Online Learning of Group-Stable Coalition Structures

Saar Cohen
Department of Computer Science
University of Oxford
saar.cohen@cs.ox.ac.uk

Michael Wooldridge
Department of Computer Science
University of Oxford
michael.wooldridge@cs.ox.ac.uk

Abstract

Coalition formation concerns partitioning agents into disjoint coalitions based on their preferences for one another. In *online learning* of coalition structures, agents' *true* preferences may be initially unknown. Thus, coalitions are repeatedly formed based on preferences learned *online* from iterative feedback derived from interactions in those coalitions. This work introduces a new *group stability* approach to online learning in coalition formation, relying only on *partial noisy* feedback observed *after* agents interact. We analyze the system in terms of common group stability notions, where no group of agents benefits by simultaneously joining other coalitions or forming a new one. Our goal is to learn coalition structures that are robust to such group deviations, while simultaneously determining whether stable outcomes exist under the underlying (unknown) preferences. To this end, we analyze algorithms through two complementary metrics: (1) the error in predicting whether a stable partition exists, and (2) a regret notion that quantifies violations of group stability over time. We propose an algorithm that combines optimistic and pessimistic estimates to explore potentially stable partitions and to establish stability from the observed feedback, and that explores by probing the most uncertain pairwise utility that keeps its pessimistic test from succeeding. We show that when no stable partition exists, our algorithm makes only a constant expected number of mistakes, while when a partition is stable with a positive margin, its error and regret grow logarithmically over time, provided its optimistic optimization problem is solved exactly. These guarantees hold under semi-bandit feedback, in which each pair of agents in a coalition reveals a noisy utility, and extend to bandit feedback, in which each agent reports only a noisy utility for its coalition. Experiments on synthetic games, on games derived from clustering and community-detection datasets, and under asymmetric utilities and heavy-tailed noise support our findings.

1 Introduction

A teacher repeatedly assigns students to groups for weekly collaborative projects without knowing their true preferences. After each project, the teacher observes only partial and noisy signals about how students benefited from their teammates, such as vague comments like "It was nice" or interactions that may be misinterpreted or affected by external factors (e.g., the project topic). Over time, the teacher aims to refine the groupings so that no small group of students would all prefer, by their *true* preferences, to move to other groups or form a new one. Such scenarios exemplify *coalition formation*, where *agents* coordinate in groups rather than alone.

The same structure arises in settings with higher stakes. A firm staffing recurring projects can learn from peer evaluations how well colleagues work together; a team in which a few employees would jointly prefer another arrangement generates transfer requests even when its total productivity is high. In federated learning and data markets, a platform may group institutions into training consortia

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

while they agree to follow its assignments, estimating pairwise compatibility from each partner's contribution to validation performance; a consortium that some institutions would jointly leave is fragile once that phase ends. In each case, a coordinator assigns the groups, but whether they persist depends on their stability against coordinated deviations, not only on total welfare. Our model applies where such assignment authority exists and additive pairwise utilities are a reasonable approximation.

A popular model for coalition formation is *hedonic games* [39], where each agent's preferences depend only on the coalition she belongs to, ignoring externalities. Outcomes are partitions into disjoint coalitions, often evaluated via *stability* notions [8, 21], which capture whether agents (individually or in groups) have incentives to remain in their assigned coalitions. Most prior work assumes that a central authority has full knowledge of agents' *true* preferences prior to forming coalitions. However, in many practical settings like our weekly projects example, these preferences are initially unknown, and only *partial* and *noisy* feedback becomes available *after* agents interact. Recent work has thus begun to explore online learning in hedonic games, but often under restrictive feedback models: Cohen and Agmon [27] studied welfare maximization under *full* feedback, and subsequent work studied *Nash stability* from the perspective of *selfish* agents, including decentralized learners, under *partial but noiseless* feedback [32, 29]. However, Nash stability only guards against unilateral deviations [17, 14]. Consequently, a partition that is stable against unilateral deviations may still be vulnerable to coordinated deviations by groups of agents.

In contrast, we introduce a new *group stability* perspective for online learning in coalition formation under *partial noisy* feedback, allowing coordinated deviations by subsets of agents. In particular, we consider k*-strong Nash stability* [17], which rules out joint deviations by groups of size at most $k \geq 1$, possibly to different coalitions. We also study the weaker k*-size core stability* [37, 43], which restricts deviations to forming a single new coalition. We further consider the stronger (ε, k)*-strong Nash stability*, which requires every deviation by a group of at most k agents to leave some deviating agent worse off by at least $\varepsilon \geq 0$. We analyze these notions in an *online learning* setting where preferences are initially unknown. At each round, a learner forms a partition, observes a *noisy* utility for every pair of agents that share a coalition, and uses this feedback to construct partitions that approach group stability over time, despite uncertainty in preferences. We call this *semi-bandit feedback*. In many applications, however, agents assess their coalition only as a whole. We therefore also study *per-agent coalition-level bandit feedback*, in which each agent reports a single noisy estimate of her total utility from her coalition rather than separate utilities for its individual members. This feedback is substantially less informative, since the report of an agent in a larger coalition aggregates the utilities of all its other members. The learner's assignments are implemented, and group stability evaluates the partitions it forms rather than describing how agents behave during learning (Appendix P).

However, even when the *true* preferences are known, such stability notions pose fundamental challenges: stable partitions may not exist, and determining their existence or computing them when they exist is often intractable [10, 17, 19, 50]. These challenges are exacerbated in our setting due to *noisy* and *incomplete* observations. Motivated by this, we aim to **(1)** learn partitions that are as close as possible to satisfying group stability, and **(2)** infer whether stable partitions exist. We evaluate performance using two metrics: *existence error*, measuring mistakes in predicting existence, and (ε, k)*-strong Nash regret*, quantifying cumulative violations of (ε, k)-strong Nash stability.

Our Contributions. Under semi-bandit feedback (Section 3), we propose *Upper and Lower Confidence Bounds for Group Stability* (**UCB-LCB-GS**), an algorithm that constructs *optimistic* and *pessimistic* estimates of group deviation gains by combining upper and lower confidence bounds on pairwise utilities. Unlike classical bandit approaches that rely solely on optimism or pessimism [62, 67, 68, 5], we use these estimates both to explore candidate partitions and to establish stability, by reasoning about worst- and best-case realizations of the unknown preferences. Optimism alone does not always yield the observations that the pessimistic test requires, so the algorithm explores through *witness probing*: when it cannot establish stability, it observes the most uncertain pair involved in the deviation that most threatens its optimistic candidate. We show that the expected number of declarations of existence is bounded by a constant when no (ε, k)-strongly Nash stable partition exists. When some partition is stable with a positive certification margin, that is, when its minimum group deviation gap exceeds ε by a positive amount, the existence error and regret grow logarithmically in the horizon, provided the optimistic max-min problem is solved exactly. We also show that a positive margin is necessary, and that purely optimistic exploration can fail even when it is positive. To handle the combinatorial family of group deviations and observation counts chosen by the learner, we relate

the optimistic and pessimistic estimates of the gap to the confidence radius of a single witness pair, and we bound the effect of approximate optimization.

We then study *per-agent coalition-level bandit feedback* (Section 4), in which, as described above, the initialization and the exploration of the semi-bandit algorithm no longer reveal the pairwise utilities, and the reports generated by the grand coalition do not even determine whether a stable partition exists. Our variant of UCB-LCB-GS for this setting initializes its pairwise estimates through a round-robin schedule of matchings, and obtains further observations through witness probes that place the selected pair in a two-agent coalition, whose two reports then measure the pair's mutual utility. Its pessimistic decision rule and its max-min problems are those of the semi-bandit algorithm. We prove corresponding soundness, existence-error and regret guarantees, with an initialization of $n-1$ or n rounds instead of one (Section 4 and Appendix F).

Empirically, we evaluate UCB-LCB-GS on synthetic games, on games derived from the Iris, karate club and jazz datasets [46, 78, 52], and under asymmetric utilities and heavy-tailed noise, with both semi-bandit and coalition-level bandit feedback (Section 5). The experiments show a trade-off between regret and reliable detection of stable partitions. Under semi-bandit feedback, UCB-LCB-GS has the lowest regret in almost all synthetic configurations and almost never declares existence wrongly, whereas optimism-based baselines detect existence sooner only by also declaring it where no stable partition exists. Under bandit feedback, UCB-LCB-GS remains sound, but its regret exceeds that of these baselines, and is several times its regret under semi-bandit feedback. Code and instructions for reproducing the experiments are available on `GitHub`.

Related Work. Offline work on hedonic games [39] and ASHGs [19] studies the existence and complexity of single-agent and group stability with known preferences [8, 77, 3, 15, 13, 17]. Online hedonic games [48, 20, 22, 21, 28, 30] assign arriving agents irrevocably, again with known preferences, and PAC learning [12, 45, 56, 72] infers static preferences from a fixed set of samples rather than from repeated interaction. Online learning for coalition formation has addressed welfare under full feedback [27], Nash stability for selfish, possibly decentralized, agents under partial but noiseless feedback [32, 29], and envy under noisy feedback [31]; in our model, a central learner forms the partitions from partial noisy feedback and targets group stability and the existence of stable partitions. Learning in matching markets restricts coalitions to pairs and typically assumes bipartite instances, monetary transfers, perfect feedback or restricted utilities [66, 57, 79, 65, 70, 40, 71, 69], whereas we allow coalitions of any size. Appendix B also covers online matching and fair division.

2 Preliminaries

We study an *online learning* version of hedonic games from a *group stability* perspective: over an *unknown* number of rounds $T \geq 1$, an *online learning algorithm* (hereafter, learner) aims to learn a partition of n agents into disjoint subsets (i.e., *coalitions*) that is stable against deviations by groups of agents, relying only on (possibly) *noisy partial* information about agents' *true* preferences. At each round, the learner picks a partition, each agent derives her utility from her coalition, and the learner observes noisy feedback on these utilities and uses it to pick the next partition.

Formally, the input to our problem is given by a finite set $N = \{1, \dots, n\}$ of n agents. Hereafter, we denote $[x] := \{1, \dots, x\}$ for $x \in \mathbb{N}$ and $[0] = \{0\}$. At any time $t \in [T]$, the learner selects a partition π^t of all agents in N, based on all available information from past rounds. For any partition π of N, we denote $|\pi|$ as the number of its coalitions and $\pi(i)$ as the coalition $C \in \pi$ such that $i \in C$.

After agents are assigned to coalitions based on the learner's chosen partition, each agent obtains a utility, determined by aggregating her utilities of other agents within her coalition. As common in the literature on online coalition formation (see, e.g., [29, 48]), we focus on *additively separable hedonic games* (ASHGs) with *symmetric* preferences[1], where any pair of agents assign the same numerical value toward each other. Formally, agent i's preferences are succinctly represented by a *cardinal utility function* $v_i : N \to \mathbb{R}$, specifying that agent i assigns a cardinal utility of $v_i(j) \in \mathbb{R}$ to any agent j. We use the convention that $v_i(i) = 0$ for any agent i. Preferences are *symmetric* if $v_i(j) = v_j(i)$ for any pair of agents $i, j \in N$, in which case we denote the ***true mutual valuation*** of agents i, j by $v_{i,j} := v_i(j) = v_j(i)$. Further, we denote the agents' ***true joint utility profile*** as

[1] Our techniques extend to *asymmetric* preferences, where two agents' valuations toward each other may differ: the learner then keeps one estimate per ordered pair, and the analysis carries over (Appendix P).

$\mathbf{v} = (v_{i,j})_{i,j \in N}$. The utility of each agent i from a coalition $C \ni i$ is then aggregated by summing her utilities from other coalition members, i.e., $u_i(C, \mathbf{v}) = \sum_{j \in C} v_i(j)$.

However, at each time t, the learner may only observe *partial* and *noisy* information about each agent's *true* mutual utilities from other members of her coalition. That is, at any time t, the learner attains one *noisy* utility feedback $V_{i,j}^t = v_{i,j} + \varepsilon_{i,j}^t$ about the *true* mutual utility of each pair of distinct agents i, j that share a coalition in π^t, where $\varepsilon_{i,j}^t$ is a σ^2-subgaussian noise with mean 0. We call this observation model *semi-bandit feedback*: the learner observes individual pairwise utilities, not only an aggregate outcome. We assume that the noises are independent across pairs and rounds, identically distributed across time, and independent of the learner's internal randomness.

Using this repeated feedback, the learner aims to learn a partition which is stable against deviations by groups of agents. Formally, given a partition π, a coalition $C \subseteq N$ is a *blocking coalition* for π if $u_i(C, \mathbf{v}) > u_i(\pi(i), \mathbf{v})$ for any agent $i \in C$. We consider following two *group stability* notions:

Definition 1 (k-Size Core Stability [37, 43])**.** *Given an integer $k \geq 1$, a partition π is* k-size core stable *(k-SCS) if it admits no blocking coalition of size at most k, i.e., every $C \subseteq N$ with $|C| \leq k$ contains an agent i with $u_i(C, \mathbf{v}) \leq u_i(\pi(i), \mathbf{v})$.*

Definition 2 (k-Strong Nash Stability [17])**.** *Given an integer $k \geq 1$, a partition π is* k-strongly Nash stable *(k-SNS) if no group of at most k agents can deviate simultaneously, possibly to different coalitions, so that all of them are strictly better off. Formally, for any partition π' obtained from π by a joint deviation of a subset $C \subseteq N$ with $|C| \leq k$ (i.e., $\pi(i) \neq \pi'(i)$ for any agent $i \in C$, while agents in C may join different coalitions in π'), there exists an agent $i \in C$ such that $u_i(\pi'(i), \mathbf{v}) \leq u_i(\pi(i), \mathbf{v})$, i.e., at least one deviating agent does not strictly improve her utility.*

Any k-strongly Nash stable partition is also k-size core stable [43]. Thus, we mainly focus on k-strong Nash stability; our results naturally extend to the latter notion. We also introduce a *stronger* notion with a margin against profitable deviations. Consider any partition π, any $k \geq 1$, and any subset of agents $C \subseteq N$ with $|C| \leq k$. Let $\Lambda_C(\pi)$ denote the set of partitions obtainable from the partition π via a group deviation performed by the agents in C. For any $\pi' \in \Lambda_C(\pi)$, we define the *group deviation gap* as the largest utility loss of any agent in C, or equivalently the negative of the smallest utility gain, when C deviates from π to π':

$$\mathrm{Gap}_C(\pi, \pi', \mathbf{v}) := \max_{i \in C} \left[u_i(\pi(i), \mathbf{v}) - u_i(\pi'(i), \mathbf{v}) \right], \tag{1}$$

which captures the joint incentive of all agents in C to deviate to π' rather than remain in their coalitions under π. Namely, if $\mathrm{Gap}_C(\pi, \pi', \mathbf{v}) < 0$, then the entire group C has an incentive to deviate; otherwise, if $\mathrm{Gap}_C(\pi, \pi', \mathbf{v}) \geq 0$, then at least one agent weakly prefers to remain in her coalition under π. Therefore, we can define a *stronger* notion than k-strong Nash stability as follows:

Definition 3 ((ε, k)-Strong Nash Stability)**.** *Given $\varepsilon \geq 0$ and an integer $k \geq 1$, a partition π is* (ε, k)-strongly Nash stable *((ε, k)-SNS) if every joint deviation by a group of at most k agents leaves at least one of its members worse off by at least ε, i.e., the* minimum group deviation gap *is at least ε:*

$$\mathrm{Gap}_k(\pi, \mathbf{v}) := \min_{C \subseteq N : |C| \leq k} \; \min_{\pi' \in \Lambda_C(\pi)} \mathrm{Gap}_C(\pi, \pi', \mathbf{v}) \geq \varepsilon. \tag{2}$$

If the partition π satisfies $\mathrm{Gap}_k(\pi, \mathbf{v}) \geq 0$, then π is also exactly k-strongly Nash stable. We denote by $\mathtt{SNS}_{\varepsilon,k}(\mathbf{v})$ the set of all (ε, k)-strongly Nash stable partitions.

For any $\varepsilon \geq 0$, $\mathtt{SNS}_{\varepsilon,k}(\mathbf{v}) \subseteq \mathtt{SNS}_{0,k}(\mathbf{v})$: (ε, k)-strong Nash stability implies exact k-strong Nash stability, which is $(0, k)$-strong Nash stability. The parameter ε is a *stability margin*: every group deviation of size at most k leaves some deviator worse off by at least ε.

However, even partitions satisfying the *weaker* notion of k-size core stability are not always guaranteed to exist, as we show in the following observation:

Observation 1. *There exists an ASHG with symmetric preferences that admits no k-size core stable partition for any $k \geq 3$, and thus admits no (ε, k)-strongly Nash stable partition for any $k \geq 3, \varepsilon \geq 0$.*

The instance is Example 1 of Aziz et al. [10], in which every partition is blocked by a coalition of at most three agents (Appendix C). Even with known utilities, stable partitions can be hard to compute. In symmetric ASHGs, deciding whether a core stable partition exists is strongly NP-hard and computing one is NP-hard [10]; every n-strongly Nash stable partition is core stable, but this

hardness does not by itself carry over to n-strong Nash stability. 1-strong Nash stability coincides with Nash stability [17]; a Nash stable partition always exists in symmetric ASHGs [19], but finding one is PLS-complete [50] (Appendix C.2).

Our first objective is to learn, from partial and noisy feedback, whether a k-strongly Nash stable partition exists. We measure the learner's performance by the cumulative number of incorrect existence decisions. Formally, at each time t, the learner outputs a binary decision $d_t \in \{0, 1\}$ based on the history up to time $t-1$, including all previously selected partitions and the corresponding noisy utility feedback. Here, $d_t = 1$ indicates that the learner predicts the existence of a k-strongly Nash stable partition (i.e., $\texttt{SNS}_{0,k}(\mathbf{v}) \neq \emptyset$), while $d_t = 0$ indicates that no such partition exists (i.e., $\texttt{SNS}_{0,k}(\mathbf{v}) = \emptyset$). We evaluate the performance of the learner under both existence and non-existence of such a partition via the *existence* and *non-existence errors*, defined as (respectively):

$$\mathcal{E}^{\exists}_{\varepsilon,T} := \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{d_t = 0\} \,\middle|\, \texttt{SNS}_{\varepsilon,k}(\mathbf{v}) \neq \emptyset\right], \quad \mathcal{E}^{\nexists}_{T} := \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{d_t = 1\} \,\middle|\, \texttt{SNS}_{0,k}(\mathbf{v}) = \emptyset\right], \quad (3)$$

where $\varepsilon \geq 0$, while the expectation is taken over the randomness in the utility feedback and, if applicable, the internal randomness of the learner. This asymmetric conditioning reflects that the existence error $\mathcal{E}^{\exists}_{\varepsilon,T}$ is evaluated with respect to the stronger notion of (ε, k)-strong Nash stability, whereas the non-existence error $\mathcal{E}^{\nexists}_{T}$ is defined with respect to *exact* k-strong Nash stability.

Further, when an (ε, k)-strongly Nash stable partition exists, we want each partition selected by the learner to be as close as possible to being (ε, k)-strongly Nash stable, despite relying only on partial and noisy feedback. To this end, we introduce the (ε, k)***-strong Nash regret***, which measures how far each selected partition is from satisfying Definition 3. Formally, the learner's (ε, k)-strong Nash regret for selecting the partitions $\{\pi^t\}_{t\in[T]}$ over T rounds is:

$$\mathcal{R}_{\varepsilon,T} := \mathbb{E}\left[\sum_{t=1}^{T} \max\left\{0, \varepsilon - \mathrm{Gap}_k(\pi^t, \mathbf{v})\right\} \,\middle|\, \texttt{SNS}_{\varepsilon,k}(\mathbf{v}) \neq \emptyset\right], \quad (4)$$

where the expectation is over the randomness in the noisy utilities and possibly the learner. When no (ε, k)-strongly Nash stable partition exists, $\Gamma_{\min} := \min_{\pi\in\Pi}(\varepsilon - \mathrm{Gap}_k(\pi, \mathbf{v})) > 0$; this includes instances with an exactly k-strongly Nash stable partition but none with margin ε. In this regime, the cumulative regret necessarily grows linearly with T, independently of the learner's decisions, and does not distinguish between algorithms. Our regret analysis therefore focuses on instances where (ε, k)-strongly Nash stable partitions exist.

The regret also controls the best partition formed: for $t^\star := \arg\max_{t\in[T]} \mathrm{Gap}_k(\pi^t, \mathbf{v})$, the violation $\max\{0, \varepsilon - \mathrm{Gap}_k(\pi^{t^\star}, \mathbf{v})\}$ is at most the average per-round violation, so its expectation is at most $\mathcal{R}_{\varepsilon,T}/T$; sublinear regret therefore makes the expected violation of the best partition formed vanish as T grows. Hence, one of our objectives is an (ε, k)***-strong Nash regret bound*** that is sublinear in the number of rounds T and polynomial in the number of agents n.

3 An Optimistic-Pessimistic Algorithm under Semi-Bandit Feedback

In this section, we devise the online learning algorithm outlined in Algorithm 1, termed *Upper and Lower Confidence Bounds for Group Stability* (**UCB-LCB-GS**). Its inputs are listed in Algorithm 1; the parameter $\beta \in (2, \infty)$ controls the width of the confidence intervals. Our approach builds on the principles of pessimism and optimism under uncertainty: we construct both optimistic and pessimistic estimates of the group deviation gap in (1) by combining upper confidence bounds (UCBs) and lower confidence bounds (LCBs) on the utilities. The pessimistic estimate evaluates whether a partition is stable under all plausible realizations of the utilities, and is used to decide whether to declare existence. The optimistic estimate evaluates whether a partition could be stable under some plausible realization, and guides exploration. Optimism alone does not always produce the observations needed by the pessimistic test, so the algorithm adds a targeted exploration step, *witness probing*.

When running UCB-LCB-GS (Algorithm 1), the learner maintains, for each pair of agents, UCB and LCB estimates of its mutual utility, computed from the pair's observations. UCB-LCB-GS operates in two phases. At time $t = 1$, it begins with an *initialization phase* specifically designed for our setting, in which the UCB and LCB estimates of every pair are initialized by forming the grand coalition (line

Algorithm 1 Upper and Lower Confidence Bounds for Group Stability (**UCB-LCB-GS**)

Input: n agents; variance σ^2; $k \geq 1$; stability margin $\varepsilon \geq 0$; uncertainty parameter $\beta \in (2, \infty)$.

1: **for each** time $t = 1, \ldots, T$ **do**
2: **if** $t = 1$ **then** The learner sets $d_1 = 0$ and forms the grand coalition, $\pi^1 = \{N\}$.
3: **else** The learner computes a pessimistic candidate $\underline{\pi}^t$ via (7).
4: **if** $G_t^-(\underline{\pi}^t) \geq \varepsilon$ **then**
5: The learner sets $d_t = 1$ and forms $\pi^t = \underline{\pi}^t$.
6: **else** The learner computes an optimistic candidate $\overline{\pi}^t$ via (9).
7: **if** $G_t^-(\overline{\pi}^t) \geq \varepsilon$ **then**
8: The learner sets $d_t = 1$ and forms $\pi^t = \overline{\pi}^t$.
9: **else** The learner sets $d_t = 0$.
10: Select $(C^\circ, \pi^\circ) \in \arg\min_{|C| \leq k,\, \pi' \in \Lambda_C(\overline{\pi}^t)} \widetilde{\mathrm{Gap}}_C(\overline{\pi}^t, \pi', \underline{\mathbf{v}}^{t-1}, \overline{\mathbf{v}}^{t-1})$.
11: Select a witness pair $e_t = \{i, j\}$, $i \in C^\circ$, $j \in (\overline{\pi}^t(i) \cup \pi^\circ(i)) \setminus \{i\}$, maximizing $b_{i,j}^{t-1}$.
12: Form $\pi^t = \overline{\pi}^t$ if $j \in \overline{\pi}^t(i)$, and $\pi^t = \pi^\circ$ otherwise.
13: The learner observes $V_{i,j}^t = v_{i,j} + \varepsilon_{i,j}^t$ for all $i \neq j$ with $j \in \pi^t(i)$.
14: The learner updates the UCBs $\{\overline{v}_{i,j}^t\}_{i,j \in N}$ and LCBs $\{\underline{v}_{i,j}^t\}_{i,j \in N}$ via (5).

2). This yields one noisy utility feedback about each mutual utility. Then, UCB-LCB-GS proceeds to a *learning phase* (lines 3-14), where it repeatedly creates partitions based on the UCB and LCB estimates and then updates those estimates using the obtained noisy feedback.

Below, we describe both phases in detail, showing how the UCBs and LCBs induce optimistic and pessimistic estimates of group deviation gaps, how the pessimistic estimate establishes that a partition is stable, and how witness probing selects the partition to form when this is not yet possible. Finally, we prove that the non-existence error is bounded by a constant, and that when some partition is stable with a positive margin, the existence error and regret grow only logarithmically in T.

Initialization Phase. Without initial exploration, the learner has no estimate for pairs it has never placed in the same coalition; their confidence intervals are unbounded, so no partition can be shown to be stable, and pairs whose co-location matters for stability may be overlooked indefinitely. To obtain one noisy utility feedback about each mutual utility, the learner forms the grand coalition (line 2). Each co-located pair i, j then yields one noisy feedback $V_{i,j}^t = v_{i,j} + \varepsilon_{i,j}^t$. Since $v_{i,j} = v_{j,i}$, the learner keeps one estimate per unordered pair. It estimates the *true* utility $v_{i,j}$ by the empirical average of the noisy utilities observed so far. Namely, let $T_{i,j}^t = \sum_{\tau=1}^t \mathbb{1}\{\pi^\tau(j) = \pi^\tau(i)\}$ be the number of rounds up to time t in which agents i, j shared a coalition. The ***empirical mean (noisy) utility*** of the pair up to time t is $\mu_{i,j}^t = \sum_{\tau=1}^t V_{i,j}^\tau \mathbb{1}\{\pi^\tau(j) = \pi^\tau(i)\}/(T_{i,j}^t \vee 1)$, where $a \vee b = \max\{a, b\}$. The learner then estimates $v_{i,j}$ via the UCB estimate $\overline{v}_{i,j}^t$ and the LCB estimate $\underline{v}_{i,j}^t$ (line 14), given under our assumption of independent σ^2-subgaussian noise by

$$\overline{v}_{i,j}^t = \mu_{i,j}^t + b_{i,j}^t, \quad \underline{v}_{i,j}^t = \mu_{i,j}^t - b_{i,j}^t, \quad b_{i,j}^t := \sqrt{\frac{2\sigma^2 \beta \log(t+1)}{T_{i,j}^t}}, \tag{5}$$

with $b_{i,j}^t := +\infty$ if $T_{i,j}^t = 0$, and $\overline{v}_{i,i}^t = \underline{v}_{i,i}^t = 0$. The estimates computed after round $t-1$ are used at round t, with radius $b_{i,j}^{t-1} = \sqrt{2\sigma^2 \beta \log t / T_{i,j}^{t-1}}$. The parameter $\beta \in (2, \infty)$ controls the degree of uncertainty; our assumption that $\beta > 2$ follows Lattimore and Szepesvári [63]. The radius $b_{i,j}^t$ widens the interval of pairs that have rarely been observed. It acts as an exploration bonus in the optimistic estimate below, and as a safety margin in the pessimistic one.

Learning Phase. At any time $2 \leq t \leq T$, the learner picks a partition π^t based on the UCB and LCB estimates computed at time $t-1$, and updates these estimates using the noisy feedback received. In particular, it uses the UCBs and LCBs from (5) to construct optimistic and pessimistic estimates of the *true* group deviation gap in (1). Formally, let $\overline{\mathbf{v}}^{t-1} = (\overline{v}_{i,j}^{t-1})_{i,j \in N}$ and $\underline{\mathbf{v}}^{t-1} = (\underline{v}_{i,j}^{t-1})_{i,j \in N}$ be the ***joint UCB profile*** and the ***joint LCB profile***, respectively. Now, consider any partition π, any subset of agents $C \subseteq N$ with $|C| \leq k$, and any $\pi' \in \Lambda_C(\pi)$. We first define the pessimistic estimate

of the *true* group deviation gap in (1):

$$\widetilde{\mathrm{Gap}}_C(\pi, \pi', \underline{\mathbf{v}}^{t-1}, \overline{\mathbf{v}}^{t-1}) := \max_{i \in C} \left[u_i(\pi(i), \underline{\mathbf{v}}^{t-1}) - u_i(\pi'(i), \overline{\mathbf{v}}^{t-1}) \right] \textbf{ (Pessimistic Estimate)} \quad (6)$$

The *pessimistic* estimate in (6) evaluates the deviation gap using LCBs for the utilities in π and UCBs for the utilities after the deviation to π'. This corresponds to a plausible realization of the utilities under which π appears as unstable as possible. Write $G_t^-(\pi) := \min_{C:|C|\le k} \min_{\pi' \in \Lambda_C(\pi)} \widetilde{\mathrm{Gap}}_C(\pi, \pi', \underline{\mathbf{v}}^{t-1}, \overline{\mathbf{v}}^{t-1})$ for the resulting estimate of $\mathrm{Gap}_k(\pi, \mathbf{v})$. The learner first picks a partition $\underline{\pi}^t$ that maximizes G_t^- (line 3):

$$\underline{\pi}^t \in \arg\max_{\pi \in \Pi} \min_{C \subseteq N:|C|\le k} \min_{\pi' \in \Lambda_C(\pi)} \widetilde{\mathrm{Gap}}_C(\pi, \pi', \underline{\mathbf{v}}^{t-1}, \overline{\mathbf{v}}^{t-1}) = \arg\max_{\pi \in \Pi} G_t^-(\pi). \quad (7)$$

To simplify the exposition and focus on the challenges in error and regret analysis, we assume access to an oracle that solves these optimization problems. Our experiments instead use a greedy local-search heuristic (Appendix I). Appendix G bounds the existence error and regret by the number of rounds on which the heuristic's optimistic candidate is insufficient; this needs no approximation guarantee, but logarithmic bounds require such rounds to be rare. Given a stability margin $\varepsilon \ge 0$, if $G_t^-(\underline{\pi}^t) \ge \varepsilon$, then under every plausible realization of the utilities, each group deviation of at most k agents from $\underline{\pi}^t$ leaves some deviating agent worse off by at least ε. In this case the learner forms $\underline{\pi}^t$ and declares, with $d_t = 1$, that it is (ε, k)-strongly Nash stable (lines 4-5).

Otherwise, the learner turns to the *optimistic* estimate of the *true* group deviation gap in (1):

$$\widetilde{\mathrm{Gap}}_C(\pi, \pi', \overline{\mathbf{v}}^{t-1}, \underline{\mathbf{v}}^{t-1}) := \max_{i \in C} \left[u_i(\pi(i), \overline{\mathbf{v}}^{t-1}) - u_i(\pi'(i), \underline{\mathbf{v}}^{t-1}) \right] \textbf{ (Optimistic Estimate)} \quad (8)$$

Unlike the *pessimistic* estimate, the *optimistic* estimate in (8) evaluates the deviation gap using UCBs for the utilities in the current partition π and LCBs for the utilities after the deviation to π'. This corresponds to a plausible realization of the utilities under which π appears as stable as possible. Writing $G_t^+(\pi)$ for the corresponding minimum over $|C| \le k$ and $\pi' \in \Lambda_C(\pi)$, the learner next picks a partition $\overline{\pi}^t$ that maximizes G_t^+ (line 6):

$$\overline{\pi}^t \in \arg\max_{\pi \in \Pi} \min_{C \subseteq N:|C|\le k} \min_{\pi' \in \Lambda_C(\pi)} \widetilde{\mathrm{Gap}}_C(\pi, \pi', \overline{\mathbf{v}}^{t-1}, \underline{\mathbf{v}}^{t-1}) = \arg\max_{\pi \in \Pi} G_t^+(\pi). \quad (9)$$

The two estimates serve different purposes. Whenever the confidence intervals contain the true utilities, the pessimistic estimate never exceeds the true gap and the optimistic estimate never falls below it. A partition that passes the pessimistic test is therefore stable, while $\overline{\pi}^t$ is the partition that is most likely to be stable given the data, and thus the natural candidate to explore. The learner also applies the pessimistic test to $\overline{\pi}^t$ (line 7); with an exact oracle this second test never succeeds when the first one fails, because $G_t^-(\overline{\pi}^t) \le G_t^-(\underline{\pi}^t)$, but with an approximate oracle it lets the learner declare existence whenever either candidate passes. If neither candidate passes, the learner predicts that no (ε, k)-strongly Nash stable partition exists by setting $d_t = 0$ (line 9), and explores.

Witness probing. A natural way to explore is to form $\overline{\pi}^t$, which costs little regret when the optimistic estimate is accurate. Establishing that a partition is stable, however, requires accurate estimates for the pairs that its deviations would bring together, and these are pairs that the partition keeps apart; forming the partition never observes them. Appendix E gives a two-agent game on which this purely optimistic exploration settles on the stable partition yet never declares existence. UCB-LCB-GS therefore uses the deviations of $\overline{\pi}^t$ to decide which pair to observe. Since $\overline{\pi}^t$ failed the pessimistic test, some deviation of $\overline{\pi}^t$ has a pessimistic gap below ε. It selects one that is most damaging under the pessimistic estimate, a minimizing pair (C°, π°) (line 10). Its pessimistic and optimistic gaps differ only through its *witness pairs*: the pairs $\{i, j\}$ with $i \in C^\circ$ and j in the coalition of i before or after the deviation. Among them, the learner selects a pair e_t with the widest confidence interval. If $\overline{\pi}^t$ already places this pair in a common coalition, the learner forms $\overline{\pi}^t$; otherwise, it forms the deviation π°, which does (line 12). Either way, the selected pair is observed. Lemma D.3 shows that the gap between the optimistic and pessimistic estimates of $\overline{\pi}^t$ is at most $\kappa_n := 2(2n-3)$ times the selected pair's confidence radius. Hence, while a partition with a positive margin exists but cannot yet be shown to be stable, each round observes a pair whose uncertainty is still large.

After forming π^t, the learner observes the feedback $V_{i,j}^t$ of every pair sharing a coalition and updates their estimates via (5) (lines 13–14).

Analysis of Non-Existence Error. When no (ε, k)-strongly Nash stable partition exists, the learner can declare existence only if its pessimistic test incorrectly accepts a partition. We bound the expected number of such declarations by a *constant* that depends only on n and β; this also bounds the non-existence error on instances without an exactly k-strongly Nash stable partition.

Theorem 3.1 (Non-Existence Error Bound)**.** *For any input with $\mathtt{SNS}_{\varepsilon,k}(\mathbf{v}) = \emptyset$, UCB-LCB-GS satisfies $\sum_{t=1}^{T} \Pr[d_t = 1] \le \frac{n(n-1)}{\beta-2}$. In particular, $\mathcal{E}_T^{\nexists} \le \frac{n(n-1)}{\beta-2}$. The bound holds for any method of computing the candidate partitions, provided the pessimistic test is evaluated exactly.*

Proof sketch. Consider a round at which every confidence interval in use contains the corresponding true utility. For any deviation $\pi' \in \Lambda_C(\pi)$, the pessimistic estimate (6) uses lower bounds for the utilities of each member of C in π and upper bounds for those in π', so it is at most the true group deviation gap. Taking minima over deviations gives $G_t^-(\pi) \le \mathrm{Gap}_k(\pi, \mathbf{v})$ for every partition π. A partition that passes the pessimistic test is therefore truly (ε, k)-strongly Nash stable, and on an instance without such a partition the learner can declare existence only at rounds where some interval fails. The number of observations of a pair is random and depends on earlier observations, because the learner chooses which pairs to observe. A supermartingale argument, combined with a union bound over the possible observation counts, shows that the expected number of rounds with a failed interval is at most $\frac{n(n-1)}{\beta-2}$ (Appendix D.1).

Analysis of Existence Error and (ε, k)-Strong Nash Regret. When a stable partition exists, we ask how often the learner fails to declare existence and how far its partitions are from being stable. Soundness alone does not address this: a learner that never declares existence is sound but has existence error T. How quickly the learner can establish stability depends on the margin by which the best partition is stable. We measure this by the *certification margin*

$$\Delta_\star := \max_{\pi \in \Pi} \mathrm{Gap}_k(\pi, \mathbf{v}) - \varepsilon. \tag{10}$$

$\Delta_\star \ge 0$ if and only if $\mathtt{SNS}_{\varepsilon,k}(\mathbf{v}) \neq \emptyset$, and $\Delta_\star > 0$ means that some partition is (ε, k)-strongly Nash stable with room to spare. A positive margin is needed. If $\Delta_\star = 0$, the best partition is stable only with zero slack, and its pessimistic estimate stays below ε whenever the confidence intervals hold. Any learner that declares existence only through such a test therefore makes $T - \mathcal{O}(1)$ existence errors (Appendix E). Let $\Gamma := \max_{\pi \in \Pi} (\varepsilon - \mathrm{Gap}_k(\pi, \mathbf{v}))^+$ denote the largest per-round regret, and $\Gamma_N := (\varepsilon - \mathrm{Gap}_k(\{N\}, \mathbf{v}))^+$ the regret of the grand coalition.

Theorem 3.2 (Existence Error Bound)**.** *For any input with $\Delta_\star > 0$, if the optimistic problem* (9) *is solved exactly, then for $T \ge 2$ the existence error of UCB-LCB-GS satisfies*

$$\mathcal{E}_{\varepsilon,T}^{\exists} \le 1 + \tbinom{n}{2} \lceil 2\kappa_n^2 \sigma^2 \beta \log T / \Delta_\star^2 \rceil + n(n-1)/(\beta - 2).$$

Theorem 3.3 (Regret Bound)**.** *Under the conditions of Theorem 3.2, the (ε, k)-strong Nash regret of UCB-LCB-GS satisfies*

$$\mathcal{R}_{\varepsilon,T} \le \Gamma_N + \Gamma \binom{n}{2} \lceil 2\kappa_n^2 \sigma^2 \beta \log T / \Delta_\star^2 \rceil + \Gamma \, \frac{n(n-1)}{\beta - 2}.$$

Proof sketch. Let $\pi^\star$ be a partition with $\mathrm{Gap}_k(\pi^\star, \mathbf{v}) = \varepsilon + \Delta_\star$. When all confidence intervals hold, the optimistic estimate of $\pi^\star$ is at least its true gap, so an exact solution of (9) satisfies $G_t^+(\overline{\pi}^t) \ge \varepsilon + \Delta_\star$. On a round without a declaration, however, $\overline{\pi}^t$ fails the pessimistic test, $G_t^-(\overline{\pi}^t) < \varepsilon$, so its two estimates differ by more than $\Delta_\star$. The witness-gap inequality (Lemma D.3) shows that this difference is at most κ_n times the confidence radius of the witness pair selected in that round. The witness pair therefore has a radius larger than $\Delta_\star / \kappa_n$, that is, fewer than $2\kappa_n^2 \sigma^2 \beta \log T / \Delta_\star^2$ observations, and the round observes it once more. Each of the $\binom{n}{2}$ pairs can be the witness pair of only that many such rounds, which gives the logarithmic term of Theorem 3.2; rounds at which some interval fails add the constant of Theorem 3.1. For the regret, a declaration made while all intervals hold forms a partition that passes the pessimistic test, which is then (ε, k)-strongly Nash stable and incurs zero regret. Every other round costs at most Γ, and the same counting bounds their number: the initialization round, at most $\binom{n}{2} \lceil 2\kappa_n^2 \sigma^2 \beta \log T / \Delta_\star^2 \rceil$ exploration rounds, and in expectation at most $\frac{n(n-1)}{\beta-2}$ rounds with a failed interval (Appendices D.2 and D.3). Both bounds grow logarithmically in T, as $\mathcal{O}(n^4 \sigma^2 \beta \log T / \Delta_\star^2)$ rounds. The dependence on k enters through $\Delta_\star$, Γ and the cost of the oracle. Witness probing makes these bounds possible. Without it, the two-agent instances of Appendix E have linear existence error and regret despite a positive certification margin.

4 Coalition-Level Bandit Feedback

In many applications, feedback is coarser than under semi-bandit feedback; for example, an employee rates the team as a whole. We therefore also study *per-agent coalition-level bandit feedback*. After the learner forms π^t, each agent i reports $R_i^t = u_i(\pi^t(i), \mathbf{v}) + \eta_i^t = \sum_{j \in \pi^t(i) \setminus \{i\}} v_{i,j} + \eta_i^t$, with mean-zero, σ^2-subgaussian noises η_i^t that are independent across agents and rounds. The learner thus receives n numbers per round instead of one per co-located pair. The metrics and d_t are unchanged.

Additive separability still reduces the learning problem to the $\binom{n}{2}$ pairwise utilities, but these are no longer observed directly: an agent in a coalition with more than one other member reports the sum of several unknown utilities. In particular, the totals reported in the grand coalition do not determine whether a stable partition exists (Proposition F.1). A two-agent coalition $\{i, j\}$, in contrast, yields $R_i^t = v_{i,j} + \eta_i^t$ and $R_j^t = v_{i,j} + \eta_j^t$, two independent observations of $v_{i,j}$. Our algorithm for this setting, Algorithm 2 in Appendix F, obtains all its observations from such coalitions. It begins with a round-robin schedule of r matchings, with $r = n - 1$ for even n and $r = n$ for odd n, whose two-agent coalitions contain every pair exactly once, yielding two independent observations of each pairwise utility. From then on, with the confidence bounds of (5) computed from these observations, the algorithm proceeds as under semi-bandit feedback. It computes the pessimistic and optimistic candidates by the max-min problems (7) and (9), declares existence if either candidate passes the pessimistic test, and otherwise selects a witness pair exactly as in Algorithm 1. Forming $\overline{\pi}^t$ or its most damaging deviation would reveal the utility of this pair only if the pair formed a coalition by itself, so, unless $\overline{\pi}^t$ already contains it as a two-agent coalition, the algorithm forms $\overline{\pi}^t$ with the two agents moved into a new coalition of their own. The stability criterion, the confidence bounds and the optimization problems are those of the semi-bandit algorithm; what changes is the way in which observations are collected (Appendix F.2).

Theorem 4.1 (Coalition-Level Feedback)**.** *Consider per-agent coalition-level bandit feedback with symmetric utilities, and let $\beta > 2$. Let $P_1, \dots, P_r$ be the round-robin initialization partitions of Algorithm 2, write $\Gamma_P := (\varepsilon - \mathrm{Gap}_k(P, \mathbf{v}))^+$ for the regret of a partition P, and let Γ, $\Delta_\star$ and $\kappa_n = 2(2n-3)$ be as in Section 3. Then Algorithm 2 satisfies the following.*

- *(a)* Soundness. *For any input with $\texttt{SNS}_{\varepsilon,k}(\mathbf{v}) = \emptyset$, $\sum_{t=1}^{T} \Pr[d_t = 1] \le \frac{2n(n-1)}{\beta-2}$, and in particular $\mathcal{E}_T^{\nexists} \le \frac{2n(n-1)}{\beta-2}$. This holds for any method of computing the candidate partitions, provided the pessimistic test is evaluated exactly.*
- *(b)* Existence error. *For any input with $\Delta_\star > 0$ and any $T \ge 2$, if the optimistic candidate maximizes the objective of* (9) *exactly at every round, then, with $y := \kappa_n^2 \sigma^2 \beta \log T / \Delta_\star^2$, $\mathcal{E}_{\varepsilon,T}^{\exists} \le \min\{T, r\} + \binom{n}{2}\lceil y \rceil + \frac{2n(n-1)}{\beta-2}$.*
- *(c)* Regret. *Under the assumptions of (b), $\mathcal{R}_{\varepsilon,T} \le \sum_{s=1}^{\min\{T,r\}} \Gamma_{P_s} + \Gamma \binom{n}{2}\lceil y \rceil + \Gamma \frac{2n(n-1)}{\beta-2}$.*

Soundness is proved exactly as for Theorem 3.1; the constant doubles because a pair can now be observed twice in a round. The bounds of (b) and (c) follow from the counting argument behind Theorems 3.2 and 3.3. Their leading term is half that of Theorem 3.2, since each probe yields two observations, but the initialization takes r rounds instead of one (Appendix F.4).

5 Experiments

Experimental Setup. Following Boehmer et al. [18], we consider two models for pairwise utilities. In the *Uniform* model, utilities are sampled uniformly from the integers in $[-100, 100]$. In the *Gaussian* model, utilities are rounded Gaussian perturbations of such integers (Appendix I). We vary one parameter at a time, T, n, k, σ, ε or β, around the default configuration $(n, k, T, \sigma, \beta, \varepsilon) = (6, 2, 30, 3, 1, 1)$, repeat these ablations with $\beta = 3$, which our theorems cover, and run the default configuration for longer horizons (Appendix K). Every configuration uses 100 seeds, and all algorithms are run on the same instances. All methods use the same local-search heuristic and an exact deviation checker. Each instance is labelled by exhaustive search as YES (an (ε, k)-strongly Nash stable partition exists), NO (no k-strongly Nash stable partition exists) or intermediate. Regret and existence error are averaged over YES instances, the non-existence error over NO instances, with ± 1 standard-error bands.

Baselines. Six baselines isolate the roles of optimism and pessimism (Appendix J). **PESSIMISTIC-ONLY** and **OPTIMISTIC-ONLY** select and test partitions with pessimistic or optimistic estimates; **UCB-ONLY** and **LCB-ONLY** use only UCBs or only LCBs for both. **MIXED** adds an optimistic fallback test to the pessimistic one, without targeted exploration. The learning baselines form a random partition when their test fails, and **RANDOM** always does so and never declares existence. Under coalition-level feedback, the learning baselines keep these decision rules but use the estimator of Section 4, after first forming the round-robin matchings; RANDOM is unchanged (Appendix L).

Empirical findings. The two objectives must be considered together: always declaring existence trivially gives zero existence error, and low regret does not show that existence is detected correctly. Across the ablations (Appendix K), UCB-LCB-GS is the only method that combines low (ε, k)-strong Nash regret with reliable existence detection. It attains the lowest regret of all methods in 92 of the 96 configurations of the two grids and in all long-horizon configurations, and it declared existence on an instance without a stable partition only once, with $\beta = 0.25$, outside the range covered by Theorem 3.1. Its regret grows slowly with the horizon and the number of agents, as our bounds predict: in the default configuration it is about 180 at $T = 30$ and 690 at $T = 1000$ with $\beta = 3$, against more than 6,000 for every baseline. Its remaining existence errors concentrate on instances with a small certification margin, which no sound learner can resolve quickly (Section 3). The baselines fail in different ways: PESSIMISTIC-ONLY establishes stability late, so its regret grows linearly; OPTIMISTIC-ONLY, UCB-ONLY and MIXED attain competitive regret on easy instances but also declare existence where no stable partition exists; and LCB-ONLY declares it on almost every round. A baseline has lower regret than UCB-LCB-GS only at the largest noise level, $\sigma = 20$. Taken together, *neither optimism nor pessimism alone is sufficient*: pessimistic tests make declarations of existence reliable, and targeted exploration makes them possible within a short horizon.

Dataset-derived games, robustness, weaker feedback and computation. On games of 8 to 12 agents built from the Iris [46], karate club [78] and jazz [52] datasets, with friends, enemies and neutrals defined by distances as in Csáji et al. [36], UCB-LCB-GS has the lowest regret of the seven methods, alone or tied, in 22 of 36 configurations under semi-bandit feedback and in 9 under bandit feedback; the baselines with lower regret declare existence on most rounds of games without a positive-margin stable partition, where UCB-LCB-GS almost never does. Its existence error is largest under aversion to enemies, where margins are small, and these games contain no NO instance (Appendix M). The picture is unchanged under asymmetric utilities, which the analysis covers, and under Student-t_5 noise, which it does not (Appendix N). On synthetic instances under coalition-level feedback, UCB-LCB-GS never declares existence on a NO instance and has lower regret than the other sound methods in every configuration, but about twice the regret of the best optimism-based baseline and, in the default configuration, 6 to 30 times its regret under pairwise feedback (Appendix L).

6 Conclusions, Limitations, and Future Work

We studied online learning in coalition formation from a *group stability* perspective under *partial noisy* feedback, with two objectives: learning partitions that approach k-strong Nash stability and inferring whether stable partitions exist. UCB-LCB-GS tests stability with pessimistic estimates and explores through witness probing. Its expected number of declarations on instances without an (ε, k)-strongly Nash stable partition is bounded by a constant, and when some partition is stable with a positive certification margin, which is necessary, its existence error and regret grow logarithmically in T; purely optimistic exploration can fail even then. These guarantees extend to coalition-level bandit feedback and, under the stated assumptions, to asymmetric utilities. Empirically, UCB-LCB-GS combines low regret with reliable existence decisions under semi-bandit feedback, whereas optimism-based baselines detect existence sooner only by also declaring it where no stable partition exists; under coalition-level feedback, its regret is higher than theirs.

These guarantees require a positive certification margin (10) and exact solutions of the optimistic problem (9); for the local-search heuristic used in practice, Appendix G bounds the effect of optimization errors but not their frequency. Exact deviation checking, which the pessimistic test requires, grows combinatorially in n and k and limits the instance sizes we can evaluate. Our model assumes additively separable, stationary utilities and a coordinator with authority to assign agents (Appendix P). Future work includes extending the framework to other classes of hedonic games, evolving preferences, endogenous participation and other solution concepts, analyzing the stability-efficiency trade-off, and studying richer models of noisy partial feedback.

Acknowledgments and Disclosure of Funding

Saar Cohen and Michael Wooldridge were supported by an AI 2050 Senior Fellowship from the Schmidt Sciences Foundation.

References

[1] M. Aleksandrov and T. Walsh. Online fair division: A survey. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13557–13562, 2020.

[2] M. Aleksandrov, H. Aziz, S. Gaspers, and T. Walsh. Online fair division: Analysing a food bank problem. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2540–2546, 2015.

[3] A. Aloisio, M. Flammini, and C. Vinci. The impact of selfishness in hypergraph hedonic games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(2):1766–1773, 2020.

[4] I. Ashlagi, Y. Azar, M. Charikar, A. Chiplunkar, O. Geri, H. Kaplan, R. Makhijani, Y. Wang, and R. Wattenhofer. Min-Cost Bipartite Perfect Matching with Delays. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017)*, volume 81 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1:1–1:20, 2017.

[5] P. Auer and R. Ortner. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65, 2010.

[6] Y. Azar, A. Chiplunkar, and H. Kaplan. Polylogarithmic bounds on the competitiveness of min-cost perfect matching with delays. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1051–1061, 2017.

[7] H. Aziz and F. Brandl. Existence of stability in hedonic coalition formation games. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, volume 2, pages 763–770, 2012.

[8] H. Aziz and R. Savani. Hedonic games. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 15, page 356–376. Cambridge University Press, 2016.

[9] H. Aziz, F. Brandt, and H. G. Seedig. Stable partitions in additively separable hedonic games. In *AAMAS*, volume 11, pages 183–190, 2011.

[10] H. Aziz, F. Brandt, and H. G. Seedig. Computing desirable partitions in additively separable hedonic games. *Artificial Intelligence*, 195:316–334, 2013.

[11] H. Aziz, F. Brandl, F. Brandt, P. Harrenstein, M. Olsen, and D. Peters. Fractional hedonic games. *ACM Transactions on Economics and Computation (TEAC)*, 7(2):1–29, 2019.

[12] M.-F. Balcan, A. D. Procaccia, and Y. Zick. Learning cooperative games. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 475–481, 2015.

[13] C. Ballester. Np-completeness in hedonic games. *Games and Economic Behavior*, 49(1):1–30, 2004.

[14] A. Balliu, M. Flammini, G. Melideo, and D. Olivetti. On non-cooperativeness in social distance games. *Journal of Artificial Intelligence Research*, 66:625–653, 2019.

[15] S. Banerjee, H. Konishi, and T. Sönmez. Core in a simple coalition formation game. *Social Choice and Welfare*, 18(1):135–153, 2001.

[16] G. Benadè, A. M. Kazachkov, A. D. Procaccia, and C.-A. Psomas. How to make envy vanish over time. In *Proceedings of the 2018 ACM Conference on Economics and Computation (EC)*, pages 593–610, 2018.

[17] V. Bilò, A. Fanelli, M. Flammini, G. Monaco, and L. Moscardelli. Nash stable outcomes in fractional hedonic games: Existence, efficiency and computation. *Journal of Artificial Intelligence Research*, 62:315–371, 2018.

[18] N. Boehmer, M. Bullinger, and A. M. Kerkmann. Causes of stability in dynamic coalition formation. *ACM Transactions on Economics and Computation*, 13(2):1–45, 2025.

[19] A. Bogomolnaia and M. O. Jackson. The stability of hedonic coalition structures. *Games and Economic Behavior*, 38(2):201–230, 2002.

[20] M. Bullinger and R. Romen. Online coalition formation under random arrival or coalition dissolution. *ACM Transactions on Algorithms*, 22(1):1–43, 2025.

[21] M. Bullinger and R. Romen. Stability in online coalition formation. *Journal of Artificial Intelligence Research*, 82:2423–2452, 2025.

[22] M. Bullinger, R. Romen, and A. Schlenga. The power of matching for online fractional hedonic games. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2163–2194. SIAM, 2026.

[23] S. Cohen. Delayed assignments in online non-centroid clustering with stochastic arrivals. In *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1091–1100, 2026.

[24] S. Cohen. Offline learning of Nash stable coalition structures with possibly overlapping coalitions. In *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 3722–3730, 2026.

[25] S. Cohen and N. Agmon. Complexity of probabilistic inference in random dichotomous hedonic games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(5):5573–5581, 2023.

[26] S. Cohen and N. Agmon. Online coalitional skill formation. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems, AAMAS*, pages 494–503, 2023.

[27] S. Cohen and N. Agmon. Online learning of partitions in additively separable hedonic games. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 2722–2730, 2024.

[28] S. Cohen and N. Agmon. Online friends partitioning under uncertainty. In *ECAI 2024*, pages 3332–3339. IOS Press, 2024.

[29] S. Cohen and N. Agmon. Decentralized online learning by selfish agents in coalition formation. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3796–3804, 2025.

[30] S. Cohen and N. Agmon. Egalitarianism in online coalition formation. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pages 2475–2477, 2025.

[31] S. Cohen and N. Agmon. Online learning of fair coalition structures. In *Proceedings of the 28th European Conference on Artificial Intelligence, ECAI-25*, pages 4937–4944, 2025.

[32] S. Cohen and N. Agmon. Online learning of coalition structures by selfish agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(13):13709–13717, 2025.

[33] S. Cohen and N. Agmon. Stability-accuracy tradeoffs in decentralized online learning of coalition structures by selfish agents. *Journal of Artificial Intelligence Research*, 87, 2026.

[34] S. Cohen, N. Teh, P. W. Goldberg, and M. J. Wooldridge. Online fair division with budget constraints. arXiv preprint arXiv:2607.23310, 2026.

[35] S. Cohen, N. Teh, and M. Wooldridge. Online fair division under an oblivious adversary. arXiv preprint arXiv:2609.28333, 2026.

[36] G. Csáji, A. Gundert, J. Rothe, and I. Schlotter. Clustering via hedonic games: New concepts and algorithms. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[37] T. Demeulemeester and J. Peters. Relaxed core stability for hedonic games with size-dependent utilities. In *48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023)*, volume 272, pages 41–1. Schloss-Dagstuhl; Leibniz, 2023.

[38] D. Dimitrov, P. Borm, R. Hendrickx, and S. C. Sung. Simple priorities and core stability in hedonic games. *Social Choice and Welfare*, 26(2):421–433, 2006.

[39] J. H. Dréze and J. Greenberg. Hedonic coalitions: Optimality and stability. *Econometrica: Journal of the Econometric Society*, pages 987–1003, 1980.

[40] M. Eichhorn, S. Banerjee, and D. Kempe. Online team formation under different synergies. In *International Conference on Web and Internet Economics*, pages 78–95. Springer, 2022.

[41] E. Elkind and M. J. Wooldridge. Hedonic coalition nets. In *AAMAS (1)*, pages 417–424. Citeseer, 2009.

[42] Y. Emek, S. Kutten, and R. Wattenhofer. Online matching: haste makes waste! In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, pages 333–344, 2016.

[43] A. Fanelli, G. Monaco, and L. Moscardelli. Relaxed core stability in hedonic games. *Artificial Intelligence*, page 104394, 2025.

[44] J. Feldman, N. Korula, V. Mirrokni, S. Muthukrishnan, and M. Pál. Online ad assignment with free disposal. In *Internet and Network Economics (WINE 2009)*, volume 5929 of *Lecture Notes in Computer Science*, pages 374–385. Springer, 2009.

[45] S. Fioravanti, M. Flammini, B. Kodric, and G. Varricchio. Pac learning and stabilizing hedonic games: towards a unifying approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(5):5641–5648, 2023.

[46] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7 (2):179–188, 1936.

[47] R. A. Fisher. Iris. UCI Machine Learning Repository, `https://doi.org/10.24432/C56C76`, 1936. Licensed under CC BY 4.0.

[48] M. Flammini, G. Monaco, L. Moscardelli, M. Shalom, and S. Zaks. On the online coalition structure generation problem. *Journal of Artificial Intelligence Research*, 72:1215–1250, 2021.

[49] E. Friedman, C.-A. Psomas, and S. Vardi. Dynamic fair division with minimal disruptions. In *Proceedings of the 16th ACM Conference on Economics and Computation (EC)*, pages 697–713, 2015.

[50] M. Gairing and R. Savani. Computing stable outcomes in symmetric additively separable hedonic games. *Mathematics of Operations Research*, 44(3):1101–1121, 2019.

[51] B. Gamlath, M. Kapralov, A. Maggiori, O. Svensson, and D. Wajc. Online matching with general arrivals. In *Proceedings of the 60th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 26–37, 2019.

[52] P. M. Gleiser and L. Danon. Community structure in jazz. *Advances in Complex Systems*, 6(4): 565–573, 2003.

[53] J. He, A. D. Procaccia, A. Psomas, and D. Zeng. Achieving a fairer future by changing the past. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 343–349, 2019.

[54] Z. Huang, N. Kang, Z. G. Tang, X. Wu, Y. Zhang, and X. Zhu. How to match when all vertices arrive online. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 17–29, 2018.

[55] Z. Huang, B. Peng, Z. G. Tang, R. Tao, X. Wu, and Y. Zhang. Tight competitive ratios of classic matching algorithms in the fully online model. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2875–2886, 2019.

[56] A. Igarashi, J. Sliwinski, and Y. Zick. Forming probably stable communities with limited interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2053–2060, 2019.

[57] M. Jagadeesan, A. Wei, Y. Wang, M. Jordan, and J. Steinhardt. Learning equilibria in matching markets from bandit feedback. *Advances in Neural Information Processing Systems*, 34:3323–3335, 2021.

[58] D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis. How easy is local search? *Journal of computer and system sciences*, 37(1):79–100, 1988.

[59] M. Karakaya. Hedonic coalition formation games: A new stability notion. *Mathematical Social Sciences*, 61(3):157–165, 2011.

[60] R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 352–358, 1990.

[61] J. Kunegis. KONECT: The Koblenz network collection. In *Proceedings of the 22nd International Conference on World Wide Web (Companion Volume)*, pages 1343–1350, 2013.

[62] T. Lattimore. Regret analysis of the finite-horizon Gittins index strategy for multi-armed bandits. In *Conference on learning theory*, pages 1214–1245. PMLR, 2016.

[63] T. Lattimore and C. Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[64] B. Li, W. Li, and Y. Li. Dynamic fair division problem with general valuations. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 375–381, 2018.

[65] L. T. Liu, H. Mania, and M. Jordan. Competing bandits in matching markets. In *International Conference on Artificial Intelligence and Statistics*, pages 1618–1628. PMLR, 2020.

[66] C. Maheshwari, S. Sastry, and E. Mazumdar. Decentralized, communication-and coordination-free learning in structured matching markets. *Advances in Neural Information Processing Systems*, 35:15081–15092, 2022.

[67] P. Ménard and A. Garivier. A minimax and asymptotically optimal algorithm for stochastic bandits. In *International Conference on Algorithmic Learning Theory*, pages 223–237. PMLR, 2017.

[68] P. Rashidinejad, B. Zhu, C. Ma, J. Jiao, and S. Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 34:11702–11716, 2021.

[69] M. Rodet and R. Gaudel. Unimodal mono-partite matching in a bandit setting. In *Complex Feedback in Online Learning Workshop at the 39th International Conference on Machine Learning*, 2022.

[70] A. Sankararaman, S. Basu, and K. A. Sankararaman. Dominate or delete: Decentralized competing bandits in serial dictatorship. In *International Conference on Artificial Intelligence and Statistics*, pages 1252–1260. PMLR, 2021.

[71] F. Sentenac, J. Yi, C. Calauzenes, V. Perchet, and M. Vojnovic. Pure exploration and regret minimization in matching bandits. In *International Conference on Machine Learning*, pages 9434–9442. PMLR, 2021.

[72] J. Sliwinski and Y. Zick. Learning hedonic games. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2730–2736, 2017.

[73] M. Springer, M. Hajiaghayi, D. Panigrahi, and M. Khani. Online algorithms for the Santa Claus problem. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 30732–30743, 2022.

[74] S.-C. Sung and D. Dimitrov. Computational complexity in additive hedonic games. *European Journal of Operational Research*, 203(3):635–639, 2010.

[75] N. Teh, S. Cohen, K. F. Celine, B. Yu, and M. J. Wooldridge. Online house allocation with subsidy. arXiv preprint arXiv:2609.29691, 2026.

[76] Y. Wang and S. C.-w. Wong. Two-sided online bipartite matching and vertex cover: Beating the greedy algorithm. In *Automata, Languages, and Programming (ICALP 2015)*, volume 9134 of *Lecture Notes in Computer Science*, pages 1070–1081. Springer, 2015.

[77] G. J. Woeginger. Core stability in hedonic coalition formation. In *SOFSEM 2013: Theory and Practice of Computer Science*, pages 33–50. Springer, 2013.

[78] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473, 1977.

[79] Y. Zhang, S. Wang, and Z. Fang. Matching in multi-arm bandit with collision. *Advances in Neural Information Processing Systems*, 35:9552–9563, 2022.

A Broader Impacts

This work is primarily theoretical and studies online learning for group-stable coalition formation under partial and noisy feedback. As such, it does not directly introduce deployable systems, datasets, or application-specific models.

Potential positive impacts include improved methods for forming group-stable coalitions in settings such as team formation and collaborative platforms, where group stability can reduce conflict and improve user satisfaction. Potential risks arise if such methods are applied in socio-technical systems involving human participants. For example, optimizing for group stability alone may overlook fairness, diversity, or individual preferences, potentially reinforcing undesirable group structures or exclusion. Additionally, in settings with strategic agents, learned models of preferences could be misused to manipulate coalition formation outcomes. Our model also assumes a coordinator who may assign participants to groups, including exploratory assignments that are not expected to be stable. In workplaces and similar settings this raises questions of consent and of the cost that exploration imposes on participants. Practical deployments would call for attention to fairness, transparency, and robustness, which we leave for future work.

B Additional Related Work

This appendix expands the discussion of related work in Section 1. We discuss offline hedonic games, online learning in coalition formation, online coalition formation, online matching and learning in matching markets, and online fair division. For each line of work, we compare the research question, the information available to the decision maker, the solution concept and the performance criterion with ours.

Offline Hedonic Games. Hedonic games have been introduced by Dréze and Greenberg [39], and later extended to the study of various solution concepts like stability, fairness, and optimality (see, e.g., [8, 77]). One major concern is designing computationally manageable classes of hedonic games, which led to an abundance of game representations. Some are *ordinal* and can *fully* express any preference over coalitions [13, 41], but may require exponential space. In contrast, *cardinal* hedonic games, based on weighted graphs [11, 19], are *not* fully expressive, yet only require *polynomial* space for reasonable weights. Our work focuses on *additively separable hedonic games* (ASHGs) with *symmetric* preferences [19], where a vast body of work evaluates the system in terms of Nash stability [9, 15, 13], a *single-agent* stability notion where no agent can improve her utility by unilaterally deviating. Bogomolnaia and Jackson [19] proved that Nash stable partitions may not exist in general ASHGs, while Sung and Dimitrov [74] showed that checking if an instance admits such partition

is NP-complete in the strong sense. Yet, for *symmetric* preferences, the existence of a Nash stable outcome is guaranteed by potential function argument [19], but computing such partitions is PLS-complete [50], where the complexity class PLS (Polynomial Local Search) consists of local search problems with polynomially verifiable local optimality [58].

In contrast, we focus on *group stability* notions, which have also been extensively studied in offline ASHGs. A large body of work studies *core stability* [74], which requires that no group of agents can form a new coalition that every member strictly prefers to her current coalition. *Strict core stability* [74] is more demanding: no group of agents can form a new coalition that every member weakly prefers and at least one member strictly prefers to her current coalition. In general, a core-stable or strictly core-stable partition is not guaranteed to exist [10]. Sung and Dimitrov [74] proved that checking whether such a core-stable or strictly core-stable partition exists is NP-hard for general ASHGs. This result was later strengthened by Aziz et al. [10], who proved that deciding whether a (strictly) core stable partition exists is strongly NP-hard even in symmetric ASHGs, and that computing one is NP-hard.

As specified in Section 1, in this work we focus on k*-strong Nash stability* [17], which rules out joint deviations by groups of size at most $k \geq 1$, possibly to different coalitions. For $k = n$, the notion of n-strong Nash stability (or *strong Nash stability* for short) is stronger than Nash stability and core stability combined. Strong Nash stability has mainly been studied in restricted subclasses of ASHGs [7, 59]. In general additively separable hedonic games, however, we prove in Appendix C that existence is not guaranteed, and the notion has received limited treatment. Moreover, k-strong Nash stability for any $1 \leq k \leq n$ has been previously examined in fractional hedonic games (FHGs) [17], where an agent's utility is the average of values she assigns to other members of her coalition. To the best of our knowledge, k-strong Nash stability for any $1 \leq k \leq n$ has not been studied in ASHGs.

We further consider the weaker version of k-size core stability [43, 37], where no group of at most k agents can deviate by forming a single new coalition. This notion has been originally introduced by Fanelli et al. [43] in the context of FHGs, where they showed that a 2-size core stable partition always exist. Demeulemeester and Peters [37] also studied k-size core stability in a more general framework that subsumes ASHGs and FHGs, and they study its relation to q-improvement core stability, which requires that no blocking coalition exists in which every agent improves by a factor of more than q. In contrast, we allow more general *group deviations* in which agents may simultaneously move to different coalitions, as captured by k-strong Nash stability.

However, as discussed in Section 1, the above works consider *offline* environments with full knowledge of preferences, while we consider *online* settings where agents' preferences are initially unknown and must be learned from *partial and noisy feedback*. Research on PAC learnability in hedonic games attempts to tackle this issue [72, 45, 12], aiming to efficiently infer preferences from a limited, fixed number of offline samples to recover core stable partitions. They also assume exact knowledge of preferences *before* making decisions, limiting practicality as agents often need time to learn their own preferences from social interactions, as in our project teams example. In particular, the PAC learning approach is unfit to our *dynamic* setting due to its *static* nature, requiring a fixed set of preferences that is known in advance. We consider situations where the agents' preferences are learned *dynamically* through repeated interactions, so that the coalitions formed improve as the preferences are learned. Learning from a fixed dataset has also been studied for Nash stability: Cohen [24] learn approximately Nash stable coalition structures, with possibly overlapping coalitions, from offline data under semi-bandit or bandit feedback, and bound the number of samples needed under coverage assumptions on the data. There, the dataset is given in advance and the learner does not choose which coalitions to observe, whereas our learner forms coalitions and collects feedback in every round. Uncertainty about preferences has also been studied from a computational perspective: Cohen and Agmon [25] analyze the complexity of computing the probability that a coalition or a partition is optimal or stable in random dichotomous hedonic games. This concerns inference under a known distribution over games rather than learning from feedback.

Online Learning in Coalition Formation. A line of work by Cohen and Agmon studies repeated coalition formation in which the agents' preferences are initially unknown and are learned from feedback. These papers differ from one another, and from ours, in who learns, what is observed after each round, and how the formed partitions are evaluated. In the model of Cohen and Agmon [27], a central learner partitions a possibly different subset of a fixed pool of agents in each round, after

which the agents' additively separable preferences, which may change over time, are revealed; the learner minimizes static or dynamic regret with respect to social cost. The learner thus observes the preferences completely after each round, and its objective is a welfare criterion rather than stability. Cohen and Agmon [32] and Cohen and Agmon [29] take the perspective of selfish agents in a fixed agent set and evaluate outcomes by Nash stability. In the former, the agents learn from noiseless semi-bandit or bandit utility feedback and pool their observations through a shared data platform. In the latter, each agent learns independently from its own noiseless bandit feedback, without coordination or information sharing. Both minimize Nash regret, which yields approximately Nash stable outcomes. In the same decentralized setting, Cohen and Agmon [33] study a tradeoff between stability and accuracy. When each agent uses a single utility observation per round, Nash regret and sample complexity are optimal up to logarithmic factors, but the error of the agents' estimates at individual steps is not controlled; using several samples per round controls this error, at the cost of weaker Nash regret and sample complexity bounds. Cohen and Agmon [31] further study the online learning of coalition structures from partial noisy feedback with a fairness objective: they minimize envy regret with respect to partitions that minimize the maximum envy, under constraints on the number and sizes of coalitions.

Our work shares with these papers the repeated formation of coalitions and the learning of cardinal utilities from interaction feedback, but it poses a different learning problem. Nash stability requires that no single agent can improve her utility by moving unilaterally to another coalition, and Nash regret measures how far the formed partitions are from satisfying this requirement. Envy-based objectives are fairness criteria: they compare the outcomes of different agents, rather than asking whether some agent can profit by deviating. The notion we study, k-strong Nash stability, is a stability notion for groups. It rules out coordinated deviations in which up to k agents move simultaneously, possibly to different coalitions, and all of them gain; for $k = 1$ it coincides with Nash stability. Nash stable partitions always exist for symmetric additively separable utilities, but k-strongly Nash stable partitions need not exist for $k \geq 3$. Our learner must therefore also determine whether a stable partition exists, and we evaluate it both by its (ε, k)-strong Nash regret and by its errors in predicting existence. Our learner is centralized, and we analyze it under pairwise semi-bandit feedback and under per-agent coalition-level bandit feedback; in the latter, the pairwise utilities on which group stability depends must be recovered from deliberately formed two-agent coalitions.

Online Coalition Formation. In other models of online coalition formation, agents arrive sequentially, and each arriving agent must be assigned to a coalition upon arrival, without the possibility of revising this assignment later [48, 20, 22, 30]; Bullinger and Romen [21] study stability notions in this model. Cohen and Agmon [28] consider arriving agents whose friendships are uncertain, and use predictions to maximize social welfare, and Cohen and Agmon [26] assign arriving agents with skills to task coalitions. Assignment problems with arrivals also arise in online clustering, where points arriving in a metric space may be assigned to clusters later, at a delay cost [23]. These models are evaluated by competitive analysis, and their uncertainty concerns which agents arrive and when. In our model all agents are present from the start, coalitions are formed repeatedly, and the uncertainty concerns the agents' utilities.

Online Matching and Learning in Matching Markets. Matchings can be viewed as a restricted class of hedonic games in which coalitions contain at most two agents. Two lines of work on matchings are related to ours, and they differ in their source of uncertainty and in their performance criterion. The first is online matching, in which the input is revealed over time and algorithms are evaluated by their competitive ratio against an optimal offline solution. In the classical model of Karp et al. [60], the vertices on one side of an unweighted bipartite graph are present from the start, while those on the other side arrive online and must be matched immediately and irrevocably; the Ranking algorithm attains the optimal competitive ratio $1 - \frac{1}{e}$. Feldman et al. [44] consider edge-weighted online bipartite matching with free disposal, in which an offline vertex may drop an earlier match in favour of a heavier one, and obtain a $(1 - \frac{1}{e})$-competitive algorithm when the capacities of the offline vertices are large. Other models let all vertices arrive online. When decisions remain immediate, Wang and Wong [76] give a 0.526-competitive algorithm for fractional matching in general graphs and show that no algorithm is better than 0.625-competitive, and Gamlath et al. [51] show that randomized integral algorithms can beat the competitive ratio $\frac{1}{2}$ of the greedy algorithm. In the fully online model of Huang et al. [54], the decision for a vertex may be delayed until its deadline, after all its neighbours have arrived; Ranking is 0.5211-competitive in this model, and no algorithm is

better than 0.6317-competitive, even on bipartite graphs [54, 55]. Decisions may also be delayed at an explicit cost, as in online matching with delays [42, 6, 4]. In all these models, the parts of the instance that have arrived are known exactly, and the difficulty lies in the unknown future input. Our setting has no arrivals: the set of agents is fixed and present from the start, the uncertainty lies in their pairwise utilities, which are learned from repeated noisy feedback, and the learner is evaluated by its regret and existence errors over many rounds rather than by a competitive ratio. Coalitions may moreover contain any number of agents.

The second line studies online learning of matchings under uncertain preferences, which is closer to our setting in that matchings are formed repeatedly and preferences are learned from bandit feedback. Most of this work considers two-sided markets, in which agents on one side learn their preferences over the other side from stochastic rewards, either through a central platform [65, 57] or in a decentralized manner [65, 70, 66, 79], and performance is measured by stable regret or by the distance from an equilibrium. Jagadeesan et al. [57] also allow monetary transfers, which are unavailable in our setting. Monopartite matchings have been learned under structural assumptions, such as rank-one or unimodal utilities [71, 69] or binary latent types [40]. In contrast, we consider coalitions of arbitrary size, partial noisy feedback and general additively separable utilities, and the stability notion we target constrains deviations by groups of agents rather than by blocking pairs.

Online Fair Division. Fair division and coalition formation both partition a set, but they partition different objects. In fair division, goods or resources are allocated to agents who value them, and allocations are evaluated by fairness criteria such as envy-freeness or proportionality. In coalition formation, the agents themselves are partitioned, each agent's utility depends on the other members of her coalition, and we evaluate partitions by their stability against deviations. Online fair division studies allocation over time [1]. Goods may arrive and be allocated immediately to a fixed set of agents, as in the food bank model of Aleksandrov et al. [2], the analysis of envy over time by Benadè et al. [16], the model of He et al. [53], which allows reallocations, and online algorithms for the max-min objective [73]. Alternatively, agents may arrive and depart while a divisible resource is shared [49, 64]. In these models, the values of the goods or agents that have arrived are known when decisions are made, and the uncertainty concerns the future input.

Recent work examines which fairness guarantees remain achievable under such uncertainty and under additional constraints. When indivisible goods arrive one at a time and are allocated immediately and irrevocably against an oblivious adversary, high-probability approximations to proportionality and its relaxations are achievable, whereas envy-based criteria such as EF1 and EFX can be guaranteed only with small probability [35]. When the agents face budget constraints, and goods that no agent can feasibly receive go to a charity, no deterministic online algorithm guarantees a fixed approximation to envy-freeness relative to budget-feasible bundles without further structure; structural conditions, resource augmentation and predictions of the types of future goods restore such guarantees [34]. In online house allocation, houses arrive over time, each agent holds exactly one house, earlier assignments may be revised, and envy-freeness is attained through monetary subsidies; learning-augmented algorithms use predictions to minimize the total subsidy [75]. In all of these models, the objects being allocated are goods or houses, fairness is measured by proportionality or envy-freeness, and predictions, where used, are predictions about the input. Our learner instead repeatedly partitions a fixed set of agents, whose pairwise utilities for one another are unknown and observed only through noisy feedback, and it aims at stability against coordinated deviations by groups of agents. Envy-based objectives in repeated coalition formation are discussed above [31].

C Existence and Complexity of Group-Stable Partitions

This appendix collects two facts about group stability with known utilities that motivate the learning problem: a stable partition need not exist (Appendix C.1), and deciding existence or computing a stable partition can be intractable (Appendix C.2).

C.1 Non-Existence for $k \geq 3$

Observation 1 shows that the existence question studied in this paper is not vacuous: already for symmetric ASHGs and groups of three agents, a stable partition may fail to exist. The instance is taken from Aziz et al. [10], who show that it has no core stable partition. We restate the observation

and verify that every blocking coalition in this instance has at most three agents, which is what the observation requires.

Observation 1. *There exists an ASHG with symmetric preferences that admits no k-size core stable partition for any $k \geq 3$, and thus admits no (ε, k)-strongly Nash stable partition for any $k \geq 3, \varepsilon \geq 0$.*

Proof. The symmetric ASHG given in Example 1 of Aziz et al. [10] consists of an agent set $N = \{1, 2, 3, 4, 5, 6\}$, whose mutual utilities are

$$\begin{aligned} v(1,2) = v(3,4) = v(5,6) &= 6, \\ v(1,6) = v(2,3) = v(4,5) &= 5, \\ v(1,3) = v(1,5) = v(3,5) &= 4, \\ v(1,4) = v(2,5) = v(3,6) &= -33, \text{ and} \\ v(2,4) = v(2,6) = v(4,6) &= -33 \end{aligned}$$

By Aziz et al. [10], no partition is core stable for this game. In this instance, any pair of agents i, j with $v_i(j) = -33$ cannot belong to the same coalition in a (potentially) stable outcome. Moreover, every player strictly prefers being in some non-singleton coalition to being alone. Hence, the set of individually rational coalitions[2] is exactly

$$\begin{gathered} \{1,2\},\{1,3\},\{1,5\},\{1,6\},\{1,2,3\},\{1,3,5\},\{1,5,6\}, \\ \{2,3\},\{3,4\},\{3,4,5\},\{3,5\},\{4,5\},\{5,6\}. \end{gathered}$$

In particular, note that every individually rational coalition has size at most 3. Observe that any blocking coalition must give each of its members strictly higher utility than in the current partition, and hence must be individually rational. Therefore, every blocking coalition has size at most 3.

Since the game admits no core stable partition and each partition is blocked by some coalition of size at most 3, the instance admits no 3-size core stable partition. Consequently, the game admits no k-size core stable partition for any $k \geq 3$. Since every k-strongly Nash stable partition is k-size core stable, and every (ε, k)-strongly Nash stable partition is k-strongly Nash stable, it admits no (ε, k)-strongly Nash stable partition for any $k \geq 3$ and $\varepsilon \geq 0$. □

C.2 Computational Complexity

Even when the utilities are known, deciding whether a stable partition exists and computing one may be computationally intractable. An n-strongly Nash stable partition satisfies the standard notion of core stability, where no group of agents can improve by forming a new coalition by itself. For core stability, deciding whether a stable partition exists is NP-hard in the strong sense even in symmetric ASHGs [10, Theorem 2], which strengthens the corresponding result of Sung and Dimitrov [74] for general ASHGs, and computing a core stable partition is NP-hard as well [10]. Since the implication runs only from n-strong Nash stability to core stability, these results do not by themselves determine the complexity of the corresponding problems for n-strong Nash stability. Further, by Bilò et al. [17], 1-strong Nash stability coincides with the standard notion of Nash stability, where no agent can improve her utility by unilaterally deviating. ASHGs with *symmetric* preferences admit a Nash stable partition due to a potential function argument [19], but computing such partitions is PLS-complete [50]. These results concern the offline problem with known utilities; Appendix H discusses their consequences for the optimization problems solved by our learner.

D Proofs for Section 3

This appendix establishes the three guarantees of Section 3. The proofs use a common distinction between rounds in which all confidence intervals in use contain the true utilities and rounds in which at least one interval fails. In the former case, the pessimistic and optimistic estimates bound the true group deviation gap from below and above, respectively. Consequently, any declaration of existence is correct, and the partition formed incurs no regret. In the latter case, we bound the expected number

[2] A coalition $C \subseteq N$ is *individually rational* if each player does as well as by being alone, i.e., $v_i(C) \geq 0$ for any agent $i \in C$.

of rounds with a failed confidence interval, taking into account that the learner chooses which pairs to observe. This bound is established once, in Lemma D.1, and is sufficient for the proof of the non-existence error bound (Appendix D.1). For the existence-error bound, however, soundness alone is insufficient: we must also show that, when a stable partition exists, the learner cannot continue indefinitely without declaring existence. We establish this in Appendix D.2 by showing that every round without a declaration observes a pair whose confidence interval is still sufficiently wide. Finally, Appendix D.3 uses the same counting argument to establish the regret bound.

Notation. Throughout, at round $t \geq 2$, the learner uses the estimates $\overline{\mathbf{v}}^{t-1}$ and $\underline{\mathbf{v}}^{t-1}$ computed from (5), with confidence radius

$$b_{i,j}^{t-1} = \sqrt{\frac{2\sigma^2 \beta \log t}{T_{i,j}^{t-1}}}.$$

Let

$$\mathcal{G}_t := \left\{ |\mu_{i,j}^{t-1} - v_{i,j}| \leq b_{i,j}^{t-1} \text{ for every pair } \{i,j\} \text{ with } T_{i,j}^{t-1} \geq 1 \right\}$$

denote the event that all confidence intervals in use at round t contain the corresponding true utilities. Let $\mathcal{G}_t^<$ denote the event defined by replacing the non-strict inequalities in $\mathcal{G}_t$ with strict ones. We write

$$\varphi_T := \sum_{t=2}^{T} \Pr[\mathcal{G}_t^c]$$

for the expected number of rounds at which at least one confidence interval fails.

For a partition π and matrices $\mathbf{x}, \mathbf{y}$, let $\widetilde{\mathrm{Gap}}_k(\pi, \mathbf{x}, \mathbf{y})$ denote the minimum of $\widetilde{\mathrm{Gap}}_C(\pi, \pi', \mathbf{x}, \mathbf{y})$ over all nonempty $C \subseteq N$ with $|C| \leq k$ and all $\pi' \in \Lambda_C(\pi)$. Thus,

$$G_t^-(\pi) = \widetilde{\mathrm{Gap}}_k\big(\pi, \underline{\mathbf{v}}^{t-1}, \overline{\mathbf{v}}^{t-1}\big), \qquad G_t^+(\pi) = \widetilde{\mathrm{Gap}}_k\big(\pi, \overline{\mathbf{v}}^{t-1}, \underline{\mathbf{v}}^{t-1}\big).$$

D.1 Proof of Theorem 3.1

A declaration $d_t = 1$ requires that the pessimistic estimate of at least one candidate partition be at least ε. Whenever all confidence intervals hold, this estimate does not exceed the corresponding true group deviation gap. Therefore, a declaration of existence is impossible on an instance that admits no (ε, k)-strongly Nash stable partition, unless some confidence interval fails. The proof consequently has two parts: first, we bound the probability that at least one confidence interval fails (Lemma D.1); second, we relate the pessimistic estimate to the true group deviation gap on the event that all confidence intervals hold (Lemma D.2).

Concentration under adaptive sampling. For a fixed number m of observations, the average of m independent σ^2-subgaussian noises deviates from its mean by more than $\sigma\sqrt{2\beta \log t/m}$ with probability at most $2t^{-\beta}$. Here, however, the number $T_{i,j}^{t-1}$ of observations of a pair is not fixed in advance. It depends on the partitions formed by the learner, which in turn depend on earlier observations, including observations of the same pair. For example, a pair whose initial observations are unfavourable may subsequently be observed less often. Thus, we cannot condition on the realized observation count and directly apply a fixed-sample concentration bound. Instead, we derive a bound that holds simultaneously for every possible number of observations of each pair, using a supermartingale, and then take a union bound over the possible observation counts.

Lemma D.1. *For every $t \geq 2$,*

$$\Pr[\mathcal{G}_t^c] \leq \Pr[(\mathcal{G}_t^<)^c] \leq n(n-1)t^{1-\beta}.$$

Consequently, for $\beta > 2$ and every T,

$$\varphi_T \leq \frac{n(n-1)}{\beta - 2}.$$

Proof. Order all observations by round and, within each round, by the index of the observed pair. Denote their resulting sequence by $p = 1, 2, \ldots$. Let $\mathcal{H}_p$ be the σ-field generated by everything that occurs before observation p, including the learner's internal randomness and its decision about

which pairs to observe in the current round. By the noise assumptions of Section 2, the noise ξ_p of observation p is independent of $\mathcal{H}_p$, has mean zero, and satisfies

$$\mathbb{E}[e^{\lambda \xi_p} \mid \mathcal{H}_p] \le e^{\lambda^2 \sigma^2 / 2} \qquad \text{for all } \lambda \in \mathbb{R}.$$

Fix a pair e and $\lambda \in \mathbb{R}$. Let

$$o_p := \mathbb{1}\{\text{observation } p \text{ concerns pair } e\}.$$

By construction, o_p is $\mathcal{H}_p$-measurable. Define

$$X_p := \exp\left(\lambda \sum_{p' \le p} o_{p'} \xi_{p'} - \frac{\lambda^2 \sigma^2}{2} \sum_{p' \le p} o_{p'}\right), \qquad X_0 := 1.$$

Since o_p is known before observation p is received, we have

$$\begin{aligned}\mathbb{E}[X_p \mid \mathcal{H}_p] &= X_{p-1}\left(o_p\, \mathbb{E}[e^{\lambda \xi_p - \lambda^2 \sigma^2/2} \mid \mathcal{H}_p] + 1 - o_p\right) \\ &\le X_{p-1}.\end{aligned}$$

Thus, (X_p) is a non-negative supermartingale with respect to the filtration generated by the successive observations and their preceding decisions.

Let τ_m denote the position of the m-th observation of pair e. This is a stopping time. By optional stopping for non-negative supermartingales,

$$\mathbb{E}[X_{\tau_m} \mathbb{1}\{\tau_m < \infty\}] \le 1.$$

On the event $\{\tau_m < \infty\}$, we have

$$X_{\tau_m} = \exp\left(\lambda S_m - \frac{m \lambda^2 \sigma^2}{2}\right),$$

where S_m is the sum of the first m noise observations of pair e. Applying Markov's inequality with $\lambda = x/\sigma^2$, and then with $\lambda = -x/\sigma^2$, gives

$$\Pr[|S_m| \ge mx,\ \tau_m < \infty] \le 2 \exp\left(-\frac{mx^2}{2\sigma^2}\right).$$

Each pair is observed at most once per round. Therefore, on the event $\{T_e^{t-1} = m\}$, we have $1 \le m \le t-1$ and

$$\mu_e^{t-1} - v_e = \frac{S_m}{m}.$$

Choose

$$x = \sigma \sqrt{\frac{2\beta \log t}{m}},$$

which equals the confidence radius b_e^{t-1} when the pair has been observed m times. Taking a union bound over the $\binom{n}{2}$ pairs and the $t-1$ possible values of their observation counts yields

$$\begin{aligned}\Pr[(\mathcal{G}_t^<)^c] &\le \sum_e \sum_{m=1}^{t-1} \Pr[|S_m| \ge mx,\ \tau_m < \infty] \\ &\le \binom{n}{2}(t-1) \cdot 2t^{-\beta} \\ &\le n(n-1)t^{1-\beta}.\end{aligned}$$

Finally, since $\mathcal{G}_t^c \subseteq (\mathcal{G}_t^<)^c$, the same upper bound applies to $\Pr[\mathcal{G}_t^c]$. Summing over rounds and using

$$\sum_{t \ge 2} t^{1-\beta} \le \int_1^\infty s^{1-\beta}\, ds = \frac{1}{\beta - 2}$$

completes the proof. □

The union bound over the possible observation counts introduces a factor of $t-1$, which the summation over rounds absorbs when $\beta > 2$. This is where the condition $\beta > 2$ enters. Under a different feedback model, the same argument applies, with the bound adjusted according to the maximum number of observations of a pair per round (Appendix F).

Estimates and the true gap. The next lemma establishes how the pessimistic and optimistic estimates compare with the true group deviation gap on $\mathcal{G}_t$. Part (a) gives the sandwich inequality needed for the soundness guarantee. Part (b) provides its strict version, which is used in Appendix E to establish the necessity of a positive certification margin. Part (c) bounds the difference between the true gap and its pessimistic estimate; this bound is used in the analyses of Appendices G and F.

Lemma D.2. *Let $t \geq 2$ and suppose that every pair has been observed. Let*

$$\bar{b}_t := \max_{i,j} b_{i,j}^{t-1}.$$

Then, for every partition π, the following hold:

(a) On $\mathcal{G}_t$,

$$G_t^-(\pi) \leq \mathrm{Gap}_k(\pi, \mathbf{v}) \leq G_t^+(\pi).$$

(b) On $\mathcal{G}_t^<$, the first inequality in (a) is strict:

$$G_t^-(\pi) < \mathrm{Gap}_k(\pi, \mathbf{v}).$$

(c) On $\mathcal{G}_t$,

$$\mathrm{Gap}_k(\pi, \mathbf{v}) - G_t^-(\pi) \leq \kappa_n \bar{b}_t.$$

Proof. (a) On $\mathcal{G}_t$, every pairwise estimate satisfies

$$\underline{v}_{i,j}^{t-1} \leq v_{i,j} \leq \overline{v}_{i,j}^{t-1}.$$

Consequently, for every $\pi' \in \Lambda_C(\pi)$ and $i \in C$,

$$\begin{aligned} u_i(\pi(i), \underline{\mathbf{v}}^{t-1}) - u_i(\pi'(i), \overline{\mathbf{v}}^{t-1}) &\leq u_i(\pi(i), \mathbf{v}) - u_i(\pi'(i), \mathbf{v}) \\ &\leq u_i(\pi(i), \overline{\mathbf{v}}^{t-1}) - u_i(\pi'(i), \underline{\mathbf{v}}^{t-1}). \end{aligned}$$

Taking the maximum over $i \in C$ and then the minimum over all admissible pairs (C, π') preserves these inequalities, giving the result.

(b) On $\mathcal{G}_t^<$, each confidence bound is strict. Since $\pi'(i) \neq \pi(i)$, at least one of the two coalitions contains another agent besides i. If the current coalition contains another agent, replacing its true pairwise utilities by their lower confidence bounds strictly decreases the first utility term. If the destination coalition contains another agent, replacing its true pairwise utilities by their upper confidence bounds strictly increases the second utility term, which is subtracted. Thus, the leftmost expression in the preceding display is strictly smaller than the corresponding true gap for every $i \in C$. Because the family of admissible deviations is finite, taking the maximum over deviators and then the minimum over deviations preserves strictness.

(c) On $\mathcal{G}_t$, each pairwise entry satisfies

$$v_{i,j} - \underline{v}_{i,j}^{t-1} \leq 2b_{i,j}^{t-1}, \qquad \overline{v}_{i,j}^{t-1} - v_{i,j} \leq 2b_{i,j}^{t-1}.$$

For a fixed deviator i, the difference between the true group-deviation term and its pessimistic estimate therefore involves at most

$$(|\pi(i)| - 1) + (|\pi'(i)| - 1) \leq 2n - 3$$

pairwise entries. Indeed, the two coalitions are distinct, both contain i, and at most one of them can be the grand coalition. Each entry contributes at most $2\bar{b}_t$, so the difference is at most $2(2n-3)\bar{b}_t = \kappa_n \bar{b}_t$. Finally, the inequalities

$$\max_i a_i - \max_i c_i \leq \max_i (a_i - c_i), \qquad \min_x f(x) - \min_x g(x) \leq \max_x (f(x) - g(x))$$

allow this bound to pass through the maximum over deviators and the minimum over admissible deviations. This establishes (c). □

We can now establish the soundness guarantee.

Theorem 3.1 (Non-Existence Error Bound)**.** *For any input with $\mathtt{SNS}_{\varepsilon,k}(\mathbf{v}) = \emptyset$, UCB-LCB-GS satisfies $\sum_{t=1}^{T} \Pr[d_t = 1] \leq \frac{n(n-1)}{\beta-2}$. In particular, $\mathcal{E}_T^{\exists} \leq \frac{n(n-1)}{\beta-2}$. The bound holds for any method of computing the candidate partitions, provided the pessimistic test is evaluated exactly.*

Proof. Round 1 never declares existence. Consider a round $t \geq 2$ at which $d_t = 1$. The learner must have formed a partition π^t satisfying $G_t^-(\pi^t) \geq \varepsilon$, whether π^t is the pessimistic or the optimistic candidate. Since every pair was observed in round 1, on $\mathcal{G}_t$, Lemma D.2(a) implies

$$\mathrm{Gap}_k(\pi^t, \mathbf{v}) \geq \varepsilon.$$

This contradicts $\mathtt{SNS}_{\varepsilon,k}(\mathbf{v}) = \emptyset$. Therefore,

$$\{d_t = 1\} \subseteq \mathcal{G}_t^c.$$

It follows that

$$\sum_{t=1}^{T} \Pr[d_t = 1] \leq \varphi_T \leq \frac{n(n-1)}{\beta - 2},$$

where the last inequality follows from Lemma D.1. Finally, $\mathtt{SNS}_{0,k}(\mathbf{v}) = \emptyset$ implies $\mathtt{SNS}_{\varepsilon,k}(\mathbf{v}) = \emptyset$, so the same bound applies to $\mathcal{E}_T^{\exists}$. The argument does not depend on how the candidate partitions are computed. □

D.2 Proof of Theorem 3.2

Soundness alone does not guarantee that the learner will declare existence when a stable partition is available: a learner that never declares existence is sound, but incurs an existence error in every round. To prove the existence-error bound, we therefore show that rounds without a declaration cannot occur too often when a stable partition with a positive certification margin exists. The argument relies on two conditions: a positive certification margin and exact optimization of the optimistic problem.

The first condition is the positive certification margin $\Delta_\star$ defined in (10). Let $\pi^\star$ be a partition satisfying

$$\mathrm{Gap}_k(\pi^\star, \mathbf{v}) = \varepsilon + \Delta_\star.$$

On $\mathcal{G}_t$, Lemma D.2(a) gives

$$G_t^+(\pi^\star) \geq \varepsilon + \Delta_\star.$$

If the optimistic problem is solved exactly, its maximizing candidate $\overline{\pi}^t$ therefore satisfies

$$G_t^+(\overline{\pi}^t) \geq G_t^+(\pi^\star) \geq \varepsilon + \Delta_\star.$$

On a round without a declaration, however, $\overline{\pi}^t$ also fails the pessimistic test, so

$$G_t^-(\overline{\pi}^t) < \varepsilon.$$

The difference between the optimistic and pessimistic estimates of this candidate must consequently exceed $\Delta_\star$.

The witness-gap inequality below relates this difference to the confidence radius of a single pair: the witness pair selected by the learner in that round. Its radius must exceed $\Delta_\star/\kappa_n$, which means that the pair has been observed fewer than

$$\frac{2\kappa_n^2 \sigma^2 \beta \log T}{\Delta_\star^2}$$

times. Since each round without a declaration adds an observation of its witness pair, each pair can be selected as a witness only logarithmically many times. Summing over the $\binom{n}{2}$ pairs gives the required bound on the number of rounds without a declaration.

Lemma D.3 (Witness-gap inequality)**.** *At every round at which UCB-LCB-GS selects a witness pair* e_t*,*

$$G_t^+(\overline{\pi}^t) - G_t^-(\overline{\pi}^t) \leq \kappa_n b_{e_t}^{t-1},$$

where $\kappa_n = 2(2n-3)$.

Proof. For each $i \in C^\circ$, define

$$c_i := u_i(\overline{\pi}^t(i), \underline{\mathbf{v}}^{t-1}) - u_i(\pi^\circ(i), \overline{\mathbf{v}}^{t-1}),$$
$$o_i := u_i(\overline{\pi}^t(i), \overline{\mathbf{v}}^{t-1}) - u_i(\pi^\circ(i), \underline{\mathbf{v}}^{t-1}).$$

By the choice of (C°, π°) as a minimizing deviation for the pessimistic gap of $\overline{\pi}^t$,

$$G_t^-(\overline{\pi}^t) = \max_{i \in C^\circ} c_i.$$

Moreover, the optimistic gap minimizes over a family that includes this same deviation. Hence,

$$G_t^+(\overline{\pi}^t) \le \max_{i \in C^\circ} o_i.$$

Combining the two inequalities and using $\max_i o_i - \max_i c_i \le \max_i (o_i - c_i)$, we obtain

$$G_t^+(\overline{\pi}^t) - G_t^-(\overline{\pi}^t) \le \max_{i \in C^\circ} (o_i - c_i).$$

For every $i \in C^\circ$, the difference between its optimistic and pessimistic estimates is

$$o_i - c_i = \sum_{j \in \overline{\pi}^t(i) \setminus \{i\}} 2b_{i,j}^{t-1} + \sum_{j \in \pi^\circ(i) \setminus \{i\}} 2b_{i,j}^{t-1}.$$

Every pair appearing in these sums is a candidate witness pair, so its confidence radius is at most $b_{e_t}^{t-1}$. The two coalitions $\overline{\pi}^t(i)$ and $\pi^\circ(i)$ are distinct and both contain i. Hence, at most one of them is the grand coalition, and their two sums contain at most $(n-1) + (n-2) = 2n-3$ terms in total. Therefore,

$$o_i - c_i \le 2(2n-3) b_{e_t}^{t-1} = \kappa_n b_{e_t}^{t-1},$$

which proves the lemma. □

The witness-gap inequality is deterministic: its proof uses neither the event $\mathcal{G}_t$ nor any assumption about how the candidate partitions are computed. It therefore also applies to the local-search heuristic studied in Appendix G and to coalition-level bandit feedback in Appendix F. Exact optimization of the optimistic problem enters the existence-error argument only through the inequality $G_t^+(\overline{\pi}^t) \ge \varepsilon + \Delta_\star$.

For arbitrary candidate-generation procedures, we measure the extent to which this inequality can fail by the *shortfall*

$$s_t := \left(\varepsilon + \Delta_\star - G_t^+(\overline{\pi}^t)\right)^+.$$

When the optimistic problem is solved exactly, $s_t = 0$ on $\mathcal{G}_t$. Let W_T be the set of rounds $t \le T$ in which a witness pair is selected. For $0 \le d < \Delta_\star$, define

$$H_d(T) := \sum_{t \in W_T} \mathbb{1}\{s_t > d\},$$

the number of witness rounds whose shortfall exceeds d. The following theorem also accommodates initialization procedures that take multiple rounds and witness probes that yield more than one observation of the selected pair, as in the coalition-level bandit setting.

Theorem D.1 (Pathwise bound for arbitrary candidates)**.** *Consider UCB-LCB-GS with arbitrary rules for computing the pessimistic and optimistic candidate partitions, and suppose that all gaps are evaluated over the complete deviation family. Assume that every pair has been observed after the first r_0 rounds, during which the learner forms partitions $P_1, \dots, P_{r_0}$ with $d_t = 0$, and that each witness round increases the observation count of its witness pair by at least $q \ge 1$. For UCB-LCB-GS, $r_0 = 1$, $P_1 = \{N\}$, and $q = 1$. Suppose further that $\Delta_\star > 0$ and $T \ge 2$. For $d \in [0, \Delta_\star)$, let*

$$y_d := \frac{2\kappa_n^2 \sigma^2 \beta \log T}{(\Delta_\star - d)^2}.$$

Then the number U_T of rounds $t \le T$ with $d_t = 0$ satisfies, on every sample path,

$$U_T \le r_0 + \binom{n}{2} \left\lceil \frac{y_d}{q} \right\rceil + H_d(T). \tag{11}$$

Consequently,

$$\mathcal{E}_{\varepsilon,T}^{\exists} \le r_0 + \binom{n}{2} \left\lceil \frac{y_d}{q} \right\rceil + \mathbb{E}[H_d(T)],$$

$$\mathcal{R}_{\varepsilon,T} \le \sum_{s=1}^{r_0} \Gamma_{P_s} + \Gamma \left(\binom{n}{2} \left\lceil \frac{y_d}{q} \right\rceil + \mathbb{E}[H_d(T)] + \varphi_T \right),$$

where $\Gamma_P := (\varepsilon - \mathrm{Gap}_k(P, \mathbf{v}))^+$ is the regret incurred by forming partition P.

Proof. Classification of rounds. A round with $d_t = 0$ is either among the first r_0 initialization rounds or occurs after every pair has been observed. In the latter case, both pessimistic tests have failed, and the algorithm selects a witness pair. Thus, if W_T denotes the set of witness rounds,

$$U_T \le r_0 + |W_T|.$$

Witness rounds with small shortfall. Consider a witness round $t \in W_T$ with $s_t \le d$. By the definition of shortfall,

$$G_t^+(\overline{\pi}^t) \ge \varepsilon + \Delta_\star - d.$$

Since the pessimistic test on $\overline{\pi}^t$ has failed, we also have $G_t^-(\overline{\pi}^t) < \varepsilon$. The witness-gap inequality therefore gives

$$\kappa_n b_{e_t}^{t-1} > \Delta_\star - d.$$

Substituting the confidence radius and rearranging,

$$T_{e_t}^{t-1} < \frac{2\kappa_n^2 \sigma^2 \beta \log t}{(\Delta_\star - d)^2} \le y_d.$$

This conclusion does not require the confidence intervals to contain the true utilities.

Counting. Fix a pair e, and let $t_1 < \cdots < t_m$ be the witness rounds for which $s_t \le d$ and $e_t = e$. Since all pairs have already been observed during initialization, $T_e^{t_1-1} \ge 1$. Each of these witness rounds increases the observation count of e by at least q, and the count never decreases. Thus,

$$T_e^{t_m-1} \ge 1 + (m-1)q.$$

Combining this with $T_e^{t_m-1} < y_d$ gives

$$m - 1 < \frac{y_d - 1}{q} < \frac{y_d}{q},$$

and hence

$$m \le \left\lceil \frac{y_d}{q} \right\rceil.$$

Summing over all $\binom{n}{2}$ pairs, at most $\binom{n}{2}\lceil y_d/q \rceil$ witness rounds have shortfall at most d. The remaining witness rounds are counted by $H_d(T)$. Therefore,

$$|W_T| \le \binom{n}{2} \left\lceil \frac{y_d}{q} \right\rceil + H_d(T).$$

Together with the classification of rounds, this establishes (11).

Existence error. Since $\Delta_\star > 0$, an (ε, k)-strongly Nash stable partition exists. Thus, every round with $d_t = 0$ contributes one unit to the existence error, giving $\mathcal{E}_{\varepsilon,T}^{\exists} = \mathbb{E}[U_T]$. Taking expectations in (11) gives the stated existence-error bound.

Regret. The first r_0 rounds contribute $\sum_{s=1}^{r_0} \Gamma_{P_s}$. For any subsequent round with $d_t = 1$ on $\mathcal{G}_t$, the formed partition satisfies $G_t^-(\pi^t) \ge \varepsilon$. By Lemma D.2(a), its true gap is at least ε, so its regret is zero. Declarations on $\mathcal{G}_t^c$ contribute at most Γ each, and their expected number is at most φ_T. Every remaining round after initialization has $d_t = 0$ and incurs regret at most Γ. By the pathwise counting argument, the number of these rounds is bounded by

$$\binom{n}{2} \left\lceil \frac{y_d}{q} \right\rceil + H_d(T).$$

Combining these contributions gives the stated regret bound. □

When the optimistic problem is solved exactly, the shortfall is zero on $\mathcal{G}_t$. Consequently, any witness round with positive shortfall must be a round in which at least one confidence interval fails. The resulting bound on $H_0(T)$ allows us to establish the existence-error guarantee.

Theorem 3.2 (Existence Error Bound)**.** *For any input with $\Delta_\star > 0$, if the optimistic problem* (9) *is solved exactly, then for $T \ge 2$ the existence error of UCB-LCB-GS satisfies*

$$\mathcal{E}_{\varepsilon,T}^{\exists} \le 1 + \tbinom{n}{2}\lceil 2\kappa_n^2 \sigma^2 \beta \log T / \Delta_\star^2 \rceil + n(n-1)/(\beta - 2).$$

Proof. Let $\pi^\star$ satisfy $\mathrm{Gap}_k(\pi^\star, \mathbf{v}) = \varepsilon + \Delta_\star$, and suppose that $\overline{\pi}^t$ maximizes G_t^+ exactly. On $\mathcal{G}_t$, Lemma D.2(a) gives

$$G_t^+(\overline{\pi}^t) \geq G_t^+(\pi^\star) \geq \varepsilon + \Delta_\star.$$

Hence, $s_t = 0$. Every witness round with $s_t > 0$ must therefore lie in $\{t \geq 2 : \mathcal{G}_t^c\}$. It follows that

$$\mathbb{E}[H_0(T)] \leq \varphi_T.$$

Applying Theorem D.1 with $d = 0$ and $r_0 = q = 1$, and then using Lemma D.1, gives

$$\begin{aligned}\mathcal{E}_{\varepsilon,T}^{\exists} &\leq 1 + \binom{n}{2}\lceil y_0 \rceil + \varphi_T \\ &\leq 1 + \binom{n}{2}\left\lceil \frac{2\kappa_n^2\sigma^2\beta\log T}{\Delta_\star^2} \right\rceil + \frac{n(n-1)}{\beta - 2}.\end{aligned}$$

□

D.3 Proof of Theorem 3.3

The regret bound follows from the same counting argument used for the existence-error bound, with each round contributing its regret rather than one unit. We distinguish three types of rounds. The initialization round forms the grand coalition and incurs regret Γ_N. On subsequent rounds when all confidence intervals hold, a declaration of existence produces a partition with zero regret, whereas a round without a declaration selects a witness pair with zero shortfall. The latter rounds are bounded in number by the same counting argument used for Theorem 3.2. All remaining rounds occur when at least one confidence interval fails, and each incurs regret at most Γ.

Theorem 3.3 (Regret Bound)**.** *Under the conditions of Theorem 3.2, the (ε, k)-strong Nash regret of UCB-LCB-GS satisfies*

$$\mathcal{R}_{\varepsilon,T} \leq \Gamma_N + \Gamma\binom{n}{2}\lceil 2\kappa_n^2\sigma^2\beta\log T/\Delta_\star^2 \rceil + \Gamma\,\frac{n(n-1)}{\beta - 2}.$$

Proof. On $\mathcal{G}_t$, every round $t \geq 2$ without a declaration is a witness round with $s_t = 0$, as established in the proof of Theorem 3.2. By the counting argument in Theorem D.1, with $d = 0$ and $q = 1$, there are at most $\binom{n}{2}\lceil y_0 \rceil$ such rounds. Each incurs regret at most Γ. Declarations on $\mathcal{G}_t$ incur zero regret, by Lemma D.2(a). All other rounds after initialization lie in $\mathcal{G}_t^c$; each incurs regret at most Γ, and their expected number is at most

$$\varphi_T \leq \frac{n(n-1)}{\beta - 2},$$

by Lemma D.1. Adding the regret Γ_N incurred in round 1 gives the stated bound. □

Finally, note that

$$\Gamma_N \leq \Gamma \leq \varepsilon + (2n-3)V, \qquad V := \max_{i \neq j} |v_{i,j}|,$$

because the utilities of an agent in two distinct coalitions involve at most $2n - 3$ other agents. Neither κ_n nor the counting argument depends on k. Its effect on the bounds enters through $\Delta_\star$, which is non-increasing in k, and through Γ, which is non-decreasing in k.

E Optimistic Exploration and the Certification Margin

This appendix examines two aspects of the analysis that are essential to the guarantees in Theorems 3.2 and 3.3. First, we show why witness probing is necessary: simply forming the optimistic candidate whenever the learner cannot declare existence may prevent it from obtaining the observations needed to establish stability. Second, we show that the assumption $\Delta_\star > 0$ is also necessary and cannot be removed from the guarantees. Both results are established using two-agent games, for which all relevant quantities can be computed explicitly.

Two-agent games. Consider an instance with $n = 2$, $k = 1$ and $v_{1,2} = v$. There are only two possible partitions: the grand coalition $G = \{\{1, 2\}\}$ and the partition into singletons $S = \{\{1\}, \{2\}\}$. Their group deviation gaps are

$$\mathrm{Gap}_1(G, \mathbf{v}) = v, \qquad \mathrm{Gap}_1(S, \mathbf{v}) = -v.$$

The only pairwise observation is obtained when the two agents form the grand coalition. Given an empirical estimate μ and confidence radius b, the pessimistic estimates of the two partitions are $\mu - b$ and $-\mu - b$, respectively, while their optimistic estimates are $\mu + b$ and $-\mu + b$.

Consider the exploration rule that forms the optimistic candidate $\overline{\pi}^t$ on every round in which the learner does not declare existence. We refer to this rule as *purely optimistic exploration*. Let ξ denote the noise in the observation obtained in round 1, so that $\mu^1 = v + \xi$.

Lemma E.1. *Under purely optimistic exploration, on the event $\{\mu^1 < -\varepsilon\}$, the learner forms the singleton partition S at every round $t \geq 2$, and its estimate of the pairwise utility is never updated.*

Proof. We prove the statement by induction on t. Suppose that only S has been formed since round 1. Then the pair has been observed exactly once, so $T^{t-1} = 1$ and $\mu^{t-1} = \mu^1 < -\varepsilon < 0$. Consequently,

$$-\mu^1 - b > \mu^1 - b \qquad \text{and} \qquad -\mu^1 + b > \mu^1 + b.$$

Thus, S is the unique maximizer of both the pessimistic and optimistic estimates. If the pessimistic test succeeds, the learner forms S and declares existence. Otherwise, purely optimistic exploration also selects S. In either case, the partition formed is S, which provides no observation of the pair. The estimate therefore remains unchanged, completing the induction. □

The next proposition shows that this behaviour can prevent the learner from detecting existence even when a stable partition exists with a positive certification margin.

Proposition E.1. *Let $\varepsilon > 0$, $M > \varepsilon$, $v = -M$, and suppose that the observation noise is Gaussian, $\mathcal{N}(0, \sigma^2)$, with $\beta > 1$. Then*

$$\mathtt{SNS}_{\varepsilon,1}(\mathbf{v}) = \{S\}, \qquad \Delta_\star = M - \varepsilon > 0.$$

Nevertheless, under purely optimistic exploration, for every T,

$$\mathcal{E}^{\exists}_{\varepsilon,T} \geq T\left(1 - e^{-\frac{(M-\varepsilon)^2}{2\sigma^2}}\right) - \sqrt{\frac{\beta}{\beta - 1}}\, e^{\frac{(M-\varepsilon)^2}{2(\beta-1)\sigma^2}}.$$

Proof. Since

$$\mathrm{Gap}_1(S, \mathbf{v}) = M \geq \varepsilon > -M = \mathrm{Gap}_1(G, \mathbf{v}),$$

the singleton partition is the unique $(\varepsilon, 1)$-strongly Nash stable partition, and the certification margin is $M - \varepsilon$.

Let

$$A := \{\xi < M - \varepsilon\}.$$

On A, we have $\mu^1 = -M + \xi < -\varepsilon$, so Lemma E.1 applies. In particular, the learner never obtains another observation of the pair. The pessimistic estimate of G is negative, and therefore, for every $t \geq 2$, the learner declares existence if and only if

$$-\mu^1 - b^{t-1} \geq \varepsilon, \qquad b^{t-1} = \sigma\sqrt{2\beta \log t}.$$

Writing $a := M - \varepsilon - \xi > 0$, this condition is equivalent to

$$\log t \leq \frac{a^2}{2\beta\sigma^2}.$$

Hence, on A, the total number of declarations of existence is at most $e^{\frac{a^2}{2\beta\sigma^2}}$.

On this event, $a = Z := M - \varepsilon - \xi$, where $Z \sim \mathcal{N}(M - \varepsilon, \sigma^2)$. Setting

$$\lambda = \frac{1}{2\beta\sigma^2} < \frac{1}{2\sigma^2},$$

the moment generating function of the square of a Gaussian random variable gives

$$\mathbb{E}\left[\mathbb{1}_A e^{a^2/(2\beta\sigma^2)}\right] \le \mathbb{E}[e^{\lambda Z^2}] = \sqrt{\frac{\beta}{\beta-1}}\, e^{\frac{(M-\varepsilon)^2}{2(\beta-1)\sigma^2}}.$$

On A^c, the learner can make at most T declarations, and the Gaussian tail bound gives

$$\Pr[A^c] \le e^{-\frac{(M-\varepsilon)^2}{2\sigma^2}}.$$

Since a declaration of existence is incorrect on every round precisely when the learner predicts non-existence on an instance admitting an $(\varepsilon, 1)$-strongly Nash stable partition, we have

$$\mathcal{E}^{\exists}_{\varepsilon,T} = T - \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{d_t = 1\}\right].$$

Combining the bounds on the expected number of declarations on A and A^c yields the claimed result. □

The lower bound grows linearly with T whenever $M - \varepsilon$ is sufficiently large relative to σ. For example, with $M = 10$, $\varepsilon = 1$, $\sigma = 3$ and $\beta = 3$, it gives

$$\mathcal{E}^{\exists}_{\varepsilon,T} \ge 0.988\, T - 11.7.$$

Purely optimistic exploration can also incur linear regret. To see this, consider $v = M > \varepsilon$, so that the grand coalition G is the stable partition. On the event

$$\{\xi < -M - \varepsilon\},$$

which has probability $\Phi(-(M+\varepsilon)/\sigma) > 0$, the initial estimate satisfies $\mu^1 = M + \xi < -\varepsilon$. By Lemma E.1, the learner consequently remains at S, whose regret is $\varepsilon + M$ in every round.

In both examples, the learner needs another observation of the pair to resolve the uncertainty, but the purely optimistic rule never selects the grand coalition once the initial observation is sufficiently unfavourable. Witness probing avoids this problem. With two agents, the only witness pair is $\{1, 2\}$, so every round without a declaration forms the grand coalition and obtains another observation of the pair. The logarithmic existence-error and regret guarantees of Theorems 3.2 and 3.3 therefore apply.

The certification margin is necessary. Witness probing addresses the failure of purely optimistic exploration, but the guarantees of Theorems 3.2 and 3.3 still require $\Delta_\star > 0$. This condition is strictly stronger than the mere existence of an (ε, k)-strongly Nash stable partition. The following proposition shows that the positive-margin assumption is not an artifact of our analysis: it cannot be dropped for any learner that declares existence only when a pessimistic test succeeds, regardless of its exploration rule.

The reason follows from part (b) of Lemma D.2. When the best partition is stable with exactly zero slack, its pessimistic estimate remains strictly below ε whenever all confidence intervals hold.

Proposition E.2. *Suppose that*

$$\max_{\pi} \mathrm{Gap}_k(\pi, \mathbf{v}) = \varepsilon,$$

so that $\Delta_\star = 0$. Then any learner that declares existence only if some partition π satisfies $G_t^-(\pi) \ge \varepsilon$, after observing every pair, has

$$\mathcal{E}^{\exists}_{\varepsilon,T} \ge T - \frac{n(n-1)}{\beta - 2}.$$

This holds regardless of the exploration rule.

Proof. On $\mathcal{G}_t^<$, Lemma D.2(b) gives

$$G_t^-(\pi) < \mathrm{Gap}_k(\pi, \mathbf{v}) \le \varepsilon$$

for every partition π. Thus, the learner cannot declare existence on this event, regardless of which partitions it has explored. Consequently,

$$\sum_{t=1}^{T} \Pr[d_t = 1] \leq \sum_{t \geq 2} \Pr[(\mathcal{G}_t^{<})^c] \leq \frac{n(n-1)}{\beta - 2},$$

where the final inequality follows from Lemma D.1. Since an (ε, k)-strongly Nash stable partition exists by assumption, the number of incorrect non-existence decisions is T minus the expected number of declarations. The stated lower bound follows. □

One might instead consider the smallest violation,

$$\min_{\pi, C, \pi'} \left(\varepsilon - \mathrm{Gap}_C(\pi, \pi', \mathbf{v})\right)$$

over deviations for which the gap is below ε. This quantity is positive on every instance, since the family of partitions and deviations is finite. It therefore cannot express the positive-margin condition required by our analysis. This is why we formulate our guarantees in terms of $\Delta_\star$.

F Coalition-Level Bandit Feedback

Section 4 introduces an extension of UCB-LCB-GS to per-agent coalition-level bandit feedback, in which agents report only their satisfaction with their coalition as a whole. This feedback model provides substantially less information about individual pairwise utilities than semi-bandit feedback, and requires changes to the way the learner collects observations. In this appendix, we develop the extension and establish its guarantees.

We first define the feedback model and examine which utilities can be recovered from its observations (Appendix F.1). This identifiability analysis motivates the initialization and exploration strategy of the adapted algorithm (Appendix F.2). We then establish the validity of its estimator and confidence intervals (Appendix F.3) and prove its soundness, existence-error and regret guarantees (Appendix F.4). Finally, we compare the two feedback models and an alternative exploration strategy (Appendix F.5), before discussing modeling extensions and limitations (Appendix F.6). The corresponding experiments build on the implementation, baseline definitions and semi-bandit results presented in Appendices I to K, and are reported in Appendix L.

F.1 Coalition-Level Bandit Feedback and Identifiability

Under our main feedback model, the learner observes noisy pairwise utilities for agents assigned to the same coalition. Since it observes the individual pairwise utilities rather than only the resulting coalition utility, we refer to this as semi-bandit feedback. We now consider a less informative model in which each agent reports only a noisy estimate of her total utility from the coalition to which she belongs. We call this *per-agent coalition-level bandit feedback*, or coalition-level feedback for short.

In round t, after the learner forms the partition π^t, each agent i reports

$$R_i^t = u_i(\pi^t(i), \mathbf{v}) + \eta_i^t = \sum_{j \in \pi^t(i) \setminus \{i\}} v_{i,j} + \eta_i^t. \tag{12}$$

The noises η_i^t have mean zero, are σ^2-subgaussian, and are independent across agents and rounds and of the learner's internal randomness. Independence across agents within a round is important: it ensures that the reports of two agents in the same two-agent coalition provide two independent observations. The performance metrics in (3)–(4) and the decision variable d_t remain unchanged.

Under coalition-level feedback, each coalition C generates $|C|$ individual reports, but each report aggregates the utilities that an agent assigns to all other members of C. Thus, unlike semi-bandit feedback, the learner does not directly observe the individual pairwise utilities in larger coalitions. The special case of a two-agent coalition is different: each of its two reports isolates the same pairwise utility, as we discuss below. It is important to distinguish this model from one in which a coalition produces only a single collective reward; our results do not cover the latter model (Appendix F.6).

The statistical target. The stability of a partition depends on the coalition utilities $u_i(C, \mathbf{v})$ for all agents i and coalitions $C \ni i$. There are $n2^{n-1}$ such agent-coalition pairs. A learner that treated every coalition as a separate arm and estimated its utility by averaging the reports received would need to visit coalitions that it may never have formed. After t rounds, agent i can have been assigned to at most t of the 2^{n-1} coalitions containing her, whereas the stability test considers all of them.

Additive separability avoids this difficulty. Every coalition utility is a known linear function of the $\binom{n}{2}$ pairwise utilities. Therefore, it suffices to estimate the pairwise utility profile $\mathbf{v}$; all coalition utilities can then be recovered by summation, just as under the semi-bandit model.

Identifiability. Although coalition-level feedback does not reveal the individual pairwise utilities in larger coalitions, it is sufficient to recover them through two-agent coalitions. Indeed, whenever $\pi^t(i) = \{i, j\}$, equation (12) becomes

$$R_i^t = v_{i,j} + \eta_i^t, \qquad R_j^t = v_{i,j} + \eta_j^t.$$

Thus, a two-agent coalition provides two independent, unbiased observations of $v_{i,j}$, where the second observation uses the symmetry $v_{j,i} = v_{i,j}$. Forming the pairs of a matching in a single round therefore allows the learner to observe several pairwise utilities simultaneously, and every pair can eventually be observed in this way.

Larger coalitions, in contrast, combine several unknown utilities in each report. In particular, forming the grand coalition, which provides an observation of every pair under semi-bandit feedback and is used to initialize Algorithm 1, reveals only the row sums $\sum_{j \neq i} v_{i,j}$. The following proposition shows that these sums do not even determine whether a stable partition exists. Thus, the initialization used under semi-bandit feedback is insufficient for coalition-level feedback.

Proposition F.1. *Let $n = 4$ and $k = 1$. Consider the symmetric utility profiles $\mathbf{v}$ and $\mathbf{v}'$ defined by*

$$v_{1,2} = v_{3,4} = 2, \qquad v_{i,j} = -1 \quad \textit{for all other pairs},$$

and $\mathbf{v}' = -\mathbf{v}$. Both profiles give every agent the same expected report, namely 0, in the grand coalition. However,

$$\max_{\pi} \mathrm{Gap}_1(\pi, \mathbf{v}) = 2, \qquad \max_{\pi} \mathrm{Gap}_1(\pi, \mathbf{v}') = 1.$$

Consequently, for $\varepsilon = 2$, a $(2, 1)$-strongly Nash stable partition exists under $\mathbf{v}$ but not under $\mathbf{v}'$, even though the reports generated by forming the grand coalition have the same distribution under both profiles.

Proof. Under $\mathbf{v}$, every agent's pairwise utilities sum to $2 - 1 - 1 = 0$. The same is true under $\mathbf{v}' = -\mathbf{v}$. Thus, the expected reports from the grand coalition are identical.

We first use the fact that, when $k = 1$, an agent in a non-singleton coalition can always deviate to being alone. The corresponding group deviation gap is exactly her current coalition utility, since the utility of a singleton is zero.

Under $\mathbf{v}$, consider the partition

$$\pi^\star = \{\{1, 2\}, \{3, 4\}\}.$$

Every agent obtains utility 2, and any unilateral deviation gives her utility at most 0. Hence,

$$\mathrm{Gap}_1(\pi^\star, \mathbf{v}) = 2.$$

No partition has a larger gap. Indeed, every agent obtains utility at most 2 in any coalition, so an agent who is not alone can deviate to being alone with a gap of at most 2. If all agents are alone, agent 1 can join agent 2 and obtain utility 2. Therefore,

$$\max_{\pi} \mathrm{Gap}_1(\pi, \mathbf{v}) = 2.$$

Under $\mathbf{v}'$, each agent has one enemy, with utility -2, and two friends, each with utility 1. Moreover, the two friends of any agent are enemies of each other. An agent obtains utility 2 only in a coalition containing both of her friends and excluding her enemy. In such a coalition, however, each of those two friends has utility at most -1. It follows that every partition contains at least one agent with utility at most 1.

If this agent is in a non-singleton coalition, she can leave and become a singleton, giving a deviation gap of at most 1. If she is alone, consider the coalition memberships of her two friends. Either some coalition contains a friend but not her enemy, in which case joining that coalition gives her utility at least 1, or both friends belong to a coalition containing her enemy, in which case joining that coalition gives her utility 0. In either case, there is a unilateral deviation with gap at most 1. Therefore, every partition has group deviation gap at most 1.

This upper bound is attained by

$$\pi' = \{\{1, 3\}, \{2, 4\}\},$$

under which every agent has utility 1 and no unilateral deviation yields a smaller group deviation gap. Thus,

$$\max_{\pi} \mathrm{Gap}_1(\pi, \mathbf{v}') = 1.$$

These values can also be verified by evaluating Gap_1 over all 15 partitions of four agents. Since the noise distribution is independent of the utility profile, the reports obtained from the grand coalition have the same distribution under $\mathbf{v}$ and $\mathbf{v}'$. □

F.2 UCB-LCB-GS under Coalition-Level Bandit Feedback

We now describe the adaptation of UCB-LCB-GS to coalition-level bandit feedback. Algorithm 2 retains the decision logic of Algorithm 1: it maintains upper and lower confidence bounds on pairwise utilities, computes the pessimistic and optimistic candidates using the max-min problems in (7) and (9), declares existence only when a candidate passes the pessimistic test, and otherwise selects a witness pair.

The difference lies in how the learner obtains observations. Under semi-bandit feedback, forming a partition reveals all pairwise utilities corresponding to agents in the same coalition. Under coalition-level feedback, a pairwise utility is observed only when its two agents form a coalition of their own. Thus, observations must be collected deliberately. Algorithm 2 uses a matching-based initialization, followed by pair probes during the learning phase. We describe these two stages in the order in which the algorithm executes them.

Algorithm 2 UCB-LCB-GS under coalition-level feedback

Input: as in Algorithm 1; a round-robin schedule $P_1, \ldots, P_r$.

1: **for** $t = 1, \ldots, T$ **do**
2: **if** $t \le r$ **then** set $d_t = 0$ and form P_t ▷ initialization
3: **else**compute $\underline{\pi}^t$ via (7)
4: **if** $G_t^-(\underline{\pi}^t) \ge \varepsilon$ **then** set $d_t = 1$ and form $\underline{\pi}^t$
5: **else**compute $\overline{\pi}^t$ via (9)
6: **if** $G_t^-(\overline{\pi}^t) \ge \varepsilon$ **then** set $d_t = 1$ and form $\overline{\pi}^t$
7: **else**set $d_t = 0$; select (C°, π°) and the witness pair $e_t = \{i, j\}$ as in Algorithm 1
8: form $\overline{\pi}^t$ if $\overline{\pi}^t(i) = \{i, j\}$, and otherwise form $\overline{\pi}^t$ with $\{i, j\}$ placed in a new coalition
9: observe $R_1^t, \ldots, R_n^t$; update the estimates of pairs that formed a two-agent coalition, using both reports

Initialization. During its first r rounds, the algorithm observes every pair exactly once (line 2). A round-robin schedule achieves this using the minimum possible number of rounds. Specifically, let $M_1, \ldots, M_r$ be matchings of the agents such that every pair belongs to exactly one matching. Such a schedule exists with $r = n - 1$ matchings when n is even, and with $r = n$ matchings when n is odd. In the latter case, one agent remains unmatched in each matching. These values are optimal because a matching contains at most $\lfloor n/2 \rfloor$ pairs, whereas there are $\binom{n}{2}$ pairs in total.

The partition P_s consists of the two-agent coalitions corresponding to the pairs in M_s, with any unmatched agent placed alone. Forming $P_1, \ldots, P_r$ therefore produces two independent observations of every pair. These initialization rounds replace the single grand-coalition round of Algorithm 1. As in the original initialization, the learner sets $d_t = 0$ throughout. If $T < r$, the horizon ends before initialization is complete.

Estimation. The algorithm estimates pairwise utilities using only reports from agents in two-agent coalitions, which it records at the end of each round (line 9). Reports from singleton coalitions contain only noise, while reports from larger coalitions combine several unknown utilities. For a pair $\{i, j\}$, let $N_{i,j}^t$ denote the number of observations collected during rounds $1, \ldots, t$ from the two agents when they form a two-agent coalition. Thus, $N_{i,j}^t$ increases by two in each such round. Let $\mu_{i,j}^t$ be the average of these observations. The confidence bounds are defined as in (5), with $N_{i,j}^t$ replacing $T_{i,j}^t$. Their validity under adaptive sampling is established in Appendix F.3.

Candidates and tests. After initialization, every pair has a finite confidence radius, and the algorithm follows the same decision procedure as Algorithm 1. It first computes the pessimistic candidate using (7) (line 3) and declares existence if this candidate passes the pessimistic test (line 4). Otherwise, it computes the optimistic candidate using (9) (line 5) and applies the same pessimistic test to that candidate (line 6). The stability criterion and both max-min optimization problems depend on the observations only through the confidence bounds. They therefore remain unchanged under the less informative feedback model.

Witness pairs and pair probes. If neither candidate passes the pessimistic test, the algorithm sets $d_t = 0$ and selects the most damaging deviation (C°, π°) of $\overline{\pi}^t$ under the pessimistic estimate. It then selects the witness pair $e_t = \{i, j\}$ exactly as in Algorithm 1 (line 7). The witness-gap inequality of Lemma D.3 depends only on the confidence bounds. Together with the positive-margin assumption and exact optimistic maximization, it implies that the selected pair has a sufficiently large confidence radius whenever the learner cannot yet declare existence on a round with valid confidence intervals.

Under semi-bandit feedback, the learner would form $\overline{\pi}^t$ or the corresponding most damaging deviation, whichever places the selected pair in the same coalition. Under coalition-level feedback, however, this does not reveal $v_{i,j}$ unless $\{i, j\}$ forms a coalition by itself. Therefore, unless $\overline{\pi}^t(i) = \{i, j\}$ already holds, the algorithm removes i and j from their current coalitions and places them together in a new two-agent coalition (line 8). All other agents retain their original coalition assignments. We call this a *pair probe*. It provides two independent observations of the selected pair, which is the key to the counting argument in Appendix F.4.

Thus, Algorithm 2 differs from Algorithm 1 in three respects: its initialization, its probe partition, and its estimator. The confidence bounds, the two max-min problems, both pessimistic tests, and the selection of the witness pair remain unchanged.

F.3 Estimation and Confidence Intervals

Under symmetric utilities, the two agents in a two-agent coalition $\{i, j\}$ report noisy observations of the same parameter $v_{i,j}$. Since their noises are independent, the estimator treats both reports as observations of this single pairwise utility. Accordingly, the observation count $N_{i,j}^t$ increases by two whenever $\{i, j\}$ forms a coalition, and $\mu_{i,j}^t$ is the average of all reports collected from this pair. This is also how the implementation updates its estimates.

The confidence radius used at round t is

$$b_{i,j}^{t-1} = \sigma \sqrt{\frac{2\beta \log t}{N_{i,j}^{t-1}}},$$

with $b_{i,j}^{t-1} = +\infty$ when $N_{i,j}^{t-1} = 0$. After the r initialization rounds, every pair has been observed twice, so all confidence radii are finite.

The estimator deliberately discards reports from larger coalitions, since each such report combines several pairwise utilities. Using these reports would require estimating $\mathbf{v}$ jointly from sums of utilities and constructing confidence sets of a different form. We do not pursue that approach here.

Whether an observation is used depends only on the partition formed by the learner, which is chosen before the noise of the round is drawn. Thus, the observations are selected adaptively, just as under pairwise feedback, but each observation remains conditionally subgaussian given all preceding information. The concentration argument of Lemma D.1 therefore continues to apply, with one modification: under coalition-level feedback, a pair can be observed twice in a single round. Consequently, its observation count before round t can take up to $2(t-1)$ possible values, rather than $t-1$.

Lemma F.1. *Under coalition-level feedback with the estimator described above,*

$$\Pr[\mathcal{G}_t^c] \le 2n(n-1)t^{1-\beta}$$

for every $t \ge 2$. Consequently, for $\beta > 2$,

$$\varphi_T \le \frac{2n(n-1)}{\beta - 2},$$

where $\mathcal{G}_t$ and φ_T are defined as in Appendix D, with $N_{i,j}^{t-1}$ replacing $T_{i,j}^{t-1}$.

Proof. We follow the proof of Lemma D.1, ordering the observations within each round by agent. Because the agents' noises are independent within a round, the noise of each observation is independent of the σ-field generated by everything preceding it, including the other report from the same round. Hence, the same supermartingale argument gives, for every pair e and every m,

$$\Pr\left[|S_m| \ge mx,\ \tau_m < \infty\right] \le 2e^{-\frac{mx^2}{2\sigma^2}}.$$

The only change is in the union bound. On the event $\{N_e^{t-1} = m\}$, we now have $1 \le m \le 2(t-1)$. Therefore,

$$\Pr[\mathcal{G}_t^c] \le \binom{n}{2} \cdot 2(t-1) \cdot 2t^{-\beta} \le 2n(n-1)t^{1-\beta}.$$

Summing over $t \ge 2$, as in Lemma D.1, yields the stated bound on φ_T. □

The sandwich inequality of Lemma D.2 and the witness-gap inequality of Lemma D.3 depend only on the confidence bounds, not on how the observations are obtained. Both therefore remain valid under coalition-level feedback.

F.4 Soundness, Existence Error, and Regret

The pessimistic test of Algorithm 2 is identical to that of Algorithm 1. Its soundness guarantee therefore carries over, with the concentration bound adjusted to account for the two observations obtained from each pair probe. The main difference concerns the rate at which uncertainty decreases. Initialization now requires r rounds instead of one, while each witness probe provides two observations of its selected pair instead of one.

The pathwise bound of Theorem D.1 was formulated to accommodate both changes. The next proposition verifies its conditions for Algorithm 2, including when the candidate partitions are computed approximately.

Proposition F.2. *Under coalition-level feedback with symmetric utilities and $\beta > 2$, the conclusions of Theorem D.1 apply to Algorithm 2 with arbitrary rules for computing the candidates, using*

$$r_0 = \min\{T, r\}, \qquad P_1, \dots, P_{r_0} \quad \text{and} \quad q = 2,$$

and with

$$\varphi_T \le \frac{2n(n-1)}{\beta - 2}.$$

When $T \le r$, the bounds hold directly because every round is an initialization round.

Proof. If $T \le r$, all rounds are part of the initialization, so $d_t = 0$ for every t. Hence, $U_T = T = r_0$, no declaration of existence is made, and the bounds in Theorem D.1 follow directly from the initialization terms.

Now suppose that $T > r$. The assumptions of Theorem D.1 are then satisfied with $r_0 = r$. First, the initial r rounds form the partitions $P_1, \dots, P_r$, during which every pair is observed. Second, every subsequent witness round increases the observation count of its selected pair by two, since the pair forms a two-agent coalition under either case of the probe rule. Thus, $q = 2$. Finally, the expected number of rounds with a failed confidence interval is bounded by Lemma F.1.

The remaining ingredients of Theorem D.1, namely the sandwich inequality and the witness-gap inequality, hold unchanged under coalition-level feedback. This verifies all the required conditions. □

We can now establish the main guarantee for coalition-level feedback.

Theorem 4.1 (Coalition-Level Feedback)**.** *Consider per-agent coalition-level bandit feedback with symmetric utilities, and let $\beta > 2$. Let $P_1, \dots, P_r$ be the round-robin initialization partitions of Algorithm 2, write $\Gamma_P := (\varepsilon - \mathrm{Gap}_k(P, \mathbf{v}))^+$ for the regret of a partition P, and let Γ, $\Delta_\star$ and $\kappa_n = 2(2n-3)$ be as in Section 3. Then Algorithm 2 satisfies the following.*

(a) Soundness. *For any input with $\texttt{SNS}_{\varepsilon,k}(\mathbf{v}) = \emptyset$, $\sum_{t=1}^{T} \Pr[d_t = 1] \le \frac{2n(n-1)}{\beta-2}$, and in particular $\mathcal{E}_T^{\exists} \le \frac{2n(n-1)}{\beta-2}$. This holds for any method of computing the candidate partitions, provided the pessimistic test is evaluated exactly.*

(b) Existence error. *For any input with $\Delta_\star > 0$ and any $T \ge 2$, if the optimistic candidate maximizes the objective of* (9) *exactly at every round, then, with $y := \kappa_n^2 \sigma^2 \beta \log T / \Delta_\star^2$, $\mathcal{E}_{\varepsilon,T}^{\exists} \le \min\{T, r\} + \binom{n}{2}\lceil y \rceil + \frac{2n(n-1)}{\beta-2}$.*

(c) Regret. *Under the assumptions of (b), $\mathcal{R}_{\varepsilon,T} \le \sum_{s=1}^{\min\{T,r\}} \Gamma_{P_s} + \Gamma\binom{n}{2}\lceil y \rceil + \Gamma\,\frac{2n(n-1)}{\beta-2}$.*

Proof. (a) Declarations are made only after initialization, when every pair has been observed, and only when a candidate partition passes the pessimistic test. The proof of Theorem 3.1 therefore applies with Lemma F.1 replacing Lemma D.1. The argument does not depend on how the candidates are computed.

(b) By Proposition F.2, Theorem D.1 applies with $r_0 = \min\{T, r\}$ and $q = 2$. When the optimistic problem is solved exactly, the shortfall s_t is zero on $\mathcal{G}_t$, as shown in the proof of Theorem 3.2. Hence,

$$\mathbb{E}[H_0(T)] \le \varphi_T \le \frac{2n(n-1)}{\beta - 2}.$$

Taking $d = 0$ in the pathwise bound gives the stated existence-error guarantee. In particular, since

$$\frac{y_0}{q} = \frac{y_0}{2} = y,$$

the resulting counting term is $\binom{n}{2}\lceil y \rceil$, as required.

(c) Applying the regret bound of Theorem D.1 with the same parameters gives the initialization contribution

$$\sum_{s=1}^{\min\{T,r\}} \Gamma_{P_s}.$$

As in the proof of Theorem 3.3, every subsequent round without a declaration on $\mathcal{G}_t$ is a witness round with zero shortfall. There are at most $\binom{n}{2}\lceil y \rceil$ such rounds, each contributing at most Γ. Declarations on $\mathcal{G}_t$ incur zero regret, while all remaining rounds lie in $\mathcal{G}_t^c$ and contribute at most Γ each. Their expected number is bounded by

$$\varphi_T \le \frac{2n(n-1)}{\beta - 2}.$$

Combining these contributions gives the claimed regret bound. □

The statistical part of these guarantees depends neither on k nor on the number of possible partitions. The probabilistic bounds involve a union bound over pairs, while the effect of the complete family of deviations is handled deterministically by the sandwich inequality. The parameter k affects the bounds through $\Delta_\star$ and Γ, and affects the computational cost through the underlying optimization problems.

F.5 Comparison with Pairwise Semi-Bandit Feedback

The two feedback models differ in what the learner can infer from a round. Under pairwise semi-bandit feedback, every pair of agents in the same coalition provides an observation, regardless of why the learner formed that coalition. Thus, a round in which the learner declares existence, or

probes a witness pair within a larger coalition, also improves the estimates of other co-located pairs. Under coalition-level feedback, only pairs forming two-agent coalitions are observed. A pair probe therefore provides observations of the selected pair, while the other coalition assignments do not directly contribute to the estimator.

The theoretical bounds capture only part of this difference. The leading terms of Theorems 4.1 and 3.2–3.3 coincide up to the factor $q = 2$, which accounts for the two independent reports obtained from each pair probe. The remaining differences arise from the r-round initialization instead of the single initialization round and the doubled concentration term. However, the worst-case analysis counts only the observations of the selected witness pair in both feedback models. It therefore does not account for the additional observations provided by semi-bandit feedback when several pairs share a coalition. Consequently, the bounds do not imply that every trajectory under coalition-level feedback must have higher regret than its semi-bandit counterpart. Nevertheless, the difference is substantial in our synthetic experiments (Appendix L).

An alternative exploration strategy. Witness probing is not the only way to obtain logarithmic guarantees under coalition-level feedback. A simpler alternative is to continue the round-robin schedule: on every round without a declaration, the learner forms the next partition in the cycle. We call this strategy *periodic matching*. It observes every pair at the same rate, and its analysis uses part (c) of Lemma D.2, rather than the witness-gap inequality.

Let

$$y := \frac{\kappa_n^2 \sigma^2 \beta \log T}{\Delta_\star^2}, \qquad \Gamma_{\mathcal{M}} := \max_s \Gamma_{P_s}.$$

Proposition F.3. *Under coalition-level feedback, consider a learner that forms $P_1, \dots, P_r$ in its first r rounds. Subsequently, it declares $d_t = 1$ and forms $\underline{\pi}^t$ whenever*

$$G_t^-(\underline{\pi}^t) \geq \varepsilon,$$

and otherwise sets $d_t = 0$ and forms the next partition in the cycle $P_1, \dots, P_r, P_1, \dots$. Suppose that $\Delta_\star > 0$, $T \geq 2$, $\beta > 2$, and that $\underline{\pi}^t$ maximizes G_t^- exactly. Then

$$\mathcal{E}_{\varepsilon,T}^{\exists} \leq r\lceil y \rceil + \varphi_T, \qquad \mathcal{R}_{\varepsilon,T} \leq \Gamma_{\mathcal{M}}\, r\lceil y \rceil + \Gamma \varphi_T.$$

Proof. Let $\bar{b}_t := \max_{i,j} b_{i,j}^{t-1}$. On $\mathcal{G}_t$, after initialization, Lemma D.2(c) gives

$$G_t^-(\pi^\star) \geq \mathrm{Gap}_k(\pi^\star, \mathbf{v}) - \kappa_n \bar{b}_t = \varepsilon + \Delta_\star - \kappa_n \bar{b}_t.$$

If round t is a round without a declaration, then, by the exact maximization of G_t^-,

$$\varepsilon > G_t^-(\underline{\pi}^t) \geq G_t^-(\pi^\star).$$

It follows that

$$\kappa_n \bar{b}_t > \Delta_\star.$$

Thus, the pair with the largest confidence radius, which is also the least-observed pair, has fewer than

$$\frac{2\kappa_n^2 \sigma^2 \beta \log T}{\Delta_\star^2} = 2y$$

observations.

Consider the j-th round without a declaration. Before that round, at least $\lfloor (j-1)/r \rfloor$ complete cycles of the matching schedule have been formed. Each complete cycle provides two observations of every pair. Consequently, the least-observed pair has received at least

$$2 \left\lfloor \frac{j-1}{r} \right\rfloor$$

observations. On $\mathcal{G}_t$, the preceding bound on its observation count therefore implies

$$2 \left\lfloor \frac{j-1}{r} \right\rfloor < 2y.$$

Hence,

$$\left\lfloor \frac{j-1}{r} \right\rfloor \leq \lceil y \rceil - 1,$$

which gives

$$j \leq r\lceil y \rceil.$$

This count also includes the first r initialization rounds, during which the learner makes no declaration.

It follows that the number of rounds without a declaration on which all confidence intervals hold is at most $r\lceil y \rceil$. Rounds without a declaration on which some confidence interval fails contribute at most φ_T in expectation. For regret, declarations on $\mathcal{G}_t$ incur zero regret, declarations outside $\mathcal{G}_t$ incur at most Γ, and every round that forms a partition from the matching schedule incurs at most $\Gamma_{\mathcal{M}}$. Combining these contributions gives both bounds. □

Periodic matching has a smaller worst-case bound on the number of rounds without a declaration: $r\lceil y \rceil$, compared with $\binom{n}{2}\lceil y \rceil$ under witness probing. The difference is a factor of approximately $n/2$. However, periodic matching does not take into account which pairs are relevant to the stability of the candidate partitions. Moreover, its analysis requires an exact pessimistic oracle, whereas the witness-probing analysis requires an exact optimistic oracle. We use witness probing because it provides a common exploration strategy under both feedback models.

F.6 Modeling Extensions and Limitations

The role of additivity. Coalition-level feedback is informative about pairwise utilities because of the additive structure of the model. Without additivity, observations from coalitions that the learner has formed would impose no restrictions on the utilities of coalitions it has never formed. In particular, a stability test involving such coalitions could not be resolved, regardless of the number of rounds. All the results in this appendix therefore rely on the additive structure, which reduces the learning problem to estimating the pairwise utility profile $\mathbf{v}$.

Asymmetric utilities. The factor of two gained from a pair probe relies on symmetric utilities. When $v_{i,j} = v_{j,i}$, the two reports from a two-agent coalition $\{i, j\}$ estimate the same parameter and can be averaged. Under asymmetric utilities, the report of agent i estimates $v_{i,j}$, whereas the report of agent j estimates $v_{j,i}$. These are different parameters and cannot be averaged.

In this case, the learner maintains a separate estimate, observation count and confidence radius for each of the $n(n-1)$ ordered pairs. A two-agent coalition adds one observation to each of its two ordered pairs. The round-robin schedule still observes every ordered pair, since both reports are collected whenever a pair forms a coalition. The witness-gap inequality also remains valid: it involves only the radii $b_{i,j}^{t-1}$ corresponding to agent i's own utilities, and the witness can be selected among ordered pairs (i, j) with $i \in C^\circ$. The corresponding pair probe observes this ordered pair once, so $q = 1$.

In the concentration argument, each of the $n(n-1)$ ordered parameters is observed at most once per round. Consequently,

$$\Pr[\mathcal{G}_t^c] \leq n(n-1)(t-1) \cdot 2t^{-\beta} \leq 2n(n-1)t^{1-\beta}.$$

The same arguments as in Appendix F.4 therefore give the soundness bound

$$\frac{2n(n-1)}{\beta - 2}$$

and, under the conditions of Theorem 4.1, the existence-error bound

$$\mathcal{E}_{\varepsilon,T}^{\exists} \leq \min\{T, r\} + n(n-1)\left\lceil \frac{2\kappa_n^2 \sigma^2 \beta \log T}{\Delta_\star^2} \right\rceil + \frac{2n(n-1)}{\beta - 2},$$

with the corresponding regret bound.

The leading term is approximately four times that of the symmetric case. This follows from having twice as many parameters to estimate, while each pair probe provides only one observation of each ordered parameter, rather than two observations of a shared parameter. As under pairwise feedback (Appendix P), asymmetric utilities can also change which instances admit a stable partition.

Collective rewards. Per-agent coalition-level reports should be distinguished from a model in which each coalition produces only one collective reward, namely the sum of its members' utilities. Under symmetric utilities, a two-agent coalition still reveals a noisy observation of $2v_{i,j}$, so the pairwise utilities remain identifiable in principle. The algorithm would, however, need a different estimator, using one observation of $2v_{i,j}$ per probe. We do not analyze this model.

Under asymmetric utilities, collective rewards do not determine whether a stable partition exists. To see this, consider two-agent games with utility profiles

$$(v_{1,2}, v_{2,1}) = (3, -1) \qquad \text{and} \qquad (v_{1,2}, v_{2,1}) = (1, 1).$$

Both profiles produce the same collective reward for every partition. In the first game, however, the grand coalition is unstable because agent 2 prefers to leave, while the singleton partition is unstable because agent 1 prefers to join agent 2. In the second game, the grand coalition is $(1, 1)$-strongly Nash stable. Thus, the collective rewards do not reveal whether a stable partition exists. Our results concern per-agent reports only.

Open questions. The factor r in the initialization and the factor $\binom{n}{2}$ in the worst-case bound arise from the particular observation schedules used here; they are not lower bounds for the problem. In principle, coalitions containing more than two agents could also be used for estimation, by jointly estimating $\mathbf{v}$ from the resulting aggregate observations. Determining the optimal dependence on n under coalition-level feedback remains open.

G Effect of Approximate Optimization

Theorems 3.2 and 3.3 assume that the optimistic max-min problem (9) is solved exactly. This assumption separates the statistical difficulty of the problem from its computational difficulty. In our experiments, we instead use a local-search heuristic, SOLVE (Appendix I), for which no approximation guarantee is known. We therefore examine how errors in solving the optimization problem affect the theoretical bounds. The pessimistic test is evaluated exactly, regardless of how its candidate partition is obtained. Thus, Theorem 3.1 remains unchanged; only the existence-error and regret bounds are affected. Appendix H discusses the computational aspects of the max-min problems.

Shortfall and optimization error. Theorem D.1 applies regardless of how the candidate partitions are computed. It bounds every round without a declaration by a logarithmic term and the number $H_d(T)$ of witness rounds on which the optimistic candidate has shortfall greater than d. We relate this shortfall to the usual optimization error

$$\delta_t := \max_{\pi \in \Pi} G_t^+(\pi) - G_t^+(\overline{\pi}^t) \;\geq\; 0,$$

which is defined at every witness round, since such a round always computes the optimistic candidate $\overline{\pi}^t$.

Corollary G.1. *On $\mathcal{G}_t$, $s_t \leq \delta_t$. Consequently, for every $d \in [0, \Delta_\star)$,*

$$H_d(T) \leq \sum_{t \in W_T} \mathbb{1}\{\delta_t > d\} + \sum_{t=2}^{T} \mathbb{1}\{\mathcal{G}_t^c\},$$

and

$$\mathcal{E}_{\varepsilon,T}^{\exists} \leq 1 + \binom{n}{2} \lceil y_d \rceil + \mathbb{E}\left[\sum_{t \in W_T} \mathbb{1}\{\delta_t > d\}\right] + \varphi_T.$$

Proof. On $\mathcal{G}_t$, Lemma D.2 gives

$$\max_{\pi \in \Pi} G_t^+(\pi) \geq G_t^+(\pi^\star) \geq \varepsilon + \Delta_\star.$$

By the definition of δ_t, it follows that

$$\varepsilon + \Delta_\star - G_t^+(\overline{\pi}^t) \leq \delta_t.$$

Since $\delta_t \geq 0$, we also have $s_t \leq \delta_t$. Consequently, a witness round with $s_t > d$ must either have $\delta_t > d$ or lie outside $\mathcal{G}_t$. Since round 1 is never a witness round, the first inequality follows. Taking expectations in Theorem D.1, with $r_0 = q = 1$, gives the second inequality. □

The additional term counts the rounds in which SOLVE returns an optimistic candidate whose gap falls short of the best achievable optimistic gap by more than d. On such a round, witness probing still observes a pair, but this pair need not be one whose uncertainty prevents the pessimistic test from succeeding. All other rounds without a declaration remain within the logarithmic budget, irrespective of the method used to compute the candidates.

No uniform approximation guarantee. Corollary G.1 reduces the question to how often SOLVE returns a candidate with optimization error above d. Theorem D.1 does not by itself yield logarithmic bounds for SOLVE, because the analysis does not control $H_d(T)$. In particular, it does not establish that the local search always returns a candidate within a fixed additive distance of the optimum. We next show that no such uniform additive guarantee holds for the max-min problem when the utility scale is unrestricted. The example also applies to confidence-bound matrices of the form used in the optimistic step.

Lemma G.1 (A failure of the local search)**.** *Let $n = 4$, $k = 2$, and let $\mathbf{v}$ be the symmetric utility profile*

$$\mathbf{v} = \begin{pmatrix} 0 & -4 & -2 & 1 \\ -4 & 0 & 1 & 4 \\ -2 & 1 & 0 & -4 \\ 1 & 4 & -4 & 0 \end{pmatrix}.$$

Let $\pi^\star = \{\{1\},\{2,4\},\{3\}\}$ and $\widehat{\pi} = \{\{1,4\},\{2,3\}\}$. Then,

$$\max_{\pi\in\Pi} \mathrm{Gap}_2(\pi,\mathbf{v}) = \mathrm{Gap}_2(\pi^\star,\mathbf{v}) = 2, \qquad \mathrm{Gap}_2(\widehat{\pi},\mathbf{v}) = -3.$$

Moreover, fix $\lambda > 0$, let $\eta := \lambda/10$, and let $\mathbf{x},\mathbf{y}$ satisfy

$$|x_{i,j} - \lambda v_{i,j}| \le \eta, \qquad |y_{i,j} - \lambda v_{i,j}| \le \eta$$

for every $i \neq j$, where $\mathbf{x}$ is symmetric. Then, SOLVE *with $R = 3$ restarts, applied to $(\mathbf{x},\mathbf{y})$, returns $\widehat{\pi}$ with probability at least p^3, where*

$$p := \frac{1}{4}\sum_{K=2}^{4} \frac{K(K-1)}{K^4} > 0.0614.$$

On this event,

$$\max_{\pi\in\Pi} \widetilde{\mathrm{Gap}}_2(\pi,\mathbf{x},\mathbf{y}) - \widetilde{\mathrm{Gap}}_2(\widehat{\pi},\mathbf{x},\mathbf{y}) \ge 5\lambda - 10\eta = 4\lambda.$$

Proof. Exact gaps. Under $\pi^\star$, the agents' utilities are $(0,4,0,4)$. A unilateral deviation by agent 1 can only join $\{2,4\}$ or $\{3\}$, obtaining utility -3 or -2, respectively. The same holds symmetrically for agent 3. Agents 2 and 4, if they deviate individually, must leave their mutual coalition; their utility in any available destination is at most 1, compared with their current utility of 4.

For a two-agent deviation involving agent 2 or 4, at least one of these agents must lose at least 3: if only one of them deviates, that agent cannot remain with its partner, while if both deviate, any reassignment either separates them or places at least one of them in a coalition with utility at most 1. The remaining pair $\{1,3\}$ can form the coalition $\{1,3\}$, giving each member utility -2 instead of 0. Their other admissible joint reassignment is to join $\{2,4\}$, which gives both utility -5. Thus, every deviation of at most two agents leaves some deviator at least 2 worse off, and the deviation in which agents 1 and 3 form $\{1,3\}$ attains this bound. Hence, $\mathrm{Gap}_2(\pi^\star,\mathbf{v}) = 2$.

To verify that this is the maximum over all partitions, we enumerate all 15 partitions of the four agents and, for each, the complete family of admissible deviations by one or two agents. The resulting values of Gap_2 are listed below; in particular, the largest is 2, attained by $\pi^\star$.

π	$\mathrm{Gap}_2(\pi,\mathbf{v})$	π	$\mathrm{Gap}_2(\pi,\mathbf{v})$
$\{\{1\},\{2\},\{3\},\{4\}\}$	-4	$\{\{1\},\{2\},\{3,4\}\}$	-8
$\{\{1\},\{2,3\},\{4\}\}$	-3	$\{\{1\},\{2,3,4\}\}$	-3
$\{\{1\},\{2,4\},\{3\}\}$	2	$\{\{1,2\},\{3\},\{4\}\}$	-8
$\{\{1,2\},\{3,4\}\}$	-9	$\{\{1,2,3\},\{4\}\}$	-7
$\{\{1,2,4\},\{3\}\}$	-3	$\{\{1,3\},\{2\},\{4\}\}$	-4
$\{\{1,3\},\{2,4\}\}$	-2	$\{\{1,3,4\},\{2\}\}$	-7
$\{\{1,4\},\{2\},\{3\}\}$	-3	$\{\{1,4\},\{2,3\}\}$	-3
$\{\{1,2,3,4\}\}$	-5		

Under $\widehat{\pi}$, every agent has utility 1. Agents 1 and 3 obtain at most 1 in any coalition, since their only positive pairwise utility is 1. Thus, any deviation involving either of them leaves at least one deviator with a gain of at most 0.

It remains to consider a joint deviation by agents 2 and 4. If they form the coalition $\{2, 4\}$, both obtain utility 4, gaining 3 each. If they move together into a coalition also containing agent 1 or agent 3, their utilities are $(0, 5)$ or $(5, 0)$, respectively; if they join both, each obtains utility 1. If they move to different coalitions, each can obtain at most 1 by joining one of the remaining singleton coalitions. Therefore, no deviation gives every deviator a gain greater than 3. Since forming $\{2, 4\}$ attains a gain of 3 for both, we obtain $\mathrm{Gap}_2(\widehat{\pi}, \mathbf{v}) = -3$.

Perturbation. Each term $u_i(\pi(i), \mathbf{x}) - u_i(\pi'(i), \mathbf{y})$ involves at most

$$(|\pi(i)| - 1) + (|\pi'(i)| - 1) \leq 2n - 3 = 5$$

pairwise utility entries, because the two coalitions are distinct, both contain i, and at most one can be the grand coalition. Each entry differs from its value under $\lambda\mathbf{v}$ by at most η. Hence, the perturbation of every such term is at most 5η. Taking the maximum over deviators and the minimum over the finite family of admissible deviations preserves this bound. Consequently,

$$\left|\widetilde{\mathrm{Gap}}_2(\pi, \mathbf{x}, \mathbf{y}) - \lambda \mathrm{Gap}_2(\pi, \mathbf{v})\right| \leq 5\eta$$

for every partition π. In particular, using the maximum value 2 and the gap -3 of $\widehat{\pi}$, we obtain

$$\begin{aligned}\max_{\pi \in \Pi} \widetilde{\mathrm{Gap}}_2(\pi, \mathbf{x}, \mathbf{y}) - \widetilde{\mathrm{Gap}}_2(\widehat{\pi}, \mathbf{x}, \mathbf{y}) \\ \geq (2\lambda - 5\eta) - (-3\lambda + 5\eta) \\ = 5\lambda - 10\eta = 4\lambda.\end{aligned}$$

The local search stays at $\widehat{\pi}$. For symmetric $\mathbf{x}$, moving a single agent changes the welfare $\mathrm{sw}(\cdot, \mathbf{x})$ by twice the change in that agent's utility. Under $\mathbf{v}$, every unilateral move from $\widehat{\pi}$ decreases the mover's utility. Moving to a new singleton decreases it by 1 for every agent. Moving to the other coalition decreases it by 7, 1, 7, and 1 for agents $1, 2, 3, 4$, respectively: agent 1 obtains $v_{1,2} + v_{1,3} = -6$, agent 2 obtains $v_{2,1} + v_{2,4} = 0$, agent 3 obtains $v_{3,1} + v_{3,4} = -6$, and agent 4 obtains $v_{4,2} + v_{4,3} = 0$, in each case instead of 1.

Under $\mathbf{x}$, each of these utility changes differs from λ times its value under $\mathbf{v}$ by at most 3η, since it involves one interaction in the current coalition and at most two in the destination coalition. Thus, every unilateral utility change is at most

$$-\lambda + 3\eta = -\frac{7\lambda}{10} < 0.$$

The corresponding welfare change is therefore at most $2(-\lambda + 3\eta) < 0$. Since SOLVE accepts a move only when it increases welfare by more than 10^{-9}, no agent moves from $\widehat{\pi}$. A restart initialized at $\widehat{\pi}$ consequently terminates after its first sweep, rather than reaching the sweep cap, and returns $\widehat{\pi}$.

A restart draws K uniformly from $\{1, \ldots, 4\}$ and assigns each of the four agents an independent uniform label from $\{0, \ldots, K - 1\}$. For a fixed K, the labels induce $\widehat{\pi}$ precisely when agents $1, 4$ receive one common label and agents $2, 3$ receive a different common label. There are $K(K - 1)$ such assignments among the K^4 possible assignments. Thus, the probability that a restart begins at $\widehat{\pi}$ is

$$p = \frac{1}{4} \sum_{K=2}^{4} \frac{K(K-1)}{K^4} > 0.0614.$$

The three restarts are independent, so with probability at least p^3 all three begin at $\widehat{\pi}$ and terminate there.

Selection. On this event, the terminal candidates consist of $\widehat{\pi}$ (possibly returned by all three restarts), the singleton partition, and the grand coalition. By the exhaustive enumeration above, the singleton and grand-coalition gaps are -4 and -5, respectively. The perturbation bound gives

$$\begin{aligned}\widetilde{\mathrm{Gap}}_2(\widehat{\pi}, \mathbf{x}, \mathbf{y}) &\geq -3\lambda - 5\eta, \\ \widetilde{\mathrm{Gap}}_2(\{\{1\}, \{2\}, \{3\}, \{4\}\}, \mathbf{x}, \mathbf{y}) &\leq -4\lambda + 5\eta, \\ \widetilde{\mathrm{Gap}}_2(\{N\}, \mathbf{x}, \mathbf{y}) &\leq -5\lambda + 5\eta.\end{aligned}$$

Since $\eta = \lambda/10$, the gap score of $\widehat{\pi}$ is at least that of the singleton partition and exceeds that of the grand coalition by at least

$$2\lambda - 10\eta = \lambda.$$

The welfare tie-break also favours $\widehat{\pi}$. In fact,

$$\mathrm{sw}(\widehat{\pi}, \mathbf{x}) \geq 4\lambda - 4\eta > 0, \qquad \mathrm{sw}(\{\{1\},\{2\},\{3\},\{4\}\}, \mathbf{x}) = 0,$$

whereas

$$\mathrm{sw}(\{N\}, \mathbf{x}) \leq -8\lambda + 12\eta < 0.$$

Thus, the 10^{-6}-weighted welfare term resolves a possible tie with the singleton partition in favour of $\widehat{\pi}$, and cannot overturn its advantage over the grand coalition. Hence, $\widehat{\pi}$ has the highest score among all candidates, and SOLVE returns it. □

Consequences for the failure. Consider the scaled instance $\lambda\mathbf{v}$ with stability margin ε. Its certification margin is

$$\Delta_\star = \max_{\pi \in \Pi} \mathrm{Gap}_2(\pi, \lambda\mathbf{v}) - \varepsilon = 2\lambda - \varepsilon,$$

which is positive if and only if $2\lambda > \varepsilon$. At a round t on $\mathcal{G}_t$ for which

$$2 \max_{i,j} b_{i,j}^{t-1} \leq \frac{\lambda}{10},$$

the optimistic and pessimistic estimates satisfy

$$|\overline{v}_{i,j}^{t-1} - \lambda v_{i,j}| \leq \eta, \qquad |\underline{v}_{i,j}^{t-1} - \lambda v_{i,j}| \leq \eta.$$

Thus, the lemma applies with $(\mathbf{x}, \mathbf{y}) = (\overline{\mathbf{v}}^{t-1}, \underline{\mathbf{v}}^{t-1})$. With probability at least p^3, the optimistic candidate is then $\widehat{\pi}$, for which

$$G_t^+(\widehat{\pi}) \leq -3\lambda + 5\eta = -\frac{5\lambda}{2}.$$

The resulting shortfall consequently satisfies

$$s_t = \left(\varepsilon + \Delta_\star - G_t^+(\overline{\pi}^t)\right)^+ \geq 2\lambda - \left(-3\lambda + \frac{\lambda}{2}\right) = \frac{9\lambda}{2}.$$

Thus, on these rounds, the optimization error and the shortfall are both large. Since λ can be arbitrarily large, no finite uniform additive approximation guarantee for SOLVE is possible over utility profiles of unrestricted scale.

This example does not establish the same unbounded scaling over the bounded true-utility class used in our experiments, where the entries of the true profiles lie in $[-100, 100]$. However, the Gaussian observation noise has unbounded support, so the estimated profiles passed to SOLVE need not remain in a deterministically bounded range even when the true utilities are bounded. We therefore make no uniform additive-approximation claim for the implemented heuristic on those random inputs either.

The non-existence guarantee does not depend on the optimizer. The pathwise bound (11) and its expected forms hold for the implemented heuristic without modification. Logarithmic existence-error and regret bounds follow when the optimistic step is solved exactly, or more generally whenever $\mathbb{E}[H_d(T)]$ is logarithmic for some $d < \Delta_\star$. For SOLVE, the present analysis does not control this term.

H Computing the Max-Min Candidates

For ease of exposition, Section 3 assumes an oracle that solves the max-min problems (7) and (9) exactly. This appendix discusses what such an oracle involves, and why the learner does not need one for its soundness.

Difficulty of exact optimization. Both problems maximize, over all partitions of N, a minimum over all deviations of at most k agents. The number of partitions grows faster than exponentially in n, and the problems are hard in general. For $k = 1$ and equal current and deviation profiles, every maximizer is a Nash stable partition, since symmetric ASHGs always admit one [19]; an exact oracle would therefore compute Nash stable partitions, which is PLS-complete [50]. The bound k on the size of deviating groups does not remove this difficulty, but it keeps the inner minimum tractable. For a fixed partition π with K coalitions, the deviation family of a group of s agents has at most $\binom{n}{s}(K+s)^s$ members, so $\widetilde{\mathrm{Gap}}_k(\pi,\cdot,\cdot)$ can be evaluated exactly in time polynomial in n for fixed k. This is what our deviation checker does (Appendix I).

What the guarantees require. The learner uses the oracle in two different ways, with different requirements.

- *The pessimistic test* must be evaluated exactly, over the complete deviation family. It is a statement about a given partition, not an optimization problem, and it is what makes declarations sound. Theorem 3.1, and the fact that a declaration made when all confidence intervals hold incurs no regret, are valid however the candidates are computed.
- *The optimistic candidate* determines which pair is probed. Theorem D.1 and Corollary G.1 (Appendix G) hold for any method of computing it. Relative to exact optimization, they add the expected number of witness rounds whose optimization error δ_t exceeds some $d < \Delta_\star$. In particular, a method whose additive error never exceeds d preserves the logarithmic bounds, with $\Delta_\star$ replaced by $\Delta_\star - d$. A multiplicative approximation guarantee does not suffice, since the gap can be negative and the bounds depend on its distance from $\varepsilon + \Delta_\star$.

Approximate optimization therefore affects only the existence-error and regret bounds, and only through the rounds on which the optimistic candidate is far from optimal.

The method used in our experiments. We compute both candidates with the local-search heuristic SOLVE (Appendix I), which rescores its local optima with the exact checker. It is shared by all algorithms, with the same restarts, number of sweeps and tie-breaking. No approximation guarantee is known for it, and Lemma G.1 shows that it admits no uniform additive guarantee once the utility scale is unrestricted; our analysis does not bound how often its error exceeds d. Its cost per call and its share of the running time are reported in Appendix O. For small n, exact optimization by enumerating all partitions is possible; we use it only to label instances for evaluation (Appendix I), not inside the learners. Exact or provably near-optimal optimization for larger instances, for example by integer programming over deviation templates, would remove the additional term in the bounds, and we leave it to future work.

I More Implementation Details

We provide the implementation details needed to reproduce the experiments.

Utility generation. Following Boehmer et al. [18], we consider two models for generating random utility profiles:

(1) ***Uniform model:*** For every pair of distinct agents $i, j \in N$, we sample the true mutual utility $v_{i,j}$ uniformly from the integers in $[-100, 100]$.

(2) ***Gaussian model:*** For every pair of distinct agents $i, j \in N$, we first sample a base utility $\mu_{i,j}$ uniformly from the integers in $[-100, 100]$. We then sample $v_{i,j}$ from a Gaussian distribution with mean $\mu_{i,j}$ and standard deviation 10, rounding the result to an integer.

In both models, $v_{i,i} = 0$ for every agent i.

Confidence bounds and estimator. For each unordered pair $\{i, j\}$, we maintain its empirical mean $\mu_{i,j}$ and the number $T_{i,j}$ of observations collected before the current round. Every round in which i and j belong to the same coalition contributes one observation. At round t, the confidence half-width

is

$$b_{i,j}^t = \sqrt{\frac{2\sigma^2 \beta \log \max(t,2)}{T_{i,j}^{t-1}}}, \tag{13}$$

which agrees with the radius used by UCB-LCB-GS in Section 3 for every $t \geq 2$.

In the analysis, an unobserved pair has an infinite confidence radius. In the implementation, we instead assign it the finite radius $B_0 = 10^3$. This exceeds the absolute value of every utility in $[-100, 100]$ by a wide margin and dominates any plausible empirical mean. Consequently, optimistic and UCB-based searches tend to place previously unobserved pairs in the same coalition early on, while no partition involving such a pair passes a pessimistic test. UCB-LCB-GS observes every pair in round 1, so B_0 has no further role for this algorithm. The baselines, however, may leave some pairs unobserved for longer. Diagonal entries are fixed to zero. In the robustness experiments with asymmetric utilities, we maintain a separate mean and observation count for each ordered pair.

Three computational tasks. The experiments involve three distinct computational tasks:

1. *Finding a candidate partition.* The max-min problems in (7) and (9) are solved approximately using the heuristic SOLVE, described below. This heuristic carries no optimality guarantee (Appendix G).
2. *Evaluating the gap of a given partition.* The quantity $\widetilde{\mathrm{Gap}}_k(\pi, \mathbf{x}, \mathbf{y})$ is evaluated exactly. This computation is used in the pessimistic test, when rescoring candidate partitions, and when computing the regret $\mathrm{Gap}_k(\pi^t, \mathbf{v})$. It is performed by the exact deviation checker described below.
3. *Determining whether a stable partition exists.* This requires computing $\max_\pi \mathrm{Gap}_k(\pi, \mathbf{v})$ over all partitions. We use this computation only to label instances for evaluation, never within the learning algorithms, and obtain it by exhaustive enumeration where feasible.

The partition heuristic SOLVE. Every algorithm uses the same heuristic, $\textsc{Solve}(\mathbf{v}_{\mathrm{curr}}, \mathbf{v}_{\mathrm{dev}})$, to approximately solve

$$\arg\max_{\pi} \min_{C \subseteq N : |C| \leq k} \min_{\pi' \in \Lambda_C(\pi)} \widetilde{\mathrm{Gap}}_C(\pi, \pi', \mathbf{v}_{\mathrm{curr}}, \mathbf{v}_{\mathrm{dev}}). \tag{14}$$

The algorithms differ only in the pair $(\mathbf{v}_{\mathrm{curr}}, \mathbf{v}_{\mathrm{dev}})$ supplied to the heuristic:

- The pessimistic step uses $(\underline{\mathbf{v}}, \overline{\mathbf{v}})$, whereas the optimistic step uses $(\overline{\mathbf{v}}, \underline{\mathbf{v}})$.
- UCB-ONLY uses $(\overline{\mathbf{v}}, \overline{\mathbf{v}})$, whereas LCB-ONLY uses $(\underline{\mathbf{v}}, \underline{\mathbf{v}})$.

The heuristic runs R independent random restarts of a best-improvement local search on the welfare

$$\mathrm{sw}(\pi, \mathbf{v}_{\mathrm{curr}}) = \sum_{i \in N} u_i(\pi(i), \mathbf{v}_{\mathrm{curr}}).$$

Each restart proceeds as follows:

1. Sample $K \sim \mathrm{Unif}\{1, \ldots, n\}$, and assign each agent independently and uniformly to one of the K labels.
2. Perform up to M sweeps. At the beginning of each sweep, draw a uniformly random ordering of the agents. For each agent in turn, evaluate the welfare obtained by moving her to each existing coalition or to a new coalition, and perform the best move if it increases welfare by more than 10^{-9}. Among moves with equal welfare, the agent's current coalition is preferred, followed by the coalition with the lowest index.
3. Terminate the restart when a complete sweep makes no move, or when M sweeps have been performed.

The R terminal partitions are considered together with the singleton partition and the grand coalition. We score all candidates using their exact gap $\widetilde{\mathrm{Gap}}_k(\pi, \mathbf{v}_{\mathrm{curr}}, \mathbf{v}_{\mathrm{dev}})$, with a 10^{-6}-weighted welfare term to break ties, and return the candidate with the highest score. Throughout the experiments, we use $R = 3$ and $M = 60$.

For symmetric $\mathbf{v}_{\mathrm{curr}}$, a restart that terminates because no agent moves returns a partition that is Nash stable for $\mathbf{v}_{\mathrm{curr}}$ up to the specified tolerance. This structural property does not, however, imply an approximation guarantee for the max-min objective (Lemma G.1).

Exact deviation checker. Given a partition π, the checker enumerates the complete deviation family $\Lambda_C(\pi)$ of Definition 2 for every coalition C with $1 \leq |C| \leq k$.

For each deviation, every member of C is assigned either to one of the coalitions remaining after C leaves or to a new coalition. Different members may choose different existing coalitions or form different new coalitions. New coalitions are labelled canonically to avoid duplicate representations. Assignments in which a member remains in the same coalition as before are discarded.

For every remaining deviation π', the checker evaluates $\widetilde{\mathrm{Gap}}_C(\pi, \pi', \mathbf{v}_{\mathrm{curr}}, \mathbf{v}_{\mathrm{dev}})$ and returns the minimum over all admissible C and π', together with a minimizing pair (C, π'). This minimizing pair is used by witness probing (line 10 of Algorithm 1); ties are resolved according to a fixed enumeration order.

The checker supports $k \leq 3$, covering all configurations in our experiments. We validated its output against a brute-force implementation of Definition 2 on randomly generated instances with $n \leq 5$ and $k \leq 3$. The same checker is used for the pessimistic test, for rescoring candidates within SOLVE, and for evaluating regret.

Using only a restricted family of deviations can lead the checker to declare an unstable partition stable. For example, let

$$n = 12, \qquad \pi = \{\{1, 2, 3\}, \{4, 5, 6\}, \{7\}, \ldots, \{12\}\}.$$

A swap between agents 1 and 4 can be profitable for both, but may be missed by a checker that independently reassigns the members of C only when the number of possible reassignments is below a fixed cap, and otherwise moves them only as a block.

We reuse results obtained with such a restricted checker only in configurations for which it is provably guaranteed to attain the same minimum as the complete checker. This holds for $k \leq 2$ when the partition contains at most seven coalitions, and therefore for all reused configurations with $n \leq 7$ and $k \leq 2$. All configurations with $k = 3$ were rerun using the exact checker.

Exact existence labels. We classify instances according to their existence status, computed from

$$g^\star := \max_{\pi} \mathrm{Gap}_k(\pi, \mathbf{v}),$$

which is obtained by enumerating all partitions using the exact checker:

- *YES* if $g^\star \geq \varepsilon$, equivalently $\mathtt{SNS}_{\varepsilon,k}(\mathbf{v}) \neq \emptyset$;
- *NO* if $g^\star < 0$, equivalently $\mathtt{SNS}_{0,k}(\mathbf{v}) = \emptyset$;
- *Intermediate* otherwise.

Following the definitions in (3) and (4), we average existence error and regret over YES instances, and non-existence error over NO instances. Intermediate instances contain an exactly k-strongly Nash stable partition, but no (ε, k)-strongly Nash stable partition. Neither error metric nor regret is evaluated on these instances; instead, we report separately the fraction of rounds in which each algorithm declares existence. No experiment in the paper relies on heuristic existence labels.

Utilities in both synthetic models are integers, so all gaps are also integers. Since $\varepsilon = 1$, the certification margin $\Delta_\star = g^\star - \varepsilon$ is integer-valued as well. YES instances with $\Delta_\star = 0$ do occur. On such instances, any learner that declares existence only through a pessimistic test makes a linear number of existence errors (Proposition E.2). We therefore report results by certification margin where relevant.

Random fallback partition. Whenever a baseline sets $d_t = 0$, it forms a random partition π_{rand}. To generate this partition, each agent is assigned independently and uniformly to one of $M_{\mathrm{rand}} = 4$ coalitions, after which empty coalitions are removed. The same value of M_{rand} is used for all baselines and is never tuned separately for an individual algorithm.

Round protocol and fairness controls. At each round t, the algorithms proceed as follows:

(i) Refresh $\overline{\mathbf{v}}^{t-1}$ and $\underline{\mathbf{v}}^{t-1}$ using the shared estimator.

(ii) Call SOLVE with the pair $(\mathbf{v}_{\mathrm{curr}}, \mathbf{v}_{\mathrm{dev}})$ prescribed for the algorithm.

(iii) Evaluate the stability test using the exact deviation checker. For UCB-LCB-GS, this involves testing $\underline{\pi}^t$ pessimistically and, if that test fails, testing $\overline{\pi}^t$ pessimistically before selecting a witness pair.

(iv) Commit to the partition π^t and declaration d_t.

(v) Form π^t and observe one value $V_{i,j}^t = v_{i,j} + \varepsilon_{i,j}^t$ for every pair sharing a coalition.

(vi) Increment $T_{i,j}$ and update $\mu_{i,j}$ using the running-mean recursion

$$\mu_{i,j} \leftarrow \mu_{i,j} + \frac{V_{i,j}^t - \mu_{i,j}}{T_{i,j}}.$$

To ensure a fair comparison, all algorithms use (a) the same estimator and confidence radius (13), (b) the same SOLVE implementation and the same values of (R, M), (c) the same exact deviation checker, (d) the same random fallback with M_{rand} coalitions, and (e) the same seeds.

Within each configuration, seed $s \in \{0, \ldots, 99\}$ determines the instance and noise sequence through one random generator. A second generator, seeded with $s + 10^4$, determines the algorithm's internal randomness, including the restarts of SOLVE and the random fallback. Noise values are drawn pair by pair in a fixed order. Thus, algorithms run with the same seed use the same utility matrix $\mathbf{v}$ and the same noise values for pairs that are co-located in both runs. They also use identical optimizer initializations whenever they reach the same restart index. This controls the sources of randomness and isolates the effect of the estimates used by each algorithm.

Hyperparameter summary. The common hyperparameters are $R = 3$ optimizer restarts, $M = 60$ local-search sweeps, $M_{\mathrm{rand}} = 4$ fallback coalitions, and an unseen-pair radius of $B_0 = 10^3$. The running-mean estimator is initialized with $\mu_{i,j} = 0$ and $T_{i,j} = 0$. None of these hyperparameters is tuned separately for individual algorithms.

Reproducibility, persistence, and plotting. Each trial is uniquely determined by its configuration and seed. We store the configuration, the exact existence label of the instance, and the per-round record of declarations d_t, regret $\max\{0, \varepsilon - \mathrm{Gap}_k(\pi^t, \mathbf{v})\}$, and round type (declaration, witness probe, or other). All figures and tables are regenerated from these stored records, without rerunning the experiments. Each algorithm is represented by the same colour and marker throughout the figures.

Standard errors and error bars. For a metric evaluated on a particular class of instances (YES for regret and existence error, and NO for non-existence error), let $X^{(1)}, \ldots, X^{(m)}$ denote its values on the m instances in that class and configuration. We report the sample mean

$$\bar{X} = \frac{1}{m} \sum_{s=1}^{m} X^{(s)}$$

and the standard error of the mean

$$\widehat{\mathrm{SEM}}(\bar{X}) = \frac{1}{\sqrt{m}} \left(\frac{1}{m-1} \sum_{s=1}^{m} \big(X^{(s)} - \bar{X}\big)^2 \right)^{1/2}.$$

The shaded bands and error bars in the figures show ± 1 standard error. This calculation does not assume that the underlying observations follow a normal distribution. The variation captured by these standard errors reflects the random instance, the noise sequence, and the algorithm's internal randomness, all of which are determined by the seed. We report the number m of instances in each class alongside the results. Comparisons between algorithms use the same instances.

Hardware. All experiments were run on a single Apple MacBook Pro equipped with an Apple M4 Max processor (16 CPU cores: 12 performance and 4 efficiency) and 128 GB of unified memory. Trials were distributed across 11 independent, single-threaded processes. The GPU and Neural Engine were not used.

Software stack. The experiments use macOS (arm64), Python 3.9.6 (PSF License), NumPy 2.0.2 (BSD-3-Clause License), and Matplotlib 3.9.4 (Matplotlib License, a BSD-compatible permissive license). No other third-party packages are required. The simulator, estimator, solver, deviation checker, existence labelling, algorithms, experiment runner, and plotting code rely only on NumPy, Matplotlib, and the Python standard library. All three dependencies are used in unmodified form and in accordance with their respective licences.

Compute resources. Resident memory remains below 40 MB per process. The dominant computational cost is the exact evaluation of gaps. Each evaluation enumerates all coalitions of at most k agents and all their admissible deviations. In addition to its local search, each call to SOLVE performs $R + 2$ such evaluations.

Appendix O reports five types of measurements:

- end-to-end runtime for each experiment;
- the learner's runtime per round;
- runtime of the local search alone;
- runtime of a single exact evaluation;
- measurements for larger values of n and k.

All reported measurements were obtained on the machine described above; none is extrapolated. The synthetic experiments required 5.9 CPU-hours, corresponding to 32 minutes of wall-clock time using 11 processes. Since trials are independent and can be distributed across processes, wall-clock time decreases nearly linearly with the number of processes used. We performed no additional hyperparameter search beyond the reported experiments.

Code, data, and licensing. The code and experimental results are available at `https://github.com/coml1149-saarco/online-group-stable-coalitions` under the MIT and CC BY 4.0 licences, respectively, subject to the third-party data exclusions described in the repository. The released code covers the environment, estimator, SOLVE, exact deviation checker, existence-labelling procedure, algorithms, experiment runners, and plotting code. It also includes the per-trial result files and a single command for regenerating every figure from the reported seeds. The synthetic experiments use only data generated by the models described above. The datasets used in Appendix M are retrieved by the code from their cited sources and are not redistributed.

I.1 Limitations of the Empirical Study

We discuss several limitations of the empirical study to clarify the scope of our findings.

Heuristic partition search. The theoretical results in Section 3 assume exact solutions to the optimistic optimization problem, whereas the experiments use the heuristic SOLVE. The non-existence guarantee holds regardless of the method used to compute candidate partitions. For the existence-error and regret guarantees, applying the analysis to the heuristic introduces the additional term in Theorem D.1, which counts the rounds on which its optimistic candidate falls short of the optimum. We do not claim that replacing SOLVE with an exact optimizer would improve every reported empirical result. Although exact optimization would meet the assumption of the theoretical bounds, it would also change the trajectories of all learners, including the baselines, in ways that our analysis does not predict.

Cost of exact gap evaluation. Exact gap evaluation enumerates every coalition C with $|C| \leq k$ and every admissible assignment of its members. Its computational cost therefore grows combinatorially with n and k. Our exact checker supports $k \leq 3$, and the experiments consider up to $n = 12$ agents. Larger instances would require either restricting k or using a restricted deviation family, which may cause unstable partitions to be incorrectly declared stable.

Synthetic instances. The two generative models are standard random ASHG instances used in the related literature [18]. They need not capture all the structure of real coalition-formation problems, such as friendship networks in social settings or complementarities in team formation. The games in

Appendix M introduce some of this structure, but their utilities are constructed from distances in the underlying data rather than observed directly.

Instance classes. Our metrics evaluate existence error and regret only on YES instances, and non-existence error only on NO instances. Randomly generated instances rarely receive a NO label. Consequently, the non-existence error is estimated from relatively few instances in each configuration; we report these counts alongside the results.

J Additional Details on Ablated Baselines

J.1 Description of Baselines

All baselines use the same heuristic solver for (7) and (9), as well as the same exact deviation checker (Appendix I). They differ only in the estimates supplied to these two components. Throughout, π_{rand} denotes the random fallback partition described in Appendix I.

The learning baselines have no initialization round. Instead, pairs that have not yet been observed are assigned the large radius B_0 (Appendix I). As a result, optimism-based searches tend to form large coalitions in the first round, whereas no partition passes a pessimistic test.

J.1.1 Pessimistic-Only (Algorithm 3)

At each round t, the algorithm computes

$$\underline{\pi}^t \in \arg\max_{\pi} \min_{C \subseteq N:\, |C| \le k} \min_{\pi' \in \Lambda_C(\pi)} \operatorname*{Gap}_C\big(\pi, \pi', \underline{\mathbf{v}}^t, \overline{\mathbf{v}}^t\big). \tag{15}$$

This is the pessimistic counterpart of (7): it uses lower confidence bounds for agents' utilities in their current coalitions and upper confidence bounds for their utilities after a deviation. If $\underline{\pi}^t$ is (ε, k)-strongly Nash stable under $(\underline{\mathbf{v}}^t, \overline{\mathbf{v}}^t)$, the algorithm sets $d_t = 1$ and forms $\pi^t = \underline{\pi}^t$. Otherwise, it sets $d_t = 0$ and forms π_{rand}. This baseline has no optimistic exploration step.

J.1.2 Optimistic-Only (Algorithm 4)

At each round t, the algorithm computes the optimistic partition

$$\overline{\pi}^t \in \arg\max_{\pi} \min_{C \subseteq N:\, |C| \le k} \min_{\pi' \in \Lambda_C(\pi)} \operatorname*{Gap}_C\big(\pi, \pi', \overline{\mathbf{v}}^t, \underline{\mathbf{v}}^t\big). \tag{16}$$

This uses upper confidence bounds for agents' utilities in their current coalitions and lower confidence bounds for their utilities after a deviation. If $\overline{\pi}^t$ is (ε, k)-strongly Nash stable under $(\overline{\mathbf{v}}^t, \underline{\mathbf{v}}^t)$, the algorithm sets $d_t = 1$ and forms $\pi^t = \overline{\pi}^t$. Otherwise, it sets $d_t = 0$ and forms π_{rand}.

J.1.3 UCB-Only (Algorithm 5)

This baseline uses upper confidence bounds in both arguments of Gap_C. It computes

$$\overline{\pi}^t \in \arg\max_{\pi} \min_{C \subseteq N:\, |C| \le k} \min_{\pi' \in \Lambda_C(\pi)} \operatorname*{Gap}_C\big(\pi, \pi', \overline{\mathbf{v}}^t, \overline{\mathbf{v}}^t\big),$$

and tests stability under $(\overline{\mathbf{v}}^t, \overline{\mathbf{v}}^t)$. If the resulting partition is (ε, k)-strongly Nash stable, it sets $d_t = 1$ and forms $\overline{\pi}^t$. Otherwise, it sets $d_t = 0$ and forms π_{rand}. Thus, UCB-ONLY represents the classical optimism-in-the-face-of-uncertainty heuristic, without using pessimistic estimates.

J.1.4 LCB-Only (Algorithm 6)

The corresponding baseline uses lower confidence bounds in both arguments of Gap_C:

$$\underline{\pi}^t \in \arg\max_{\pi} \min_{C \subseteq N:\, |C| \le k} \min_{\pi' \in \Lambda_C(\pi)} \operatorname*{Gap}_C\big(\pi, \pi', \underline{\mathbf{v}}^t, \underline{\mathbf{v}}^t\big).$$

It tests stability under $(\underline{\mathbf{v}}^t, \underline{\mathbf{v}}^t)$ and forms the resulting partition if the test succeeds; otherwise, it forms π_{rand}.

Using lower confidence bounds in both arguments is neither pessimistic nor optimistic. The confidence widths of the utilities in the current coalition and those after a deviation largely cancel, so the resulting test is not conservative. Empirically, LCB-ONLY declares existence on almost every round, including on instances without a stable partition (Table 2).

J.1.5 Mixed (Algorithm 7)

The MIXED baseline first computes the pessimistic candidate from (15). If $\underline{\pi}^t$ is (ε, k)-strongly Nash stable under $(\underline{\mathbf{v}}^t, \overline{\mathbf{v}}^t)$, it sets $d_t = 1$ and forms $\underline{\pi}^t$. Otherwise, it computes the optimistic candidate from (16). If $\overline{\pi}^t$ is (ε, k)-strongly Nash stable under $(\overline{\mathbf{v}}^t, \underline{\mathbf{v}}^t)$, it sets $d_t = 1$ and forms $\overline{\pi}^t$. If neither test succeeds, it sets $d_t = 0$ and forms π_{rand}.

Unlike UCB-LCB-GS, MIXED does not use optimism for active exploration after the pessimistic test fails. Instead, it uses the optimistic candidate only as a secondary stability test.

Relation to UCB-LCB-GS. The MIXED baseline and UCB-LCB-GS have a similar structure: both compute pessimistic and optimistic candidates, and both first apply the pessimistic test to the pessimistic candidate. They therefore agree whenever this first test succeeds. Their behaviour differs, however, when it fails, in two ways that affect the soundness and regret analyses. Consequently, the guarantees for UCB-LCB-GS do not transfer directly to MIXED.

(a) *The second test.* UCB-LCB-GS applies the pessimistic test to its optimistic candidate, whereas MIXED applies the optimistic test and declares existence if it succeeds. Passing the optimistic test does not establish stability under the true utilities: $\overline{\mathbf{v}}^t$ raises the estimated utilities in the current coalition, while $\underline{\mathbf{v}}^t$ lowers those after a deviation. Thus, a partition whose gap is at least ε under $(\overline{\mathbf{v}}^t, \underline{\mathbf{v}}^t)$ need not be (ε, k)-strongly Nash stable under the true utilities. The soundness guarantee of Theorem 3.1 relies on the pessimistic test and therefore does not extend to MIXED.

(b) *Exploration.* When neither test succeeds, MIXED forms π_{rand}. UCB-LCB-GS instead forms the optimistic candidate or its most damaging deviation, ensuring that the selected witness pair, i.e., the most uncertain pair among those preventing the candidate from passing the pessimistic test, is observed (Section 3). The existence-error and regret bounds rely on this targeted observation (Lemma D.3). By forming a random partition, MIXED observes pairs without targeting the uncertainty responsible for the failed test; a logarithmic bound for this baseline would therefore require a separate analysis.

Both differences are reflected in the experiments. Regarding (a), MIXED declares existence on most rounds of NO and intermediate instances in the synthetic grid (Table 2), and on 1.1 to 3.3 rounds per NO instance under asymmetric utilities (Appendix N). Regarding (b), its regret exceeds that of UCB-LCB-GS in every configuration of the two grids except those with $\sigma = 20$, as well as in every long-horizon configuration.

At $\sigma = 20$, however, UCB-LCB-GS rarely establishes stability within 30 rounds and continues probing. In this regime, MIXED achieves lower regret by settling on partitions that pass the optimistic test (Appendix K.5). We therefore regard MIXED as the closest ablation of UCB-LCB-GS: it uses the same two candidates in the same order, but replaces the second pessimistic test with an optimistic one and witness probing with a random partition. Its guarantees should not be interpreted as matching those of UCB-LCB-GS.

J.2 Pseudo-codes

We collect the pseudo-code for the five learning baselines below. RANDOM sets $d_t = 0$ and forms π_{rand} in every round. All five learning baselines use the same estimator, partition optimizer, and deviation checker, and receive one noisy utility observation $V_{i,j}^t = v_{i,j} + \varepsilon_{i,j}^t$ for each pair of distinct agents sharing a coalition. Before a pair is first observed, its estimates use the radius B_0 (Appendix I).

Algorithm 3 PESSIMISTIC-ONLY baseline

Input: n agents; variance σ^2; $k \geq 1$; stability margin $\varepsilon \geq 0$; uncertainty parameter $\beta \in (2, \infty)$.

for each time $t = 1, \ldots, T$ **do**
  Compute $\underline{\pi}^t$ via (7).
  if $\widetilde{\mathrm{Gap}}_C(\underline{\pi}^t, \pi', \underline{\mathbf{v}}^{t-1}, \overline{\mathbf{v}}^{t-1}) \geq \varepsilon$ for all $C \subseteq N$, $|C| \leq k$, and all $\pi' \in \Lambda_C(\underline{\pi}^t)$ **then**
    Set $d_t = 1$ and form $\pi^t = \underline{\pi}^t$.
  else
    Set $d_t = 0$ and form $\pi^t = \pi_{\mathrm{rand}}$.
  For each pair of distinct agents i, j with $\pi^t(i) = \pi^t(j)$, observe $V_{i,j}^t = v_{i,j} + \varepsilon_{i,j}^t$.
  Update the UCBs and LCBs via (5).

Algorithm 4 OPTIMISTIC-ONLY baseline

Input: n agents; variance σ^2; $k \geq 1$; stability margin $\varepsilon \geq 0$; uncertainty parameter $\beta \in (2, \infty)$.

for each time $t = 1, \ldots, T$ **do**
  Compute $\overline{\pi}^t$ via (9).
  if $\widetilde{\mathrm{Gap}}_C(\overline{\pi}^t, \pi', \overline{\mathbf{v}}^{t-1}, \underline{\mathbf{v}}^{t-1}) \geq \varepsilon$ for all $C \subseteq N$, $|C| \leq k$, and all $\pi' \in \Lambda_C(\overline{\pi}^t)$ **then**
    Set $d_t = 1$ and form $\pi^t = \overline{\pi}^t$.
  else
    Set $d_t = 0$ and form $\pi^t = \pi_{\mathrm{rand}}$.
  For each pair of distinct agents i, j with $\pi^t(i) = \pi^t(j)$, observe $V_{i,j}^t = v_{i,j} + \varepsilon_{i,j}^t$.
  Update the UCBs and LCBs via (5).

Algorithm 5 UCB-ONLY baseline

Input: n agents; variance σ^2; $k \geq 1$; stability margin $\varepsilon \geq 0$; uncertainty parameter $\beta \in (2, \infty)$.

for each time $t = 1, \ldots, T$ **do**
  Compute $\overline{\pi}^t$ via

$$\arg\max_{\pi} \min_{\substack{C \subseteq N \\ |C| \leq k}} \min_{\pi' \in \Lambda_C(\pi)} \mathrm{Gap}_C(\pi, \pi', \overline{\mathbf{v}}^{t-1}, \overline{\mathbf{v}}^{t-1}).$$

  if $\widetilde{\mathrm{Gap}}_C(\overline{\pi}^t, \pi', \overline{\mathbf{v}}^{t-1}, \overline{\mathbf{v}}^{t-1}) \geq \varepsilon$ for all $C \subseteq N$, $|C| \leq k$, and all $\pi' \in \Lambda_C(\overline{\pi}^t)$ **then**
    Set $d_t = 1$ and form $\pi^t = \overline{\pi}^t$.
  else
    Set $d_t = 0$ and form $\pi^t = \pi_{\mathrm{rand}}$.
  For each pair of distinct agents i, j with $\pi^t(i) = \pi^t(j)$, observe $V_{i,j}^t = v_{i,j} + \varepsilon_{i,j}^t$.
  Update the UCBs and LCBs via (5).

Algorithm 6 LCB-ONLY baseline

Input: n agents; variance σ^2; $k \geq 1$; stability margin $\varepsilon \geq 0$; uncertainty parameter $\beta \in (2, \infty)$.

for each time $t = 1, \ldots, T$ **do**
  Compute $\underline{\pi}^t$ via

$$\arg\max_{\pi} \min_{\substack{C \subseteq N \\ |C| \leq k}} \min_{\pi' \in \Lambda_C(\pi)} \mathrm{Gap}_C(\pi, \pi', \underline{\mathbf{v}}^{t-1}, \underline{\mathbf{v}}^{t-1}).$$

  if $\widetilde{\mathrm{Gap}}_C(\underline{\pi}^t, \pi', \underline{\mathbf{v}}^{t-1}, \underline{\mathbf{v}}^{t-1}) \geq \varepsilon$ for all $C \subseteq N$, $|C| \leq k$, and all $\pi' \in \Lambda_C(\underline{\pi}^t)$ **then**
    Set $d_t = 1$ and form $\pi^t = \underline{\pi}^t$.
  else
    Set $d_t = 0$ and form $\pi^t = \pi_{\mathrm{rand}}$.
  For each pair of distinct agents i, j with $\pi^t(i) = \pi^t(j)$, observe $V_{i,j}^t = v_{i,j} + \varepsilon_{i,j}^t$.
  Update the UCBs and LCBs via (5).

Table 1: Default configuration ($n = 6$, $k = 2$, $\sigma = 3$, $\varepsilon = 1$), both utility models pooled, 100 seeds per model. Regret and existence error are averaged over YES instances (mean $\pm$ standard error). Non-existence error is averaged over the 178 NO instances of the asymmetric-utility robustness study, where false declarations can occur. Lower is better in every column.

	$T = 30$, $\beta = 1$		$T = 1000$, $\beta = 3$		non-existence
Algorithm	regret	existence error	regret	existence error	error (NO)
UCB-LCB-GS	183 ± 9	8.1 ± 0.8	688 ± 93	144.1 ± 20.9	0.00
PESSIMISTIC-ONLY	2756 ± 97	18.1 ± 0.6	36271 ± 3894	240.5 ± 25.3	0.00
OPTIMISTIC-ONLY	337 ± 37	1.2 ± 0.2	7361 ± 1493	47.8 ± 10.1	3.12
UCB-ONLY	568 ± 62	2.8 ± 0.4	12661 ± 1885	85.2 ± 13.0	1.64
LCB-ONLY	3583 ± 63	0.0 ± 0.0	119427 ± 2094	0.0 ± 0.0	30.00
MIXED	295 ± 34	0.9 ± 0.2	6088 ± 1520	39.3 ± 10.2	3.15
RANDOM	4648 ± 56	30.0 ± 0.0	155215 ± 1569	1000.0 ± 0.0	0.00

Algorithm 7 MIXED baseline

Input: n agents; variance σ^2; $k \geq 1$; stability margin $\varepsilon \geq 0$; uncertainty parameter $\beta \in (2, \infty)$.

for each time $t = 1, \ldots, T$ **do**
  Compute $\underline{\pi}^t$ via (7).
  if $\widetilde{\mathrm{Gap}}_C(\underline{\pi}^t, \pi', \underline{\mathbf{v}}^{t-1}, \overline{\mathbf{v}}^{t-1}) \geq \varepsilon$ for all $C \subseteq N$, $|C| \leq k$, and all $\pi' \in \Lambda_C(\underline{\pi}^t)$ **then**
    Set $d_t = 1$ and form $\pi^t = \underline{\pi}^t$.
  else
    Compute $\overline{\pi}^t$ via (9).
    if $\widetilde{\mathrm{Gap}}_C(\overline{\pi}^t, \pi', \overline{\mathbf{v}}^{t-1}, \underline{\mathbf{v}}^{t-1}) \geq \varepsilon$ for all $C \subseteq N$, $|C| \leq k$, and all $\pi' \in \Lambda_C(\overline{\pi}^t)$ **then**
      Set $d_t = 1$ and form $\pi^t = \overline{\pi}^t$.
    else
      Set $d_t = 0$ and form $\pi^t = \pi_{\mathrm{rand}}$.
  For each pair of distinct agents i, j with $\pi^t(i) = \pi^t(j)$, observe $V_{i,j}^t = v_{i,j} + \varepsilon_{i,j}^t$.
  Update the UCBs and LCBs via (5).

K Empirical Analysis: Per-Ablation Discussion

This appendix provides a detailed analysis of the synthetic experiments summarized in Section 5. We vary one parameter at a time around the default configuration $(n, k, T, \sigma, \beta, \varepsilon) = (6, 2, 30, 3, 1, 1)$, over the values

- $T \in \{15, 30, 45, 60\}$ and $n \in \{4, 5, 6, 7\}$;
- $k \in \{1, 2, 3\}$ and $\sigma \in \{1, 3, 5, 10, 20\}$;
- $\varepsilon \in \{0.5, 1, 2, 5, 10\}$ and $\beta \in \{0.25, 0.5, 1, 2, 4\}$.

Every configuration uses 100 seeds, and all algorithms are run on the same instances. Since our theorems require $\beta > 2$, we repeat each ablation except the one over β with $\beta = 3$. We also evaluate the default configuration over longer horizons (Appendix K.7). Regret and existence error are averaged over YES instances; the number of YES instances varies with n, k and ε, and is reported in the figure captions where it changes. Shaded bands show ± 1 standard error, computed over the YES instances of each configuration; their lower edges are truncated at zero, since both metrics are non-negative. Table 1 summarizes the default configuration.

K.1 Horizon T

The horizon determines how many rounds the learner has to explore and establish stability. Since regret and existence error are cumulative, both can increase with T; the important question is how quickly they accumulate. Figures 1 and 2 show a marked difference between UCB-LCB-GS and the baselines. On the Uniform model, the regret of UCB-LCB-GS increases only from 162 at $T = 15$ to 198 at $T = 60$ with $\beta = 1$, and from 180 to 246 with $\beta = 3$. This corresponds to approximately 0.8

and 1.5 additional regret per round, respectively. By comparison, the regret of PESSIMISTIC-ONLY more than doubles, from 1,915 to 3,993. The regret of LCB-ONLY and RANDOM grows roughly linearly, reaching 7,194 and 9,224, respectively. OPTIMISTIC-ONLY and MIXED perform between UCB-LCB-GS and PESSIMISTIC-ONLY, but their regret accumulates several times faster than that of UCB-LCB-GS.

This difference illustrates the benefit of targeted exploration. Rather than repeatedly forming arbitrary or uninformative partitions, UCB-LCB-GS directs its observations towards the uncertain pairs that prevent its candidate from passing the pessimistic test. Consequently, it can establish stability with substantially less cumulative regret.

The existence error of UCB-LCB-GS also increases with T, from 4.8 to 14.0 rounds on the Uniform model. However, as a fraction of the horizon, it decreases from 32% to 23%. Most remaining errors arise on instances with small certification margins, for which establishing stability requires many observations of the relevant pairs (Appendix K.7). In contrast, PESSIMISTIC-ONLY incurs existence error on between 43% and 82% of the rounds (45% to 86% on the Gaussian model), while RANDOM incurs it on every round. The optimism-based baselines and LCB-ONLY achieve much lower existence error because they declare existence readily. Table 2 reveals the other side of this behaviour: these methods also make declarations on instances without a stable partition.

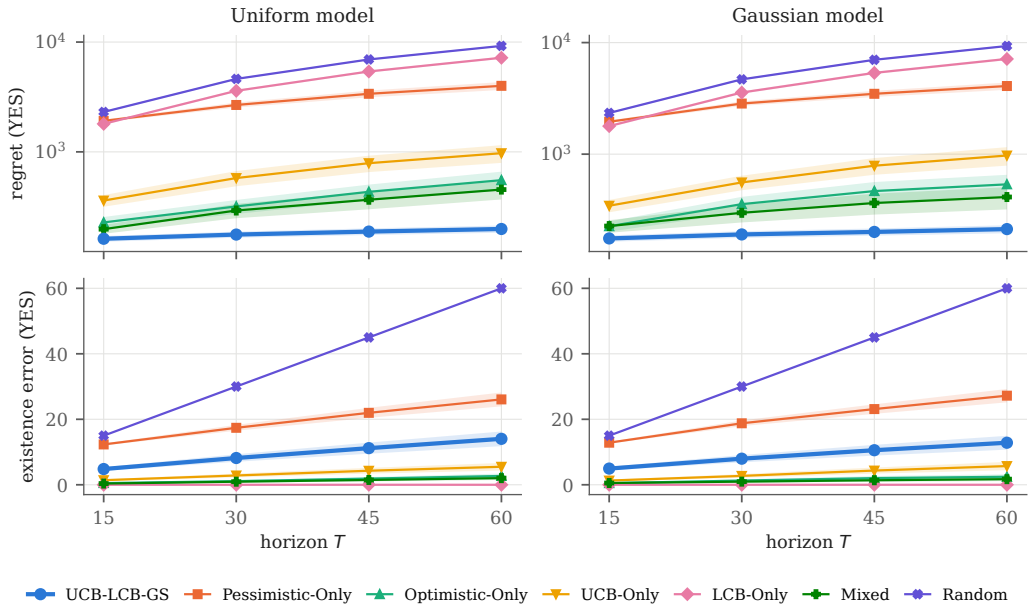

Figure 1: Ablation over the horizon T ($\beta = 1$; other parameters at their default values). Top: regret on YES instances (logarithmic scale). Bottom: existence error on YES instances. Left: Uniform model (98 YES instances). Right: Gaussian model (94 YES instances). Shaded bands: ± 1 standard error.

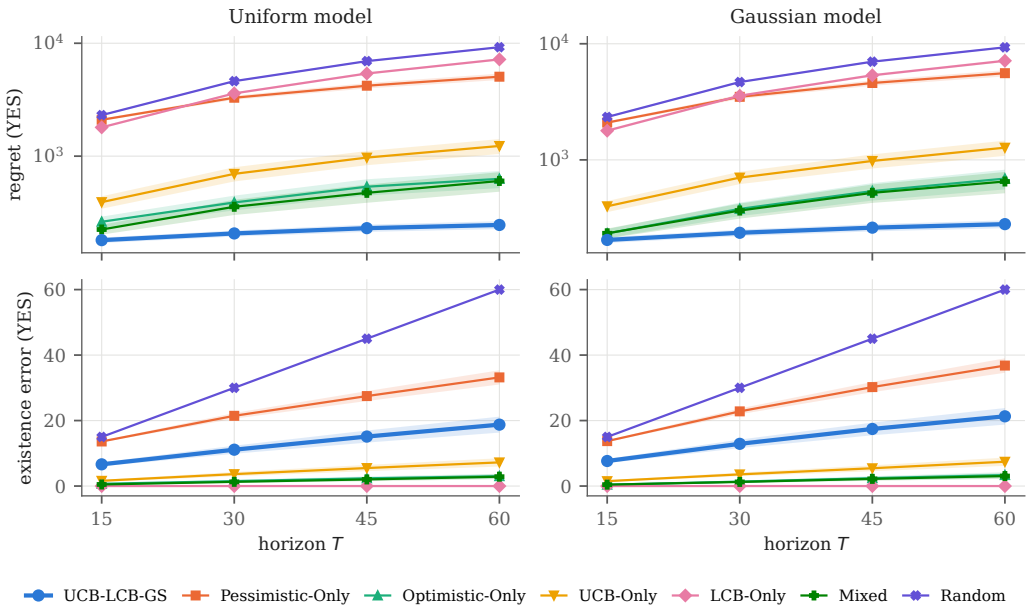

Figure 2: As Figure 1, with $\beta = 3$, the smallest integer value covered by our theorems.

K.2 Stability Margin ε

The stability margin determines which partitions count as stable and, consequently, which instances are labelled YES. As ε increases from 0.5 to 10, the number of YES instances decreases from 98 to 73 on the Uniform model and from 94 to 71 on the Gaussian model. These instances move to the intermediate class: they still admit an exactly k-strongly Nash stable partition, but none with margin ε. Thus, Figures 3 and 4 average over progressively smaller and more demanding sets of instances. Moreover, since regret measures the shortfall from ε, increasing this threshold can increase regret even when the formed partitions remain unchanged.

Against this increasingly demanding criterion, UCB-LCB-GS is affected relatively mildly. On the Uniform model, its regret rises from 171 to 250, while its existence error remains between 7.6 and 8.2 rounds. With $\beta = 3$, its regret rises from 200 to 335. The effect is more pronounced for UCB-ONLY, whose regret more than doubles and whose existence error rises from 2.0 to 5.8 rounds. These results suggest that the combined optimistic-pessimistic approach is less sensitive to the increasing stability requirement than this single-estimate baseline. The distinction is important because a more demanding stability margin affects both the set of instances being evaluated and the threshold used to compute regret.

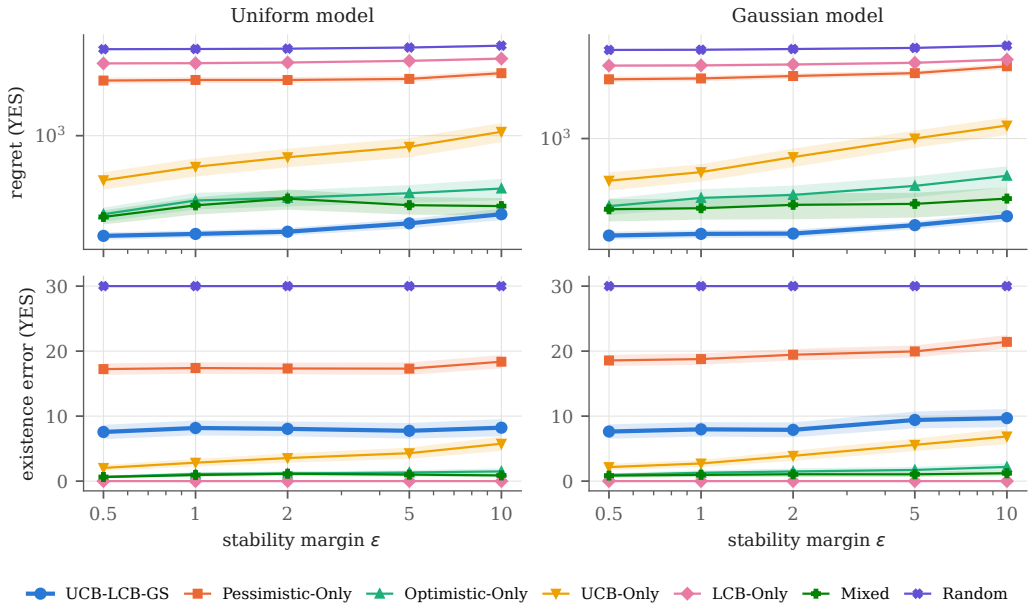

Figure 3: Ablation over the stability margin ε ($\beta = 1$). Panels as in Figure 1. The number of YES instances decreases with ε (Uniform: from 98 to 73; Gaussian: from 94 to 71), so each point averages over a different set of instances.

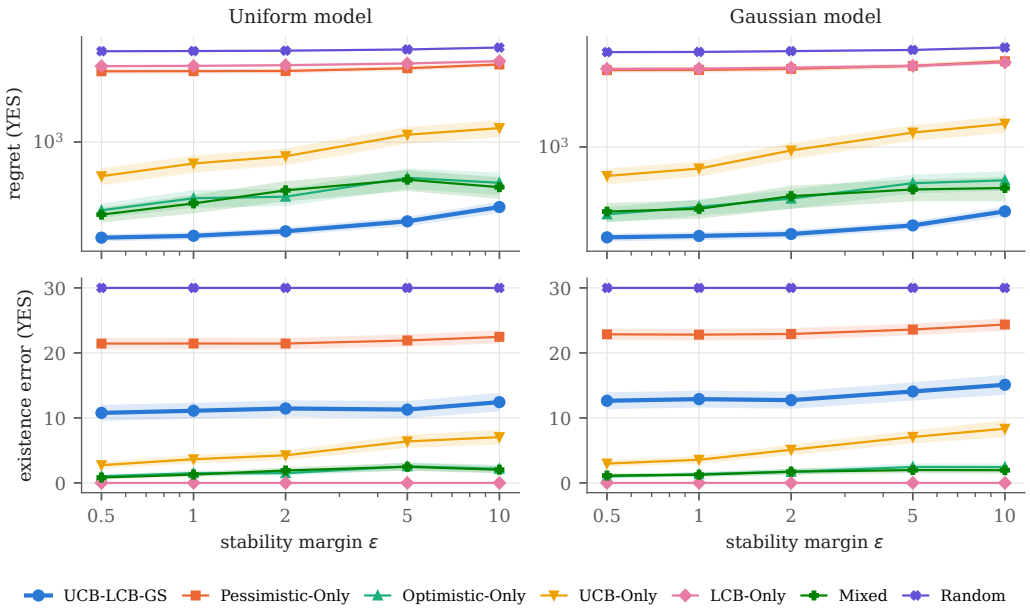

Figure 4: As Figure 3, with $\beta = 3$.

K.3 Number of Agents n

Increasing the number of agents increases both the number of unknown pairwise utilities, $\binom{n}{2}$, and the number of possible deviations. Our bounds account for this growth polynomially in n. Figures 5 and 6 show that the empirical regret of UCB-LCB-GS also increases, but remains substantially below that of the baselines with sound existence tests. From $n = 4$ to $n = 7$, its regret rises from 94 to 214 on the Uniform model and from 93 to 236 on the Gaussian model. Its existence error rises from approximately 3–5 rounds at $n = 4$ to about 8 rounds at $n = 7$. By comparison, the regret of PESSIMISTIC-ONLY increases from 1,253 to 3,680, and that of RANDOM from 2,870 to 5,345 on the Uniform model. Thus, although learning becomes more demanding as the number of agents increases, witness probing continues to provide a substantial advantage over purely pessimistic or uninformed exploration.

The relative ordering changes only among baselines with similar regret. The instances differ across values of n, and a few NO instances appear for some values of n, so the comparisons across problem sizes are not paired comparisons on identical instances. With $\beta = 3$, the regret of UCB-LCB-GS at $n = 7$ is 292 on the Uniform model and 319 on the Gaussian model, still far below that of every sound baseline.

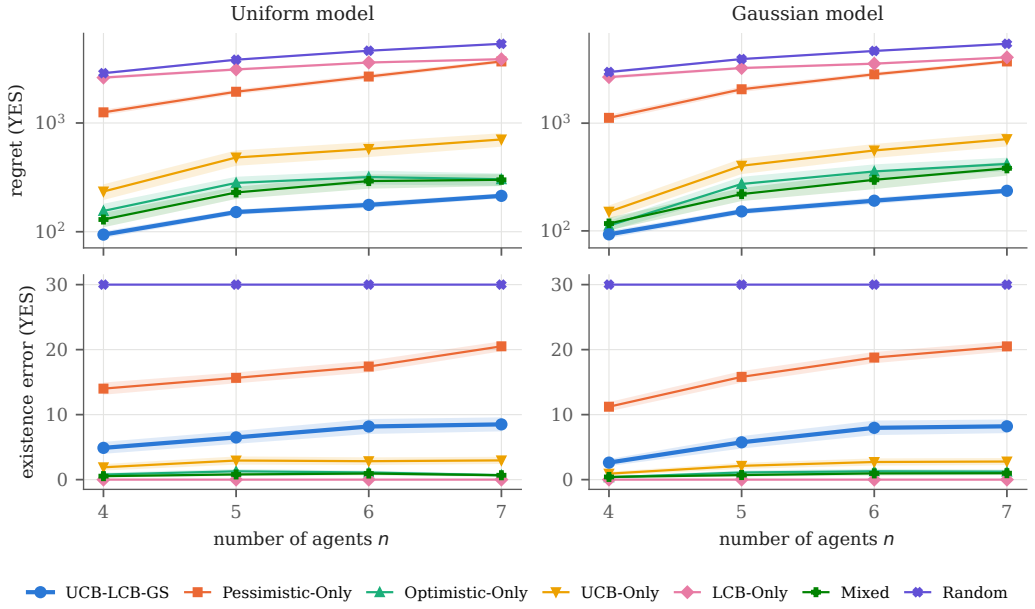

Figure 5: Ablation over the number of agents n ($\beta = 1$). Panels as in Figure 1. Each value of n uses its own 100 instances per model.

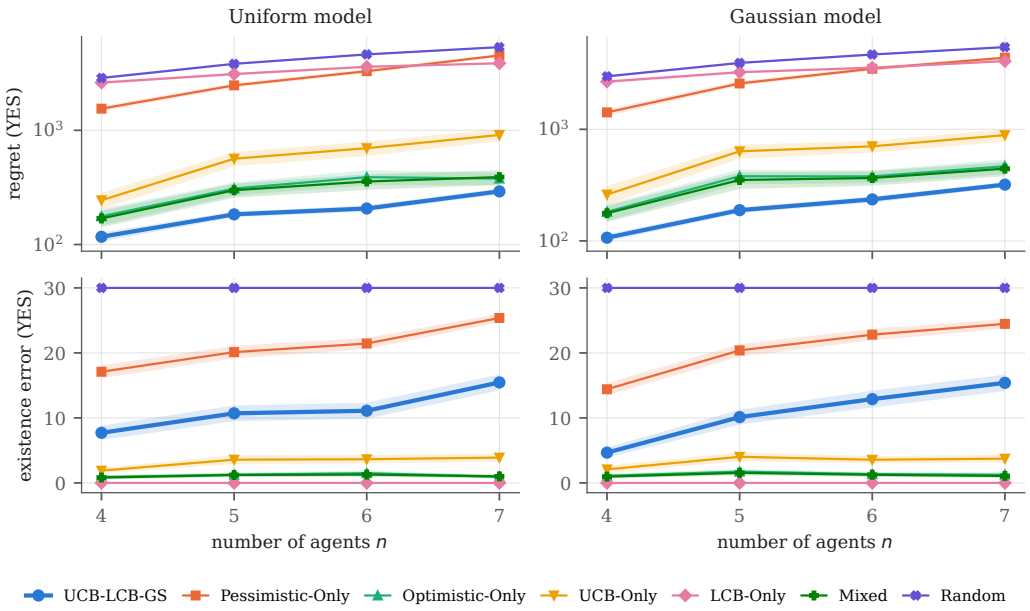

Figure 6: As Figure 5, with $\beta = 3$.

K.4 Deviating-Coalition Cap k

The parameter k determines the maximum size of a deviating group. Increasing k makes stability harder to achieve: the certification margin is non-increasing in k, and the family of deviations grows. Nevertheless, the empirical effect on our random instances is relatively modest. Few partitions that resist deviations by two agents fail when deviations by three are allowed. On the Uniform model, the number of YES instances is nearly unchanged (99, 98 and 98 for $k = 1, 2, 3$, respectively), as are the results in Figures 7 and 8. The regret of UCB-LCB-GS rises from 159 to 188, and its existence error from 6.1 to 9.1 rounds on the Uniform model.

The main consequence of increasing k is therefore computational rather than statistical in these experiments. Exact deviation checking with $k = 3$ costs about four times as much per round as with $k = 2$ at $n = 6$ (Appendix O), and the cost grows much faster with n. Thus, the relatively small change in the observed regret should not be interpreted as indicating that larger deviation groups are computationally inexpensive.

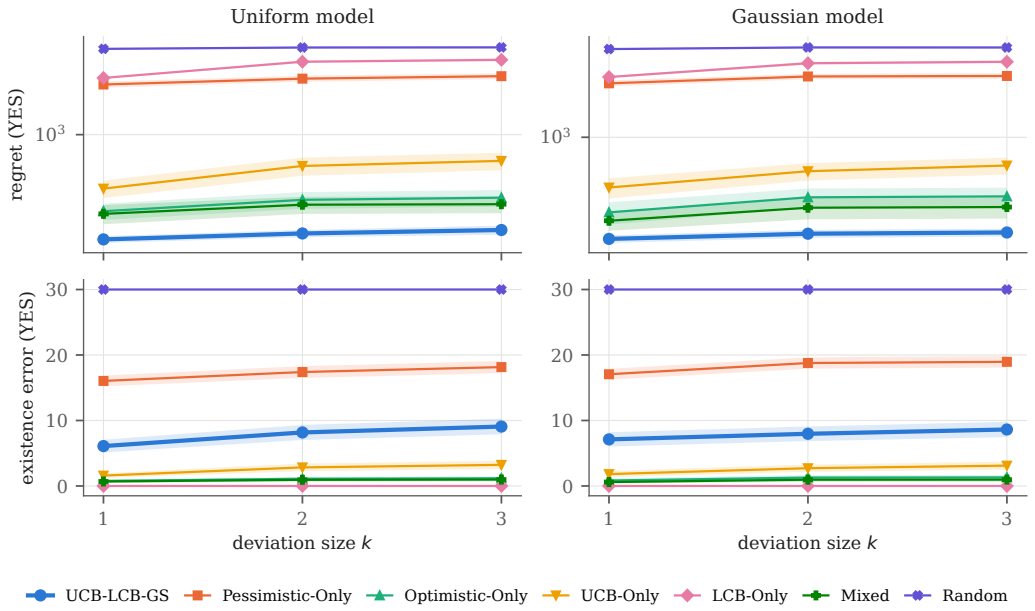

Figure 7: Ablation over the deviating-coalition cap k ($\beta = 1$). Panels as in Figure 1.

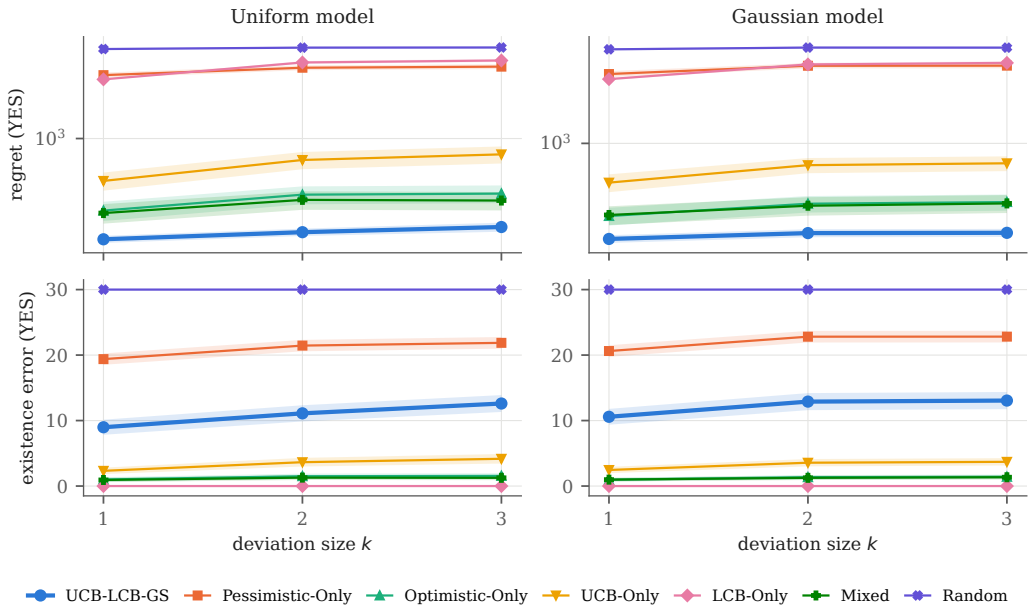

Figure 8: As Figure 7, with $\beta = 3$.

K.5 Noise Standard Deviation σ

The noise standard deviation scales the confidence radii and therefore affects how many observations are needed before the learner can establish stability. The theoretical bounds grow quadratically with σ. Figures 9 and 10 show that noise has the strongest effect on the empirical performance of UCB-LCB-GS among the parameters considered. On the Uniform model, increasing σ from 1 to 20 raises its regret from 148 to 856 and its existence error from 4.0 to 27.2 rounds out of 30.

This is also the only ablation in which baselines attain lower regret than UCB-LCB-GS. At $\sigma = 20$, MIXED incurs regret 774 and OPTIMISTIC-ONLY 790, compared with 856 for UCB-LCB-GS. The same ordering holds with $\beta = 3$ at $\sigma = 20$, while the methods are nearly tied at $\sigma = 10$.

The reason for this reversal is particularly informative. At such high noise, UCB-LCB-GS rarely establishes stability within the 30-round horizon. It therefore continues probing uncertain pairs, and these probes incur regret when the resulting partition is far from stable. OPTIMISTIC-ONLY and MIXED, in contrast, do not use targeted exploration. They can settle on partitions that appear stable under optimistic estimates and declare existence without first obtaining the observations needed for a pessimistic guarantee. Their lower regret in this regime must therefore be read alongside their existence decisions: the optimism-based baselines frequently declare existence on instances without a stable partition, as reflected in Table 2. This is a concrete example of the trade-off between regret and reliable existence detection, rather than evidence that optimistic exploration is uniformly preferable. We did not run longer horizons at this noise level.

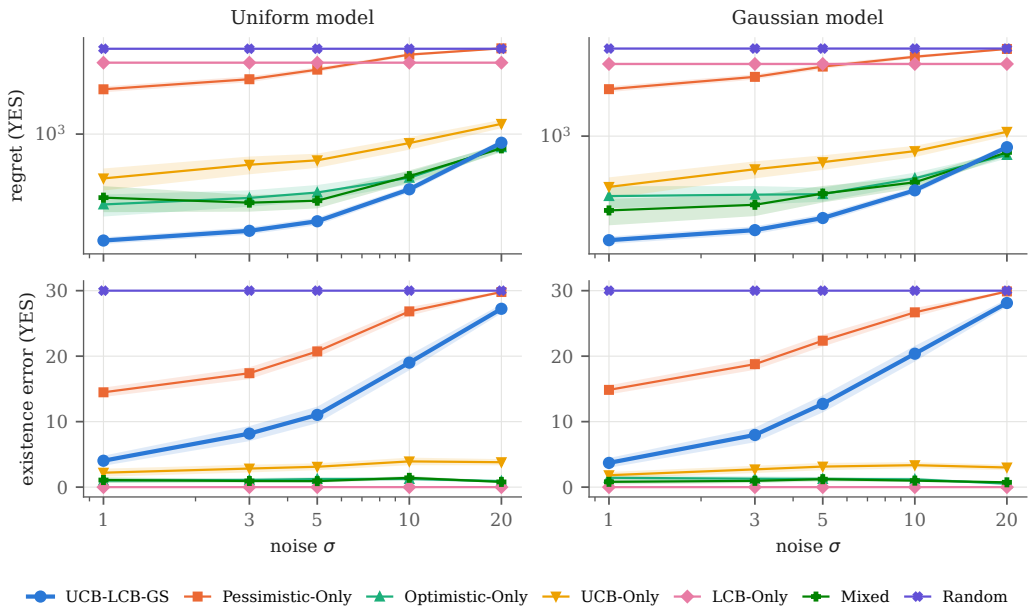

Figure 9: Ablation over the noise standard deviation σ ($\beta = 1$). Panels as in Figure 1. At $\sigma = 20$, MIXED and OPTIMISTIC-ONLY have lower regret than UCB-LCB-GS, which rarely establishes stability within 30 rounds.

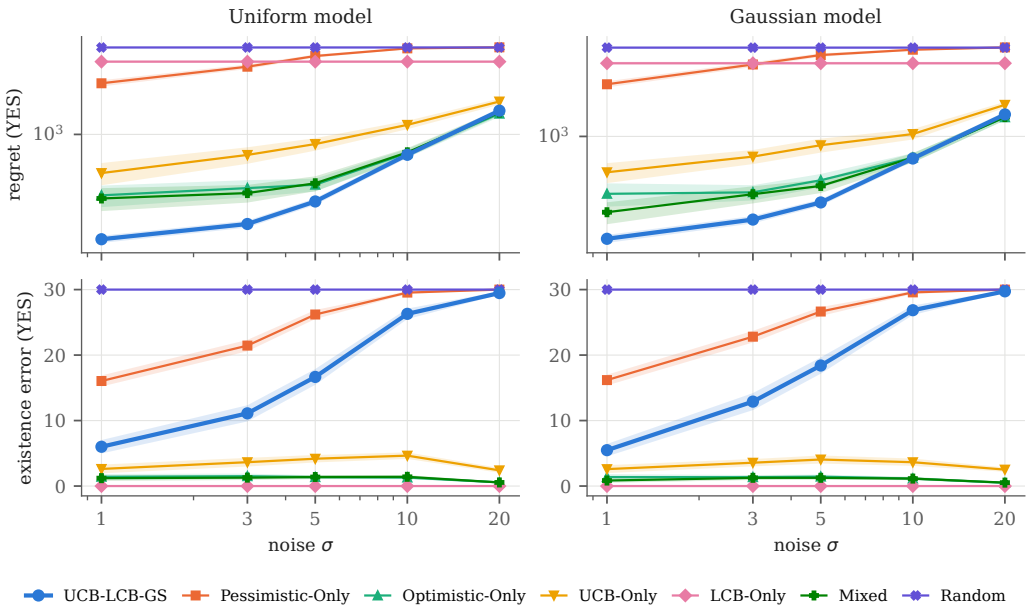

Figure 10: As Figure 9, with $\beta = 3$.

K.6 Uncertainty Parameter β

The parameter β determines the width of the confidence intervals, scaling their radii by $\sqrt{\beta}$. Larger values make the pessimistic test more conservative and the optimistic estimate more exploratory. Our theorems require $\beta > 2$. Thus, in this ablation only $\beta = 4$ lies within the theoretical range; the repetitions of the other ablations with $\beta = 3$ provide additional theory-aligned results.

As β increases from 0.25 to 4 (Figure 11), the regret of UCB-LCB-GS on the Uniform model rises from 150 to 230, and its existence error from 4.4 to 12.8 rounds. PESSIMISTIC-ONLY exhibits the same trend, while LCB-ONLY and RANDOM are largely unaffected. Wider confidence intervals make the learner more cautious: this can improve the reliability of its declarations, but may require more rounds of exploration before stability is established. Conversely, smaller values of β lead to earlier declarations, at the risk of accepting a partition without sufficient statistical evidence. Indeed, the only false declaration of UCB-LCB-GS in our grid occurred at $\beta = 0.25$: a single round on one of the three Gaussian NO instances (Table 2).

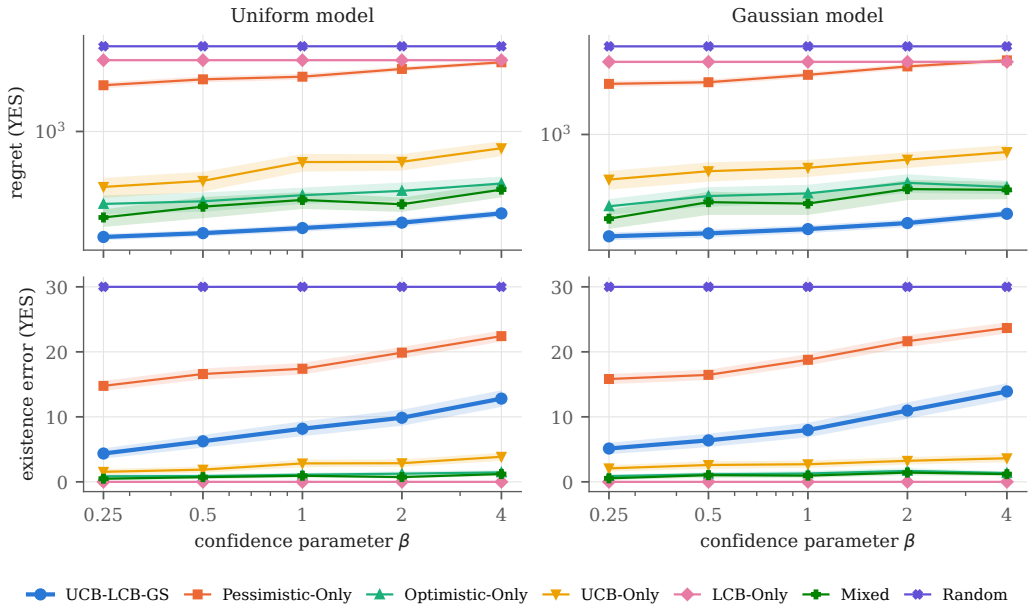

Figure 11: Ablation over the uncertainty parameter β. Panels as in Figure 1. The theorems require $\beta > 2$.

K.7 Long Horizons

Figures 12 and 13 extend the default configuration to $T = 1000$ rounds. They show cumulative regret and the cumulative number of rounds without a declaration as functions of time.

- *Regret.* The cumulative regret of UCB-LCB-GS increases mainly during the early rounds and then grows much more slowly, whereas the baselines continue to accumulate regret. At $T = 1000$ with $\beta = 3$, its average regret across both models is about 690, compared with approximately 6,100 for MIXED, the best baseline. This substantial difference illustrates the benefit of targeted exploration over the longer horizon: UCB-LCB-GS increasingly concentrates its observations on the pairs that matter for establishing stability, rather than repeatedly incurring regret through uninformative partitions. The observed growth is consistent with the logarithmic behaviour of our bounds, subject to the exact-optimization assumption in the theoretical guarantee.
- *Existence error.* The cumulative number of rounds without a declaration continues to increase, but at a decreasing rate. The remaining errors are concentrated on instances with small certification margins. When the best partition has a gap only slightly above ε, the learner needs many observations of the relevant pairs before it can establish stability. When the margin is zero, a pessimistic test cannot establish existence on rounds with valid confidence intervals (Proposition E.2).

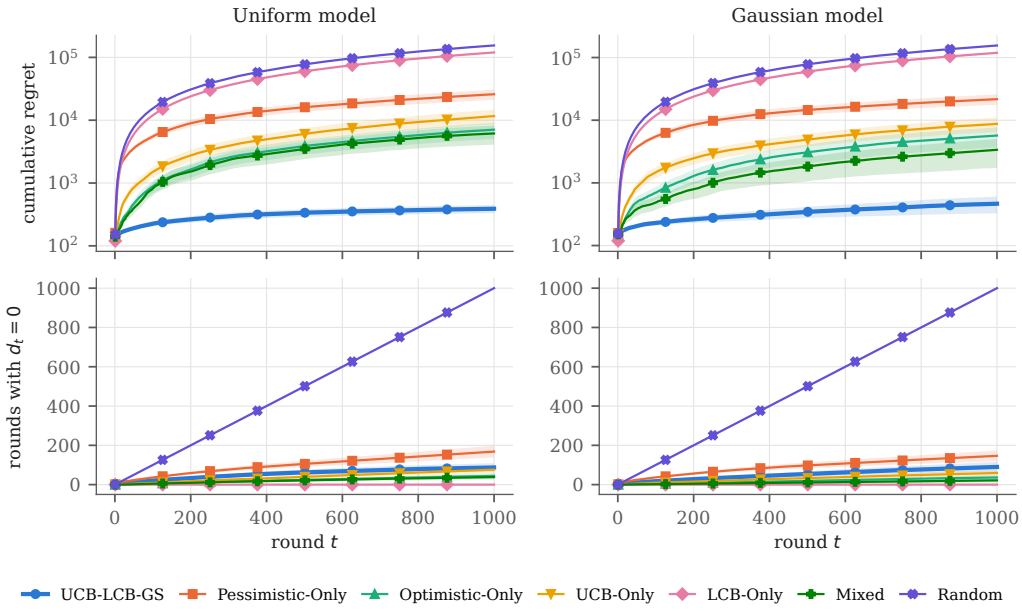

Figure 12: Default configuration with $T = 1000$ and $\beta = 1$: cumulative regret (top, logarithmic scale) and cumulative number of rounds with $d_t = 0$ (bottom) on YES instances, as functions of the round. Shaded bands: ± 1 standard error.

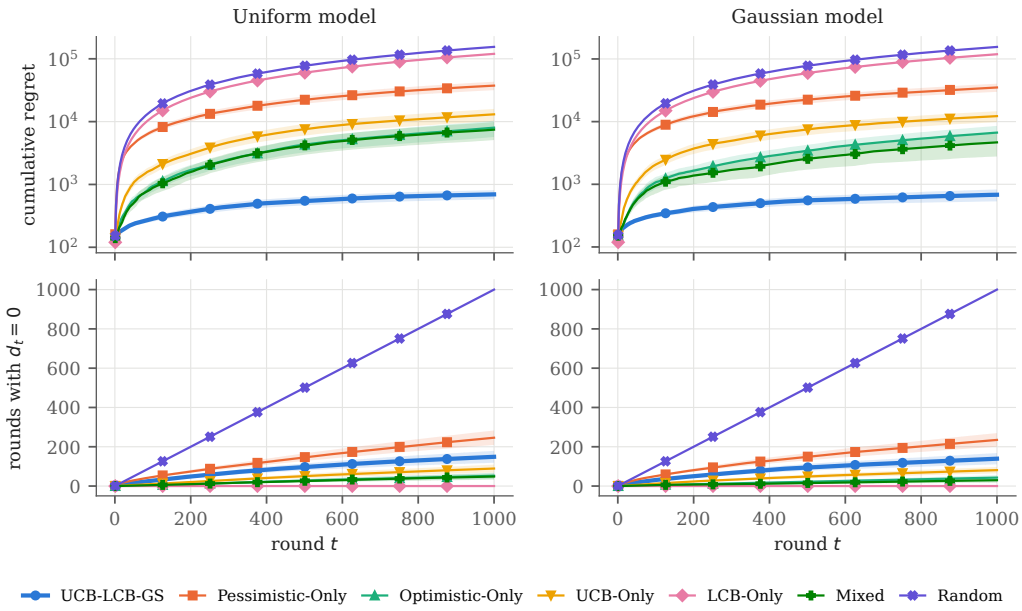

Figure 13: As Figure 12, with $\beta = 3$.

K.8 Instances without a Stable Partition

Table 2 complements the YES-instance results by reporting declarations on NO instances, which determine the non-existence error, and on intermediate instances, which are not scored by either error metric. A clear difference emerges between the methods. The optimism-based baselines and LCB-ONLY declare existence on most rounds of these instances. UCB-LCB-GS, in contrast, declared existence on a NO instance only once across the two grids, in the configuration with $\beta = 0.25$. This was outside the range covered by our theorems. PESSIMISTIC-ONLY and RANDOM never declared existence on a NO instance.

These results help explain why the low existence error of the optimism-based baselines should not be interpreted in isolation. Their frequent declarations reduce existence error on YES instances, but the same behaviour leads to incorrect declarations on NO instances and declarations on intermediate instances. UCB-LCB-GS avoids this systematic trade-off by requiring the pessimistic test to succeed before declaring existence.

Table 2: Declarations on instances without an (ε, k)-strongly Nash stable partition, pooled over all cells of each grid. "NO": mean number of rounds with $d_t = 1$ per NO instance (the non-existence error). "int.": fraction of rounds with $d_t = 1$ on intermediate instances, which neither metric scores. The counts are instance–cell pairs; the same instance appears in several cells.

	historical grid		$\beta = 3$ grid	
Algorithm	NO	int.	NO	int.
UCB-LCB-GS	0.01	0.00	0.00	0.00
PESSIMISTIC-ONLY	0.00	0.00	0.00	0.00
OPTIMISTIC-ONLY	19.80	0.80	25.53	0.85
UCB-ONLY	13.16	0.57	17.55	0.61
LCB-ONLY	31.22	1.00	31.45	1.00
MIXED	19.99	0.80	22.97	0.84
RANDOM	0.00	0.00	0.00	0.00
instance–cell pairs	74	197	62	177

Effect of the exact deviation checker. The baselines' trajectories in configurations with $k \leq 2$ were initially computed using a restricted family of deviations. We reuse these results because that family provably attains the same minimum as the complete family on all partitions arising in these configurations (Appendix I). For $k = 3$, we reran all baselines using the exact checker and the same seeds. The reruns reproduced the earlier declarations and regret in every trial except one: the regret of a RANDOM trial with $k = 3$, whose partitions were scored differently by the two checkers.

K.9 Cross-Ablation Summary

Taken together, the experiments show that the main advantage of UCB-LCB-GS is its ability to combine low regret with reliable existence decisions. Across the 96 configurations of the two grids, it achieves the lowest regret among all methods in 92; the four exceptions are precisely the configurations with $\sigma = 20$. It also has the lowest existence error in every configuration among the methods that avoid systematic declarations on instances without a stable partition. Thus, its advantage is not simply that it is conservative: unlike PESSIMISTIC-ONLY, it uses targeted exploration to establish stability substantially earlier, while avoiding the unsupported declarations made by the optimism-based baselines.

The comparisons between methods make the roles of the two components clear. Pessimistic tests make declarations of existence reliable, whereas witness probing supplies the observations needed to pass those tests. PESSIMISTIC-ONLY lacks the latter and consequently incurs much higher regret and existence error. OPTIMISTIC-ONLY, UCB-ONLY, MIXED and LCB-ONLY can achieve lower existence error by declaring existence readily, but this comes at the cost of declarations on NO and intermediate instances. At $\sigma = 20$, the cost of witness probing is particularly high, and the optimism-based methods obtain lower regret by settling earlier. The corresponding declarations, however, make this a regret-versus-reliability trade-off rather than an unambiguous improvement over UCB-LCB-GS.

The parameter sweeps also reveal which aspects of the problem drive performance. Increasing T naturally increases cumulative regret and existence error, although UCB-LCB-GS accumulates both substantially more slowly than the baselines. Increasing n, σ, β or ε generally raises its regret and, in most cases, its existence error over the tested ranges. Increasing k has a more modest effect on these metrics in our instances, although it substantially increases the computational cost of exact deviation checking. Finally, moving from $\beta = 1$ to the theory-aligned value $\beta = 3$ raises the regret of UCB-LCB-GS in the default configuration by 17% on the Uniform model and 24% on the Gaussian model, and its existence error by 36% and 62%, respectively. Despite this additional conservatism, the comparison with the baselines remains unchanged.

L Experimental Evaluation under Coalition-Level Bandit Feedback

The theoretical results of Appendix F show that coalition-level feedback preserves the soundness guarantee, but makes learning more costly: UCB-LCB-GS still has a bounded expected number of false declarations, although the bound is larger, and requires more rounds to establish stability. This difference is particularly relevant for witness probing, since each probe now provides information about only one pair. We examine the empirical cost of this feedback model and how it affects the comparison with baselines that do not require a sound stability test. We first describe the adaptation of the baselines, then evaluate the methods under coalition-level feedback, and finally compare the two feedback models.

Setup. We evaluate Algorithm 2 on the same parameter grid, instances and seeds as under pairwise semi-bandit feedback in Section 5 and Appendix K. This includes the ablations around the default configuration with $\beta = 1$, their repetitions with $\beta = 3$, and the default configuration with $T \in \{300, 1000\}$. Instance labels and the deviation checker are unchanged (Appendix I), since existence depends on the underlying game rather than the feedback model. The learning baselines retain their decision rules (Appendix J) but use the estimator of Appendix F.3, which obtains observations only from two-agent coalitions. They therefore all begin with the same round-robin matching schedule as UCB-LCB-GS and subsequently follow their respective rules, including the random fallback when applicable. RANDOM remains unchanged because its decisions do not depend on feedback. To isolate the effect of the decision rule from that of exploration, we also evaluate the learning baselines with a common exploration schedule in the default configuration: on every second round without a declaration, they form the next partition in the matching cycle instead of following their usual partition-selection rule.

Soundness and the regret trade-off. Table 3 and Figures 15–17 reveal a clear distinction between UCB-LCB-GS and the optimism-based baselines. UCB-LCB-GS never declared existence on a NO instance in any configuration. It also declared existence on intermediate instances, which neither error metric scores, in only one configuration: with $\beta = 0.25$ under the Gaussian model, it did so in 7% of the rounds across the three intermediate instances.

Among the methods that avoid declarations on NO instances, namely UCB-LCB-GS, PESSIMISTIC-ONLY and RANDOM, UCB-LCB-GS achieves the lowest regret in all 96 configurations of the two grids and in every long-horizon configuration. It also has the lowest existence error in all but two configurations, both with $\sigma = 20$ and $\beta = 3$, in which all three methods fail to declare existence on any YES instance. Thus, witness probing substantially improves over purely pessimistic exploration, while retaining the latter's cautious decision rule.

The comparison changes when we consider the optimism-based baselines. Under coalition-level feedback, OPTIMISTIC-ONLY, UCB-ONLY, LCB-ONLY and MIXED all achieve lower regret than UCB-LCB-GS in every configuration. The regret of UCB-LCB-GS is between 1.2 and 2.6 times that of the best of these baselines, with a median ratio of 2.0. These baselines also have lower existence error on YES instances, but this does not mean that they identify existence more reliably. They do not spend rounds on targeted probes; instead, they form partitions that appear stable under their respective estimates and may declare existence without a pessimistic guarantee. Indeed, they declare existence on an average of 8 to 22 rounds per NO instance, and on most rounds of intermediate instances.

The difference reflects the cost of obtaining reliable evidence under coalition-level feedback. Under pairwise feedback, a round observes every pair that shares a coalition, so an exploration round can improve several pairwise estimates simultaneously. Under coalition-level feedback, a witness

probe reveals only the selected pair. UCB-LCB-GS must therefore spend substantially more rounds collecting the observations needed by its pessimistic test, and these additional probes incur regret. Its advantage over the other sound methods remains clear, but maintaining reliable existence decisions now comes at a substantially higher empirical regret than under pairwise feedback.

Table 3: Coalition-level feedback, default configuration, both utility models pooled, 100 seeds per model (mean $\pm$ standard error over YES instances). The last column is the non-existence error pooled over the NO instances of both grids. RANDOM is identical under both feedback models.

	$T = 30$, $\beta = 1$		$T = 1000$, $\beta = 3$		non-existence
Algorithm	regret	existence error	regret	existence error	error (NO)
UCB-LCB-GS	1478 ± 79	12.1 ± 0.7	18076 ± 2453	178.7 ± 22.5	0.00
PESSIMISTIC-ONLY	2208 ± 122	14.7 ± 0.8	44218 ± 4044	293.8 ± 26.5	0.00
OPTIMISTIC-ONLY	870 ± 25	5.5 ± 0.2	7796 ± 1122	23.1 ± 7.4	21.73
UCB-ONLY	962 ± 35	6.1 ± 0.2	9382 ± 1367	39.4 ± 9.6	17.35
LCB-ONLY	958 ± 37	6.2 ± 0.2	6258 ± 905	26.3 ± 5.6	8.38
MIXED	859 ± 24	5.4 ± 0.1	7689 ± 1205	21.8 ± 7.8	21.88
RANDOM	4648 ± 56	30.0 ± 0.0	155215 ± 1569	1000.0 ± 0.0	0.00

Shared exploration. The common matching schedule provides an additional comparison between exploration and decision-making. With this schedule, the regret and existence error of the baselines change only moderately (Table 4). The additional exploration is particularly helpful to PESSIMISTIC-ONLY at the longer horizon, but does not change which methods declare existence on the basis of tests that are not sound. This is expected: the exploration schedule affects which utilities are observed, whereas the reliability of a declaration is determined by the decision rule. The results therefore suggest that merely adding exploration is not sufficient to reproduce the behaviour of UCB-LCB-GS; it must be coupled with a sound pessimistic test.

Table 4: Coalition-level feedback: baselines with their own exploration and with the shared matching schedule (regret / existence error on YES instances, Uniform model, default configuration). UCB-LCB-GS is shown for reference.

Algorithm	$T = 30$, $\beta = 1$	$T = 30$, $\beta = 3$	$T = 300$, $\beta = 1$	$T = 300$, $\beta = 3$
UCB-LCB-GS	1442 / 11.6	1827 / 15.4	5631 / 52.0	8162 / 78.1
PESSIMISTIC-ONLY, own	2136 / 14.2	2964 / 19.6	12782 / 84.1	17983 / 118.0
PESSIMISTIC-ONLY, shared	1987 / 12.9	2668 / 17.4	10911 / 69.5	15005 / 97.0
OPTIMISTIC-ONLY, own	849 / 5.3	890 / 5.3	1741 / 9.8	2633 / 10.6
OPTIMISTIC-ONLY, shared	838 / 5.3	879 / 5.3	1750 / 10.0	2233 / 9.9
UCB-ONLY, own	942 / 5.9	1005 / 6.1	2578 / 14.5	3097 / 16.2
UCB-ONLY, shared	941 / 6.0	988 / 6.0	2482 / 13.9	3177 / 15.6
LCB-ONLY, own	1020 / 6.6	994 / 6.1	2538 / 15.8	2714 / 13.9
LCB-ONLY, shared	986 / 6.2	967 / 5.9	2342 / 14.6	2511 / 12.4
MIXED, own	856 / 5.4	865 / 5.1	1357 / 7.3	2421 / 9.5
MIXED, shared	836 / 5.2	860 / 5.2	1576 / 8.5	2160 / 8.9

The cost of coalition-level feedback. Figure 14 compares UCB-LCB-GS under the two feedback models on the same instances and seeds. In the default configuration, its regret under coalition-level feedback is 6 to 30 times its regret under pairwise feedback, with the ratio increasing with the horizon. Across the 96 configurations of the two grids, the ratio ranges from approximately 3 to 11. The increase in existence error is considerably smaller: between 1.2 and 1.7 times in the default configuration. Thus, the learner establishes stability only moderately later under coalition-level feedback, but incurs substantially more regret along the way. This distinction is explained by the observations available under the two models. Pairwise feedback continues to reveal the utilities of all co-located pairs, including after stability has been established, whereas coalition-level feedback provides information about an individual pair only when that pair is deliberately probed in a two-agent coalition. Consequently, the learner may spend many additional rounds forming partitions that are further from stable before obtaining enough information to make a reliable declaration.

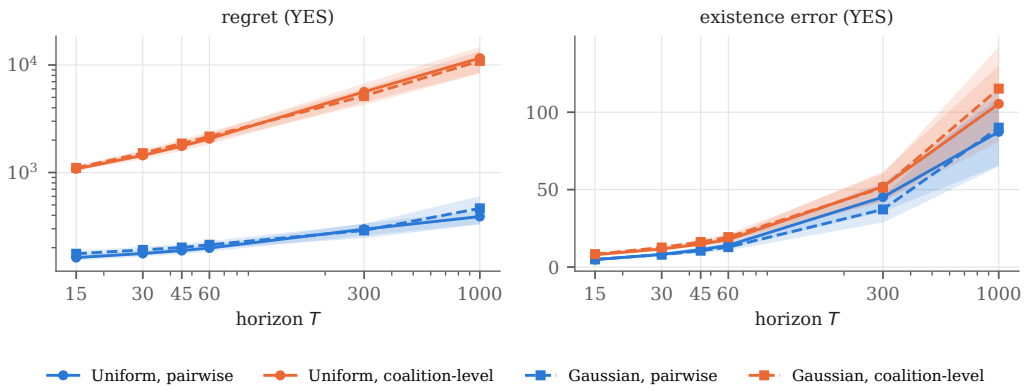

Figure 14: UCB-LCB-GS under pairwise (blue) and coalition-level (orange) feedback in the default configuration ($n = 6$, $k = 2$, $\sigma = 3$, $\beta = 1$, $\varepsilon = 1$), on the same instances and seeds. Solid lines: Uniform model; dashed lines: Gaussian model. Left: cumulative regret on YES instances (logarithmic scale). Right: existence error on YES instances. Shaded bands show ± 1 standard error over the YES instances.

Computation. The learner's runtime per round under coalition-level feedback is similar to that under pairwise feedback, amounting to a few milliseconds in the default configuration, since both algorithms solve the same max-min problems and use the same deviation checker. The complete set of coalition-level feedback experiments (54,400 trials) took 38 minutes on 11 parallel processes (Appendix O).

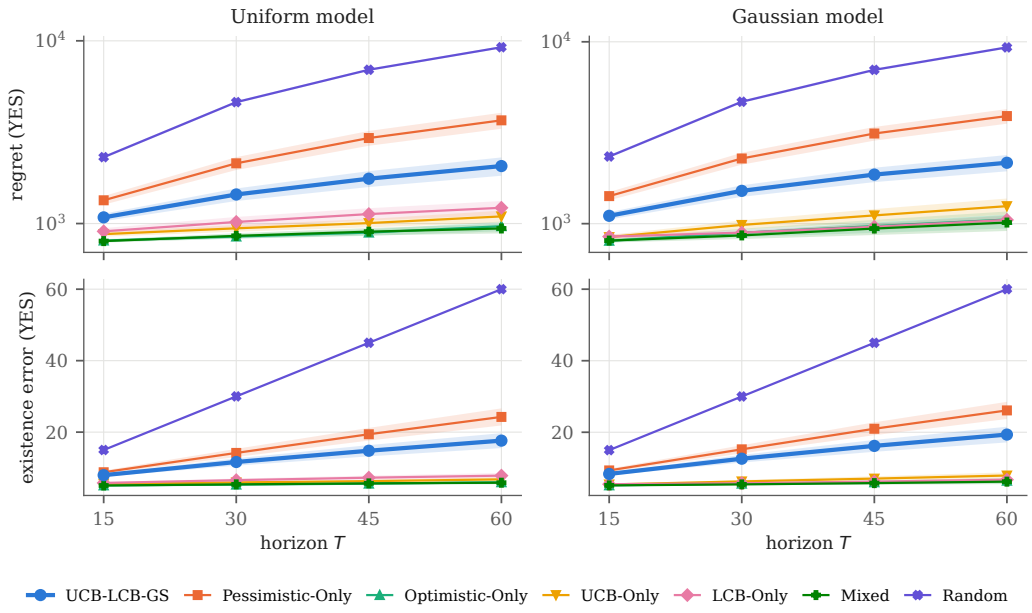

Figure 15: Coalition-level feedback: regret (top, logarithmic scale) and existence error (bottom) on YES instances as functions of the horizon T, with the other parameters at their default values ($\beta = 1$). Shaded bands show ± 1 standard error over the YES instances of each configuration.

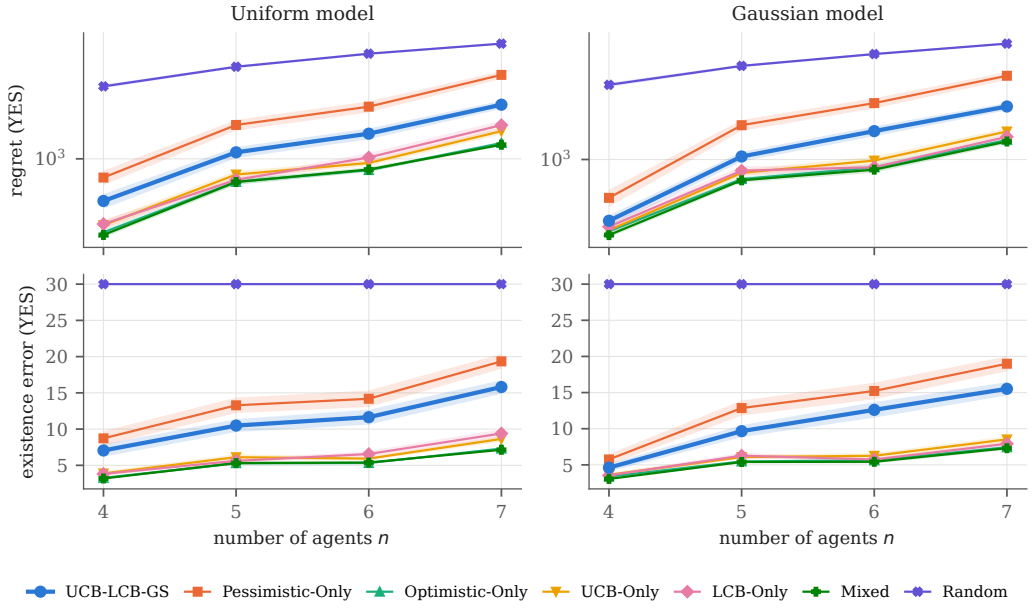

Figure 16: Coalition-level feedback: as Figure 15, as functions of the number of agents n. Larger n lengthens the initialization ($r = n - 1$ or n rounds) and increases the number of pairs that may need probing.

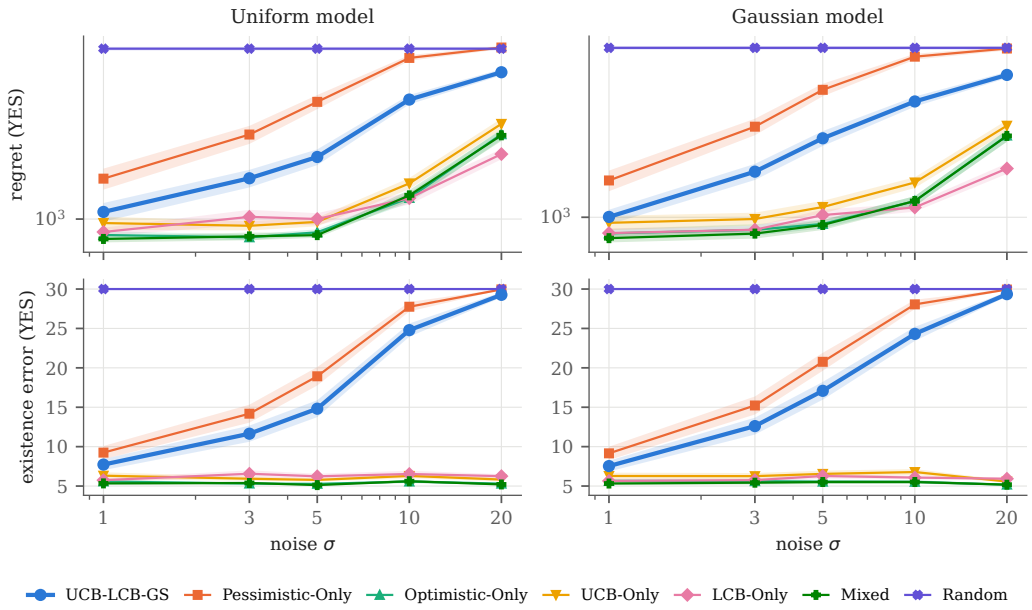

Figure 17: Coalition-level feedback: as Figure 15, as functions of the noise level σ. With $\sigma = 20$, no sound method establishes stability within 30 rounds on most instances.

M Experiments on Games Derived from Real-World Datasets

The synthetic experiments of Section 5 generate each pairwise utility independently, and therefore do not capture the structure that may arise from underlying similarities or relationships between agents. To examine learning in more structured games, we also consider games derived from three datasets commonly used in clustering and community detection: the Iris measurements [46], Zachary's karate club network [78], and the jazz collaboration network [52]. Following the construction of Csáji et al. [36], we use distances in these datasets to identify friends, enemies and neutral pairs, and use these relationships to define additively separable utilities. These are constructed utilities, rather than preferences elicited from the underlying entities. We evaluate UCB-LCB-GS and all six baselines under both semi-bandit and coalition-level bandit feedback.

M.1 Datasets

Iris contains four measurements (sepal and petal length and width, in centimetres) of 150 flowers, with 50 flowers from each of three species [46]. We use the version of the dataset in the UCI Machine Learning Repository that corrects two records [47], which is also the version used by Csáji et al. [36]. Zachary's karate club network contains 34 members and 78 ties between members who interacted outside the club; following a dispute, the club split into two groups of 17 members [78]. The jazz collaboration network of Gleiser and Danon [52], in the version maintained by the Koblenz Network Collection [61], contains 198 nodes and $2{,}742$ edges. Each node represents a band, and two bands are adjacent if they have a musician in common. In both networks, nodes are identified only by numbers. Our code retrieves the datasets from their respective sources; we do not redistribute them.

M.2 Construction of the Games

Friends, enemies and neutrals. Following Csáji et al. [36], let $d(x, y)$ denote the Euclidean distance between the four measurements of two flowers, or the length of a shortest path between two nodes of a network. Let diam denote the largest distance in the corresponding full dataset. For thresholds $f \le e$, agent y is a *friend* of x if $d(x, y) \le f \cdot \mathrm{diam}$, an *enemy* if $d(x, y) > e \cdot \mathrm{diam}$, and *neutral* otherwise. These relations are symmetric.

We use $(f, e) = (0.25, 0.35)$, the only setting considered by Csáji et al. [36] with $f < e$, and hence the only one that allows neutral pairs. The diameter of the karate club network is 5, so its hop distances are integers and none lies between 1.25 and 1.75. Consequently, its games contain no neutral pairs: two members are friends exactly when they are connected by an edge, and enemies otherwise. In Iris, approximately 10% of the pairs are neutral. The jazz network has diameter 6, so two nodes are friends if they are adjacent, neutral if their distance is 2, and enemies otherwise; approximately 55% of its pairs are neutral.

Utilities. Given friends, enemies and neutral pairs, a game of type (α_f, α_e) is the symmetric ASHG with

$$v_{x,y} = \begin{cases} \alpha_f & \text{if } x \text{ and } y \text{ are friends,} \\ \alpha_e & \text{if } x \text{ and } y \text{ are enemies,} \\ 0 & \text{if } x \text{ and } y \text{ are neutral.} \end{cases}$$

This construction follows Dimitrov et al. [38]. We consider the three preference types of Csáji et al. [36]: *appreciation of friends*, $(n, -1)$, under which an agent prioritizes having more friends and, among coalitions with the same number of friends, fewer enemies; *aversion to enemies*, $(1, -n)$, which reverses these priorities; and *balanced* preferences, $(1, -1)$.

We multiply the utilities by the largest integer c for which they all lie in $[-100, 100]$. Thus, $c = \lfloor 100/n \rfloor$ for appreciation of friends and aversion to enemies, and $c = 100$ for balanced preferences.

This scaling preserves the labels of our instances, although it changes the numerical gaps. For $c > 0$, scaling all utilities by c scales every utility difference in the definition of the gap by the same factor. Since multiplication by a positive constant commutes with the maxima and minima in that definition,

$$\mathrm{Gap}_k(\pi; c\mathbf{v}) = c\,\mathrm{Gap}_k(\pi; \mathbf{v}) \quad \text{and} \quad \mathrm{Gap}_k(\pi; c\mathbf{v}) \ge \varepsilon \iff \mathrm{Gap}_k(\pi; \mathbf{v}) \ge \varepsilon/c.$$

In our games, the unscaled utilities and gaps are integers, $c \ge 1$, and $\varepsilon = 1$. Hence, $0 < \varepsilon/c \le 1$, and an integer is at least ε/c if and only if it is at least 1. Similarly, an integer is negative before scaling if

and only if it remains negative afterwards. Thus, scaling preserves all three instance labels: YES, intermediate and NO.

It does, however, affect the difficulty of learning because it determines the magnitudes of the gaps relative to the observation noise $\sigma = 3$ and the stability margin $\varepsilon = 1$. The benchmark utilities are bounded in absolute value by 100, matching the range of the synthetic Uniform model, but their distribution and the resulting gaps are different. Regret is measured in utility units, and the three preference types have different scales; we therefore compare regret across methods within each type, rather than directly across types.

Ties and strict stability. Because these games use only three possible utility values, agents are often indifferent between different coalitions. For example, an agent who is neutral towards every member of another coalition loses nothing by joining it. Under balanced preferences, an agent with equally many friends and enemies in her coalition loses nothing by leaving. Under aversion to enemies, an agent is indifferent between coalitions containing the same number of friends and no enemies. Any such move gives a deviation gap of at most 0.

Consequently, a game may admit partitions that are exactly k-strongly Nash stable but none that is (ε, k)-strongly Nash stable for $\varepsilon > 0$. Such a game is labelled intermediate (Appendix I). The distinction is important for interpreting the results: a declaration on an intermediate game is not a non-existence error, since an exactly stable partition does exist, but it does not establish the positive margin required by our existence criterion.

M.3 Sampling and Protocol

Sampling. Each game contains $n \in \{8, 10, 12\}$ sampled agents. We calculate distances and the diameter using the full dataset, so the relationship between two sampled agents does not depend on the other agents selected. Each game is therefore the restriction of the game constructed by Csáji et al. [36] on the full dataset to the sampled agents.

We sample Iris flowers stratified by species and karate club members stratified by the two groups, uniformly within each class, so that every class is represented. When n is not divisible by three, the remaining Iris flowers are assigned to species selected uniformly at random. The jazz network has no class labels, so we use two sampling procedures: *Uniform* sampling selects n nodes uniformly at random, whereas *Snowball* sampling starts from a uniformly chosen node and adds its neighbours in random order, breadth first, until n nodes have been selected. Each combination of dataset, sampling procedure and size uses 100 seeds. For each seed, the same sampled agents are used to construct all three preference types.

Learning. All experiments use the default parameters of the synthetic experiments: $T = 30$, $k = 2$, $\sigma = 3$, $\beta = 1$ and $\varepsilon = 1$. As in the synthetic experiments, $\beta = 1$ lies outside the range $\beta > 2$ covered by our theorems. Under semi-bandit feedback, we run UCB-LCB-GS with witness probing and the six baselines of Section 5. Under coalition-level bandit feedback, we run the corresponding bandit variant and adapt the same six baselines: they retain their decision rules, use the estimator based on two-agent coalitions, and begin with the round-robin matching schedule (Appendix L).

We use the same implementations, local-search heuristic, exact deviation checker and hyperparameters as in the synthetic experiments; no method is tuned specifically to these games. All methods and both feedback models use the same games and environment and algorithm seeds. The noise realizations are not shared observation by observation, however, because the two feedback models observe different quantities.

Metrics. As elsewhere in the paper, regret and existence error are averaged over YES games, and non-existence error over NO games. Neither metric is defined on intermediate games; for these, we report the frequency with which each method declares existence as a separate diagnostic. Standard errors are computed over the games contributing to each statistic, and we report the corresponding sample counts.

M.4 Exact Labels

We determine the label of every game exactly. A unilateral deviation is a group deviation of size one, so

$$\mathrm{Gap}_2(\pi) \le \mathrm{Gap}_1(\pi)$$

for every partition π. Define

$$S_\varepsilon = \{\pi : \mathrm{Gap}_1(\pi) \ge \varepsilon\}, \qquad S_0 = \{\pi : \mathrm{Gap}_1(\pi) \ge 0\}.$$

Every partition with $\mathrm{Gap}_2(\pi) \ge \varepsilon$ belongs to S_ε, and every partition with $\mathrm{Gap}_2(\pi) \ge 0$ belongs to S_0. We first compute Gap_1 for all partitions, and then evaluate Gap_2 on S_ε. A game is labelled YES if some partition in this set has $\mathrm{Gap}_2(\pi) \ge \varepsilon$; the maximum over this set gives its largest gap. If no such partition exists, we evaluate Gap_2 on S_0: the game is intermediate if some partition there has a non-negative gap, and NO otherwise. No partition outside these sets can change the label. As a validation, for $n = 8$ we also computed Gap_2 for every partition of all $1{,}200$ games, and obtained identical labels.

Table 5 summarizes the resulting classification. Of the $3{,}600$ games, $2{,}271$ are YES instances and $1{,}329$ are intermediate; none is a NO instance. Thus, these games provide no instances on which to evaluate non-existence error. The distribution of labels is closely related to the frequency of ties in the constructed utilities. Under appreciation of friends, an agent with a friend in her coalition strictly prefers it to any coalition without friends. Ties are therefore relatively rare, and games derived from Iris, the karate club and snowball samples of jazz are almost all YES instances. Under balanced preferences and aversion to enemies, indifference is more common, so fewer games are YES, particularly in the karate club, where sampled members have relatively few ties with one another. Uniform samples of jazz are mostly intermediate under all three types because more than half of their pairs are neutral.

We also examined alternative thresholds to understand the role of the neutral zone. With $(f, e) = (0.2, 0.2)$, there are no neutral pairs, and all 300 uniform jazz samples are YES under appreciation of friends; the corresponding numbers under balanced preferences and aversion to enemies are 134 and 127. With $(f, e) = (0.4, 0.4)$, which also makes pairs at distance two in the networks friends, the number of YES instances increases further: among uniform jazz samples it is 300, 270 and 228 under the three preference types, respectively, and among karate club samples it is 300, 232 and 202. None of these alternative threshold settings produces a NO instance.

The intermediate games are not misclassified. Their largest gap is exactly 0: they admit exactly 2-strongly Nash stable partitions, but every such partition has a deviation in which no deviator loses and at least one is indifferent. Such a deviation is not profitable, since exact k-strong Nash stability requires only that at least one deviator does not strictly improve (Definition 2). These games are therefore exactly stable, but none has a positive margin of $\varepsilon = 1$. On rounds with valid confidence intervals, UCB-LCB-GS cannot declare existence on these games through its pessimistic test, whereas methods that declare based on less conservative estimates may do so. Such declarations are not counted by either error metric, because an exactly stable partition exists, but they provide a useful diagnostic of how readily the different methods make existence decisions. Intermediate games are particularly common here because utilities take only three values.

The preference type also has a strong effect on the size of the certification margins. Under balanced preferences, every gap is a multiple of $c = 100$, so every YES instance has a largest gap of at least 100. Under aversion to enemies, the largest gap of a YES instance is often just one friend utility, 8 to 12, with medians between 10 and 24 across configurations. Under appreciation of friends, the median largest gap ranges from 10 in the karate club to 288 in snowball samples of the jazz network. With observation noise $\sigma = 3$, margins of only one friend utility are substantially harder to establish than the larger margins typical of the balanced and appreciation-of-friends games. These differences help explain the variation in learning performance across preference types.

M.5 Example Games

Figures 18 and 19 illustrate one game with $n = 12$ from each configuration, using the first seed. For configurations in which UCB-LCB-GS declared existence under semi-bandit feedback with appreciation of friends, the figures also show the partition formed at its last declaration. We recomputed the exact gap of each displayed partition.

Table 5: Number of YES instances among the 100 games of each configuration ($k = 2$, $\varepsilon = 1$); all other games are intermediate instances, and none is a NO instance. Neutral: average share of neutral pairs under the thresholds $(f, e) = (0.25, 0.35)$ used for learning. The last two columns give the number of YES instances among the 300 games of a row, labelled again under the other two thresholds of Csáji et al. [36], which have no neutral zone; no learning runs use them.

			$(f, e) = (0.25, 0.35)$				$(0.2, 0.2)$	$(0.4, 0.4)$
dataset	neutral	type	$n = 8$	$n = 10$	$n = 12$	all	all	all
Iris	10%	friends	97	100	100	297	300	300
		enemies	84	88	91	263	220	289
		balanced	74	79	70	223	194	281
Karate club	0%	friends	100	100	100	300	300	300
		enemies	44	28	21	93	93	232
		balanced	44	22	14	80	80	202
Jazz, uniform	54%	friends	18	20	20	58	300	300
		enemies	10	13	7	30	134	270
		balanced	10	13	6	29	127	228
Jazz, snowball	29%	friends	100	100	100	300	300	300
		enemies	100	100	100	300	252	300
		balanced	99	99	100	298	214	298

In Iris (Figure 18(a,c)), the four sampled setosa flowers are friends of one another and enemies or neutral towards the other flowers, reflecting the separation of setosa from the other species in the measurements. Versicolor and virginica overlap: among their eight sampled flowers, 20 of the 28 pairs are friends, 5 are neutral and 3 are enemies, including several pairs across species. The resulting relation matrix therefore has one block of mutual friends for setosa and a larger, mostly friendly block containing versicolor and virginica. The partition declared by UCB-LCB-GS consists of precisely these two coalitions and has gap 168. Panel (a) displays only petal length and width, so it gives an approximate view of the flowers' positions; the relationships themselves are determined using all four measurements.

In the karate club (Figure 18(b,d)), friends are sampled members connected by an edge, and all other sampled pairs are enemies. Most sampled members of the first group are connected to one another, whereas the sampled members of the second group have no ties among themselves, apart from one member connected to the first group. The partition declared by UCB-LCB-GS places the members of the first group and that connected member together, and leaves the remaining members alone. Its gap is 8. Note that panel (b) displays the full karate club network, whereas the game itself contains only the relationships among the 12 sampled members.

The two jazz sampling procedures produce markedly different structures (Figure 19). A uniform sample draws nodes from across the network: among the 66 pairs in the first-seed game, 10 are adjacent and hence friends, 21 are at distance two and hence neutral, and 35 are enemies. This game is intermediate under all three preference types. A snowball sample, by contrast, forms a connected neighbourhood. In its first-seed game, 38 pairs are friends, 28 are neutral and none are enemies. All three preference types then admit a positive-margin stable partition. Under appreciation of friends, UCB-LCB-GS declares the grand coalition, whose exact gap is 288. Thus, the sampling procedure affects not only the relationships within a game but also whether a positive-margin stable partition exists, and, as the next sections show, how difficult it is to learn one. These figures illustrate the construction of the games; they do not represent the preferences or behaviour of the flowers, club members or bands themselves.

M.6 Learning under Semi-Bandit Feedback

We first compare the seven methods under semi-bandit feedback. Figures 20 and 21 show regret and existence error on YES games as functions of n. The corresponding values, together with standard errors and the number of YES games contributing to each result, are reported in Tables 7 and 8. Declarations on intermediate games are reported separately in Table 6.

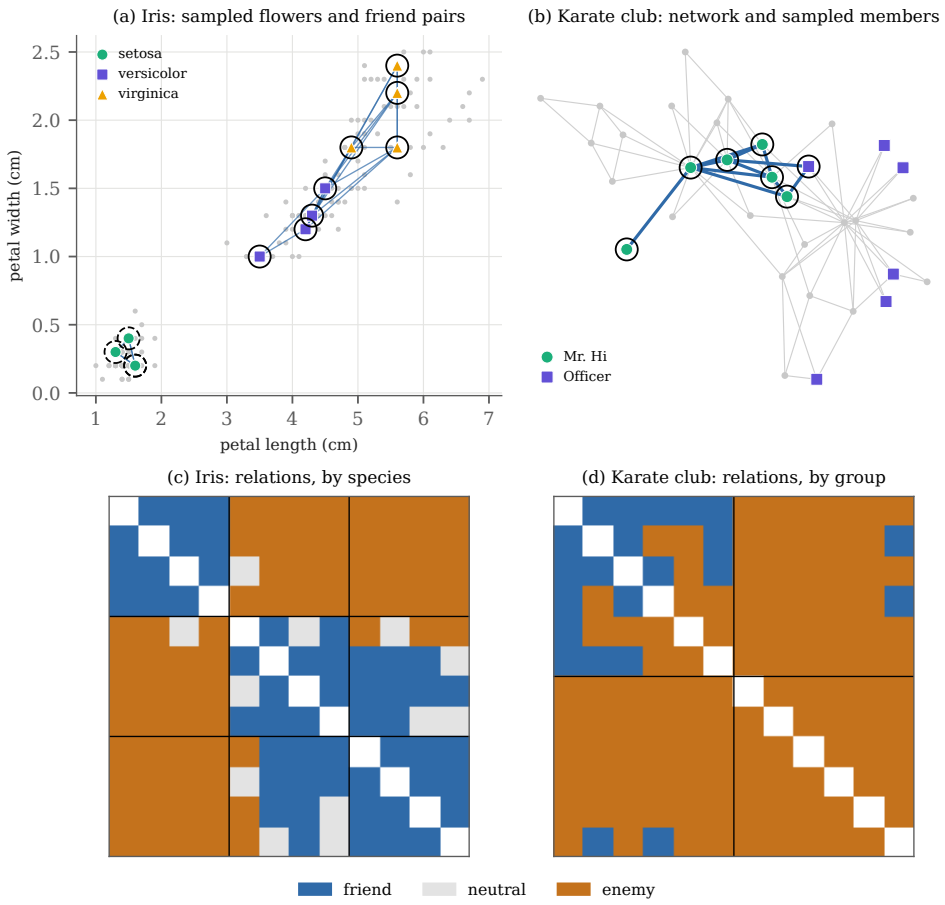

Figure 18: Games with $n = 12$ from Iris and the karate club (first seed). (a) The 150 flowers plotted by petal length and width, two of the four measurements used to determine distances; the 12 sampled flowers are coloured by species, and lines join friend pairs. (b) The full karate club network (grey), with the 12 sampled members coloured by group; blue lines indicate friend pairs in the game, corresponding to ties between sampled members. In (a) and (b), circles mark coalitions with more than one member in the partition declared by UCB-LCB-GS under semi-bandit feedback and appreciation of friends. The exact gaps are 168 in (a) and 8 in (b). Panels (c) and (d) show the relations between sampled agents, ordered by class.

The comparison with PESSIMISTIC-ONLY and RANDOM highlights the contribution of witness probing. Both methods have substantially higher regret than UCB-LCB-GS in every configuration: at least 6 and 7 times as high, respectively, wherever the regret of UCB-LCB-GS is positive. Moreover, the mean existence error of UCB-LCB-GS is no higher than that of PESSIMISTIC-ONLY in any configuration, and is usually lower. Thus, the improvement over purely pessimistic selection is not obtained by sacrificing reliable decisions: optimistic candidate selection and targeted probes allow UCB-LCB-GS to establish stability earlier, while retaining a pessimistic test.

The comparison with all seven methods is more nuanced. UCB-LCB-GS has strictly the lowest mean regret in 13 of the 36 configurations, all under appreciation of friends or balanced preferences, and ties for the lowest regret in another 9. These ties arise when several methods form identical partitions. This occurs, for example, under balanced preferences, where UCB-LCB-GS and several baselines declare a stable partition from the second round onwards, and on snowball samples under appreciation of friends, where several methods attain zero regret. In the remaining 14 configurations, another method has lower regret: LCB-ONLY, MIXED, OPTIMISTIC-ONLY or UCB-ONLY in all 12 configurations with aversion to enemies, and MIXED in two configurations with balanced preferences and $n = 8$, by less than 1%.

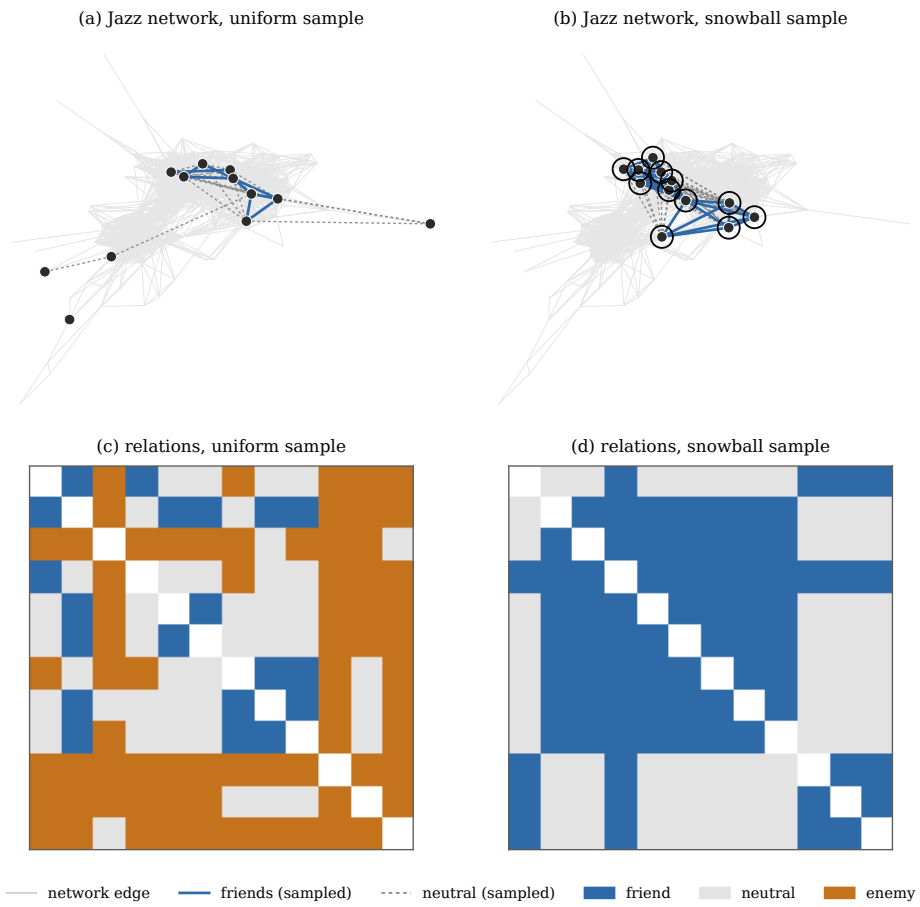

Figure 19: Games with $n = 12$ from the jazz network (first seed), under uniform and snowball sampling. Panels (a) and (b) show the network (light grey) and sampled nodes; blue lines join friend pairs (adjacent nodes), and dotted lines indicate neutral pairs (distance two). Enemy pairs are not drawn. In (b), circles mark the partition declared by UCB-LCB-GS under semi-bandit feedback and appreciation of friends: the grand coalition, with exact gap 288. For the intermediate game in (a), the algorithm made no declaration. Panels (c) and (d) show the relations between sampled nodes.

The aversion-to-enemies results help explain where UCB-LCB-GS loses this comparison. In these games, the certification margins are often only one or two friend utilities. UCB-LCB-GS continues probing until its pessimistic test has sufficient evidence to establish such a margin; on uniform jazz samples, it rarely obtains enough evidence within the 30-round horizon. The other methods can attain lower regret by forming partitions suggested by their optimistic or two-sided estimates without requiring a successful pessimistic test. Their lower regret, however, comes with less reliable existence decisions.

Indeed, the lower-regret methods declare existence much more often on games without a positive-margin stable partition. OPTIMISTIC-ONLY, UCB-ONLY, MIXED and LCB-ONLY have existence errors of at most about six rounds on YES games, largely because they declare existence readily. The same behaviour appears on intermediate games: LCB-ONLY declares existence in all 30 rounds, OPTIMISTIC-ONLY and MIXED in an average of 29 rounds, and UCB-ONLY in 19 rounds (Table 6). In particular, under aversion to enemies, these methods form and declare partitions that look stable under their respective estimates without requiring a pessimistic test. Their lower regret therefore does not reflect more reliable detection of positive-margin stability. The results illustrate why regret and existence decisions must be considered together.

UCB-LCB-GS itself made a small number of declarations on intermediate games under semi-bandit feedback: 46 declarations in the 39,870 rounds played on these games, distributed across 10 of the

1,329 intermediate games. In eight of these games, it declared existence in at most three rounds, all before round six. The other two cases are the same jazz sample under two different preference types, with 17 declarations each. We recomputed the exact gap of every partition formed in these rounds, and all had gap exactly 0. Thus, these declarations were made on partitions that are exactly 2-strongly Nash stable, as required by the exact stability criterion, but have no positive margin of $\varepsilon = 1$. They are not non-existence errors, which concern games without an exactly stable partition, and they do not contradict Theorem 3.1, whose assumptions include $\beta > 2$.

They nevertheless warrant attention because the experiments use $\beta = 1$. The confidence intervals were not recorded in these runs, so we cannot attribute these declarations to a particular failed interval. We can only note that, at $\beta = 1$, the bound $n(n-1)t^{1-\beta}$ of Lemma D.1 on the probability that some confidence interval fails is not informative. Neither PESSIMISTIC-ONLY nor RANDOM declared existence on any intermediate game.

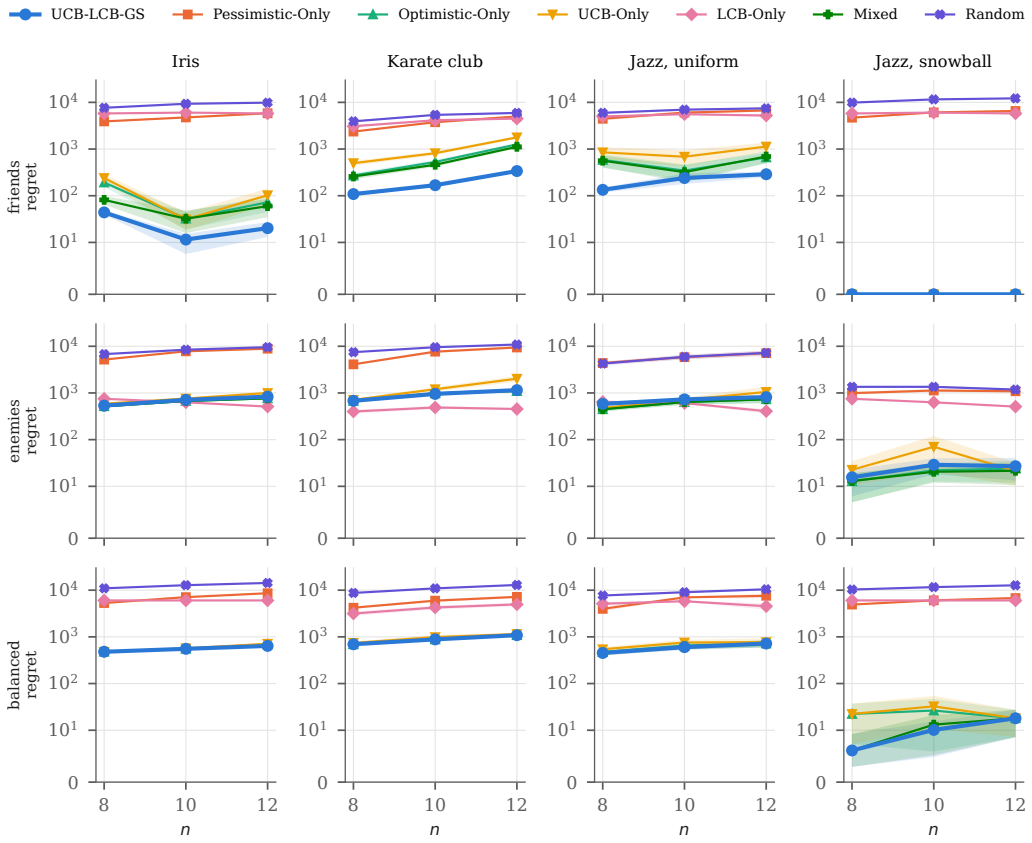

Figure 20: Regret on YES games under semi-bandit feedback, as a function of n, for the four dataset configurations (columns) and the three preference types (rows), on a symmetric logarithmic scale. Each point averages over the YES games among 100 games; points with fewer than 5 YES games are omitted (Table 7 reports all values). Shaded bands show ± 1 standard error.

M.7 Learning under Coalition-Level Bandit Feedback

Under coalition-level feedback, each agent reports only a noisy aggregate of her utility from the current coalition. Consequently, the learner can recover a pairwise utility only when the corresponding agents form a two-agent coalition. Figures 22 and 23 and Tables 9 and 10 compare all seven methods under this feedback model.

Every learning method begins with the $n - 1$ round-robin matchings, since no pairwise estimate is available before the corresponding pair has formed a two-agent coalition. No method declares existence during these initialization rounds. Their contribution to existence error is therefore at least

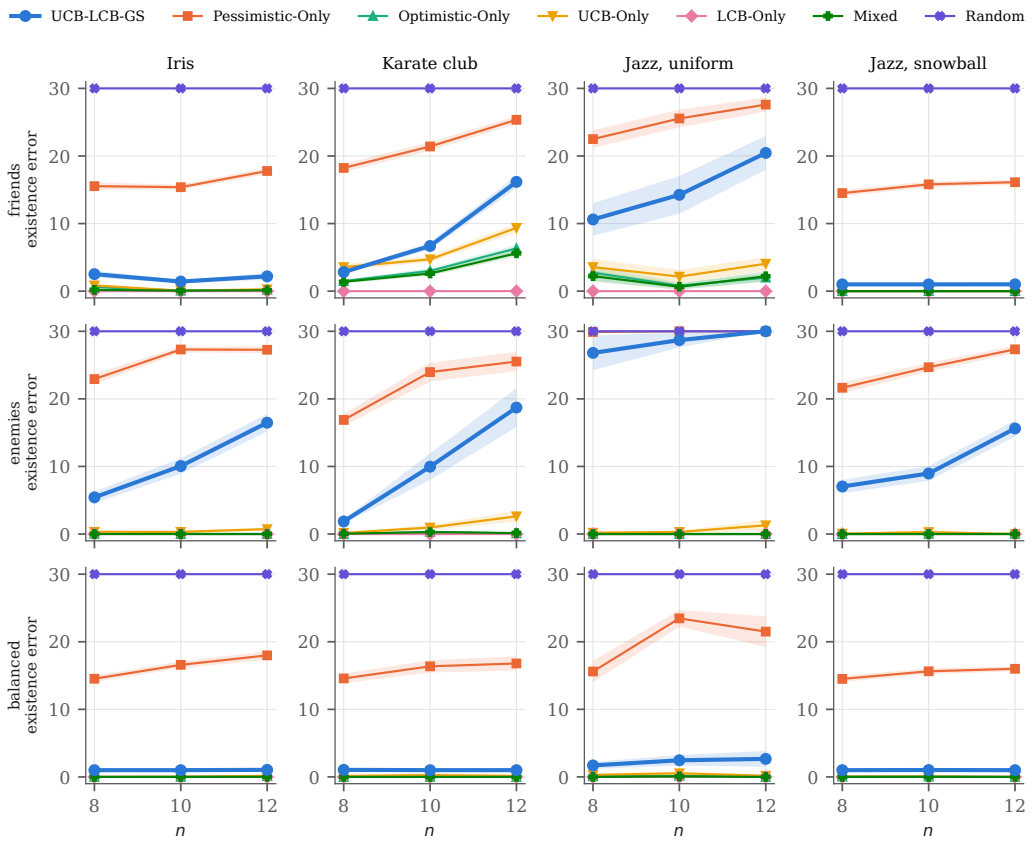

Figure 21: Existence error on YES games under semi-bandit feedback, in rounds out of 30; layout as in Figure 20.

Table 6: Declarations on intermediate games: average number of rounds, out of 30, in which each method declares existence on a game that has an exactly 2-strongly Nash stable partition but no $(\varepsilon, 2)$-strongly Nash stable one, pooled over $n \in \{8, 10, 12\}$; int.: number of intermediate games. These declarations are not non-existence errors, which are defined on games without an exactly stable partition. Rows without intermediate games are omitted; abbreviations as in Table 7.

feedback	dataset	type	int.	UCB-LCB-GS	PESS.	OPT.	UCB-O.	LCB-O.	MIXED	RANDOM
semi-bandit	Iris	friends	3	0.00	0.00	24.0	23.0	30.0	29.7	0.00
semi-bandit	Iris	enemies	37	0.00	0.00	30.0	26.5	30.0	30.0	0.00
semi-bandit	Iris	balanced	77	0.00	0.00	30.0	26.8	30.0	29.8	0.00
semi-bandit	Karate club	enemies	207	0.04	0.00	29.9	7.7	30.0	29.9	0.00
semi-bandit	Karate club	balanced	220	0.00	0.00	29.5	14.9	30.0	29.6	0.00
semi-bandit	Jazz, uniform	friends	242	0.08	0.00	27.4	18.7	30.0	27.2	0.00
semi-bandit	Jazz, uniform	enemies	270	0.00	0.00	29.4	23.6	30.0	29.4	0.00
semi-bandit	Jazz, uniform	balanced	271	0.07	0.00	29.1	23.3	30.0	29.0	0.00
semi-bandit	Jazz, snowball	balanced	2	0.00	0.00	30.0	30.0	30.0	30.0	0.00
bandit	Iris	friends	3	0.00	0.00	22.7	17.7	21.7	22.7	0.00
bandit	Iris	enemies	37	0.00	0.00	21.4	21.1	18.9	21.4	0.00
bandit	Iris	balanced	77	0.00	0.00	20.9	20.4	20.1	20.9	0.00
bandit	Karate club	enemies	207	0.00	0.00	20.8	3.1	11.6	20.8	0.00
bandit	Karate club	balanced	220	0.00	0.00	20.6	13.8	13.1	20.6	0.00
bandit	Jazz, uniform	friends	242	0.00	0.00	20.1	17.2	18.6	20.1	0.00
bandit	Jazz, uniform	enemies	270	0.00	0.00	20.5	19.2	14.2	20.4	0.00
bandit	Jazz, uniform	balanced	271	0.00	0.00	20.1	18.2	18.6	20.1	0.00
bandit	Jazz, snowball	balanced	2	0.00	0.00	22.0	22.0	21.5	22.0	0.00

7, 9 or 11 rounds for $n = 8, 10, 12$, respectively. These rounds also contribute substantially to regret: every agent is paired with only one other agent, and the resulting matching is usually far from stable.

The initialization cost is particularly visible under balanced preferences and on snowball samples under appreciation of friends. In these configurations, UCB-LCB-GS declares existence in almost every round after initialization, as do most of the learning baselines. Their regret is therefore largely determined by the initial matching schedule, and several methods attain exactly the same regret. Once the initial pairwise estimates are available, these games are sufficiently easy for several methods to identify stable partitions quickly.

Across all configurations, UCB-LCB-GS never has strictly the lowest regret under coalition-level feedback. It ties for the lowest value in 9 configurations, all among the relatively easy cases described above, while a baseline has lower regret in the remaining 27, most often MIXED. Nevertheless, UCB-LCB-GS retains an advantage over the methods that avoid unsupported declarations: its mean existence error is no higher than that of PESSIMISTIC-ONLY in every configuration, and its regret is lower than that of both PESSIMISTIC-ONLY and RANDOM in 30 configurations. It ties with PESSIMISTIC-ONLY in another four. Only in two configurations, snowball samples of the jazz network under aversion to enemies with $n = 10$ and $n = 12$, does its regret exceed that of PESSIMISTIC-ONLY, by 15% and 33%, respectively.

These two cases illustrate the cost of witness probing particularly clearly. In the corresponding jazz games, most pairs are friends and almost none are enemies, so partitions formed by the optimism-based baselines are already close to stable after initialization. The margins required by UCB-LCB-GS's pessimistic test, however, are often one friend utility, and the learner rarely establishes them within 30 rounds. It consequently spends almost every remaining round on probes. Each probe removes the two agents of a witness pair from their original coalitions and places them together in a separate two-agent coalition, which can incur substantial regret in these games. PESSIMISTIC-ONLY does not perform this targeted exploration and instead forms its pessimistic candidate, which explains why it can achieve lower regret in these two cases. This is a genuine limitation of the exploration strategy under coalition-level feedback, not evidence that the baseline has a more reliable existence test.

As under semi-bandit feedback, the lower regret of OPTIMISTIC-ONLY, UCB-ONLY, MIXED and LCB-ONLY comes with frequent declarations on intermediate games: between 15 and 21 of the 30 rounds on average (Table 6). By contrast, UCB-LCB-GS, PESSIMISTIC-ONLY and RANDOM make no declarations on these games. Thus, even in this more difficult feedback model, the distinction between methods that require pessimistic evidence and those that do not remains visible. UCB-LCB-GS pays a substantial regret cost for its probes, but it does not obtain its existence decisions by declaring on games that have no positive-margin stable partition.

M.8 Feedback Model and Game Size

Figure 24 compares UCB-LCB-GS under the two feedback models. On the same YES games, its regret under coalition-level feedback exceeds that under semi-bandit feedback in every configuration, by 106 to 5,194 on average. The corresponding increase in existence error ranges from 0 to 16 rounds. The difference reflects several costs: the $n - 1$ initialization rounds, the slower accumulation of pairwise estimates because only two-agent coalitions provide individual pairwise observations, and the regret incurred by the additional probes needed to obtain those observations. The experiments do not separate these contributions, however, and the two feedback models also use different noise realizations. The comparison should therefore be interpreted as a comparison of the two implemented feedback models, rather than as an isolation of any one of these costs.

The results also show how learning varies with the number of agents. Existence error increases with n in most configurations, most visibly in the karate club under appreciation of friends, where it rises from 2.8 to 16.2 rounds between $n = 8$ and $n = 12$ under semi-bandit feedback. Under balanced preferences, it remains between one and three rounds. Larger games have more pairwise utilities to learn and more possible deviations to consider, which can make stability harder to establish. However, the sampled agents differ across values of n, so these comparisons describe the particular sampled games and do not isolate a causal effect of increasing the number of agents.

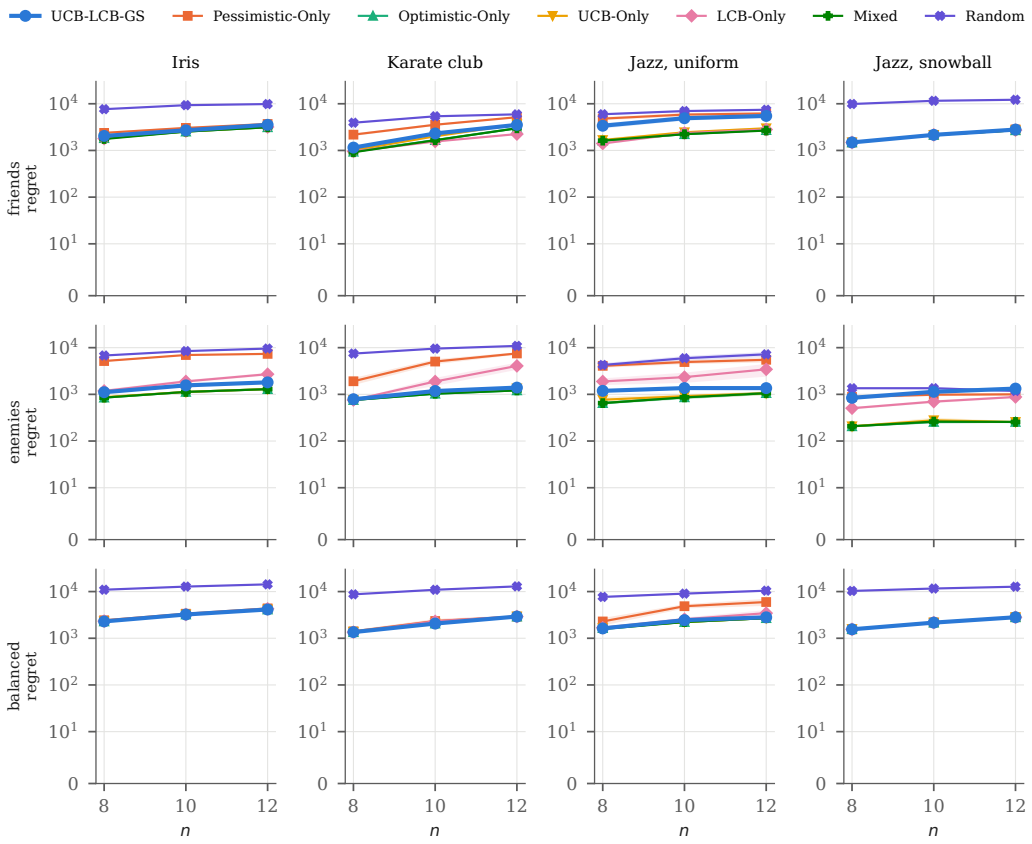

Figure 22: Regret on YES games under coalition-level bandit feedback; layout as in Figure 20.

M.9 The Declared Partitions

For Iris and the karate club, the known classes allow us to describe the partitions declared by UCB-LCB-GS under semi-bandit feedback. We consider the last round in which it declared existence. These classes are used only for this descriptive analysis; they are not treated as stable partitions of the constructed games and do not enter the performance evaluation.

In Iris, the sampled setosa flowers form exactly one coalition in 88%–100% of the declared partitions, depending on the preference type and game size. Versicolor and virginica, whose measurements overlap, are rarely separated (Figure 18(a)). This is consistent with the relationships induced by the feature distances: setosa is separated from the other two species, while versicolor and virginica have substantial overlap.

The karate club results are less closely aligned with the two historically observed groups. Under appreciation of friends and $n = 12$, the declared partitions contain an average of 4.7 coalitions, rather than two. This reflects the construction of the utilities: two sampled members without a tie are enemies, and members with few ties to the other sampled members often remain alone (Figure 18(b)). The resulting partitions therefore reflect the local network structure among the sampled members, rather than reproducing the historical split of the club.

M.10 Limitations

These experiments examine constructed games, not preferences elicited from the underlying entities. The utilities are derived from distances in the datasets following Csáji et al. [36], and should not be interpreted as statements about the preferences or behaviour of flowers, club members or bands. The results also depend on the distance thresholds, preference types, utility scaling and sampling procedures. Moreover, the games contain only small subsets of the original datasets.

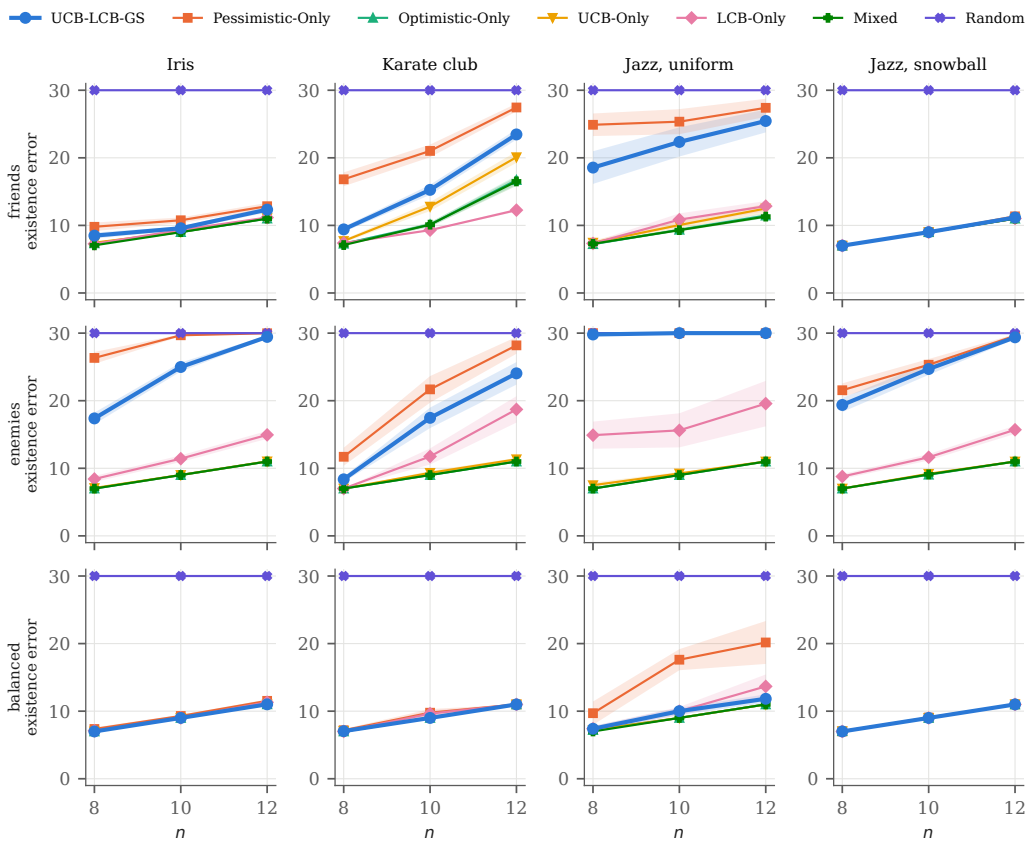

Figure 23: Existence error on YES games under coalition-level bandit feedback, in rounds out of 30; layout as in Figure 20. Every learning method spends the first $n-1$ rounds on round-robin matchings without declaring existence.

Because the utilities take only three values, indifference is common. Many games are consequently intermediate instances, and none is a NO instance. These experiments therefore do not measure non-existence error. Finally, as in the synthetic experiments, $\beta = 1$ lies outside the range covered by our theorems, and the learner uses local search rather than the exact optimization assumed by the theoretical analysis. The results should be interpreted with these limitations in mind.

M.11 Complete Results

Tables 7 to 10 report the regret and existence error of all seven methods for every configuration, preference type and game size, together with the number of YES games contributing to each value. The figures in Appendices M.6 and M.7 are generated from these results.

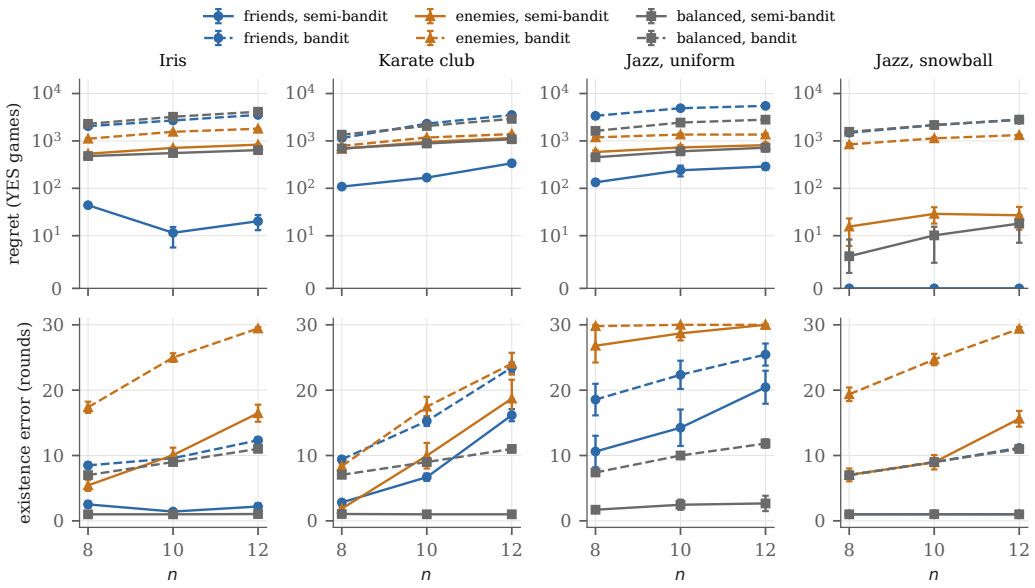

Figure 24: UCB-LCB-GS under both feedback models as a function of n. Top: regret on YES games (symmetric logarithmic scale). Bottom: existence error, in rounds out of 30. Colours: preference type; solid lines: semi-bandit feedback; dashed lines: coalition-level bandit feedback. Points with fewer than 5 YES games are omitted. Error bars: ± 1 standard error.

Table 7: Regret on YES games under semi-bandit feedback ($T = 30$, $k = 2$, $\sigma = 3$, $\beta = 1$, $\varepsilon = 1$): mean and, in parentheses, standard error over the YES games among the 100 games of each row; YES gives their number. Abbreviations: PESS., OPT., UCB-O. and LCB-O. denote PESSIMISTIC-ONLY, OPTIMISTIC-ONLY, UCB-ONLY and LCB-ONLY.

dataset	type	n	YES	UCB-LCB-GS	PESS.	OPT.	UCB-O.	LCB-O.	MIXED	RANDOM
Iris	friends	8	97	44 (5)	3,905 (134)	192 (39)	240 (47)	5,790 (0)	82 (17)	7,654 (107)
		10	100	12 (4)	4,756 (123)	32 (13)	31 (13)	6,030 (0)	32 (16)	9,320 (85)
		12	100	20 (7)	5,838 (167)	72 (27)	103 (40)	5,790 (0)	60 (25)	9,850 (75)
Iris	enemies	8	84	536 (8)	5,180 (178)	519 (7)	589 (26)	750 (0)	516 (6)	6,789 (49)
		10	88	712 (11)	7,794 (144)	671 (7)	758 (37)	630 (0)	674 (8)	8,438 (64)
		12	91	833 (9)	8,870 (154)	767 (6)	998 (90)	510 (0)	765 (5)	9,566 (55)
Iris	balanced	8	74	479 (11)	5,306 (184)	480 (11)	480 (11)	6,030 (0)	479 (11)	10,953 (83)
		10	79	554 (12)	7,091 (219)	554 (12)	569 (20)	6,030 (0)	554 (12)	12,786 (93)
		12	70	640 (12)	8,594 (309)	694 (41)	710 (42)	6,030 (0)	648 (14)	14,276 (84)
Karate club	friends	8	100	108 (5)	2,365 (106)	270 (35)	499 (61)	3,080 (141)	258 (34)	3,922 (138)
		10	100	168 (10)	3,756 (151)	529 (54)	814 (73)	4,139 (158)	461 (45)	5,390 (148)
		12	100	338 (23)	4,992 (144)	1,258 (92)	1,780 (116)	4,494 (144)	1,116 (82)	5,935 (129)
Karate club	enemies	8	44	679 (2)	4,103 (212)	673 (0)	712 (31)	400 (38)	684 (11)	7,468 (58)
		10	28	953 (13)	7,664 (456)	983 (66)	1,201 (132)	489 (37)	980 (67)	9,540 (91)
		12	21	1,150 (17)	9,419 (523)	1,095 (39)	2,020 (287)	453 (23)	1,109 (39)	10,907 (132)
Karate club	balanced	8	44	694 (9)	4,212 (187)	692 (9)	742 (51)	3,160 (309)	692 (9)	8,721 (130)
		10	22	883 (13)	5,962 (388)	883 (12)	988 (106)	4,255 (427)	883 (13)	10,925 (177)
		12	14	1,087 (14)	7,188 (458)	1,087 (14)	1,151 (67)	4,959 (399)	1,087 (14)	12,930 (276)
Jazz, uniform	friends	8	18	134 (17)	4,453 (363)	584 (188)	850 (246)	4,990 (313)	563 (158)	5,963 (290)
		10	20	240 (62)	6,040 (354)	345 (128)	684 (257)	5,580 (246)	320 (127)	6,978 (261)
		12	20	289 (42)	6,712 (335)	661 (167)	1,130 (248)	5,214 (264)	693 (199)	7,442 (347)
Jazz, uniform	enemies	8	10	583 (49)	4,380 (325)	448 (46)	478 (42)	642 (55)	448 (47)	4,255 (418)
		10	13	728 (83)	5,845 (621)	638 (77)	709 (91)	607 (23)	639 (77)	5,940 (674)
		12	7	814 (109)	7,127 (901)	733 (126)	1,052 (314)	407 (48)	731 (124)	7,206 (886)
Jazz, uniform	balanced	8	10	452 (50)	3,976 (409)	472 (47)	542 (67)	5,130 (458)	452 (50)	7,680 (318)
		10	13	602 (80)	7,016 (303)	650 (96)	758 (108)	5,799 (231)	625 (88)	9,038 (411)
		12	6	719 (126)	7,572 (827)	720 (126)	770 (161)	4,530 (671)	719 (126)	10,447 (478)
Jazz, snowball	friends	8	100	0 (0)	4,700 (144)	0 (0)	0 (0)	5,790 (0)	0 (0)	9,878 (103)
		10	100	0 (0)	6,090 (175)	0 (0)	0 (0)	6,030 (0)	0 (0)	11,572 (101)
		12	100	0 (0)	6,538 (157)	0 (0)	0 (0)	5,790 (0)	0 (0)	12,184 (99)
Jazz, snowball	enemies	8	100	16 (8)	991 (62)	13 (6)	23 (12)	750 (0)	13 (6)	1,352 (43)
		10	100	29 (11)	1,127 (73)	22 (9)	70 (50)	630 (0)	20 (8)	1,354 (68)
		12	100	27 (14)	1,087 (76)	24 (12)	21 (11)	510 (0)	22 (11)	1,182 (68)
Jazz, snowball	balanced	8	99	6 (3)	4,915 (151)	22 (15)	22 (15)	6,030 (0)	6 (3)	10,318 (104)
		10	99	10 (5)	6,052 (176)	26 (21)	33 (22)	6,030 (0)	13 (8)	11,622 (95)
		12	100	18 (9)	6,788 (164)	18 (9)	18 (9)	6,030 (0)	18 (9)	12,719 (100)

Table 8: Existence error on YES games under semi-bandit feedback, in rounds out of 30; layout and abbreviations as in Table 7.

dataset	type	n	YES	UCB-LCB-GS	PESS.	OPT.	UCB-O.	LCB-O.	MIXED	RANDOM
Iris	friends	8	97	2.5 (0.5)	15.5 (0.5)	0.6 (0.1)	0.8 (0.2)	0.0 (0.0)	0.2 (0.1)	30.0 (0.0)
		10	100	1.4 (0.2)	15.4 (0.4)	0.1 (0.0)	0.1 (0.0)	0.0 (0.0)	0.1 (0.0)	30.0 (0.0)
		12	100	2.2 (0.5)	17.8 (0.5)	0.2 (0.1)	0.3 (0.1)	0.0 (0.0)	0.1 (0.1)	30.0 (0.0)
Iris	enemies	8	84	5.4 (0.9)	22.9 (0.8)	0.0 (0.0)	0.3 (0.1)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
		10	88	10.1 (1.1)	27.3 (0.4)	0.0 (0.0)	0.3 (0.1)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
		12	91	16.5 (1.3)	27.3 (0.5)	0.0 (0.0)	0.7 (0.3)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
Iris	balanced	8	74	1.0 (0.0)	14.5 (0.5)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
		10	79	1.0 (0.0)	16.6 (0.5)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
		12	70	1.0 (0.0)	18.0 (0.6)	0.1 (0.1)	0.2 (0.1)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
Karate club	friends	8	100	2.8 (0.2)	18.2 (0.6)	1.5 (0.2)	3.6 (0.4)	0.0 (0.0)	1.4 (0.2)	30.0 (0.0)
		10	100	6.7 (0.6)	21.4 (0.6)	3.0 (0.4)	4.7 (0.5)	0.0 (0.0)	2.6 (0.3)	30.0 (0.0)
		12	100	16.2 (0.9)	25.4 (0.6)	6.3 (0.5)	9.4 (0.7)	0.0 (0.0)	5.6 (0.5)	30.0 (0.0)
Karate club	enemies	8	44	1.9 (0.3)	16.9 (0.9)	0.0 (0.0)	0.2 (0.1)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
		10	28	10.0 (2.0)	24.0 (1.4)	0.3 (0.2)	1.0 (0.4)	0.0 (0.0)	0.3 (0.2)	30.0 (0.0)
		12	21	18.7 (2.9)	25.5 (1.4)	0.1 (0.1)	2.6 (0.8)	0.0 (0.0)	0.1 (0.1)	30.0 (0.0)
Karate club	balanced	8	44	1.0 (0.0)	14.6 (0.7)	0.0 (0.0)	0.1 (0.1)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
		10	22	1.0 (0.0)	16.4 (0.9)	0.0 (0.0)	0.3 (0.3)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
		12	14	1.0 (0.0)	16.8 (1.1)	0.0 (0.0)	0.1 (0.1)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
Jazz, uniform	friends	8	18	10.6 (2.4)	22.5 (1.3)	2.7 (1.1)	3.6 (1.2)	0.0 (0.0)	2.2 (0.8)	30.0 (0.0)
		10	20	14.2 (2.8)	25.6 (1.3)	0.8 (0.5)	2.1 (1.0)	0.0 (0.0)	0.7 (0.4)	30.0 (0.0)
		12	20	20.4 (2.5)	27.6 (1.1)	2.0 (0.6)	4.0 (1.0)	0.0 (0.0)	2.2 (0.8)	30.0 (0.0)
Jazz, uniform	enemies	8	10	26.8 (2.6)	29.9 (0.1)	0.0 (0.0)	0.2 (0.1)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
		10	13	28.7 (1.1)	30.0 (0.0)	0.0 (0.0)	0.3 (0.2)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
		12	7	30.0 (0.0)	30.0 (0.0)	0.0 (0.0)	1.3 (0.8)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
Jazz, uniform	balanced	8	10	1.7 (0.5)	15.6 (1.6)	0.1 (0.1)	0.3 (0.2)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
		10	13	2.5 (0.8)	23.5 (1.2)	0.2 (0.1)	0.5 (0.2)	0.0 (0.0)	0.1 (0.1)	30.0 (0.0)
		12	6	2.7 (1.2)	21.5 (2.3)	0.0 (0.0)	0.2 (0.2)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
Jazz, snowball	friends	8	100	1.0 (0.0)	14.5 (0.5)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
		10	100	1.0 (0.0)	15.8 (0.4)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
		12	100	1.0 (0.0)	16.1 (0.4)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
Jazz, snowball	enemies	8	100	7.0 (1.0)	21.6 (0.8)	0.0 (0.0)	0.1 (0.0)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
		10	100	9.0 (1.1)	24.7 (0.7)	0.0 (0.0)	0.3 (0.3)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
		12	100	15.6 (1.2)	27.3 (0.6)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
Jazz, snowball	balanced	8	99	1.0 (0.0)	14.5 (0.5)	0.1 (0.0)	0.1 (0.0)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
		10	99	1.0 (0.0)	15.6 (0.4)	0.1 (0.1)	0.1 (0.1)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)
		12	100	1.0 (0.0)	16.0 (0.4)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	30.0 (0.0)

Table 9: Regret on YES games under coalition-level bandit feedback ($T = 30$, $k = 2$, $\sigma = 3$, $\beta = 1$, $\varepsilon = 1$): mean and, in parentheses, standard error over the YES games among the 100 games of each row; YES gives their number. Abbreviations: PESS., OPT., UCB-O. and LCB-O. denote PESSIMISTIC-ONLY, OPTIMISTIC-ONLY, UCB-ONLY and LCB-ONLY.

dataset	type	n	YES	UCB-LCB-GS	PESS.	OPT.	UCB-O.	LCB-O.	MIXED	RANDOM
Iris	friends	8	97	2,045 (91)	2,376 (142)	1,830 (32)	1,848 (31)	1,794 (30)	1,758 (23)	7,654 (107)
		10	100	2,703 (83)	3,039 (122)	2,537 (19)	2,549 (21)	2,620 (45)	2,533 (19)	9,320 (85)
		12	100	3,538 (142)	3,691 (153)	3,136 (13)	3,159 (19)	3,178 (30)	3,134 (13)	9,850 (75)
Iris	enemies	8	84	1,112 (21)	5,170 (201)	852 (5)	872 (15)	1,184 (93)	851 (5)	6,789 (49)
		10	88	1,557 (22)	6,960 (63)	1,127 (5)	1,125 (5)	1,893 (142)	1,127 (5)	8,438 (64)
		12	91	1,807 (13)	7,361 (45)	1,292 (6)	1,297 (9)	2,704 (186)	1,293 (6)	9,566 (55)
Iris	balanced	8	74	2,295 (28)	2,430 (82)	2,295 (28)	2,295 (28)	2,329 (42)	2,295 (28)	10,953 (83)
		10	79	3,228 (23)	3,351 (81)	3,228 (23)	3,228 (23)	3,228 (23)	3,228 (23)	12,786 (93)
		12	70	4,131 (23)	4,366 (88)	4,134 (24)	4,134 (24)	4,273 (69)	4,128 (24)	14,276 (84)
Karate club	friends	8	100	1,151 (73)	2,179 (144)	939 (37)	1,011 (56)	957 (40)	915 (36)	3,922 (138)
		10	100	2,310 (155)	3,552 (192)	1,666 (61)	2,010 (95)	1,555 (47)	1,655 (57)	5,390 (148)
		12	100	3,531 (140)	5,149 (152)	2,989 (124)	3,612 (154)	2,241 (72)	2,984 (128)	5,935 (129)
Karate club	enemies	8	44	786 (14)	1,907 (330)	756 (7)	756 (7)	757 (8)	756 (7)	7,468 (58)
		10	28	1,178 (34)	5,041 (632)	1,032 (10)	1,118 (82)	1,886 (372)	1,032 (10)	9,540 (91)
		12	21	1,395 (32)	7,517 (467)	1,215 (10)	1,342 (79)	4,065 (711)	1,215 (10)	10,907 (132)
Karate club	balanced	8	44	1,355 (63)	1,407 (94)	1,350 (62)	1,380 (75)	1,350 (62)	1,350 (62)	8,721 (130)
		10	22	2,064 (118)	2,342 (253)	2,064 (118)	2,073 (121)	2,305 (239)	2,064 (118)	10,925 (177)
		12	14	2,918 (168)	2,918 (168)	2,918 (168)	2,918 (168)	2,918 (168)	2,918 (168)	12,930 (276)
Jazz, uniform	friends	8	18	3,382 (433)	4,807 (383)	1,628 (123)	1,670 (133)	1,410 (73)	1,622 (118)	5,963 (290)
		10	20	4,899 (554)	5,863 (499)	2,231 (81)	2,450 (122)	2,353 (183)	2,229 (81)	6,978 (261)
		12	20	5,483 (463)	6,195 (372)	2,730 (115)	2,983 (154)	2,804 (226)	2,666 (71)	7,442 (347)
Jazz, uniform	enemies	8	10	1,188 (45)	4,082 (371)	649 (38)	768 (125)	1,888 (311)	649 (38)	4,255 (418)
		10	13	1,363 (73)	4,920 (494)	864 (75)	914 (97)	2,338 (577)	861 (74)	5,940 (674)
		12	7	1,360 (65)	5,496 (689)	1,063 (87)	1,056 (91)	3,438 (1,047)	1,056 (88)	7,206 (886)
Jazz, uniform	balanced	8	10	1,627 (77)	2,290 (424)	1,648 (116)	1,648 (116)	1,680 (112)	1,577 (80)	7,680 (318)
		10	13	2,441 (154)	4,856 (511)	2,226 (82)	2,226 (82)	2,556 (232)	2,225 (82)	9,038 (411)
		12	6	2,812 (115)	5,970 (1,043)	2,696 (147)	2,696 (147)	3,453 (561)	2,695 (147)	10,447 (478)
Jazz, snowball	friends	8	100	1,486 (12)	1,486 (12)	1,486 (12)	1,486 (12)	1,486 (12)	1,486 (12)	9,878 (103)
		10	100	2,157 (23)	2,157 (23)	2,157 (23)	2,157 (23)	2,157 (23)	2,157 (23)	11,572 (101)
		12	100	2,798 (75)	2,827 (95)	2,703 (26)	2,703 (26)	2,729 (34)	2,703 (26)	12,184 (99)
Jazz, snowball	enemies	8	100	842 (56)	879 (68)	207 (8)	207 (8)	505 (29)	207 (8)	1,352 (43)
		10	100	1,134 (53)	983 (67)	256 (17)	279 (39)	699 (50)	261 (22)	1,354 (68)
		12	100	1,329 (24)	1,000 (60)	256 (11)	256 (11)	879 (44)	256 (11)	1,182 (68)
Jazz, snowball	balanced	8	99	1,555 (14)	1,555 (14)	1,567 (19)	1,567 (19)	1,555 (14)	1,555 (14)	10,318 (104)
		10	99	2,163 (24)	2,184 (32)	2,163 (24)	2,163 (24)	2,163 (24)	2,163 (24)	11,622 (95)
		12	100	2,817 (27)	2,821 (28)	2,817 (27)	2,817 (27)	2,817 (27)	2,817 (27)	12,719 (100)

Table 10: Existence error on YES games under coalition-level bandit feedback, in rounds out of 30; layout and abbreviations as in Table 9.

dataset	type	n	YES	UCB-LCB-GS	PESS.	OPT.	UCB-O.	LCB-O.	MIXED	RANDOM
Iris	friends	8	97	8.5 (0.5)	9.8 (0.6)	7.3 (0.1)	7.4 (0.1)	7.3 (0.1)	7.1 (0.0)	30.0 (0.0)
		10	100	9.6 (0.3)	10.8 (0.4)	9.0 (0.0)	9.0 (0.0)	9.3 (0.1)	9.0 (0.0)	30.0 (0.0)
		12	100	12.3 (0.4)	12.8 (0.5)	11.0 (0.0)	11.1 (0.0)	11.2 (0.1)	11.0 (0.0)	30.0 (0.0)
Iris	enemies	8	84	17.4 (0.9)	26.3 (0.9)	7.0 (0.0)	7.1 (0.1)	8.4 (0.4)	7.0 (0.0)	30.0 (0.0)
		10	88	25.0 (0.7)	29.7 (0.2)	9.0 (0.0)	9.0 (0.0)	11.4 (0.5)	9.0 (0.0)	30.0 (0.0)
		12	91	29.4 (0.2)	30.0 (0.0)	11.0 (0.0)	11.0 (0.0)	14.9 (0.6)	11.0 (0.0)	30.0 (0.0)
Iris	balanced	8	74	7.0 (0.0)	7.4 (0.2)	7.0 (0.0)	7.0 (0.0)	7.1 (0.1)	7.0 (0.0)	30.0 (0.0)
		10	79	9.0 (0.0)	9.3 (0.2)	9.0 (0.0)	9.0 (0.0)	9.0 (0.0)	9.0 (0.0)	30.0 (0.0)
		12	70	11.0 (0.0)	11.5 (0.2)	11.0 (0.0)	11.0 (0.0)	11.3 (0.1)	11.0 (0.0)	30.0 (0.0)
Karate club	friends	8	100	9.4 (0.4)	16.8 (1.0)	7.3 (0.1)	7.7 (0.2)	7.4 (0.1)	7.1 (0.0)	30.0 (0.0)
		10	100	15.3 (0.7)	21.0 (0.9)	10.1 (0.3)	12.8 (0.7)	9.3 (0.1)	10.1 (0.3)	30.0 (0.0)
		12	100	23.5 (0.8)	27.4 (0.6)	16.7 (0.7)	20.1 (0.8)	12.3 (0.2)	16.5 (0.7)	30.0 (0.0)
Karate club	enemies	8	44	8.4 (0.4)	11.7 (1.3)	7.0 (0.0)	7.0 (0.0)	7.0 (0.0)	7.0 (0.0)	30.0 (0.0)
		10	28	17.5 (1.5)	21.7 (2.0)	9.0 (0.0)	9.3 (0.3)	11.8 (1.2)	9.0 (0.0)	30.0 (0.0)
		12	21	24.0 (1.7)	28.2 (1.2)	11.0 (0.0)	11.3 (0.2)	18.7 (2.0)	11.0 (0.0)	30.0 (0.0)
Karate club	balanced	8	44	7.0 (0.0)	7.2 (0.2)	7.0 (0.0)	7.1 (0.1)	7.0 (0.0)	7.0 (0.0)	30.0 (0.0)
		10	22	9.0 (0.0)	9.8 (0.5)	9.0 (0.0)	9.0 (0.0)	9.6 (0.5)	9.0 (0.0)	30.0 (0.0)
		12	14	11.0 (0.0)	11.0 (0.0)	11.0 (0.0)	11.0 (0.0)	11.0 (0.0)	11.0 (0.0)	30.0 (0.0)
Jazz, uniform	friends	8	18	18.6 (2.4)	24.9 (1.7)	7.2 (0.2)	7.4 (0.3)	7.3 (0.2)	7.3 (0.2)	30.0 (0.0)
		10	20	22.4 (2.2)	25.4 (1.9)	9.3 (0.2)	10.1 (0.6)	10.8 (0.9)	9.3 (0.2)	30.0 (0.0)
		12	20	25.4 (1.7)	27.4 (1.3)	11.4 (0.4)	12.5 (0.7)	12.8 (0.7)	11.3 (0.2)	30.0 (0.0)
Jazz, uniform	enemies	8	10	29.8 (0.2)	30.0 (0.0)	7.0 (0.0)	7.5 (0.4)	14.9 (2.0)	7.0 (0.0)	30.0 (0.0)
		10	13	30.0 (0.0)	30.0 (0.0)	9.0 (0.0)	9.2 (0.2)	15.6 (2.5)	9.0 (0.0)	30.0 (0.0)
		12	7	30.0 (0.0)	30.0 (0.0)	11.0 (0.0)	11.0 (0.0)	19.6 (3.4)	11.0 (0.0)	30.0 (0.0)
Jazz, uniform	balanced	8	10	7.4 (0.4)	9.7 (1.7)	7.2 (0.2)	7.2 (0.2)	7.4 (0.4)	7.0 (0.0)	30.0 (0.0)
		10	13	10.0 (0.5)	17.6 (1.6)	9.0 (0.0)	9.0 (0.0)	10.0 (0.6)	9.0 (0.0)	30.0 (0.0)
		12	6	11.8 (0.7)	20.2 (3.2)	11.0 (0.0)	11.0 (0.0)	13.7 (1.8)	11.0 (0.0)	30.0 (0.0)
Jazz, snowball	friends	8	100	7.0 (0.0)	7.0 (0.0)	7.0 (0.0)	7.0 (0.0)	7.0 (0.0)	7.0 (0.0)	30.0 (0.0)
		10	100	9.0 (0.0)	9.0 (0.0)	9.0 (0.0)	9.0 (0.0)	9.0 (0.0)	9.0 (0.0)	30.0 (0.0)
		12	100	11.2 (0.1)	11.3 (0.2)	11.0 (0.0)	11.0 (0.0)	11.0 (0.0)	11.0 (0.0)	30.0 (0.0)
Jazz, snowball	enemies	8	100	19.4 (1.0)	21.6 (1.0)	7.0 (0.0)	7.0 (0.0)	8.8 (0.4)	7.0 (0.0)	30.0 (0.0)
		10	100	24.7 (0.9)	25.3 (0.8)	9.1 (0.1)	9.2 (0.2)	11.6 (0.5)	9.1 (0.1)	30.0 (0.0)
		12	100	29.4 (0.3)	29.6 (0.2)	11.0 (0.0)	11.0 (0.0)	15.7 (0.6)	11.0 (0.0)	30.0 (0.0)
Jazz, snowball	balanced	8	99	7.0 (0.0)	7.0 (0.0)	7.0 (0.0)	7.0 (0.0)	7.0 (0.0)	7.0 (0.0)	30.0 (0.0)
		10	99	9.0 (0.0)	9.1 (0.1)	9.0 (0.0)	9.0 (0.0)	9.0 (0.0)	9.0 (0.0)	30.0 (0.0)
		12	100	11.0 (0.0)	11.0 (0.0)	11.0 (0.0)	11.0 (0.0)	11.0 (0.0)	11.0 (0.0)	30.0 (0.0)

N Robustness Study

Our analysis considers symmetric utilities and subgaussian observation noise. We examine the empirical performance of the algorithms when either assumption is relaxed, using the default configuration ($n = 6$, $k = 2$, $T = 30$, $\sigma = 3$, $\beta = 1$, $\varepsilon = 1$) and 100 seeds per utility model. The reference setting uses symmetric utilities and Gaussian noise.

Asymmetric utilities. We independently sample $v_{i,j}$ and $v_{j,i}$ from the same marginal distributions as in the synthetic models. Thus, agent i's observation of j carries $v_{i,j}$, whereas agent j's observation carries $v_{j,i}$. Accordingly, the learner maintains a separate estimate for each ordered pair. The analysis extends to this estimator, as discussed in Appendix P.

Asymmetric games frequently lack a stable partition: 89 of the 100 games in each utility model are NO instances. The remaining 11 games per model are YES instances, giving 22 YES instances across both models. Consequently, regret and existence-error estimates are less precise than in the symmetric setting, but the large number of NO instances provides a more demanding empirical test of false existence declarations.

Heavy-tailed noise. We replace the Gaussian observation noise with

$$\xi = \sigma\sqrt{\tfrac{3}{5}}\, Z, \qquad Z \sim t_5,$$

where t_5 denotes the Student-t distribution with 5 degrees of freedom. The scaling ensures that ξ has mean zero and variance σ^2. Unlike Gaussian noise, its tails decay polynomially, and its absolute moments of order 5 or higher are infinite. In particular, the noise is not subgaussian, so our theoretical guarantees do not apply. We retain the utilities of the symmetric reference setting, changing only the noise distribution.

Results. Figures 25 and 26 compare the three settings. The first reports regret and existence error on YES instances, while the second examines false declarations on NO instances. Together, they show that UCB-LCB-GS maintains a favourable combination of regret and reliable existence decisions even when the assumptions underlying the analysis are changed, although the different settings call for some qualifications.

On YES instances (Figure 25), UCB-LCB-GS achieves the lowest regret in every tested setting. The advantage is particularly pronounced with asymmetric utilities: its regret is 550 under the Uniform model and 616 under the Gaussian model, compared with at least 1,157 for every baseline. These averages are based on only 11 YES instances per model and should therefore be interpreted with the corresponding uncertainty in mind. Its existence error under asymmetric utilities is approximately 15 and 17 rounds, respectively. This lies between the error of PESSIMISTIC-ONLY and that of the optimism-based baselines, illustrating that the regret advantage does not come from making the fewest existence errors on YES instances alone.

The comparison with the NO instances is particularly informative. UCB-LCB-GS avoids false declarations on all of them, as do PESSIMISTIC-ONLY and RANDOM. By contrast, the optimism-based baselines frequently declare existence despite the absence of a stable partition. Thus, the low regret of UCB-LCB-GS is accompanied by reliable decisions on both YES and NO instances, rather than being achieved simply by declaring existence more readily.

Under Student-t_5 noise, the empirical performance of UCB-LCB-GS and most baselines remains close to that under Gaussian noise. The largest change, approximately 20%, occurs in the regret of some optimism-based baselines. For UCB-LCB-GS, regret changes only from 177 to 178 on the Uniform model and from 191 to 188 on the Gaussian model. Its existence error changes by less than 0.2 rounds in either model. These results indicate that, in the tested configuration, the algorithm is empirically robust to this particular heavy-tailed noise distribution. They do not, however, establish theoretical guarantees under heavy-tailed noise.

Figure 26 provides the complementary comparison on NO instances. It covers the asymmetric settings, which contain 89 NO instances per model, and the symmetric Gaussian setting under both noise distributions, which contains 3 NO instances. The symmetric Uniform settings contain no NO instances and are therefore omitted. UCB-LCB-GS makes no false declaration in any of these

settings. Neither do PESSIMISTIC-ONLY and RANDOM. Under asymmetric utilities, OPTIMISTIC-ONLY, UCB-ONLY and MIXED declare existence on between 1.1 and 3.3 rounds per NO instance, depending on the method and utility model, while LCB-ONLY declares existence on every round. Even on the three symmetric Gaussian NO instances, the optimism-based baselines declare existence on between 12 and 20 of the 30 rounds.

Taken together, the results highlight the benefit of combining pessimistic stability tests with the exploration strategy of UCB-LCB-GS. The algorithm retains the lowest regret on YES instances across all tested settings, while avoiding false declarations on NO instances. The optimism-based baselines can achieve lower existence error on YES instances, but their behaviour on NO instances reveals the cost of relying on tests that do not establish stability. These findings are empirical: in particular, the results under Student-t_5 noise do not extend the theoretical guarantees beyond the subgaussian setting.

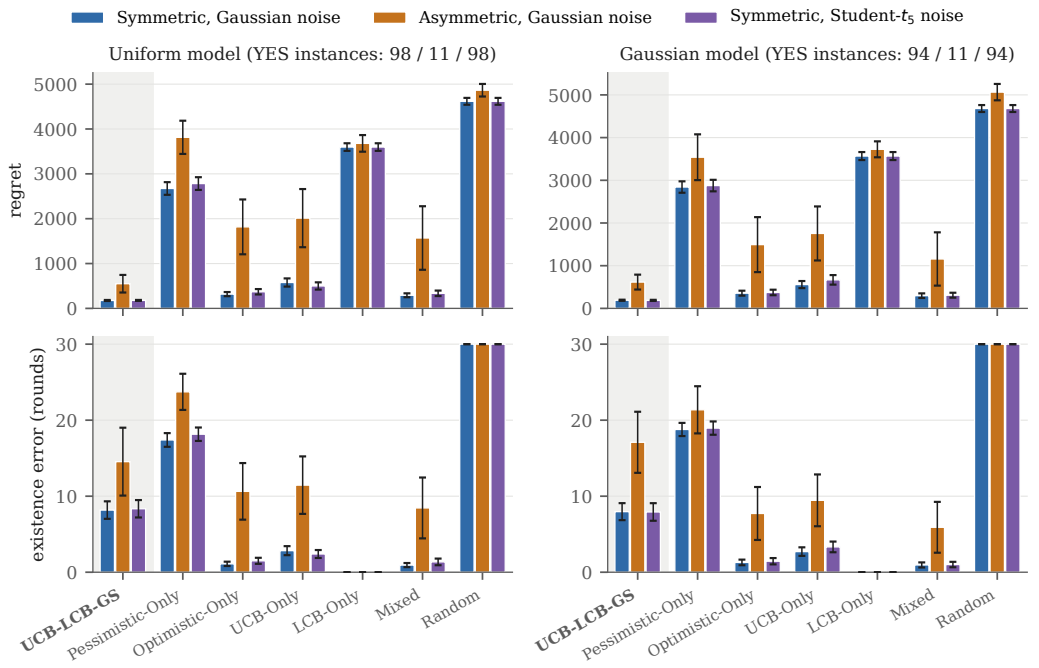

Figure 25: Robustness study, YES instances ($n = 6$, $k = 2$, $T = 30$, $\sigma = 3$, $\beta = 1$, $\varepsilon = 1$, 100 seeds per utility model). Top: regret. Bottom: existence error, in rounds out of 30. Left: Uniform model. Right: Gaussian model. For each method, the three bars correspond to symmetric utilities with Gaussian noise (reference, blue), asymmetric utilities with Gaussian noise (orange), and symmetric utilities with variance-matched Student-t_5 noise (purple). The shaded column marks UCB-LCB-GS. The number of YES instances in each setting is shown in the panel titles. Whiskers indicate ± 1 standard error over the YES instances in each setting.

O Computational Cost

The theoretical analysis assumes that the max-min optimization problems are solved by an oracle. In our experiments, we use local search to obtain candidate partitions and evaluate their gaps exactly (Appendix I). The main computational cost per round comes from this exact evaluation: for each candidate partition and stability test, the deviation checker enumerates every group of at most k agents and all admissible deviations. The number of such deviations grows combinatorially with n and k, reflecting the stability criterion itself rather than an artefact of the implementation. Table 11 reports the measured learner runtime per round.

The complete synthetic experiments (31,400 new trials) required 5.9 CPU-hours, corresponding to 32 minutes of wall-clock time on 11 parallel processes. The robustness study required 0.02 CPU-hours. The dataset-derived experiments (Appendix M), comprising 50,400 trials, took 57 minutes on 11

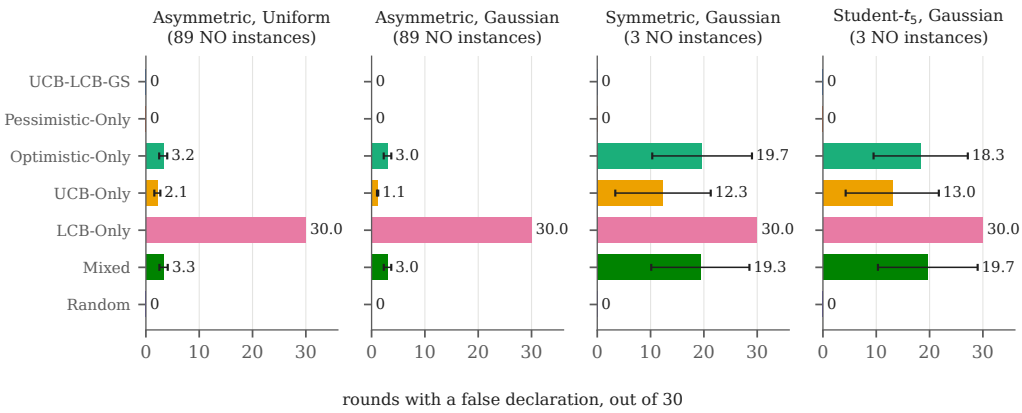

Figure 26: Robustness study on NO instances: average number of rounds, out of 30, in which each method declares existence despite the absence of a k-strongly Nash stable partition. Only settings containing NO instances are shown; the symmetric Uniform settings contain none. Values are printed at the ends of the bars, with 0 indicating that a method never declares existence. Whiskers indicate ± 1 standard error over the NO instances in each setting.

Table 11: Measured learner time per round of UCB-LCB-GS with the exact deviation checker (milliseconds; median and 90th percentile over trials, $T \leq 60$), with 11 trials running in parallel on the machine described in Appendix I.

instances	n	k	median	90th percentile
synthetic	4	2	1.2	2.1
synthetic	5	2	2.4	4.4
synthetic	6	1	1.3	2.3
synthetic	6	2	4.8	8.1
synthetic	6	3	20.9	37.2
synthetic	7	2	7.9	12.8
datasets	8	2	12.2	24.4
datasets	10	2	27.4	56.7
datasets	12	2	60.4	116.3

processes (10.5 CPU-hours); exact labelling of their 3,600 games took an additional 37 minutes. No learning trial used more than 31 MB of memory, and exact labelling at $n = 12$ required less than 1 GB per process. Finally, the coalition-level feedback experiments (54,400 trials) took 38 minutes on 11 parallel processes.

Scaling of exact evaluation. Table 12 reports the time required for a single exact gap evaluation, measured outside the learning loop. For a deviating group of size s, the checker considers every choice of s agents and every admissible assignment of their members to the existing coalitions or to newly formed ones. For a partition with K coalitions, the number of assignments grows roughly as

$$\binom{n}{s}(K+s)^s.$$

Thus, partitions with many coalitions are particularly expensive to evaluate.

The measured cost also increases sharply with the maximum deviation size. For $k = 2$, a single evaluation takes at most 0.8 s up to $n = 32$ across all three partition shapes, and a complete call to the partition heuristic takes approximately one second. For $k = 3$, the cost grows much faster: evaluating the singleton partition takes 3.6 s at $n = 16$, 49 s at $n = 24$ and 312 s at $n = 32$.

These measurements indicate that exact evaluation of the stability condition remains practical for instances with a few dozen agents when $k = 2$. For $k = 3$, it is limited to approximately $n \leq 16$ when every round must be evaluated quickly. The reported runtimes are specific to our implementation and hardware; they do not establish a lower bound on the computational complexity of exact evaluation.

Table 12: Time of one exact evaluation of $\mathrm{Gap}_k(\pi, \mathbf{v})$ for three partition shapes, and of one call of the partition heuristic (three restarts), measured in a single process on an otherwise idle machine. Utilities are random integers in $[-100, 100]$. "–": not run because a single evaluation took more than two seconds.

k	n	singletons	triples	halves	heuristic call
2	6	1.8 ms	0.7 ms	0.7 ms	5.2 ms
2	8	4.0 ms	1.4 ms	0.8 ms	8.1 ms
2	10	6.7 ms	2.2 ms	1.1 ms	15.9 ms
2	12	13.9 ms	3.4 ms	1.6 ms	29.3 ms
2	16	46.0 ms	10.9 ms	3.0 ms	85.3 ms
2	20	114.2 ms	23.2 ms	4.8 ms	200.9 ms
2	24	242.1 ms	41.9 ms	7.2 ms	418.2 ms
2	32	803.4 ms	126.8 ms	14.0 ms	1.2 s
3	6	4.3 ms	2.2 ms	2.3 ms	13.5 ms
3	8	32.7 ms	11.5 ms	6.3 ms	83.9 ms
3	10	151.7 ms	38.7 ms	13.6 ms	229.9 ms
3	12	523.1 ms	79.2 ms	24.5 ms	661.9 ms
3	16	3.6 s	469.5 ms	62.5 ms	–
3	20	15.3 s	1.5 s	128.0 ms	–
3	24	49.1 s	3.7 s	232.8 ms	–
3	32	312.2 s	20.7 s	588.0 ms	–

Local search versus exact evaluation. To identify the source of the computational cost, we also measured local search separately, using three restarts on the same instances. For $k = 2$, local search takes 1.2 ms at $n = 6$, 8.8 ms at $n = 12$, 35 ms at $n = 20$ and 233 ms at $n = 32$. At these sizes, it accounts for between 17% and 30% of a complete heuristic call, reaching 40% at $n = 8$; the remaining time is spent on exact evaluation of the candidate partitions. For $k = 3$, local search accounts for only about 1% of the total runtime at $n = 12$.

Thus, the principal computational bottleneck is not the search for promising partitions, but the verification of their stability. This verification requires considering every group of at most k agents and every admissible assignment of its members to existing or newly formed coalitions.

Earlier measurements with a restricted deviation family. An earlier version of our experimental pipeline enumerated only a restricted family of deviations. This family is guaranteed to be exact for $k \leq 2$ only on partitions with at most seven coalitions (Appendix I). Using this restricted checker, we ran experiments with $k = 2$ for values of n up to 256. At $n = 256$, a single learner round took approximately 1.9 hours using one seed. From around $n = 128$ onwards, local search, rather than gap evaluation, accounted for most of the runtime.

These measurements are not directly comparable with those obtained using the complete deviation checker: the restricted family may miss profitable deviations and incorrectly classify unstable partitions as stable. We include the earlier measurements only to illustrate the computational cost of local search at much larger values of n, not as evidence of the runtime or accuracy of the exact pipeline.

P Modeling Assumptions and Scope

Our model isolates the statistical problem of learning group-stable partitions from partial and noisy feedback. We clarify below the assumptions underlying this formulation, the extensions supported by our analysis, and the aspects of coalition formation that remain outside its scope.

Assignment during learning and strategic deviations. In hedonic games, stability notions are solution concepts for evaluating coalition structures; they do not specify a mechanism for implementing assignments. Our online model follows this distinction. At each round, the learner selects and implements a partition, observes noisy feedback, and updates its estimates. Coordinated deviations do not take place as part of the learning protocol. Instead, k-strong Nash stability is the criterion used

to assess whether a partition formed by the learner admits a profitable coordinated deviation by a group of at most k agents.

The model therefore assumes that the learner's assignments are implemented, including exploratory assignments that may not be stable. Agents do not voluntarily accept or reject assignments within the model, and strategic behaviour enters only through the stability criterion. Allowing endogenous participation, whereby agents may refuse assignments or leave coalitions during learning, would require a different model of online coalition formation.

Pairwise additive utilities. Additively separable hedonic games represent the exponentially many possible coalition utilities through only $\binom{n}{2}$ pairwise utility parameters. This representation underlies our confidence intervals, the comparison between the estimated and true deviation gaps (Lemma D.2), and the counting argument for witness probing (Theorem D.1).

With higher-order interactions, such as complementarities among three or more agents, coalition utilities can no longer be recovered from pairwise parameters alone. Extending our approach to such settings would require additional structure, for example, sparse higher-order interactions or a parametric utility model, together with new concentration and regret analyses. Our current guarantees do not cover these models.

Symmetric and asymmetric preferences. Symmetry primarily reduces the number of parameters that the learner must estimate. Under asymmetric preferences, the utility that agent i assigns to agent j may differ from the reverse valuation. The learner therefore maintains a separate estimate for each ordered pair (i, j), and obtains an observation of that utility whenever the two agents share a coalition.

The proofs in Appendix D do not rely on symmetry and extend to this estimator. The number of estimated utilities becomes $n(n-1)$ rather than $\binom{n}{2}$. Accordingly, the constant in Lemma D.1 doubles to $2n(n-1)/(\beta-2)$, and $\binom{n}{2}$ is replaced by $n(n-1)$ in the existence-error and regret bounds.

Symmetry does, however, affect the existence of stable partitions. Symmetric ASHGs always admit a Nash stable partition [19], whereas asymmetric games need not. Moreover, stable partitions need not exist for $k \geq 3$, even under symmetric preferences (Observation 1). Thus, although the learning procedure and its analysis extend to asymmetric utilities, the set of instances in which a stable partition exists can change. Appendix F.6 discusses asymmetric utilities under coalition-level feedback.

Stationarity and noise. We assume throughout that pairwise utilities remain fixed over time and that observation noise is independent and σ^2-subgaussian. If preferences evolve, the learner would need to discount older observations and the regret analysis would require a different treatment of non-stationarity. Similarly, heavy-tailed noise falls outside the subgaussian assumption and is not covered by our theoretical guarantees. We examine such noise only empirically in the robustness study above.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and the "Our Contributions" paragraph in Section 1 state the model, the algorithm UCB-LCB-GS with witness probing, its guarantees under pairwise and coalition-level feedback, and the scope of the experiments. These claims correspond to Definitions 1–3, Algorithm 1, Theorems 3.1–3.3 and 4.1 (proved in Appendices D and F), and the experiments in Section 5 and Appendices L and K–O. The guarantees are stated together with their assumptions, including exact optimization of the optimistic problem; the effect of the heuristic used in practice is analyzed in Appendix G.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 6, Appendix P (modeling assumptions: assignment authority, additive and symmetric utilities, stationarity, subgaussian noise), Appendices G and H (no approximation guarantee for the heuristic solver, with a counterexample), Appendix F (limitations under coalition-level feedback), Appendix I.1 (empirical limitations), Appendix M (the utilities of the dataset-derived games are constructed from distances in the data rather than elicited, and these games contain no instance without a stable partition, so the non-existence error is not measured on them), Appendix O (computational cost of exact deviation checking), and Appendix E (necessity of a positive certification margin).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: All theoretical results are numbered and cross-referenced. Observation 1 is proved in Appendix C. Theorems 3.1, 3.2 and 3.3 are proved in full in Appendix D, which also contains the concentration inequality for adaptively collected observations (Lemma D.1) and the witness-gap inequality (Lemma D.3). Appendix E proves that purely optimistic exploration can fail and that a positive certification margin is necessary, Appendix G analyzes approximate optimization, and Appendix F proves the results for coalition-level feedback (Theorem 4.1, Propositions F.1, F.2 and F.3). The assumptions are stated in the theorems and in Section 2: symmetric ASHGs, independent σ^2-subgaussian noise, $\beta > 2$, and, for the existence-error and regret bounds, a positive certification margin (10) and exact solution of the optimistic problem (9).

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Section 5 and Appendix I specify the utility models, the parameter grid, the seeds (100 per configuration), the solver and its hyperparameters, the exact deviation checker, the exact labelling of instances, and the metrics. Appendix J gives the pseudo-code of every baseline. Appendices L, M and N specify the coalition-level feedback experiments, the construction and sampling of the dataset-derived games, and the robustness settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may

be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The code is publicly available at `https://github.com/coml1149-saarco/online-group-stable-coalitions`, with instructions to reproduce every experiment and figure (Appendix I). The synthetic data are generated by the models of Section 5. The datasets used in Appendix M are publicly available from the sources cited there; the repository contains the code that retrieves them and constructs the games, rather than copies of the data.

   Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: Section 5 and Appendix I list the grid, the seeds, and every hyperparameter (none tuned per algorithm); Appendices L, M and N give the settings of the additional studies.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: Every configuration uses 100 seeds with the same instances for all algorithms. Tables and figures report means with ± 1 standard errors, computed over the instances of the class on which each metric is defined (Appendix I, "Statistical significance and error bars", which gives the formula), and the numbers of instances per class are reported.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

Answer: [Yes]

Justification: Appendix I ("Hardware" and "Compute resources") and Appendix O report the machine (Apple M4 Max, 16 CPU cores, 128 GB memory), the memory use, the learner's time per round, the time of the local search and of a single exact evaluation, and the total CPU time and wall-clock time of each experiment.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The synthetic experiments involve no human data. Appendix M reuses three public benchmark datasets: measurements of flowers, and two networks derived from relations between people, namely ties between members of a karate club and musicians shared by jazz bands. In these networks nodes are identified only by numbers, which does not make the data anonymous; we use no other information about individuals, make no attempt to identify them, and do not present the derived utilities as their preferences.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Appendix A discusses potential positive and negative societal impacts. The paper is foundational research on online learning of group-stable coalition structures, evaluated on synthetic instances and on games derived from public clustering and community-detection datasets. It releases no model or data set and does not target a deployed application. We outline beneficial applications, such as stable team formation, and risks, such as unfair assignments when a coordinator imposes coalitions on agents, or the misreading of utilities derived from similarity in data as preferences of the underlying entities.

    Guidelines:

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

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper releases no data set, pre-trained model, scraped corpus or generative model. The artifact to be released is the experimental code with the per-trial result files (Appendix I), which carry no foreseeable risk of misuse. The third-party datasets are not redistributed.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The synthetic utility models follow Boehmer et al. [18], and the counterexample in Appendix C follows Aziz et al. [10]; both are cited. The game construction of Appendix M follows Csáji et al. [36], and the datasets are cited there with their sources. Iris is distributed by the UCI Machine Learning Repository under the CC BY 4.0 licence [47]. For the karate club network [78] and the jazz network [52], which we obtain from the Koblenz Network Collection [61], we found no licence that covers the data; we cite the original publications, as the collection requests. The split of the karate club into two groups is read from a source file of NetworkX (BSD-3-Clause License). We do not redistribute any of these datasets; the code retrieves them from their sources. We use Python (PSF License), NumPy (BSD-3-Clause License) and Matplotlib (Matplotlib License) within the terms of their licenses (Appendix I).

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The new asset is the experimental code base, including the exact deviation checker, the instance-labelling code and the construction of the dataset-derived games. It is publicly available with documentation under the MIT license at `https://github.com/coml1149-saarco/online-group-stable-coalitions` (Appendix I).

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper involves no crowdsourcing and no research with human subjects: we recruit no participants and collect no new data. Appendix M reuses existing public benchmark datasets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper involves no research with human subjects: we recruit no participants and collect no new data. Appendix M reuses existing public benchmark datasets, two of which are derived from relations between people.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: No large language model is used as a component of the core methodology. The contribution consists of the algorithm UCB-LCB-GS (Section 3, Algorithm 1), its theoretical analysis, and experiments on synthetic and real-data-derived games (Section 5 and the appendices); none of these involve LLMs.

    Guidelines:

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