NExT-Game 2026: New Frontiers in Game-Theoretic Learning – ICML 2026 Workshop

Failure Modes in AI Retraining Dynamics

author names withheld

Under Review for NExT-Game 2026

Abstract

Modern AI systems are increasingly retrained on data generated through interaction with users. Three forces are at play: (i) the users who strategically adapt their behavior, (ii) a prompting interface which obscures user intent, and (iii) the fact that AI is typically retrained "greedily," ignoring exploration-exploitation tradeoffs. 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. Second, a non-degenerate subset of problem instances exhibit *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 model produces outputs, users respond, and the resulting data is used to update the model. This feedback loop underlies reinforcement learning from human feedback, personalization systems, and interactive assistants.

A central challenge in such systems is that the data used for retraining is *endogenous*: it is produced by users who strategically adapt their behavior to the current model. Moreover, this interaction is mediated through a prompt interface, which obscures user intent. Users do not directly reveal their underlying objective, but instead choose prompts that best elicit useful responses from the current model. As the model changes, users may adapt by changing which prompts they issue—even when their underlying intent remains the same.

As a result, the model observes only prompt-level feedback, without any direct signal of the underlying intents that generated it. Because users adapt their prompts to the current model, the relationship between prompts and underlying intents shifts over time, even when user objectives remain fixed. The model therefore updates on feedback that is a function of its own induced behavior rather than purely of the underlying intent distribution. This induces a fundamentally game-theoretic learning process. The model's behavior shapes user responses, which in turn determine the data used for retraining.

We study a stylized model of this process. Users have latent intents and interact with the model via prompts, while the model maps prompts to answers and updates based on observed rewards. The model receives only bandit-style feedback at the prompt level, observing rewards only for actions it takes. This induces a natural benchmark: a static game over prompting and answer policies, whose optimal Nash equilibria represent ideal outcomes of interaction.

We show that retraining can fail in two qualitatively distinct ways: (i) *Optimization failure*: the system fails to converge to an optimal Nash equilibrium, and (ii) *Model Deterioration*: more severely, the system converges to an outcome that is strictly worse than the initial policy. In this case, retraining actively harms

performance rather than improving it.[1] Both failure types arise from the same underlying source: the interaction between limited exploration and strategically generated data. Because the model only learns from actions it takes, and because user behavior adapts to the model, early randomness or structural constraints can prevent the system from ever observing critical information. The prompt interface plays a key role in this process by aggregating feedback across heterogeneous intents, further limiting the model's ability to recover from early errors.

We formalize these phenomena and provide results for both. Optimization failure is ubiquitous: we show that it occurs with positive probability for essentially every non-degenerate problem instance. We lower-bound the failure probability optimization failure under several "mechanisms" causing it. While these lower bounds are very weak in general, they get (somewhat) stronger with additional structure. Regarding model deterioration, we exhibit a set problem instances where it always occurs, and show that this set is non-degenerate (in some technical sense). We also show that model deterioration can lose as much as a constant fraction of (a) the "best Nash" utility, (b) the initial utility.

RELATED WORK: exogenous distribution shift. A central assumption in supervised learning is that training and test data are drawn i.i.d. from a common distribution. A large body of work studies settings where this assumption fails, leading to distribution shift between training and deployment. A standard decomposition attributes such shifts to changes in the input distribution $P(x)$, the label distribution $P(y)$, or the conditional relationship $P(y \mid x)$, corresponding to covariate shift [3, 8, 17, 18, 20], label shift [1, 2, 9, 14, 15], and concept shift [5, 16, 19], respectively. We refer to [12] for a survey. More generally, shifts may involve changes in the support of the distribution (support shift), where test inputs lie outside the training domain, or arise from differences between source and target domains (domain shift). While certain structured shifts such as covariate or label shift under appropriate overlap and invariance assumptions admit principled correction methods (e.g., reweighting or prior adjustment), more general forms, particularly changes in $P(y \mid x)$ or support mismatch, can invalidate standard generalization guarantees [4].

While our work also considers a certain type of concept shift, in particular the shift caused by the change in user behavior, it is different from the above line of work in a number of ways which we highlight. Most importantly, these works focus on exogenous distribution shift: the source of the shift is an external factor independent of the model and they assume a bound on the magnitude of the shift [10, 11]. In contrast, the concept shift in our model is endogenous: the change in user behavior is driven by the change in the model and is not necessarily bounded in size. Additionally, the focus of these works is whether learning is possible using an algorithm designed for their setting. In contrast, we focus on the greedy dynamics arising from the users interacting with the model, and study the failures caused by this dynamics.

Endogenous distribution shift (arising from response to the deployed model). *Strategic classification* [7] 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 [7] considered a one-shot interaction, subsequent works have looked at the dynamics arising from the interaction. In particular, [21] study how the frequency with which the model and agent adapt determines the order of play in the Stackelberg equilibrium the dynamic converges to. Importantly, their work assumes that the incentives of the model and the agent are not aligned in general. In contrast, we assume that they are aligned, but show that the inability of the model to observe the intent behind each prompt causes the greedy dynamics to fail.

1. The second phenomenon is strictly stronger than the first. If a deterioration failure occurs, then the converged system utility must be strictly lower than some arbitrary starting state. But this starting state itself must be weakly lower than that of any optimal Nash equilibrium, as this is a shared-utility game (see Lemma 19). Therefore, any deteriorated game cannot have converged to an optimal Nash, so deterioration implies optimization failure. On the other hand, optimization failure does not imply deterioration; for example, the system may improve and converge to some Nash which is not the optimal Nash.

Performative prediction [13] posits that the data distribution itself depends on the deployed model, treating this dependence abstractly without specifying the underlying mechanism. This framework captures endogenous distribution shifts that may affect both the feature distribution and the conditional relationship between inputs and labels, and highlights that optimizing empirical risk on a fixed training distribution may be misaligned with the objective induced after deployment, leading to a focus on notions of performative stability and the behavior of retraining dynamics. This line of work generally does not directly model the way in which the agents react to the changes in the model. Instead, they assume *bounded sensitivity*: the change in the agent's behavior can be bounded based on the amount of change in the model [6, 13]. In contrast, we assume that the agents use best-response and that the model greedily optimizes over the entire past data (as opposed to just the previous episode) without knowing the agents' intent behind each prompt.

2. Model and Preliminaries

Learning via prompt interface. We study a strategic interaction between a population of users and a model that is retrained over time. 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, but the designer only observes feedback at the prompt level.

Formally, let I denote the set of human intents, A the set of answers, and $[K]$ the set of prompts. For non-degeneracy, we assume $|I|, |A|, K \geq 2$. The human population chooses a *prompting policy* $\pi_H : I \to [K]$ (i.e., a mapping from from intents to prompts), while the designer chooses a *answer policy* $\pi_D : [K] \to A$ (i.e., a mapping from prompts to answers).

Throughout the paper, we will typically speak in terms of intents rather than users. This is without loss of generality for our purposes: whether a single user is represented by one intent or by a set of intents, the strategic behavior of the human population is still captured by a mapping from each intent to a preferred prompt. Thus the relevant strategic object is the intent-level prompting policy.

Human behavior. Given an answer policy π_D, users best-respond: choose a prompting policy π_H that chooses a utility-maximizing prompt for each intent (let $\texttt{BR}(\pi_D)$ be the set of all such policies, see (1)).

Rewards. For each intent–answer pair (i, a) and each round t, the realized reward is an independent Bernoulli sample with mean $\mu_{i,a} \in [0, 1]$. In Section 3, the means are bounded away from 0 and 1, i.e., $\mu_{i,a} \in [\alpha, 1-\alpha]$ for some $\alpha > 0$. For a policy pair (π_H, π_D), define its expected utility by $u(\pi_H, \pi_D) := \sum_{i \in I} \mu_{i,a(i)}$, where $a(i) = \pi_D(\pi_H(i))$. In Section 4, we posit deterministic rewards ($\mu_{i,a} \in \{0, 1\}$).

Designer learning. The designer has access to all these rewards, in addition to the corresponding prompt and answer, but *not* to the underlying intents. Thus, the designer maintains empirical estimates $\hat{\mu}_{p,a}$ over prompt–answer pairs, and updates them using only the feedback it observes. Because feedback is aggregated across all intents mapped to a prompt, these estimates depend on the current human policy. Their initial beliefs are seeded via one round of full feedback from a surjective initial prompting policy.

Learning dynamics are as follows (see Figure 1): the designer first chooses an answer policy by greedily optimizing over empirical prompt-level rewards. Users then best-respond to this answer policy (as per above). The designer observes only aggregated prompt-level feedback from these interactions, updates its empirical estimates, and repeats the process.

Equilibrium benchmark. We evaluate the dynamics against the static game induced over policy pairs (π_H, π_D), with expected reward as the common payoff. An *optimal Nash equilibrium* is a Nash equilibrium of this game that maximizes the shared payoff. Equivalently, it is a policy pair that maximizes total expected reward (see Lemma 19). Let u^* denote the optimal Nash value.

3. Optimization Failures

In this section we show that the system exhibits optimization failure: gets stuck in suboptimal behavior with some positive-constant probability. 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.

Definition 1 Optimization failure*, denoted* `FAIL`*, is the event that for some* $\eta > 0$ *and all sufficiently large* T *we have* $\frac{1}{T}\sum_{t\in[T]} u(\pi_H^t, \pi_D^t) \le u^* - \eta$.

Any optimization failure happens with $\eta \ge \eta_0$, for some $\eta_0 > 0$ determined by the problem instance. Note that $\mathbb{P}[\texttt{FAIL}] > 0$ implies that policy regret is $\Omega(T)$; for more details, see Appendix C.1.

Our first result is that $\mathbb{P}[\texttt{FAIL}] > 0$ for all problem instances, under a minimal non-degeneracy condition.

Theorem 2 *Suppose some answer policy is not a part of any optimal Nash equilibrium. Then:*

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

The condition in the theorem is extremely permissive: in the degenerate case where every answer policy appears in some optimal Nash, each round in our dynamics trivially yields expected utility u^*.

Proof [Proof Sketch] The argument proceeds by identifying a self-reinforcing mis-estimation event that permanently locks the dynamics. The key is to construct an event in which the initial full-feedback data makes a fixed "decoy" policy (a policy which is not a part of any optimal Nash) $\bar{\pi}_D$ appear strictly optimal at every prompt, while assigning zero estimated value to every deviation from it. Because rewards are Bernoulli and independent across (i, a) pairs, this event factorizes cleanly across all intent–answer pairs, yielding the simple product lower bound. The technical crux is the dynamical consequence of this event. Once $\bar{\pi}_D$ is selected, the bandit feedback structure ensures that only realized prompt–answer pairs are ever updated. Thus, any action that is initially assigned value 0 is never revisited and never updated. Meanwhile, the chosen actions retain strictly positive empirical value under averaging. This creates a strict and permanent separation in the estimated utilities, so the greedy update rule never deviates. In this sense, the proof reduces a complex adaptive process to a one-shot "locking" event. ■

While the failure probability in Theorem 2 is quite small, it suffices to imply sublinear policy regret (which is a standard notion of a learning failure). That said, in the remainder of this section we provide stronger lower bounds on failure probability under additional structure.

Failure via Required Answers. We begin by considering what happens if there is a *required answer*, an answer which must be mapped to in order to achieve the optimal Nash.

Required answers capture settings where some type of response is indispensable for optimal performance. For example, in a tutoring system, some prompts may need to elicit the correct answer rather than a hint; in a medical assistant, some intents may require a referral rather than generic advice. If the required answer is not used, no rearrangement of prompts can recover the optimal outcome.

Definition 3 (Required Answer) *An answer* a *is a* required answer *if, in every optimal Nash equilibrium of the game,* a *is mapped to from some prompt.*

If a required answer exists, this simplifies the conditions needed for a failure event; we only require that this *particular* answer has zero estimated value in the initial data, and some other prompt has estimated value > 0. Thus the failure event remains exponentially small in $|I|$ and n_0, but no longer in $|A|$. We provide further details, as well as further improvements via an *grouping* structure, in Appendix B.

4. Model Deterioration

We now show a strictly stronger pathology than failure to reach the optimal Nash equilibrium: the dynamics may converge to a fixed point whose utility is strictly lower than that of the initial policy. In this section, we consider the special case where the rewards are *deterministic*, and show that even here, the system can exhibit deterioration. Specifically, for any game size, deterioration will occur on a positive measure set of instances, and can decrease system utility by up to 50 percent.

For our benchmark, let $u_0 := u(\pi_H^0, \pi_D^0)$ denote the utility of the initial outcome, where π_H^0 is the initial prompting policy and π_D^0 is the designer policy after initialization.

Definition 4 (Model deterioration) *We say the dynamics exhibit* model deterioration *if the time-averaged utility converges, and* $\lim_{T\to\infty} \frac{1}{T} \sum_{t\in[T]} u(\pi_H^t, \pi_D^t) < u_0$.

We begin by showing an example of a game exhibiting model deterioration and providing intuition.

Theorem 5 (Existence of deterioration) *There exists a retraining game in which the dynamics converge to a fixed point with strictly lower utility than the initial policy.*

The witness is as follows:

	a_1	a_2	a_3
i_1	1.00	0.98	0.99
i_2	0.07	0.30	0.01
i_3	0.00	0.00	1.00

Initialize $\pi_H^0(i_1) = \pi_H^0(i_2) = p_1, \quad \pi_H^0(i_3) = p_2$.

The deterioration mechanism hinges on a subtle interaction between averaging and selective feedback. Initially, the designer prefers an action based on aggregated rewards across multiple intents. However, once the human reassigns intents across prompts, the feedback becomes selectively biased: each prompt now receives data from a different subset of intents than before. This causes the empirical averages to shift in a way that favors a different action. This shift, while actually harming the system utility, can be irreversible because of bandit feedback. Once an action is no longer played, its estimate is frozen, while the competing action continues to receive updates and refine its estimate. This creates a one-way transition: the system can switch away from a high-performing configuration due to transient averaging effects, but cannot return because it no longer gathers the necessary information. The result is a stable fixed point that is strictly worse than the initial state.

Deterioration is a Positive-Measure Outcome. Next, we show that not only can deterioration exist, it can occur with positive probability in a game of any constant size.

Definition 6 (Non-degenerate set) *A set of retraining game instances is* non-degenerate *if it has positive Lebesgue measure as a subset of* $[0,1]^{|I|\cdot|A|}$.

Proving this requires two steps. First, we generalize the example above into a non-degenerate set of 3 by 2 by 3 games for which equivalent failure dynamics occur. Second, we show how to embed this instance many times into any larger game, giving us our final non-degeneracy result.

The inequalities defining deterioration for general 3 by 2 by 3 games encode the same qualitative mechanism as the example: an initial ranking that induces a switch, followed by a reallocation of feedback that prevents reversal. Crucially, these conditions define an open region of the parameter space. This implies

robustness: deterioration is not a knife-edge phenomenon, but persists under small perturbations of the rewards. From a geometric perspective, the proof identifies a region of $[0, 1]^9$ in which the induced dynamics follow the same trajectory. The openness of this region ensures that the phenomenon has positive measure, and therefore occurs with nonzero probability under any continuous distribution over games.

Next, we extend this construction to larger games by embedding the $3 \times 2 \times 3$ instance as a building block. The key idea is that the deterioration mechanism is local: it depends only on the relative ordering of a small set of intents, prompts, and answers, and is unaffected by the presence of additional actions as long as they are sufficiently separated in value.

Concretely, we partition the intents, prompts, and answers into blocks, and replicate the $3 \times 2 \times 3$ construction within each block. By assigning cross-block rewards to be uniformly lower than within-block rewards (but still allowing them to vary within a constant-size interval), we ensure that both the designer and the human strictly prefer to remain within blocks at every round. So, the dynamics decouple across blocks, and each block evolves independently according to the same deterioration trajectory as the base instance.

Theorem 7 *For any constants $|I| \geq 3$, $K \geq 2$, and $|A| \geq 3$, there is a non-degenerate set of retraining games of size $(|I|, K, |A|)$ that exhibit model deterioration.*

Maximum Harm. We can also quantify how much the model can deteriorate by. Define the *harm* of the deterioration as $H := W_0 - W_\infty$, where W_0 is the utility at the first round and W_∞ is the limit utility. When the limit does not exist H is undefined. However, note that model deterioration is defined in terms of the limit utility, and therefore any game that experiences model deterioration has a well-defined deterioration harm.

Theorem 8 *For any $|I|, |A|, K$ and ϵ, there exists a retraining game with dimensions $|I|, |A|, K$ where the deterioration harm is at least $\frac{1}{2} - \epsilon$.*

Model Deterioration is specific to prompt interface + bandit feedback. Finally, we show that the human-AI prompt interface combined with bandit feedback is precisely what allows us to realize model deterioration. When removing the bandit feedback, the prompt interface, or both, the dynamics are always weakly utility-improving.

The table isolates the two ingredients needed for deterioration. With full feedback, the designer continues to observe the value of unplayed actions, so harmful transitions can be corrected. Without the prompt interface, feedback is not pooled across strategically changing mixtures of intents, so greedy updates are monotone with respect to the relevant benchmark. Deterioration requires both: prompt-level aggregation creates biased estimates, and bandit feedback makes those biased estimates persistent.

	Prompt Game	Matrix Game
Bandit Feedback	Non-monotone; at least minmax	Monotone; at least minmax
Full Feedback	Monotone; at least worst Nash	Monotone; at least worst Nash

We provide complete details in Appendix C.12.

References

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

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

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

[4] 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*, pages 129–136. JMLR Workshop and Conference Proceedings, 2010.

[5] 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.

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

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

[8] 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.

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

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

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

[12] 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.

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

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

[15] 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.

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

[17] 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.

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

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

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

[21] 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

Appendix A. Model Dynamics: A Summary

Learning Dynamics

Fix an initial prompting policy π_H^0; assume it is surjective.
Let $I_p^t := \{\, i \in I : \pi_H^t(i) = p \,\}$. for each prompt p and round $t \in \mathbb{N}$.

Initialization (full feedback). For each prompt-answer pair (p, a):
the designer observes n_0 independent reward samples for every intent $i \in I_p^0$, and sets

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

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

1. **Designer update:** a new answer policy π_D^t such that

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

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

$$\mathtt{BR}(\pi_D) := \{\, \text{prompting policies } \pi_H : \pi_H(i) \in \arg\max_{p \in [K]} \mu_{i,\pi_D(p)} \ \forall i \in I \,\}. \tag{1}$$

3. **Bandit feedback.** For each prompt p such that $I_p^t \neq \emptyset$:
Let $a = \pi_D^t(p)$. 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

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

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

Figure 1: Learning dynamics in our model.

Appendix B. More on Optimization Failures (Section 3)

Theorem 9 $\mathbb{P}[\texttt{FAIL}] \geq \exp\left(-\Theta\left(n_0 \cdot |I|\right)\right)$ *for any problem instance with* ≥ 1 *required answer.*

Proof [Proof Sketch] The existence of a required answer sharpens the argument in Theorem 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 intent. This significantly simplifies the probabilistic structure: the event now decomposes across intents rather than across all (i, a) pairs. The dynamics now only need to learn avoid a^*. If a^* is never selected in the initial step, then it is never updated, and its estimate remains identically zero forever. Because every prompt always has some alternative action with strictly positive estimated value, the greedy policy will never revisit a^*. Thus, the failure mechanism again reduces to a one-shot elimination event, but now one that depends only on suppressing a single answer rather than an entire policy. This illustrates how mild structural assumptions can dramatically increase failure probability: the difficulty is no longer discovering a correct policy, but merely avoiding a single indispensable action. ■

Failure via Grouping Structure. We now consider another structure in the retraining space: that there are groups of intents that are all more similar to each other than to other intents, and thus prefer the same set of answers over any other answer. 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 category, many answers may be useful, while answers from other categories may be essentially irrelevant. Grouping formalizes the case where intents and answers naturally cluster into such domains.

Under this assumption, we can strengthen our failure lower bounds in two ways: (i) When the optimal Nash follows the grouping structure, our anti-concentration results, which previously depended exponentially on the total number on intents, need only depend exponentially on the number of intents in each group. (ii) Counterintuitively, the optimal Nash *need not* follow the grouping structure. When it does not, we can show much higher failure probabilities via concentration arguments rather than anti-concentration.

Definition 10 *Fix* $I_g \subset I$ *and* $A_g \subset A$. *A pair* (I_g, A_g) *is called a* Δ-robust group *if for all intents* $i \in I_g$ *and all answers* $a \in A_g$, $a' \notin A_g$ *we have* $\mu_{i,a} \geq \Delta$ *and* $\mu_{i,a'} = 0$.

Definition 11 *A problem instance is* (G, Δ)*-grouped if there exist* Δ*-robust groups* (I_g, A_g) *for* $g \in [G]$ *such that* $\{I_g\}_{g\in[G]}$ *partitions* I *and* $\{A_g\}_{g\in[G]}$ *partitions* A. *Let* $A_{\min} := \min_{g\in[G]} |A_g|$.

Definition 12 *An answer* $a \in A_g$ *is* ε*-safe for group* (I_g, A_g) *if for all* $i \in I_g$, $\mu_{i,a} \geq \varepsilon$. *An answer is* contained *in a group* g *if* $a \in A_g$.

Given an initial prompting policy that respects groupings, in addition to assigning intents and answers to groups, we can also assign each prompt p to a group g, corresponding to the group that all of its intents in the initial prompting policy are from. Therefore given an initial prompting policy we also have a notion of a "group" of prompt p. A prompting policy (resp. answer) policy *respects initial prompt groupings* if it always maps intents to prompts in the same group (resp. always maps prompts to answers in the same group).

Theorem 13 *Suppose the game is* (G, Δ)*-grouped, and suppose the initial prompting policy* p_0 *respects the groups. Let* g^* *contain a required answer* a^*, *and let there be an* ε*-safe answer* a_s *for* g^* *(for some absolute constant* ϵ*). Let* I_{g^*} *be the set of intents in* g^*, *let* K_g *be the number of prompts initially assigned to* g^*. *Suppose* $n_0 = \Omega((\log K)/(\Delta\, A_{\min}))$. *Then* $\mathbb{P}[\texttt{FAIL}] \geq \exp\left(-\Theta\left(n_0 \cdot |I_{g^*}|\right)\right)$.

Proof [Proof Sketch] The grouping structure changes the nature of the argument in two ways. First, it localizes the anti-concentration requirement: instead of needing a coordinated failure across all intents, it suffices to suppress the required answer within a single group. Second, it introduces a dynamical invariance: once the system respects group boundaries, it continues to do so indefinitely (Lemma 23). This effectively decomposes the global learning problem into independent subproblems at the group level.

The proof decomposes failure into three events, each capturing a different aspect of the dynamics: (i) a structural event ensuring that behavior remains confined within groups, (ii) a stability event ensuring that a safe but suboptimal answer remains attractive within a group, and (iii) an anti-concentration event ensuring that the required answer is initially underestimated. These events depend on disjoint sets of reward draws, allowing their probabilities to multiply. This decomposition reflects the underlying mechanism: grouping 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 group, rather than the full system. A more detailed statement of the bound is found in Theorem 22. ■

Group-Harmed Games. Thus far, grouping has acted as a stabilizing structure to reason about low-probability failure events: when the dynamics respect groups, they restrict the space of possible outcomes and allow us to evaluate failure separately in each group. We now show that in some cases, grouping is a failure in and of itself. There are grouped games where, if the dynamics respect groups, the outcome is a failure with probability 1.

Perhaps surprisingly, the optimal Nash equilibrium need not respect the grouping. Intuitively, if one group g^* has sufficiently high achievable utility relative to the others, it may be optimal to reallocate prompts away from other groups in order to better differentiate within g^*. In particular, it can be optimal to assign zero prompts to some group, thereby violating the grouping structure.

Group-harmed games. We call a game (G, Δ) *group-harmed* if it is (G, Δ) grouped and every optimal Nash equilibrium does not respect the grouping.

Theorem 14 *If a game is (G, Δ) group-harmed and the initial prompting policy respects the groups, then*

$$\mathbb{P}[\texttt{FAIL}] \;\geq\; 1 - \exp(-\Delta n_0 A_{\min} + \log K)\,.$$

In particular, if $\Delta n_0 A_{\min} \geq c \log K$ for a sufficiently large constant c, then $\mathbb{P}[\texttt{FAIL}] \geq 1 - o(1)$.

We will first show an example of a group-harmed game, and then a more robust characterization.

Theorem 15 *There exist (G, Δ)-group harmed games.*

Proof We give an explicit example with two groups and two prompts. Let $I_1 = \{i_1\}, I_2 = \{i_2, i_3\}$, and let $A_1 = \{a\}, A_2 = \{b, c\}$. Fix parameters $\delta, \varepsilon > 0$ with $\delta + \varepsilon < 1$. Define rewards by $\mu_{i_1,a} = \delta, \mu_{i_1,b} = \mu_{i_1,c} = 0$, and $\mu_{i_2,b} = 1, \mu_{i_2,c} = \varepsilon, \mu_{i_3,b} = \varepsilon, \mu_{i_3,c} = 1$, with all cross-group rewards equal to zero. Thus game is $(\{I_1, I_2\}, \Delta)$-grouped for any $\Delta \leq \min\{\delta, \varepsilon\}$.

The best allocation that respects groups obtains value at most $\delta + 1 + \varepsilon$. However, assigning both prompts to A_2, with one prompt mapped to b and the other to c, obtains value 2. Since $\delta + \varepsilon < 1$, we have $\delta + 1 + \varepsilon < 2$. Thus every grouping-respecting allocation is strictly suboptimal, so every optimal Nash equilibrium violates the grouping. Hence the game is group-harmed. ■

If a game is group-harmed, the dynamics and the optimum are misaligned. As shown in Lemma 23, the dynamics tend to preserve grouping, while in a group-harmed game the optimal Nash necessarily violates it. This leads to failure with high probability.

We now characterize when a game is group-harmed.

Definition 16 (Group value) *For a group g, let $v_n(g)$ denote the maximum total utility that can be obtained from the intents and answers in g when allocating n prompts to that group.*

Definition 17 (Subset value) *For a set of groups S, let $v_n(S)$ denote the maximum total utility that can be obtained from the intents and answers in S when allocating n prompts to that subset.*

Theorem 18 *A game is group-harmed if and only if there exists a group g such that*

$$v_p(g) < v_K((I, A) \setminus g) - v_{K-p}((I, A) \setminus g), \quad \textit{for every prompt } p \in [K - G + 1].$$

The condition says that group g is never worth allocating prompts to. For any possible number p of prompts assigned to g, the best value obtainable from g is strictly smaller than the marginal value of assigning those same prompts to the rest of the system. Thus, every optimal allocation abandons g, and therefore violates the grouping structure.

The proof is an allocation argument. If the inequality holds, then any allocation giving p prompts to g can be improved by reallocating those prompts to the remaining groups. Conversely, if no such group exists, then every group can be assigned some positive number of prompts without decreasing total value, yielding an optimal allocation that respects the grouping.

Appendix C. Proofs and Additional Content

Lemma 19 *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. ■

C.1. Additional Details On Regret

Remark 20 (Strict suboptimality) *By Lemma 19, any policy that is not part of an optimal Nash equilibrium has strictly suboptimal utility. Thus, whenever optimization failure occurs, the limiting outcome is bounded away from the optimal value by a fixed positive η.*

Note that an optimization failure on the scale of η can equivalently be seen as attaining η policy regret.

Remark 21 (Policy regret interpretation) *Consider the cumulative gap to the optimal Nash value u^*:*

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

Since $\pi_H^t \in BR(\pi_D^t)$ for all t, we can rewrite this as

$$\sum_{t=1}^{T} \big(u^* - u(BR(\pi_D^t), \pi_D^t)\big).$$

By Lemma 19, we have

$$u^* = \max_{\pi_D} u(BR(\pi_D), \pi_D).$$

Substituting this, the expression becomes

$$\sum_{t=1}^{T} \Big(\max_{\pi_D} u(BR(\pi_D), \pi_D) - u(BR(\pi_D^t), \pi_D^t)\Big),$$

which is exactly the regret with respect to the best fixed policy under this benchmark.

C.2. Proof of Theorem 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

$$\mathbb{P}(D) = \mathbb{P}\left(\forall(p,a) \in R_p(\pi_H, \bar{\pi}_D), \hat{\mu}_1(p,a) > 0 \ \cap \ \forall(p,a) \in \bar{R}_p(\pi_H, \bar{\pi}_D), \hat{\mu}_1(p,a) = 0\right) \tag{2}$$

$$\geq \mathbb{P}\left(\forall(i,a) \in R_i(\pi_H, \bar{\pi}_D), r_1(i,a) = 1 \ \cap \ \forall(i,a) \in \bar{R}_i(\pi_H, \bar{\pi}_D), r_1(i,a) = 0\right). \tag{3}$$

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

$$\mathbb{P}(D) \geq \prod_{(i,a) \in R_i(\pi_H, \bar{\pi}_D)} \mathbb{P}(r_1(i,a) = 1) \cdot \prod_{(i,a) \in \bar{R}_i(\pi_H, \bar{\pi}_D)} \mathbb{P}(r_1(i,a) = 0) \tag{4}$$

$$\geq \prod_{(i,a) \in R_i(\pi_H, \bar{\pi}_D)} \alpha \cdot \prod_{(i,a) \in \bar{R}_i(\pi_H, \bar{\pi}_D)} \alpha \tag{5}$$

$$= \alpha^{|I| \cdot |A|}. \tag{6}$$

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.3. Proof of Theorem 9

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.4. Proof of Theorem 13

We begin by restating Theorem 13 with more detail:

Theorem 22 *Suppose the game is (G, Δ)-grouped, and suppose the initial prompting policy p_0 respects the groups. Let g^* be a group containing both an ϵ-safe answer a_s and a required answer a^*. Let I_{g^*} be the set of intents in g^*, let K_g be the number of prompts initially assigned to g^*, and let*

$$A_{\min} := \min_h |A_h|.$$

Then the probability of failure is at least

$$(1 - e^{-\Delta n_0 A_{\min}})^{K-K_g}(1 - e^{-n_0\epsilon^2/2})^{K_g} \left(\frac{e^{-\lambda n_0}}{\sqrt{4n_0\epsilon(1-\epsilon/2)}} \right)^{|I_{g^*}|},$$

where

$$\lambda := \frac{(1-\alpha-\epsilon/2)^2}{(1-\alpha)\alpha}.$$

In particular, if

$$n_0 \geq \max\left\{ \frac{\log(2(K-K_g))}{\Delta A_{\min}}, \frac{2\log(2K_g)}{\epsilon^2} \right\},$$

then

$$\mathbb{P}[\text{failure}] \geq \frac{1}{4} \left(\frac{e^{-\lambda n_0}}{\sqrt{4n_0\epsilon(1-\epsilon/2)}} \right)^{|I_{g^*}|}.$$

Proof We define three events.

- $Grouped$: for every prompt p initially assigned to some group $h \neq g^*$, the initial data contains nonzero utility for at least one answer in A_h.
- $Safe$: for every prompt p initially assigned to g^*, the estimated utility of a_s satisfies

  $$\hat{u}(p, a_s) \geq \epsilon/2$$

  after initialization and at every later round.
- $Underestimate$: for every intent $i \in I_{g^*}$, the initial empirical reward of a^* satisfies

  $$\hat{u}(i, a^*) < \epsilon/2.$$

We first show that if all three events occur, then the dynamics fail. Since $Safe$ and $Underestimate$ occur, every prompt initially assigned to g^* strictly prefers a_s to a^* after the initial data. Since $Grouped$ occurs, Lemma 23 implies that prompts outside g^* continue to be mapped only to answers in their own groups. In particular, no prompt outside g^* is mapped to a^*. Thus, after initialization, no prompt is mapped to a^*: prompts in g^* prefer a_s, and prompts outside g^* respect their own groups.

We now show by induction that the Designer never maps any prompt to a^*. Suppose that, through round $r-1$, the Designer has never mapped any prompt to a^*. Since feedback after initialization is bandit feedback, the estimate for a^* has not changed. Hence, for prompts in g^*, $Underestimate$ still gives

$$\hat{u}(p, a^*) < \epsilon/2,$$

while $Safe$ gives

$$\hat{u}(p, a_s) \geq \epsilon/2.$$

So prompts in g^* are not mapped to a^*. For prompts outside g^*, mapping to a^* would violate their initial groupings, which cannot occur by Lemma 23. Therefore the Designer does not map any prompt to a^* in round r. By induction, a^* is never played.

Since a^* is a required answer, every optimal Nash equilibrium maps some prompt to a^*. Therefore, on the event

$$Grouped \cap Safe \cap Underestimate,$$

the dynamics cannot converge to an optimal Nash equilibrium. Hence this event implies failure.

It remains to lower bound the probability of these events. We couple the reward process by drawing $r_t(i, a)$ for every intent–answer pair (i, a) and every round t, whether or not that pair is realized by the dynamics. The dynamics only observe realized draws, but the unrealized draws are still well-defined random variables. Thus events such as $Safe$ and $Underestimate$ are well-defined even for pairs that may never be selected.

The three events depend on disjoint collections of reward draws: $Grouped$ depends only on in-group answers for groups outside g^*, $Safe$ depends only on draws of a_s in group g^*, and $Underestimate$ depends only on draws of a^* in group g^*. Therefore the events are independent, and

$$\mathbb{P}(Grouped \cap Safe \cap Underestimate) = \mathbb{P}(Grouped)\,\mathbb{P}(Safe)\,\mathbb{P}(Underestimate).$$

We now bound these three probabilities. For $Grouped$, fix a prompt initially assigned to a group $h \neq g^*$. There are at least $A_{\min}$ in-group answers in A_h. Since every such answer has mean at least Δ, the probability that none of them appears with nonzero reward in the n_0 initial samples is at most

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

Taking the product over the $K - K_g$ prompts outside g^* gives

$$\mathbb{P}(Grouped) \geq (1 - e^{-\Delta n_0 A_{\min}})^{K - K_g}.$$

Next, because a_s is ϵ-safe for group g^*, its true mean is at least ϵ for every intent in g^*. Thus, by Hoeffding's inequality, for each prompt initially assigned to g^*,

$$\mathbb{P}\left(\hat{u}(p, a_s) \geq \epsilon/2\right) \geq 1 - e^{-2n_0(\epsilon/2)^2} = 1 - e^{-n_0\epsilon^2/2}.$$

Applying this over the K_g prompts in group g^* gives

$$\mathbb{P}(Safe) \geq (1 - e^{-n_0\epsilon^2/2})^{K_g}.$$

Finally, for each intent $i \in I_{g^*}$, the mean reward of a^* is at most $1 - \alpha$. By the binomial anti-concentration bound,

$$\mathbb{P}\left(\hat{u}(i, a^*) < \epsilon/2\right) \geq \frac{e^{-\lambda n_0}}{\sqrt{4n_0\epsilon(1 - \epsilon/2)}}, \qquad \lambda := \frac{(1 - \alpha - \epsilon/2)^2}{(1 - \alpha)\alpha}.$$

The reward draws are independent across intents, so

$$\mathbb{P}(Underestimate) \geq \left(\frac{e^{-\lambda n_0}}{\sqrt{4n_0\epsilon(1 - \epsilon/2)}}\right)^{|I_{g^*}|}.$$

Combining the bounds gives

$$\mathbb{P}[\text{failure}] \geq (1 - e^{-\Delta n_0 A_{\min}})^{K - K_g} (1 - e^{-n_0 \epsilon^2/2})^{K_g} \left(\frac{e^{-\lambda n_0}}{\sqrt{4 n_0 \epsilon (1 - \epsilon/2)}} \right)^{|I_{g^*}|}.$$

For the simplified bound, if

$$n_0 \geq \frac{\log(2(K - K_g))}{\Delta A_{\min}},$$

then

$$(1 - e^{-\Delta n_0 A_{\min}})^{K - K_g} \geq \frac{1}{2}.$$

Similarly, if

$$n_0 \geq \frac{2 \log(2 K_g)}{\epsilon^2},$$

then

$$(1 - e^{-n_0 \epsilon^2/2})^{K_g} \geq \frac{1}{2}.$$

Therefore,

$$\mathbb{P}[\text{failure}] \geq \frac{1}{4} \left(\frac{e^{-\lambda n_0}}{\sqrt{4 n_0 \epsilon (1 - \epsilon/2)}} \right)^{|I_{g^*}|}.$$

■

Lemma 23 *If the game is (G, Δ)-grouped, $Grouped$ occurs and the initial prompting policy respects the groupings, then both the Designer and the Human will respect the initial prompt groupings every round for the entirety of the game.*

C.5. Proof of Lemma 23

Proof

We will proceed via induction on the number of rounds, showing that in each round, both the Designer and Human respect prompt groupings regardless of tiebreaking.

For the base case, we consider the first Designer policy after seeing the initial data, and the first Human policy after the initial prompting policy. Note that by assumption, the initial prompting policy respects the grouping. Therefore every prompt p is mapped to only by intents that are part of the same group (and, definitionally, mapped to by intents in its own group). Furthermore, because $Grouped$ occurs, the Designer's initial data shows that every prompt in group g has nonzero utility with at least some answer in group g. Furthermore, by the fact that the game is (G, Δ) grouped, each prompt p in group g must observe zero utility for every answer that is not in the same group. Therefore the Designer's initial policy will respect prompt groupings, regardless of tiebreaking. If the Designer's policy respects prompt groupings, then the Human policy in that same round will as well, also regardless of tiebreaking. Therefore, our base case is proven: in the first round, the Designer and the Human both respect the groupings regardless of tiebreaking.

Our inductive hypothesis is that at some round $r - 1$, the Designer and the Human have respected prompt groupings at that round and all previous rounds, and furthermore their preferences for doing so were strict.

Our inductive step is to prove that, if the inductive hypothesis holds, then the Designer and the Human will respect prompt groupings in round r and their preferences for doing so are strict. To see this, note that, if

the Human respected prompt groupings in all previous rounds, and the Designer previously strictly preferred to respect the prompt groupings at round $r-1$, then the Designer will also strictly prefer this at round r. Assume for contradiction that this is not true. Then there is some prompt $\hat{p}$ in group $\hat{g}$, and some answer $\bar{a}$ in group $\bar{g}$, such that the Designer would map $\hat{p}$ to $\bar{a}$ under some tiebreaking. The estimated utility of this choice must be 0, as by the inductive hypothesis all intents that have ever mapped to prompt $\hat{p}$ must be in group $\hat{g}$, and such prompts have zero utility with $\bar{a}$. In round $r-1$, the Designer strictly preferred to map $\hat{p}$ to some answer $\hat{a}$ in $\hat{g}$, and so $\hat{u}_r(\hat{p}, \hat{a}) = \gamma > 0$. If the estimated utility at round $r-1$ was $\gamma > 0$, the estimated utility at round r is at least $\frac{\gamma(r-1)}{r} > 0$. Therefore this is a better choice for the Designer to map $\hat{p}$ to under any tiebreaking, so we have reached a contradiction. Thus, the Designer strictly prefers to respect prompt groupings in round r. Finally, if the Designer respects prompt groupings in round i, then the best-responding Human will strictly prefer to respect prompt groupings in round i as well. This proves the inductive step. ■

C.6. Proof of Theorem 18

Proof ($\Rightarrow$) Suppose the game is group-harmed. Then in the optimal Nash equilibrium, some prompt p is mapped to by intents i_1 and i_2 in two different groups g_1 and g_2. The answer that p is mapped to can be in at most one of these groups. Assume w.l.o.g. that in the optimal Nash, p maps to an answer $a \notin g_2$. Then i_2 receives utility 0 in this equilibrium. Since users best-respond, every prompt available to i_2 must also yield utility 0; otherwise i_2 would deviate to a prompt with strictly positive utility. Because all within-group utilities are strictly positive and all cross-group utilities are 0, this implies that no prompt maps to an answer in group g_2. Therefore it must be that no prompts map to any answer in g_2. Thus the value of the optimal Nash is

$$v_K((I, A) \setminus g_2).$$

Consider any grouping-respecting allocation. Such an allocation must assign at least one prompt to each group, and hence there exist some

$$1 \leq p \leq K - G + 1$$

prompts which are mapped to only by intents in g_2. The total value of this allocation is at most

$$v_p(g_2) + v_{K-p}((I, A) \setminus g_2).$$

Since the optimal allocation excludes g_2, every such allocation must be strictly worse, giving

$$v_p(g_2) + v_{K-p}((I, A) \setminus g_2) < v_K((I, A) \setminus g_2),$$

which yields the desired condition with witness group g_2.

($\Leftarrow$) Suppose there exists a group g satisfying the condition. Then for every grouping-respecting allocation assigning p prompts to g,

$$v_p(g) + v_{K-p}((I, A) \setminus g) < v_K((I, A) \setminus g).$$

Thus any grouping-respecting policy is strictly suboptimal. Therefore the optimal Nash must violate the grouping, and the game is group-harmed. ■

C.7. Proof of Theorem 14

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

We first show that $Grouped$ implies optimization failure. Since the initial prompting policy respects the groups, Lemma 23 implies that, on $Grouped$, both the Designer and the Human respect the initial prompt groupings in every round. Therefore every policy pair reached by the dynamics is grouping-respecting.

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

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

Thus the dynamics exhibit optimization failure.

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

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

By a union bound over the K prompts,

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

Since $Grouped$ implies `FAIL`, this gives

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

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

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

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

C.8. Auxiliary Proofs for Theorem 7

Here, we characterize a full class of deterioration games and show that the deterioration pattern contains a nonempty open subset of the game space, and therefore has positive measure.

Let

$$R = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \in [0,1]^9.$$

Consider the following strict inequalities:

$$b + e > a + d, \tag{7}$$
$$a + d > c + f, \tag{8}$$
$$a > c, \tag{9}$$
$$c > b, \tag{10}$$
$$d > f, \tag{11}$$
$$e > f, \tag{12}$$
$$\frac{a+d}{2} > e, \tag{13}$$
$$i > g, \tag{14}$$
$$i > h, \tag{15}$$
$$\frac{c+i}{2} > g, \tag{16}$$
$$\frac{c+i}{2} > h. \tag{17}$$

Theorem 24 *If a game satisfies inequalities (7)–(17), then, from the initial prompting policy*

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

the dynamics deteriorate.

Proof Consider the case where the initial prompting policy is $i_1 \mapsto p_1, i_2 \mapsto p_1, i_3 \mapsto p_2$. Then the initial data for the designer will appear as:

$$\begin{pmatrix} \frac{a+d}{2} & \frac{b+e}{2} & \frac{c+f}{2} \\ g & h & i \end{pmatrix}$$

By (7) and (8), the designer will map p_1 to a_2, and by (14) and (15), they will map p_2 to a_3. Thus, the initial answer policy will be $p_1 \mapsto a_2, p_2 \mapsto a_3$. By (10), (12), and (15), the human will now map i_1 to p_2, i_2 to p_1, and i_3 to p_2. So the next prompting policy is unchanged at $i_1 \mapsto p_2, i_2 \mapsto p_1, i_3 \mapsto p_2$. The feedback that the designer gets is:

$$\begin{pmatrix} -- & e & -- \\ -- & -- & \frac{c+i}{2} \end{pmatrix}$$

Furthermore, as the human always best responds, as long as the designer continues to play $i_1 \mapsto p_2, i_2 \mapsto p_1, i_3 \mapsto p_2$, this is the feedback they will receive every round. We will now prove that this feedback will eventually cause them to switch to a very specific policy. Until they switch, their utility matrix will always be of the form

$$\begin{pmatrix} \frac{a+d}{2} & \frac{b+te}{t+2} & \frac{c+f}{2} \\ g & h & \frac{tc+(t+1)i}{2t+1} \end{pmatrix}$$

By (16) and (17), feedback of this form will always cause the designer to map p_2 to a_3. But by (8) and (13), feedback of this form will eventually cause the designer to map p_1 to a_1 (switching it from a_2).

After this occurs, the human will best-respond with the strategy $i_1 \mapsto p_1$ (by (9)), $i_2 \mapsto p_1$ (by (11)), and $i_3 \mapsto p_2$ (by (14)). The bandit feedback to the designer will get this round and whenever they play the mapping $p_1 \mapsto a_1, p_2 \mapsto a_3$ is

$$\begin{pmatrix} \frac{a+d}{2} & -- & -- \\ -- & -- & i \end{pmatrix}$$

Note that this feedback will not change the estimate of the payoff of p_1, so the designer will continue mapping it to i_1. Furthermore, by (14) and (15), this feedback also cannot make p_2 prefer to map to a_1 or a_2. Therefore this system converges to this state and in the limit gets time-averaged utility of $a + d + i$.

Note that the utility of the initial human policy paired with the initial designer policy is $b + e + i$. By (7), the converged-upon utility is strictly lower. Furthermore, note that as the human's action is the same as their initial action, this also implies that the final converged upon state is not a Nash in the policy game. ■

C.8.1. Proof of Theorem 7

Proof The inequalities (7)–(17) are strict linear inequalities, so their feasible region is open. It remains only to show that this region is nonempty. For example,

$$R = \begin{pmatrix} 1.000 & 0.998 & 0.999 \\ 0.001 & 0.500 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{pmatrix}$$

satisfies all inequalities strictly. Hence the feasible region contains an open neighborhood of this matrix. Therefore it has positive measure in $[0, 1]^9$, and under any continuous distribution with full support, robust deterioration occurs with positive probability. ■

C.9. Proof of Proposition 8

Proof Using (13), we have

$$e < \frac{a+d}{2}.$$

Therefore

$$H = b + e - a - d < b + \frac{a+d}{2} - a - d = b - \frac{a+d}{2}.$$

Since $a > c > b$ by (9) and (10), we have $a > b$, and since $d \geq 0$,

$$b - \frac{a+d}{2} < b - \frac{b}{2} = \frac{b}{2} \leq \frac{1}{2}.$$

Thus $H < 1/2$.

For tightness, fix $\epsilon > 0$ and consider

$$R_\epsilon = \begin{pmatrix} 1 & 1-2\epsilon & 1-\epsilon \\ \epsilon & 1/2 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

For sufficiently small $\epsilon > 0$, this matrix satisfies all inequalities strictly. Its harm is

$$H = (1 - 2\epsilon) + \frac{1}{2} - (1 + \epsilon) = \frac{1}{2} - 3\epsilon.$$

Taking ϵ arbitrarily small gives harm arbitrarily close to $1/2$. ■

C.10. Proof of Theorem 5

Proof Consider three intents, two prompts, and three answers with reward matrix

$$R = \begin{pmatrix} 1.00 & 0.98 & 0.99 \\ 0.07 & 0.30 & 0.01 \\ 0.00 & 0.00 & 1.00 \end{pmatrix}.$$

Initialize

$$\pi_H^0(i_1) = \pi_H^0(i_2) = p_1, \quad \pi_H^0(i_3) = p_2.$$

Step 1: Initial estimates. The Designer maintains a 2×3 matrix of empirical means, which is initialized with full feedback based on the initial prompt mapping as follows:

$$\hat{U}^0 = \begin{pmatrix} 0.535_2 & 0.64_2 & 0.50_2 \\ 0.00_1 & 0.00_1 & 1.00_1 \end{pmatrix}.$$

The subscripts denote the number of samples that have been seen thus far for each entry (as the estimates at every time step are the mean of all previous samples).

Thus

$$\pi_D^0 = (a_2, a_3).$$

Step 2: Human best response. Given (a_2, a_3):

$$(i_1, i_2, i_3) \to (p_2, p_1, p_2).$$

Step 3: Updated estimates. After one round of bandit feedback: - (p_1, a_2) receives reward 0.30, - (p_2, a_3) receives rewards $0.99, 1.00$.

The updated estimate matrix is:

$$\hat{U}^1 = \begin{pmatrix} 0.535_2 & 0.526_3 & 0.50_2 \\ 0.00_1 & 0.00_1 & 0.997_3 \end{pmatrix}.$$

The key observation is that only realized prompt–answer pairs are updated; all others remain fixed.

Step 4: Switch. Now on p_1:

$$0.535 > 0.526 > 0.50,$$

so the Designer switches:

$$\pi_D^1 = (a_1, a_3).$$

Step 5: Human best response. Given (a_1, a_3):

$$(i_1, i_2, i_3) \to (p_1, p_1, p_2).$$

Step 6: Stability. From this point, (p_1, a_1) continues to receive $(1.00, 0.07)$ and remains at 0.535, (p_1, a_2) is never played again and remains at 0.526, and (p_2, a_3) becomes closer and closer to 1.

Thus the policies are at a fixed point, and the estimate matrix stabilizes at

$$\hat{U}^\infty = \begin{pmatrix} 0.535 & 0.526 & 0.50 \\ 0.00 & 0.00 & 1.00 \end{pmatrix},$$

The value of the estimate of $(p2, a3)$ will never precisely reach 1, but it is the limit of the sequence.

Step 7: Utility comparison. Initial utility:

$$0.98 + 0.30 + 1.00 = 2.28.$$

Final utility:

$$1.00 + 0.07 + 1.00 = 2.07.$$

Thus the dynamics strictly decrease utility. ■

C.11. Proof of Theorem 25

Block construction. Let R be any base $3 \times 2 \times 3$ game satisfying inequalities (7)–(17). Choose $\epsilon > 0$ such that every intra-block payoff relevant to the deterioration is strictly larger than 2ϵ.

For any $D \geq 1$, construct a larger game G_D with $3D$ intents, $2D$ prompts, and $3D$ answers. Partition intents, prompts, and answers into D blocks, each containing 3 intents, 2 prompts, and 3 answers. Within each block, payoffs are given by a copy of R. Across blocks, all payoffs are at most ϵ.

Theorem 25 *For every $D \geq 1$, there exists a $(3D) \times (2D) \times (3D)$ game containing D dynamically isolated robust deterioration.*

Positive probability at scale. This block construction is robust to small perturbations. The base deterioration is defined by strict inequalities, and the block isolation condition has slack:

$$R_{\text{intra}} > 2\epsilon \qquad \text{while} \qquad R_{\text{inter}} \leq \epsilon.$$

Therefore the set of large games satisfying the deterioration conditions also contains an open neighborhood of the constructed game. In particular, under any continuous distribution with full support over the space of $(3D) \times (2D) \times (3D)$ reward tensors, a game with D dynamically isolated robust deterioration occurs with strictly positive probability.

Proof Consider an initial prompting policy that is block-contained and, within each block, uses the initial prompting policy from Lemma 24. We show that the dynamics never leave blocks.

For the Designer, the initial estimates of within-block answers are formed from intra-block payoffs, all of which are strictly larger than 2ϵ for the relevant actions. Any cross-block answer has payoff at most ϵ. Therefore, on every prompt, the Designer strictly prefers an answer from the same block.

For the Human, as long as the Designer chooses answers within each block, any intent obtains payoff greater than 2ϵ from some prompt in its own block and payoff at most ϵ from prompts in other blocks. Hence the Human strictly prefers to remain within its own block.

By induction, both players remain within blocks forever. Conditional on this isolation, the dynamics inside each block are exactly the dynamics of the base $3 \times 2 \times 3$ deterioration example. Therefore each block deteriorates, independently of the others. ■

C.12. 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**; at least minmax	Monotone; at least minmax
Full Feedback	Monotone; at least worst Nash	Monotone; at least worst Nash

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

C.13. 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 26 (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 27 (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

$$u_t \equiv w(r_1) \qquad \textit{for all } t \geq 1.$$

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.

However, the limiting value can still be substantially suboptimal.

Proposition 28 (Counterexample to "limit $\geq$ worst Nash") *The limiting value of the Bandit Best-Response Matrix Game need not exceed the value of the worst pure Nash equilibrium.*

Proof Consider

$$U = \begin{pmatrix} 2 & 6 & 10 \\ 2 & 5 & 1 \end{pmatrix}.$$

The row-maxima are

$$w(1) = 10, \qquad w(2) = 5.$$

The unique pure Nash equilibrium is $(1, 3)$ with value 10.

Reveal column $j_0 = 1$, so

$$U(\cdot, 1) = (2, 2).$$

Suppose the designer's tie-breaking rule selects $r_1 = 2$. Then the human best-responds on row 2, yielding utility 5, and by the previous theorem the process locks there forever:

$$u_t \equiv 5.$$

This is strictly below the Nash value 10. ■

The previous proposition shows that no Nash-based lower bound is possible in general. The exact guarantee is instead the minimum row-maximum value.

Corollary 29 (Tight universal lower bound) *In the Bandit Best-Response Matrix Game,*

$$u_\infty = w(r_1) \geq \min_i w(i) = \min_i \max_c U(i, c),$$

and this bound is tight.

Proof The equality

$$u_\infty = w(r_1)$$

follows from the lock-in theorem. Since

$$w(r_1) \geq \min_i w(i),$$

the lower bound follows immediately.

For tightness, consider

$$U = \begin{pmatrix} 1 & 0 \\ 0.9 & 100 \end{pmatrix}.$$

Then

$$w(1) = 1, \qquad w(2) = 100.$$

Revealing column 1 gives $(1, 0.9)$, so FTL selects row 1. The human best-responds with column 1, and the process locks at

$$u_\infty = w(1) = 1 = \min_i w(i).$$

Thus the bound cannot be improved in general. ■

C.14. 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 30 (Full-feedback best-response dynamics)

Consider a finite common-payoff game with row actions R, column actions C, and shared payoff $U(r, c)$. At each round t, the row player chooses

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

with arbitrary initialization, and 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, and

- *the dynamics converge to a value at least as large as the value of the worst pure Nash equilibrium.*

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_∞.

Now suppose for contradiction that

$$u_\infty < v_{\min}^{\mathrm{NE}},$$

where $v_{\min}^{\mathrm{NE}}$ denotes the value of the worst pure Nash equilibrium.

Since the utilities converge monotonically, every action pair occurring infinitely often must have utility exactly u_∞. Consider one such pair $(r^\star, c^\star)$.

Because u_∞ lies below every pure Nash value, $(r^\star, c^\star)$ is not a Nash equilibrium. Since the column player best-responds every round, $c^\star$ is a best response to $r^\star$. Therefore there exists some row $\bar{r}$ and some $\gamma > 0$ such that

$$U(\bar{r}, c^\star) \geq U(r^\star, c^\star) + \gamma.$$

Because $(r^\star, c^\star)$ occurs infinitely often, the column action $c^\star$ also occurs infinitely often. On each such round, row $\bar{r}$ gains at least γ more payoff than $r^\star$. Hence the cumulative payoff advantage of $\bar{r}$ over $r^\star$ diverges to $+\infty$.

Consequently, for sufficiently large time, the empirical payoff of $\bar{r}$ exceeds that of $r^\star$, contradicting that FTL continues to select $r^\star$ infinitely often.

Therefore

$$u_\infty \geq v_{\min}^{\mathrm{NE}}.$$

■

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

Theorem 31 *In a standard matrix game without bandit feedback:*

- *the dynamics are always weakly utility-improving, and*
- *the dynamics converge to a value at least as large as the value of the worst pure Nash equilibrium.*

Proof Apply Lemma 30 with rows as the designer's actions, columns as the human's actions, and payoff matrix U. ■

Theorem 32 *In our prompt-interface game without bandit feedback:*

- *the dynamics are always weakly utility-improving, and*
- *the dynamics converge to a value at least as large as the value of the worst Nash equilibrium.*

Proof Apply Lemma 30 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 $u(\pi_H, \pi_D)$. ■