility. Assume for contradiction that this is not true. Then there exists some optimal Nash S, and an optimal policy pair P, such that the utility of P is strictly higher than S. But from above, P is also a Nash. As we assumed S to be an *optimal* Nash, this derives a contradiction. □

B.1 Additional Details on Regret

We first relate our notion of regret to the standard notion of *policy regret* against a reactive environment. Policy regret differs from ordinary external regret in that a comparator is evaluated against the counterfactual behavior that the environment would have exhibited had that comparator actually been played. In our setting, the reactive environment is the human population: after observing the designer's current answer policy, humans myopically best-respond to it. For any designer policy $\pi_D : [K] \to A$, define its induced value

$$V(\pi_D) := u(\pi_H, \pi_D) \qquad \text{for any } \pi_H \in \mathtt{BR}(\pi_D).$$

This is well-defined. Indeed, for every intent i, all prompts in

$$\arg\max_{p \in [K]} \mu_{i,\pi_D(p)}$$

give the same utility, and hence all human best responses to a fixed π_D have the same total utility.

Definition B.2 (Policy regret)**.** *Consider the reactive environment in which, whenever the designer plays an answer policy π_D, the human population plays a best response in $\mathtt{BR}(\pi_D)$. The policy regret of a sequence of designer policies $\pi_D^1, \ldots, \pi_D^T$ against the best fixed designer policy is*

$$\mathtt{regret}_{\mathrm{pol}}(T) := \max_{\pi_D : [K] \to A} \sum_{t=1}^{T} V(\pi_D) - \sum_{t=1}^{T} V(\pi_D^t).$$

We define policy regret in terms of expected utility; the randomness of the reward samples affects the sequence of policies selected by the dynamics.

The counterfactual in this definition is important. If the designer had instead played some fixed policy π_D throughout, we do not evaluate π_D against the sequence of human policies generated by the actual dynamics. Rather, the humans would have reacted to the counterfactual designer policy, and hence would have played a best response to π_D in every round.

Lemma B.3. *For the repeated retraining game,*

$$\max_{\pi_D : [K] \to A} V(\pi_D) = u^*.$$

Consequently,

$$\mathtt{regret}_{\mathrm{pol}}(T) = \sum_{t=1}^{T} \left(u^* - u(\pi_H^t, \pi_D^t)\right) = \mathtt{regret}(T).$$

Proof. For every fixed designer policy π_D, humans best-respond, so

$$V(\pi_D) = \max_{\pi_H : I \to [K]} u(\pi_H, \pi_D).$$

Therefore,

$$\max_{\pi_D} V(\pi_D) = \max_{\pi_D} \max_{\pi_H} u(\pi_H, \pi_D) = \max_{\pi_H, \pi_D} u(\pi_H, \pi_D) = u^*,$$

where the final equality follows from Lemma B.1. Moreover, since $\pi_H^t \in \texttt{BR}(\pi_D^t)$ at every retraining step,

$$V(\pi_D^t) = u(\pi_H^t, \pi_D^t).$$

Substituting these two identities into the definition of policy regret gives

$$\texttt{regret}_{\text{pol}}(T) = Tu^* - \sum_{t=1}^{T} u(\pi_H^t, \pi_D^t) = \texttt{regret}(T).$$

□

We can therefore interpret optimization failure directly in terms of policy regret. Recall that FAIL occurs if there exists some $\eta > 0$ such that, for all sufficiently large T,

$$\frac{1}{T}\sum_{t=1}^{T} u(\pi_H^t, \pi_D^t) \le u^* - \eta.$$

Using the lemma above, this is equivalent to

$$\frac{\texttt{regret}_{\text{pol}}(T)}{T} \ge \eta$$

for all sufficiently large T. Equivalently,

$$\texttt{FAIL} = \left\{ \liminf_{T\to\infty} \frac{\texttt{regret}_{\text{pol}}(T)}{T} > 0 \right\}.$$

Thus, optimization failure is precisely the event that the greedy retraining dynamics incur asymptotically linear policy regret against the best fixed designer policy. In particular, on the event FAIL, $\texttt{regret}(T) = \Omega(T)$. Finally, suppose $\mathbb{P}[\texttt{FAIL}] > 0$. Since FAIL is the event that the long-run regret rate is bounded away from zero, there must exist some $\eta > 0$ such that, with positive probability, $\texttt{regret}(T) \ge \eta T$ for all sufficiently large T. Since $\texttt{regret}(T) \ge 0$ always, it follows that

$$\mathbb{E}[\texttt{regret}(T)] = \Omega(T).$$

C Proofs for Optimization Failures (Section 3)

C.1 Proof of Theorem 3.2

Realized-set notation. For a policy pair (π_H, π_D), let $R_i(\pi_H, \pi_D)$ denote the set of realized intent–answer pairs, and $\bar{R}_i(\pi_H, \pi_D)$ the unrealized ones. Similarly, let $R_p(\pi_H, \pi_D)$ and $\bar{R}_p(\pi_H, \pi_D)$ denote the realized and unrealized prompt–answer pairs.

Proof. We call an answer policy which is not a part of any optimal Nash a *decoy* policy.

We will lower bound the probability of failure by finding the probability that the system immediately converges to the designer always selecting some specific decoy answer policy $\bar{\pi}_D$. Define the event D as the event that the initial data for the designer shows, for each prompt p, a reward > 0 for selecting answer $\bar{\pi}_D(p)$, and a reward of 0 for selecting any other answer. For some initial prompting policy π_H, we can write this probability as

$$\mathbb{P}(D) = \mathbb{P}\left(\forall (p,a) \in R_p(\pi_H, \bar{\pi}_D), \hat{\mu}_1(p,a) > 0 \ \cap \ \forall (p,a) \in \bar{R}_p(\pi_H, \bar{\pi}_D), \hat{\mu}_1(p,a) = 0\right) \quad (10)$$

$$\ge \mathbb{P}\left(\forall (i,a) \in R_i(\pi_H, \bar{\pi}_D), r_1(i,a) = 1 \ \cap \ \forall (i,a) \in \bar{R}_i(\pi_H, \bar{\pi}_D), r_1(i,a) = 0\right). \quad (11)$$

By independence of the reward draws across (i,a),

$$\mathbb{P}(D) \ge \prod_{(i,a)\in R_i(\pi_H,\bar{\pi}_D)} \mathbb{P}(r_1(i,a)=1) \cdot \prod_{(i,a)\in \bar{R}_i(\pi_H,\bar{\pi}_D)} \mathbb{P}(r_1(i,a)=0) \quad (12)$$

$$\ge \prod_{(i,a)\in R_i(\pi_H,\bar{\pi}_D)} \alpha \cdot \prod_{(i,a)\in \bar{R}_i(\pi_H,\bar{\pi}_D)} \alpha \quad (13)$$

$$= \alpha^{|I|\cdot|A|}. \quad (14)$$

For n_0 initial samples per intent, this becomes $\alpha^{n_0 \cdot |I| \cdot |A|}$.

Finally, we prove that if D occurs, then the designer will continue playing $\bar{\pi}_D$ forever. Recall that the designer selects their mapping from prompts to actions by greedy optimization of the current empirical reward matrix $\hat{\mu}$, and that they receive bandit feedback in all rounds but the first. Thus, they only update the (p, a) pairs realized under the current policy pair. For every prompt p, there are two cases:

- The designer receives no feedback for p, as the current prompting policy does not use it. Thus the designer does not change the fact that they prefer to map p to $\bar{\pi}_D(p)$, because the reward matrix has not changed.
- The designer receives feedback for $(p, \bar{\pi}_D(p))$. Regardless of the feedback, the averaged reward value remains strictly above 0, since the initial empirical value was strictly positive and all future rewards lie in $\{0, 1\}$. Meanwhile, every estimate (p, a) for $a \neq \bar{\pi}_D(p)$ remains equal to 0, since those actions are never played and hence never updated. Thus, the estimated reward of $(p, \bar{\pi}_D(p))$ remains strictly higher than (p, a) for any $a \neq \bar{\pi}_D(p)$, so the designer continues to map p to $\bar{\pi}_D(p)$.

□

C.2 Proof of Theorem 3.4

Proof. Let a^* be a required answer. Define the event F that: (i) for every $i \in I$, all n_0 initial samples of a^* are 0, and (ii) for every $i \in I$, there exists $a \neq a^*$ with at least one initial sample equal to 1.

We first lower bound $\mathbb{P}(F)$. By independence,

$$\mathbb{P}(F) = \mathbb{P}\big(\forall i,\ \hat{\mu}_i^0(a^*) = 0\big)\, \mathbb{P}\big(\forall i,\ \exists a \neq a^* : \hat{\mu}_i^0(a) > 0\big)\,.$$

Since $\mu_{i,a^*} \leq 1 - \alpha$,

$$\mathbb{P}\big(\forall i,\ \hat{\mu}_i^0(a^*) = 0\big) \geq \alpha^{n_0 |I|}.$$

For each i,

$$\mathbb{P}\big(\exists a \neq a^* : \hat{\mu}_i^0(a) > 0\big) \geq 1 - e^{-\alpha n_0 (|A|-1)},$$

so

$$\mathbb{P}(F) \geq \alpha^{n_0 |I|} \left(1 - e^{-\alpha n_0 (|A|-1)}\right)^{|I|}.$$

We now show that F implies failure. Let

$$I_p^0 := \{i : \pi_H^0(i) = p\}.$$

Since π_H^0 is surjective, $I_p^0 \neq \emptyset$.

On F, for every prompt p,

$$\hat{\mu}_{p,a^*}(0) = 0.$$

Moreover, there exists $i_p \in I_p^0$ and $a_p \neq a^*$ such that (i_p, a_p) has a positive initial sample. Since rewards are nonnegative, this implies

$$\hat{\mu}_{p,a_p}(0) > 0.$$

Thus a^* is not a maximizer at any prompt initially, so it is never selected in round 1.

If a^* is never selected, its estimate remains identically zero under bandit feedback. Meanwhile, each a_p either remains unplayed (and hence positive) or is updated by averaging with nonnegative rewards, so

$$\hat{\mu}_{p,a_p}(t) > 0 \quad \text{for all } t.$$

Thus a^* is never selected at any round.

Since a^* is required, no optimal Nash equilibrium avoids it. Hence all policy pairs reached by the dynamics are bounded away from u^* by some $\eta > 0$, implying optimization failure.

Therefore,

$$\mathbb{P}[\texttt{FAIL}] \geq \mathbb{P}(F) \geq \alpha^{n_0|I|} \left(1 - e^{-\alpha n_0(|A|-1)}\right)^{|I|},$$

which yields

$$\mathbb{P}[\texttt{FAIL}] \geq \exp(-\Theta(n_0|I|)).$$

□

C.3 Proof of Theorem 3.9

We begin by restating Theorem 3.9 with more detail:

Theorem C.1. *Suppose the problem instance admits a Δ-robust biclustering (I_b, A_b), $b \in [B]$, with $A_{\min} := \min_{b\in[B]} |A_b|$, and the initial prompting policy π_P^{init} is consistent with this biclustering. Let $a^* \in A_{b^*}$ be a required answer, suppose $|A_{b^*}| \geq 2$, and suppose that for some constant $\alpha > 0$, $\mu_{i,a^*} \leq 1 - \alpha$ for every $i \in I_{b^*}$. Let $K^* := |\pi_P^{\text{init}}(I_{b^*})|$. Then*

$$\mathbb{P}[\texttt{FAIL}] \geq (1 - e^{-\Delta n_0 A_{\min}})^{K-K^*}(1 - K^* e^{-n_0 D(\Delta/2\|\Delta)}) \left(\frac{e^{-\lambda n_0}}{\sqrt{4n_0\Delta(1-\Delta/2)}}\right)^{|I_{b^*}|},$$

where $\lambda := (1 - \alpha - \Delta/2)^2/((1-\alpha)\alpha)$. In particular, if

$$n_0 \geq \max \frac{\log(2(K-K^*))}{\Delta A_{\min}}, \frac{\log(2K^*)}{D(\Delta/2\|\Delta)},$$

then

$$\mathbb{P}(\texttt{FAIL}) \geq \exp(-\Theta(n_0|I_{b^*}|))$$

for constant $\Delta, \alpha > 0$.

Proof. Fix any $a_s \in A_{b^*} \setminus a^*$. By Δ-robustness, $\mu_{i,a_s} \geq \Delta$ for every $i \in I_{b^*}$. We define three events.

- GROUPED: for every prompt p initially assigned to some bicluster $b \neq b^*$, the initial data contains positive utility for at least one answer in A_b.
- SAFE: for every prompt p initially assigned to b^*, the estimated utility of a_s satisfies $\widehat{\mu} * p, a_s{}^t > \Delta/2$ after initialization and at every later retraining step.
- UNDERESTIMATE: for every intent $i \in I * b^*$, the initial empirical reward of a^* satisfies $\widehat{\mu}^0_{i,a^*} < \Delta/2$.

We first show that these events imply failure. Together, GROUPED and SAFE ensure that every prompt initially has a strictly positive estimate for some answer in its own bicluster, so Lemma C.3 implies that the dynamics remain consistent with the biclustering. In particular, prompts outside b^* never map to a^*.

For prompts initially assigned to b^*, UNDERESTIMATE implies that the initial prompt-level estimate of a^* is below $\Delta/2$, while SAFE keeps the estimate of a_s above $\Delta/2$. Thus a^* is not initially selected at any such prompt. Since feedback is bandit, its estimates then remain frozen below $\Delta/2$, while those of a_s remain above $\Delta/2$. Hence a^* is never played. Since a^* is required, every policy pair reached on GROUPED ∩ SAFE ∩ UNDERESTIMATE is strictly suboptimal. Since the policy space is finite, this event implies FAIL.

It remains to bound the probability of these events. For GROUPED, fix a prompt initially assigned to $b \neq b^*$. There are at least $A_{\min}$ answers in A_b, each with mean at least Δ for every intent assigned to the prompt. Therefore,

$$\mathbb{P}[\text{GROUPED}] \geq (1 - e^{-\Delta n_0 A_{\min}})^{K-K_{b^*}}.$$

Next consider SAFE. Fix a prompt p initially assigned to b^*, and consider all reward observations of a_s incorporated into $\widehat{\mu} * p, a_s$. As long as SAFE has held, bicluster invariance implies that each

such observation comes from an intent in $I * b^*$, and therefore has conditional mean at least Δ. This sequence stochastically dominates an i.i.d. sequence of $\mathrm{Bernoulli}(\Delta)$ random variables. Since the estimate begins with at least n_0 observations, Theorem C.2 with $\alpha = \Delta$ and $q = \Delta/2$ gives

$$\mathbb{P}[\widehat{\mu}^t_{p,a_s} > \Delta/2 \text{ for every } t] \geq 1 - e^{-n_0 D(\Delta/2\|\Delta)}.$$

Note also that $D(\Delta/2\|\Delta) \geq \frac{\Delta}{8}$, and therefore

$$\mathbb{P}[\widehat{\mu}^t_{p,a_s} > \Delta/2 \text{ for every } t] \geq 1 - e^{-\frac{n_0\Delta}{8}}.$$

A union bound over the K_{b^*} prompts gives

$$\mathbb{P}[\textsc{Safe} \mid \textsc{Grouped}, \textsc{Underestimate}] \geq 1 - K_{b^*} e^{-\frac{n_0\Delta}{8}}.$$

Finally, consider UNDERESTIMATE. For each $i \in I_{b^*}$, we have $\mu_{i,a^*} \leq 1 - \alpha$. Therefore, the probability that all n_0 initial samples of a^* for intent i are zero is

$$\mathbb{P}\big[\widehat{\mu}^0_{i,a^*} = 0\big] = (1 - \mu_{i,a^*})^{n_0} \geq \alpha^{n_0}.$$

Since $\Delta > 0$, this event implies $\widehat{\mu}^0_{i,a^*} < \Delta/2$, and hence

$$\mathbb{P}\big[\widehat{\mu}^0_{i,a^*} < \Delta/2\big] \geq \alpha^{n_0}.$$

The initialization draws are independent across intents, so

$$\mathbb{P}[\textsc{Underestimate}] \geq \alpha^{n_0|I_{b^*}|}.$$

Moreover, GROUPED and UNDERESTIMATE depend on disjoint initialization draws. Combining the three bounds,

$$\mathbb{P}[\textsc{Fail}] \geq (1 - e^{-\Delta n_0 A_{\min}})^{K - K_{b^*}} \left(1 - K_{b^*} e^{-n_0\Delta/8}\right) \alpha^{n_0|I_{b^*}|}.$$

If

$$n_0 \geq \max\{\frac{\log(2(K - K_{b^*}))}{\Delta A_{\min}}, \frac{8\log(2K_{b^*})}{\Delta}\},$$

then the first two factors are each at least $1/2$, yielding, for constant α:

$$\mathbb{P}[\textsc{Fail}] \geq \frac{1}{4}\alpha^{n_0|I_{b^*}|} = \exp(-\Theta(n_0|I_{b^*}|)),$$

□

Theorem C.2 (Time-uniform lower bound for never dipping below)**.** *Let* $(X_t)_{t\geq 1}$ *be i.i.d.* $\mathrm{Bernoulli}(\alpha)$ *with* $\alpha \in (0,1)$*. Fix* $q \in (0,\alpha)$*. Then, for every* $n \geq 1$*,*

$$\mathbb{P}\left(\forall t \geq n : \frac{1}{t}\sum_{s=1}^{t} X_s > q\right) \geq 1 - \exp(-nD(q\|\alpha)),$$

where

$$D(q\|\alpha) = q\ln\frac{q}{\alpha} + (1-q)\ln\frac{1-q}{1-\alpha}$$

is the Bernoulli KL divergence.

Proof. Let

$$S_t := \sum_{s=1}^{t} X_s.$$

For any $\lambda < 0$, define

$$M_t(\lambda) := \exp(\lambda S_t - t\psi(\lambda)), \qquad \psi(\lambda) := \ln\left(\alpha e^{\lambda} + 1 - \alpha\right).$$

Since the X_t are i.i.d. Bernoulli(α), $(M_t(\lambda))_{t\geq 0}$ is a nonnegative martingale with

$$M_0(\lambda) = 1.$$

Choose

$$\lambda^* := \ln \frac{q(1-\alpha)}{\alpha(1-q)}.$$

Since $q < \alpha$, we have $\lambda^* < 0$. Moreover, the standard Bernoulli Chernoff calculation gives

$$\lambda^* q - \psi(\lambda^*) = D(q\|\alpha).$$

Suppose that for some $t \geq n$,

$$\frac{S_t}{t} \leq q.$$

Because $\lambda^* < 0$, this implies

$$\lambda^* S_t \geq \lambda^* q t.$$

Therefore

$$M_t(\lambda^*) \geq \exp\big(t\big(\lambda^* q - \psi(\lambda^*)\big)\big) = \exp(tD(q\|\alpha)) \geq \exp(nD(q\|\alpha)).$$

Hence

$$\left\{\exists t \geq n : \frac{S_t}{t} \leq q\right\} \subseteq \left\{\sup_{t\geq 0} M_t(\lambda^*) \geq \exp(nD(q\|\alpha))\right\}.$$

By Ville's inequality for nonnegative martingales,

$$\mathbb{P}\left(\sup_{t\geq 0} M_t(\lambda^*) \geq \exp(nD(q\|\alpha))\right) \leq \exp(-nD(q\|\alpha)).$$

Thus

$$\mathbb{P}\left(\exists t \geq n : \frac{S_t}{t} \leq q\right) \leq \exp(-nD(q\|\alpha)).$$

Taking complements gives

$$\mathbb{P}\left(\forall t \geq n : \frac{S_t}{t} > q\right) \geq 1 - \exp(-nD(q\|\alpha)).$$

□

Lemma C.3. *Suppose the instance admits a Δ-robust biclustering (I_b, A_b), $b \in [B]$, and the initial prompting policy is consistent with this biclustering. Suppose furthermore that, for every prompt p initially assigned to bicluster b, there is some answer $a \in A_b$ whose initial estimated utility at p is strictly positive. Then, in every round, both the Designer's and the Human's policies are consistent with the biclustering.*

C.4 Proof of Lemma C.3

Proof. We proceed by induction on the rounds.

First consider the Designer's policy after initialization. Fix a prompt p initially assigned to bicluster b. By assumption, some answer $a \in A_b$ has strictly positive estimated utility at p. Since the initial prompting policy is consistent with the biclustering, all intents mapped to p lie in I_b. By Δ-robustness, every answer outside A_b has utility zero for these intents, and hence has estimated utility zero at p. Thus the Designer strictly prefers an answer in A_b. The Human then also strictly prefers to map every

intent in I_b to a prompt assigned an answer in A_b, since such answers have utility at least Δ, while answers outside A_b have utility zero. Hence both policies are consistent in the first round.

Now suppose both policies have been consistent through round $t-1$. Fix a prompt p in bicluster b. Because the Human has always been consistent, every reward observation incorporated into the Designer's estimates at p comes from an intent in I_b. Therefore every answer outside A_b continues to have estimated utility zero at p. Moreover, the Designer has previously selected an answer in A_b with strictly positive estimated utility. Since rewards are nonnegative, incorporating additional observations cannot make the cumulative reward of that answer zero, so its estimated utility remains strictly positive. Thus the Designer again strictly prefers an answer in A_b at round t.

Given a Designer policy consistent with the biclustering, the Human again strictly prefers a consistent prompting policy, because answers within an intent's bicluster yield utility at least Δ and answers outside it yield utility zero. Hence both policies remain consistent at round t, completing the induction. □

C.5 Proof of Theorem 3.14

Proof. We first show that $u^*_{\text{consistent}}(\mathcal{B})$ is exactly the maximum utility of a policy pair consistent with $\mathcal{B}$. Consider any policy pair (π_P, π_A) consistent with $\mathcal{B}$. By Definition 3.8, there is a partition $(P_b : b \in [B])$ of the prompts such that

$$\pi_P(I_b) \subseteq P_b \qquad \text{and} \qquad \pi_A(P_b) \subseteq A_b$$

for every $b \in [B]$. Let $p_b := |P_b|$. Then $p_b \geq 1$ for every b and

$$\sum_{b \in [B]} p_b = K.$$

Because bicluster b interacts only through the p_b prompts in P_b, its contribution to the total utility is at most $u^*_{p_b}(\{b\})$. Hence

$$u(\pi_P, \pi_A) \leq \sum_{b \in [B]} u^*_{p_b}(\{b\}) \leq u^*_{\text{consistent}}(\mathcal{B}).$$

Conversely, fix any $p_1, \ldots, p_B \geq 1$ summing to K, and partition the prompts into sets P_b with $|P_b| = p_b$. For each bicluster b, take a policy pair attaining $u^*_{p_b}(\{b\})$ in the game restricted to I_b, A_b, and P_b. Combining these policies across the disjoint prompt sets gives a policy pair consistent with $\mathcal{B}$ whose total utility is

$$\sum_{b \in [B]} u^*_{p_b}(\{b\}).$$

Thus $u^*_{\text{consistent}}(\mathcal{B})$ is exactly the largest utility achievable by a policy pair consistent with $\mathcal{B}$. Next, consider any policy pair (π_P, π_A) in which π_P is a best response to π_A. Suppose that π_A uses at least one answer from every bicluster. For each b, define

$$P_b := \{p \in [K] : \pi_A(p) \in A_b\}.$$

The sets $(P_b : b \in [B])$ form a partition of $[K]$, and each is nonempty. Fix $i \in I_b$. Every answer in A_b gives i strictly positive utility, whereas every answer outside A_b gives utility zero. Since P_b is nonempty, every best response of i must lie in P_b. Hence

$$\pi_P(I_b) \subseteq P_b \qquad \text{and} \qquad \pi_A(P_b) \subseteq A_b$$

for every b, so the policy pair is consistent with $\mathcal{B}$. Therefore, any best-response policy pair that is not consistent must omit some answer bicluster A_b entirely. Such a policy obtains utility at most

$$u^*_K(\mathcal{B} \setminus \{b\}).$$

Conversely, any policy attaining $u^*_K(\mathcal{B} \setminus \{b\})$ in the restricted game can be used in the full game while simply never using an answer from A_b. Therefore the largest utility achievable by an inconsistent best-response policy pair is

$$\max_{b \in [B]} u^*_K(\mathcal{B} \setminus \{b\}).$$

It follows that

$$u^* = \max\left\{u^*_{\text{consistent}}(\mathcal{B}), \max_{b\in[B]} u^*_K(\mathcal{B}\setminus\{b\})\right\}.$$

Hence no optimal Nash equilibrium is consistent with $\mathcal{B}$ if and only if

$$\max_{b\in[B]} u^*_K(\mathcal{B}\setminus\{b\}) > u^*_{\text{consistent}}(\mathcal{B}),$$

which is equivalent to the existence of some $b \in [B]$ such that

$$u^*_K(\mathcal{B}\setminus\{b\}) > u^*_{\text{consistent}}(\mathcal{B}).$$

□

C.6 Proof of Theorem 3.11

Proof. Let $Grouped$ be the event that, for every prompt p initially assigned to bicluster b, the initial data contains nonzero utility for at least one answer in A_b.

We first show that $Grouped$ implies optimization failure. Since the initial prompting policy is consistent with the biclustering, Lemma C.3 implies that, on $Grouped$, both the Designer and the Human are consistent with the initial biclusters in every round. Therefore every policy pair reached by the dynamics is consistent with the biclustering.

Because the game instance is nash-devious, every biclustering-consistent policy pair is strictly suboptimal. Equivalently, no biclustering-consistent policy pair is an optimal Nash equilibrium. Since the policy space is finite, there exists some $\eta > 0$ such that every biclustering-consistent policy pair has utility at most $u^* - \eta$, where u^* is the optimal Nash value. Hence, on $Grouped$, for every round t,

$$u(\pi_H^t, \pi_D^t) \le u^* - \eta.$$

Thus the dynamics exhibit optimization failure.

It remains to lower bound $\mathbb{P}(Grouped)$. Fix a prompt p initially assigned to bicluster b. There are at least $A_{\min}$ answers in A_b, and each has mean reward at least Δ for every intent initially mapped to p. Therefore the probability that none of these answers in A_b obtains a nonzero reward in the n_0 initial samples is at most

$$(1-\Delta)^{n_0 A_{\min}} \le e^{-\Delta n_0 A_{\min}}.$$

By a union bound over the K prompts,

$$\mathbb{P}(Grouped) \ge 1 - Ke^{-\Delta n_0 A_{\min}} = 1 - \exp(-\Delta n_0 A_{\min} + \log K).$$

Since $Grouped$ implies `FAIL`, this gives

$$\mathbb{P}(\texttt{FAIL}) \ge 1 - \exp(-\Delta n_0 A_{\min} + \log K).$$

Finally, if $\Delta n_0 A_{\min} \ge c \log K$, then

$$\mathbb{P}(\texttt{FAIL}) \ge 1 - K^{1-c}.$$

Thus for any constant $c > 1$, the failure probability is $1 - o(1)$ as $K \to \infty$. □

D Proofs for Model Deterioration (Section 4)

This section contains the detailed proofs and implementation conventions for Section 4.

D.1 Concrete Example

We prove the deterioration claim for the witness in Theorem 4.3. The reward matrix is

$$R_\star = \begin{pmatrix} 1 & 0.98 & 0.99 \\ 0.05 & 0.50 & 0 \\ 0 & 0 & 0.50 \end{pmatrix},$$

with initial human policy

$$H^0 = (p_1, p_1, p_2).$$

Under this initial split, the designer's full-feedback estimates are

$$\widehat{\mu}^0 = \begin{array}{c|ccc} & a_1 & a_2 & a_3 \\ \hline p_1 & (1+0.05)/2 & (0.98+0.50)/2 & (0.99+0)/2 \\ p_2 & 0 & 0 & 0.50 \end{array} = \begin{array}{c|ccc} & a_1 & a_2 & a_3 \\ \hline p_1 & 0.525 & 0.740 & 0.495 \\ p_2 & 0 & 0 & 0.500. \end{array}$$

Thus the initial designer policy is

$$D^0(p_1) = a_2, \qquad D^0(p_2) = a_3,$$

and the initial utility is

$$U^0 = 0.98 + 0.50 + 0.50 = 1.98.$$

Given $D^0 = (a_2, a_3)$, the human best response is

$$H^1 = (p_2, p_1, p_2),$$

because $0.99 > 0.98$, $0.50 > 0$, and $0.50 > 0$. The designer therefore receives the following on-path feedback matrix:

$$F(D^0, H^1) = \begin{array}{c|ccc} & a_1 & a_2 & a_3 \\ \hline p_1 & - & 0.50 & - \\ p_2 & - & - & (0.99+0.50)/2 \end{array} = \begin{array}{c|ccc} & a_1 & a_2 & a_3 \\ \hline p_1 & - & 0.500 & - \\ p_2 & - & - & 0.745. \end{array}$$

A blank entry means that the answer was not played at that prompt, so its previous estimate is unchanged. As long as the designer continues to play $D^0 = (a_2, a_3)$, the human continues to best-respond with $H^1 = (p_2, p_1, p_2)$, so this same feedback is received every round. After t transient feedback rounds, and before the designer switches, its estimate matrix has the form

$$\widehat{\mu}^t = \begin{array}{c|ccc} & a_1 & a_2 & a_3 \\ \hline p_1 & 0.525 & \dfrac{1.48+0.50t}{t+2} & 0.495 \\ p_2 & 0 & 0 & \dfrac{0.50+1.49t}{1+2t}. \end{array}$$

The p_1, a_2 entry converges to 0.50. Since $0.50 < 0.525$, there is a finite time at which

$$\frac{1.48+0.50t}{t+2} < 0.525.$$

At that time a_1 strictly beats both a_2 and a_3 at p_1, so the designer switches to

$$D^\infty(p_1) = a_1, \qquad D^\infty(p_2) = a_3.$$

Given this final designer policy, the human returns to the initial prompt split H^0, since $1 > 0.99$, $0.05 > 0$, and $0.50 > 0$. The final on-path feedback matrix is

$$F(D^\infty, H^0) = \begin{array}{c|ccc} & a_1 & a_2 & a_3 \\ \hline p_1 & (1+0.05)/2 & - & - \\ p_2 & - & - & 0.50 \end{array} = \begin{array}{c|ccc} & a_1 & a_2 & a_3 \\ \hline p_1 & 0.525 & - & - \\ p_2 & - & - & 0.500. \end{array}$$

This state is stable. At p_1, the played answer a_1 has long-run feedback 0.525, which exceeds the frozen estimate 0.495 for a_3 and the post-transient estimate of a_2, which is below 0.525 by the time the switch occurs. At p_2, answer a_3 has feedback 0.500, while a_1 and a_2 remain frozen at 0. The final utility is

$$U^\infty = 1 + 0.05 + 0.50 = 1.55.$$

Thus the process deteriorates by

$$U^0 - U^\infty = 1.98 - 1.55 = 0.43.$$

All comparisons are strict, so this is a robust deterioration.

D.2 Canonical Deterioration Region

Let

$$R = \begin{pmatrix} \mu_{11} & \mu_{12} & \mu_{13} \\ \mu_{21} & \mu_{22} & \mu_{23} \\ \mu_{31} & \mu_{32} & \mu_{33} \end{pmatrix} \in [0,1]^9$$

and suppose the initial human policy is

$$H^0(i_1) = H^0(i_2) = p_1, \qquad H^0(i_3) = p_2.$$

Assume the following strict inequalities:

$$\mu_{12} + \mu_{22} > \mu_{11} + \mu_{21}, \quad \mu_{11} + \mu_{21} > \mu_{13} + \mu_{23}, \tag{15}$$

$$\mu_{11} > \mu_{13}, \qquad \mu_{13} > \mu_{12}, \qquad \mu_{21} > \mu_{23}, \qquad \mu_{22} > \mu_{23}, \quad \frac{\mu_{11} + \mu_{21}}{2} > \mu_{22}, \tag{16}$$

$$\mu_{33} > \mu_{31}, \qquad \mu_{33} > \mu_{32}, \qquad \frac{\mu_{13} + \mu_{33}}{2} > \mu_{31}, \quad \frac{\mu_{13} + \mu_{33}}{2} > \mu_{32}. \tag{17}$$

We show that these inequalities force a robust deterioration.

Under the initial routing, the designer's full-feedback estimate matrix is

$$\widehat{\mu}^0 = \begin{array}{c|ccc} & a_1 & a_2 & a_3 \\ \hline p_1 & (\mu_{11} + \mu_{21})/2 & (\mu_{12} + \mu_{22})/2 & (\mu_{13} + \mu_{23})/2 \\ p_2 & \mu_{31} & \mu_{32} & \mu_{33}. \end{array}$$

The inequality $\mu_{12} + \mu_{22} > \mu_{11} + \mu_{21}$ makes a_2 strictly beat a_1 at p_1. Together with $\mu_{11} + \mu_{21} > \mu_{13} + \mu_{23}$, it also implies $\mu_{12} + \mu_{22} > \mu_{13} + \mu_{23}$, so a_2 strictly beats a_3 at p_1. Meanwhile $\mu_{33} > \mu_{31}, \mu_{32}$ makes a_3 strictly greedy at p_2. Hence the initial designer policy is

$$D^0 = (a_2, a_3).$$

Given D^0, the human compares a_2 against a_3. The inequalities $\mu_{13} > \mu_{12}$, $\mu_{22} > \mu_{23}$, and $\mu_{33} > \mu_{32}$ force the strict best response

$$H^1 = (p_2, p_1, p_2).$$

The designer then receives the following on-path feedback matrix:

$$F(D^0, H^1) = \begin{array}{c|ccc} & a_1 & a_2 & a_3 \\ \hline p_1 & - & \mu_{22} & - \\ p_2 & - & - & (\mu_{13} + \mu_{33})/2. \end{array}$$

As long as the designer continues to play $D^0 = (a_2, a_3)$, the human continues to best-respond with $H^1 = (p_2, p_1, p_2)$, so this same feedback is received every round. After t transient feedback rounds, and before the designer switches, the designer's estimate matrix has the form

$$\widehat{\mu}^t = \begin{array}{c|ccc} & a_1 & a_2 & a_3 \\ \hline p_1 & (\mu_{11} + \mu_{21})/2 & \dfrac{\mu_{12} + \mu_{22} + t\mu_{22}}{t+2} & (\mu_{13} + \mu_{23})/2 \\ p_2 & \mu_{31} & \mu_{32} & \dfrac{\mu_{33} + t(\mu_{13} + \mu_{33})}{1 + 2t}. \end{array}$$

The p_1, a_2 entry converges to μ_{22}. Since $(\mu_{11} + \mu_{21})/2 > \mu_{22}$, after finitely many transient rounds the designer strictly prefers a_1 to a_2 at p_1. Also, $\mu_{11} + \mu_{21} > \mu_{13} + \mu_{23}$ implies $(\mu_{11} + \mu_{21})/2 > (\mu_{13} + \mu_{23})/2$, so a_1 strictly beats the frozen estimate of a_3. At p_2, the played estimate is a weighted average of μ_{33} and $(\mu_{13} + \mu_{33})/2$, both of which are strictly larger than μ_{31} and μ_{32}. Thus a_3 remains strictly greedy at p_2 throughout the transient. The designer therefore switches to

$$D^\infty = (a_1, a_3).$$

Given D^∞, the human returns to H^0: intent i_1 chooses p_1 because $\mu_{11} > \mu_{13}$, intent i_2 chooses p_1 because $\mu_{21} > \mu_{23}$, and intent i_3 chooses p_2 because $\mu_{33} > \mu_{31}$. The final played-feedback matrix is

$$F(D^\infty, H^0) = \begin{array}{c|ccc} & a_1 & a_2 & a_3 \\ \hline p_1 & (\mu_{11} + \mu_{21})/2 & - & - \\ p_2 & - & - & \mu_{33}. \end{array}$$

This state is analytically stable. At p_1, the played answer a_1 has long-run feedback $(\mu_{11} + \mu_{21})/2$, which strictly exceeds both the frozen estimate $(\mu_{13} + \mu_{23})/2$ for a_3 and the post-transient estimate of a_2, whose limit is μ_{22}. At p_2, the played answer a_3 has long-run feedback μ_{33}, which strictly exceeds both frozen alternatives μ_{31} and μ_{32}.

The initial utility is

$$U^0 = \mu_{12} + \mu_{22} + \mu_{33},$$

while the final utility is

$$U^\infty = \mu_{11} + \mu_{21} + \mu_{33}.$$

Since $\mu_{12} + \mu_{22} > \mu_{11} + \mu_{21}$, the final stable state is a strict deterioration. Because all inequalities are strict, the same path persists under sufficiently small perturbations of R.

D.3 Volume and Relabelings

The canonical region has exact volume

$$\frac{827}{15{,}482{,}880}$$

inside $[0, 1]^9$. One way to compute this is to integrate out the variables in the order suggested by the inequalities. The constraints imply

$$\frac{\mu_{11}}{2} < \mu_{12} < \mu_{13} < \mu_{11}, \qquad 0 < \mu_{21} < 2\mu_{12} - \mu_{11}, \qquad 0 < \mu_{23} < \mu_{21}, \qquad \mu_{11} + \mu_{21} - \mu_{12} < \mu_{22} < \frac{\mu_{11} + \mu_{21}}{2}.$$

Integrating over μ_{23} and μ_{22} contributes

$$\mu_{21} \left(\mu_{12} - \frac{\mu_{11} + \mu_{21}}{2} \right),$$

and integrating over μ_{21} contributes $(2\mu_{12} - \mu_{11})^3/12$. For the last row, the feasible length for each of μ_{31} and μ_{32}, conditional on μ_{13} and μ_{33}, is

$$\min \left\{ \mu_{33}, \frac{\mu_{13} + \mu_{33}}{2} \right\},$$

so integrating over μ_{33} gives

$$\frac{1 + 3\mu_{13} + 3\mu_{13}^2 - 3\mu_{13}^3}{12}.$$

The remaining polynomial integral over $\mu_{11}, \mu_{12}, \mu_{13}$ is rational and equals

$$\frac{827}{15{,}482{,}880}.$$

The canonical inequalities are written for one labeling of the three intents and three answers. As an unlabeled matrix property, we may permute the three intent rows and the three answer columns, giving $3! \cdot 3! = 36$ relabeled regions. These regions are disjoint. To see this, note that the strict inequalities identify the roles of the rows and columns uniquely. In any satisfying labeling, the singleton row is the row whose singleton-prompt answer is strictly better than both alternatives; the first pooled row is the row in which the final pooled-prompt answer strictly beats the singleton-prompt answer and the singleton-prompt answer strictly beats the initial pooled-prompt answer; and the second pooled row is the row that remains on the pooled prompt during the transient. Once these rows are fixed, the three columns are fixed by their roles as the final pooled-prompt answer, the initial pooled-prompt answer, and the singleton-prompt answer. If a second relabeling satisfied the same canonical inequalities, at least one row or column would have to play two different strict roles. That would force one of the strict comparisons defining those roles to hold in both directions.

Thus no game lies in two relabeled canonical regions, and the probability of satisfying the canonical inequalities up to arbitrary row and column relabeling is

$$36 \cdot \frac{827}{15{,}482{,}880} = \frac{827}{430{,}080}.$$

For the fixed simulation convention $H^0 = (p_1, p_1, p_2)$, only the $2! \cdot 3! = 12$ relabelings that preserve the initial two-plus-one prompt split are directly compatible, giving probability

$$\frac{827}{1{,}290{,}240}.$$

D.4 Embedding in Larger Games

We prove Theorem 4.5. Let R be any base $3 \times 2 \times 3$ game satisfying the strict canonical inequalities. Because all comparisons in the base proof are strict, choose $\eta > 0$ such that every relevant within-bicluster comparison has slack at least 4η.

Fix $D \geq 1$, $|I| \geq 3D$, and $|A| \geq 3D$. Construct a larger game with $|I|$ intents, $2D$ prompts, and $|A|$ answers by choosing D active biclusters:

$$I_1, \ldots, I_D, \qquad P_1, \ldots, P_D, \qquad A_1, \ldots, A_D,$$

where $|I_\ell| = 3$, $|P_\ell| = 2$, and $|A_\ell| = 3$. Within each bicluster $I_\ell \times A_\ell$, set the rewards equal to a copy of R. There may also be extra intents I^{extra} and extra answers A^{extra}. Across distinct active biclusters, and from active-bicluster intents to extra answers, set rewards to be at most η.

It remains to make the extra intents inert. Assign each extra intent to the singleton prompt of some active bicluster, namely the prompt corresponding to p_2 in the base construction. For an extra intent assigned to bicluster ℓ, give it reward above $1 - \eta$ for that bicluster's singleton answer a_3^ℓ, and reward below η for every other answer. In the initial prompting policy, send this extra intent to the singleton prompt of bicluster ℓ. Thus extra intents reinforce the singleton prompt's preference for a_3^ℓ, but they never enter the switching prompt. Since only strict inequalities are imposed, all these choices leave a nonempty open set.

The dynamics never leave biclusters. For the designer, the switching prompt in each active bicluster contains exactly the two base intents assigned to p_1, so its estimates and updates are exactly those in the base proof. The singleton prompt may contain extra intents, but these extra intents make a_3^ℓ even more strongly preferred and give low reward to all other answers. Any cross-bicluster or extra

answer has prompt-level estimate at most η, so it is strictly dominated by the within-bicluster answer selected in the bicluster trajectory. The same remains true after bandit updates, since such answers are never played on an active prompt and their estimates remain low.

For the human, as long as the designer chooses answers within each active bicluster according to the bicluster trajectory, each active intent obtains a payoff from some prompt in its own bicluster that exceeds the payoff from any prompt outside its bicluster by the isolation margin. Every cross-bicluster prompt plays an answer worth at most η to that active intent. Extra intents strictly prefer their assigned singleton prompt, because it plays a_3^ℓ, which gives them reward above $1-\eta$. Thus every human remains in its assigned bicluster or singleton prompt. By induction over time, active biclusters remain dynamically isolated, and extra intents never affect the switching prompts.

Conditional on this isolation, the switching prompt in each active bicluster evolves exactly as in the base $3 \times 2 \times 3$ deterioration game, and the singleton prompt remains strictly assigned to a_3^ℓ. Hence every active bicluster undergoes the same robust deterioration, independently of the others. The extra intents receive the same singleton answer before and after the deterioration, so their utility contribution cancels when comparing W_0 and W_∞. The separation conditions and the base canonical inequalities are all strict, so the constructed set contains an open neighborhood in $[0,1]^{|I||A|}$. Therefore the set of $(|I|, 2D, |A|)$ games exhibiting robust deterioration has positive Lebesgue measure.

The same bicluster construction preserves the additive harm lower bound as Theorem 4.4: choose one bicluster within ϵ of the $1/2$ additive supremum and choose the remaining biclusters from any isolated strict canonical region. Since harms add across dynamically isolated biclusters, the larger game has deterioration harm at least $1/2 - \epsilon$.

D.5 One-Switch Certificates

We say that a deterioration follows a *one-switch trajectory* if it has the qualitative form

$$(H_0, D_0) \rightsquigarrow (H_1, D_0) \rightsquigarrow (H_\infty, D_\infty),$$

where D_∞ differs from D_0 at exactly one prompt. Here, "switch" refers to the change in the designer policy. The *return-only one-switch* subfamily additionally requires $H_\infty = H_0$.

For a fixed candidate tuple $(D_0, H_1, D_\infty, H_\infty)$, the trajectory is certified by the following strict inequalities:

$$\begin{aligned}
\widehat{\mu}^0_{p,D^0(p)} &> \widehat{\mu}^0_{p,a} && \text{for all prompts } p \text{ and answers } a \neq D^0(p),\\
r_{i,D^0(H^1(i))} &> r_{i,D^0(q)} && \text{for all intents } i \text{ and prompts } q \neq H^1(i),\\
\widehat{\mu}^0_{p,D^\infty(p)} &> \widehat{\mu}^{H^1}_{p,D^0(p)} && \text{for the switching prompt } p,\\
\widehat{\mu}^0_{p,D^\infty(p)} &> \widehat{\mu}^0_{p,a} && \text{for the switching prompt and other frozen alternatives},\\
\widehat{\mu}^{H^1}_{p,D^0(p)} &> \widehat{\mu}^0_{p,a} && \text{for unchanged prompts } p \text{ and } a \neq D^0(p),\\
r_{i,D^\infty(H^\infty(i))} &> r_{i,D^\infty(q)} && \text{for all intents } i \text{ and prompts } q \neq H^\infty(i),\\
u(H^0, D^0) &> u(H^\infty, D^\infty).
\end{aligned}$$

Here $\widehat{\mu}^0$ is the initial full-feedback estimate under H^0, and $\widehat{\mu}^{H^1}$ is the prompt-level average induced by the transient routing. The inequalities are linear in the reward matrix. Whenever they have a strict solution, they define an open set of reward matrices that all exhibit the same robust one-switch deterioration.

For the general one-switch family, we additionally require final designer stability: for every prompt p used under H_∞, $\widehat{\mu}^{H\infty}p, D\infty(p) > \widehat{\mu}p, a$ for every $a \neq D\infty(p)$, where $\widehat{\mu}p, a$ is the alternative's frozen estimate at the time of the switch. For the return-only family, $H\infty = H_0$, so this condition is automatic.

The general one-switch family is the union over all candidate tuples satisfying these conditions, while the return-only one-switch family restricts to tuples with $H_\infty = H_0$.

D.6 WORST-CASE HARM

Because the path inequalities and the initial/final utilities are linear, worst-case harm over a one-switch family can be computed by linear programming over the corresponding inequality regions, taking the supremum over their closures. The resulting bounds are:

family	$\sup(U^0 - U^\infty)$	$\sup(U^0/U^\infty)$
canonical trap path	$1/2$	$3/2$
return-only one-switch	$1/2$	2
general one-switch	$1/2$	$2.$

For the canonical path, the additive harm is

$$H = \mu_{12} + \mu_{22} - \mu_{11} - \mu_{21}.$$

Using $(\mu_{11} + \mu_{21})/2 > \mu_{22}$,

$$H < \mu_{12} + \frac{\mu_{11} + \mu_{21}}{2} - \mu_{11} - \mu_{21} = \mu_{12} - \frac{\mu_{11} + \mu_{21}}{2}.$$

Since $\mu_{11} > \mu_{13} > \mu_{12}$, we have $\mu_{11} > \mu_{12}$, and since $\mu_{21} \geq 0$,

$$\mu_{12} - \frac{\mu_{11} + \mu_{21}}{2} < \mu_{12} - \frac{\mu_{12}}{2} = \frac{\mu_{12}}{2} \leq \frac{1}{2}.$$

Thus canonical additive harm is strictly below $1/2$, with supremum $1/2$. The supremum is approached by matrices tending to

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

from within the strict canonical region.

The canonical multiplicative bound is $3/2$. Since $U^0 = U^\infty + H$, it is enough to show $H \leq U^\infty/2$ in the closure. The same inequalities give

$$H = \mu_{12} + \mu_{22} - \mu_{11} - \mu_{21} \leq \mu_{12} - \frac{\mu_{11} + \mu_{21}}{2}.$$

Because $\mu_{12} \leq \mu_{11}$ in the closure and $\mu_{21}, \mu_{33} \geq 0$,

$$\mu_{12} - \frac{\mu_{11} + \mu_{21}}{2} \leq \frac{\mu_{11} + \mu_{21} + \mu_{33}}{2} = \frac{U^\infty}{2}.$$

Therefore $U^0/U^\infty \leq 3/2$. The ratio $3/2$ is approached by taking $\mu_{11}, \mu_{12}, \mu_{13}$ close to 1, $\mu_{21}, \mu_{23}, \mu_{31}, \mu_{32}, \mu_{33}$ close to 0, and μ_{22} close to $1/2$ from within the strict canonical region.

The broader one-switch additive bound has the same intuition. In a robust one-switch deterioration, all but at most one intent are protected by the final best-response inequalities: every intent can still access its initial answer at some prompt, and if it does not remain on the switching prompt receiving the new answer, its final chosen prompt gives reward at least as high as that initial answer. Thus any additive loss is concentrated on a single possibly harmed intent h.

Let x be the old answer at the switching prompt and y the new answer. If the switching prompt initially served one intent h, then the switch condition implies, in the closure,

$$y_h \geq \frac{x_h + x_m}{2}$$

for some transiently co-routed intent m. Since $x_m \geq 0$, the loss on that intent is at most

$$x_h - y_h \leq \frac{x_h}{2} \leq \frac{1}{2}.$$

Moreover $U^\infty \geq y_h \geq x_h/2$, so the additive harm is at most U^∞, implying $U^0/U^\infty \leq 2$.

If the switching prompt initially served two intents h, m, the switch condition instead gives

$$\frac{y_h + y_m}{2} \geq x_h,$$

so

$$x_h - y_h \leq \frac{y_m - y_h}{2} \leq \frac{1}{2}.$$

The final utility is at least $y_h + y_m$, because intent h receives y_h and the final best-response condition gives intent m reward at least y_m. Hence the additive harm is again at most U^∞, and $U^0/U^\infty \leq 2$. The examples found by the linear programs show that both broader one-switch bounds are tight as suprema over the closure of strict regions.

D.7 Experimental Conventions

The experiments use deterministic tie-breaking: the designer and human choose the lowest-index maximizer when a tie occurs. Following the definition in Section 4, a deterioration is labeled robust only if no tie or equality is encountered along the path or in the final stability certificate. Non-robust deteriorations are therefore not open sets of reward matrices; they can depend on the deterministic tie-breaking convention.

Stability is checked analytically. For a candidate final designer policy and human policy, the human policy must be a myopic best response to the designer policy. For each prompt used under the current human routing, the long-run reward of the played answer given the prompting policy must be at least the designer's current estimate for every alternative answer. If the inequality is strict everywhere, then the dynamics are analytically proven to be stable; if equality is needed, the outcome is tie-dependent.

The reported code is in

`https://anonymous.4open.science/r/greedy-dynamics-experiments-DC93/`.

The main simulation is implemented in `greedy_dynamics.robust_trap_search`; comparison with the canonical trap inequalities is implemented in `greedy_dynamics.compare_trap_deterioration`; and one-switch coverage is implemented in `greedy_dynamics.one_switch_catalogue`.

D.8 Additional Content from Section 4

Section 4 showed that model deterioration arises from the interaction between two ingredients:

- *prompt-level aggregation*, which causes estimates to mix strategically changing populations of intents, and
- *bandit feedback*, which allows incorrect estimates to persist once actions are no longer explored.

In the main text, we summarized this decomposition via the following four-way classification:

	Prompt Game	**Matrix Game**
Bandit Feedback	**Non-monotone**	Monotone
Full Feedback	Monotone	Monotone

In this appendix we prove the remaining claims in this table.

D.9 Bandit Feedback without Prompt Aggregation

We first study the matrix-game analogue of our dynamics. Here, the human directly selects actions rather than interacting through a prompt interface, so the only remaining ingredient is bandit feedback.

Definition D.1 (Bandit Best-Response Matrix Game)**.** *Fix a payoff matrix $U \in \mathbb{R}^{m\times n}$ with shared utilities. Play proceeds in rounds $t = 1, 2, \ldots$ as follows:*

1. ***Initialization.*** *Before play begins, a single column $j_0 \in [n]$ is fully revealed: the designer observes $\{U(i, j_0) : i \in [m]\}$.*
2. ***AI move.*** *In round t, the designer selects a row $r_t \in [m]$ using the* Follow-the-Leader *(FTL) rule: it chooses a row with the highest empirical mean payoff observed so far. For each row i, its empirical mean is*
$$\widehat{\mu}_i(t) = \frac{1}{|S_i(t)|} \sum_{x\in S_i(t)} x,$$
where $S_i(t)$ is the multiset of payoffs from rounds (including initialization) in which row i was observed. We initialize
$$\widehat{\mu}_i(0) = U(i, j_0).$$
3. ***Human move.*** *After seeing r_t, the human plays a best response:*
$$c_t \in \arg\max_{c\in[n]} U(r_t, c).$$
4. ***Payoff and feedback.*** *The realized utility is*
$$u_t = U(r_t, c_t).$$
The designer receives only bandit feedback: it observes (r_t, c_t, u_t) and updates only the played row.

For each row i, define its row-maximum value by
$$w(i) := \max_c U(i, c).$$

The key phenomenon is immediate lock-in: once the designer plays a row, the human reveals the best possible payoff obtainable from that row, which permanently reinforces the designer's estimate.

Theorem D.2 (Immediate lock-in and weak monotonicity)**.** *Suppose FTL uses a fixed tie-breaking rule, and let*
$$r_1 \in \arg\max_i U(i, j_0)$$
be the row selected at the first round. Then
$$u_t \equiv w(r_1) \qquad \text{for all } t \geq 1.$$
In particular, the realized utilities are weakly increasing (indeed, constant after the first play).

Proof. The human best-responds on r_1, so
$$u_1 = \max_c U(r_1, c) = w(r_1) \geq U(r_1, j_0) = \max_i U(i, j_0).$$

After the bandit update,
$$\widehat{\mu}_{r_1}(1) = \frac{1}{2}\big(U(r_1, j_0) + w(r_1)\big) \geq U(r_1, j_0) \geq \widehat{\mu}_i(1) \qquad \forall i \neq r_1.$$
Thus r_1 remains an FTL maximizer. Since ties are broken consistently, FTL again selects r_1.

Inductively, only row r_1 is ever updated. Each update averages in the same payoff $w(r_1)$, while every other row remains fixed at its initialization value. Therefore r_1 remains an FTL maximizer forever, so
$$r_t = r_1 \qquad \text{and} \qquad u_t = w(r_1)$$
for all $t \geq 1$. □

Thus, unlike the prompt-interface setting, the matrix game cannot exhibit deterioration: the process immediately stabilizes at a fixed row.

D.10 FULL FEEDBACK

We now turn to the full-feedback setting. Here, after each round, the designer updates the empirical estimate of *every* action against the realized opponent behavior, rather than only the played action.

This eliminates the mechanism responsible for deterioration: no action can become permanently "frozen out," since every action continues to receive updates forever. As a result, the dynamics become monotone in realized utility.

We first prove a general lemma for finite common-payoff games, and then apply it both to standard matrix games and to our prompt-interface game.

Lemma D.3 (Full-feedback best-response dynamics)**.** *Consider a finite common-payoff game with row actions R, column actions C, and shared payoff $U(r, c)$. Choose r_1 arbitrarily. At each round $t \geq 2$, the row player chooses*

$$r_t \in \arg\max_{r \in R} \frac{1}{t-1} \sum_{s<t} U(r, c_s),$$

In all rounds $t \geq 1$, the column player best-responds:

$$c_t \in \arg\max_{c \in C} U(r_t, c).$$

Then the realized utilities

$$u_t := U(r_t, c_t)$$

are weakly increasing

Proof. Since c_t is a best response to r_t,

$$u_t = \max_c U(r_t, c).$$

Let r be any row satisfying

$$\max_c U(r, c) < u_t.$$

Then in particular

$$U(r, c_t) < U(r_t, c_t).$$

Since r_t was chosen by FTL before observing c_t,

$$\sum_{s<t} U(r_t, c_s) \geq \sum_{s<t} U(r, c_s).$$

Adding the strict inequality

$$U(r_t, c_t) > U(r, c_t),$$

we obtain

$$\sum_{s \leq t} U(r_t, c_s) > \sum_{s \leq t} U(r, c_s).$$

Thus no row whose row-maximum value is strictly below u_t can be selected in round $t+1$. Therefore

$$u_{t+1} \geq u_t,$$

so the realized utilities are weakly increasing.

Because the game is finite, the sequence is bounded and therefore converges to some limit u_∞. □

We now obtain the remaining two quadrants of the table as immediate corollaries.

Theorem D.4. *In a standard matrix game without bandit feedback, the dynamics are always weakly utility-improving.*

Proof. Apply Lemma D.3 with rows as the designer's actions, columns as the human's actions, and payoff matrix U. □

Theorem D.5. *In our prompt-interface game without bandit feedback, the dynamics are always weakly utility-improving*

Proof. Apply Lemma D.3 to the common-payoff game whose row actions are designer policies $\pi_D : [K] \to A$, whose column actions are human prompting policies $\pi_H : I \to [K]$, and whose payoff is $u(\pi_H, \pi_D)$. □