Learning a Stackelberg Leader's Incentives from Optimal Commitments

Yurong Chen[1], Xiaotie Deng[2], Jiarui Gan[3], Yuhao Li[4]
[1]Inria, École Normale Supérieure, PSL Research University
[2]Peking University
[3]University of Oxford
[4]Columbia University
yurong.chen@inria.fr, xiaotie@pku.edu.cn,
jiarui.gan@cs.ox.ac.uk, yuhaoli@cs.columbia.edu

Abstract

Stackelberg equilibria, as functions of the players' payoffs, can inversely reveal information about the players' incentives. In this paper, we study to what extent one can learn about the leader's incentives by actively querying the leader's optimal commitments against strategically designed followers. We show that, by using polynomially many queries and operations, one can learn a payoff function that is *strategically equivalent* to the leader's, in the sense that: 1) it preserves the leader's preference over almost all strategy profiles; and 2) it preserves the set of all possible (strong) Stackelberg equilibria the leader may engage in, considering all possible follower types. As an application, we show that the information acquired by our algorithm is sufficient for a follower to induce the best possible Stackelberg equilibrium by imitating a different follower type. To the best of our knowledge, we are the first to demonstrate that this is possible without knowing the leader's payoffs beforehand.

Contents

1 Introduction

Strategy commitment naturally arises in various scenarios due to asymmetric timing, commitment advantages, or hierarchical structures. In a Stackelberg equilibrium, the *leader*, commits to a strategy that maximizes her payoff conditioned on the follower's best-responding behavior [von Stackelberg, 1934, Von Stengel and Zamir, 2010]. Over the past decade, Stackelberg games and the Stackelberg equilibrium have been adopted in applications across various domains to obtain optimal strategies to commit to, including security [Tambe, 2011], exam design [Li and Conitzer, 2017], transportation management [Staňková and Boudewijn, 2015, Böhnlein et al., 2023], multi-agent learning [Yang et al., 2023], as well as persuasion and contract design [Dughmi, 2017, Dütting et al., 2021].

Despite its rationality, optimal commitment exposes the leader's incentives, leaving a passage for interested parties to probe into the leader's incentives—and potentially exploit this information—from samples of the leader's optimal commitments. Especially noteworthy are scenarios where a learner can obtain various samples of the leader's optimal commitments against different types of followers (i.e., followers with different payoffs). This arises, for example, in web platforms such as recommender systems or e-commerce websites, where the platform (the leader) provides customized recommendations or services based on the user's (the follower's) profile. The platform may interact simultaneously with millions of users. A learner (e.g., a competing platform or a data aggregation company) can create multiple accounts with strategically crafted user profiles to acquire the platform's optimal commitments against different user types. Similarly, regulatory organizations may also want to understand and assess commitment optimizers' motives to ensure that their incentives align with fairness, transparency, or other regulatory objectives [Meßmer and Degeling, 2023, Koshiyama et al., 2024]. This leads to the following *inverse game theory* and *active learning* problem we study in this paper:

> *What can we learn about the leader's incentive based on actively obtained samples of optimal commitments, especially in sample-efficient and computationally tractable ways?*

We consider this problem in bi-matrix games and adopt a query model. In the model, the learner can query an *equilibrium oracle* to obtain the leader's optimal commitment against any given *follower type* (i.e., a payoff function or matrix of the follower). The oracle abstracts the process where the learner creates a new follower type to interact with the leader. The follower reports their type to the leader, and the leader commits optimally w.r.t. the reported payoffs. The type-reporting step can either be direct, where the follower explicitly submits their payoffs, e.g., advertisers specifying targeting preferences in online ad platforms or job seekers uploading resumes to hiring agencies. Alternatively, it can be realized indirectly via a learning process [Letchford et al., 2009, Blum et al., 2014a, Haghtalab et al., 2016, Peng et al., 2019]: e.g., online platforms learn customized recommendations by interacting with users.[1]

1.1 Main Result

Indeed, to recover the exact payoffs of the leader through such a query model is hope in vain because of cases that are indistinguishable due to the limited information exposure of the oracle. Hence, we calibrate our objective according to several types of strategic equivalences. In general, we would like to recover the leader's preference order over as many strategy profiles as possible and to learn sufficiently about the leader's commitment behavior in the Stackelberg scenario.

Our main result (Theorem 5.4) is an efficient approach which, by interacting with the equilibrium oracle, learns a function $\tilde{u}^L$ equivalent to the leader's original payoff u^L in the following senses.

[1]Such learning processes can converge rapidly due to intense user interactions or the use of heuristics [Covington et al., 2016].

- First, $\tilde{u}^L$ preserves the leader's preference over (mixed) strategy profiles in which the follower's actions involved do not *obviously dominate* each other w.r.t. the leader's payoff (*in-component equivalence*). The notion of obvious dominance is a stronger form of dominance than standard dominance notions, making it unlikely to arise frequently in practice. In such cases, $\tilde{u}^L$ preserves the leader's preference over all strategy profiles.

- Second, $\tilde{u}^L$ preserves the leader's preference over strategy profiles where the follower's responses are *pure* and outside a "hard set" (*cross-component*), regardless of obvious dominance. Notably, restricting the follower's responses to pure strategies is common in studies of the (strong) Stackelberg equilibrium, given that a pure best response always exists. The leader's commitments reveal little information about actions in the hard set as the leader's preference over these actions does not influence the equilibria of the game. Therefore, what we recover matches the best possible in the model considered.

- Indeed, the actions in the hard set are insignificant for determining possible equilibria of the game. This makes $\tilde{u}^L$ *inducibility-equivalent* to u^L: it preserves the set of *inducible Stackelberg equilibria*. An equilibrium is inducible if it arises for *some* follower type. Hence, the set can be viewed as a "footprint" of the leader's commitment behavior, consisting of all possible equilibria the leader may engage in. Using $\tilde{u}^L$, we can determine the inducibility of every strategy profile under the leader's actual payoffs u^L and, for the inducible ones, efficiently compute a follower type to induce them.

The inducibility-equivalence between $\tilde{u}^L$ and u^L, is closely related to the following *equilibrium inducing problem*: Suppose a follower can hide their true type, say u^F, and make the leader believe that they are of some other type $\tilde{u}^F$. How to find a $\tilde{u}^F$ so that the Stackelberg equilibrium s of game $(u^L, \tilde{u}^F)$ (which the leader believes they are playing) has the maximum possible $u^F(s)$?

The problem is also phrased as payoff manipulation or imitative deception in Stackelberg games [Gan et al., 2019b,a, Nguyen and Xu, 2019, Birmpas et al., 2021, Chen et al., 2025], where a follower imitates the best-responding behavior of another type to maneuver the game into a different equilibrium. Despite the growing interest in this problem, it remains unknown whether such manipulations are possible when the manipulator does not have any information about the leader's payoffs beforehand but has to learn through interactions. Holding the leader's incentive confidential is, therefore, the last potential barrier against such manipulations. As an application of our result, optimal manipulation is polynomially learnable. Therefore, the information barrier may be unreliable due to the inevitable information exposure via optimal commitments. Our results imply that due caution is necessary when one intends to utilize the power of optimal commitment (e.g., providing customized service based on user profiles).

1.2 Technique Overview

Since the leader's payoff function is linear in their mixed strategy for each follower action, it can be represented as a finite set of linear functions $\{u^L(\cdot, i)\}_{i=1}^n$, for n follower actions. To learn a "surrogate" function that preserves the value order of u^L, the task is twofold: 1) learning the gradient direction of each function, and 2) learning their relative growth rates and offsets. A follower type partitions the convex function domain (i.e., the leader's strategy space) into a collection of polytopal subsets $\{S_i\}_{i=1}^n$ (best response regions), each associated with a follower action. This results in a piecewise linear function f, such that $f(\mathbf{x}) = u^L(\mathbf{x}, j)$ for $\mathbf{x} \in S_j$, which effectively maps each commitment $\mathbf{x}$ to the utility it yields for the leader along with the follower's best response to $\mathbf{x}$.

While f is not directly observable, Stackelberg equilibria correspond to its maxima. Hence, the core difficulty lies in constructing best response (BR) partitions, so that strategy profiles carrying useful information about u^L are exposed as Stackelberg equilibria. (We do *not* have direct access to utilities of strategy profiles.) Along this line, our approach features the following key novelties.

- First, to propagate the pairwise information between actions, we introduce *non-dominance graph* (Definition 5). This graph allows us to derive relative information between any two actions connected by a path, propagating the pairwise information across an entire connected component. Consequently, we achieve a strong in-component equivalence, which helps us to understand the maximum extent of information that can be learned in our model.

- Second, to infer the gradient of $u^L(\cdot, i)$, we need to identify multiple strategies yielding the same u^L value. The normal vector of the hyperplane defined by these points gives the gradient. To achieve this, we design a parametrized type, where each parameter is responsible for identifying one strategy (see Fig. 3). Our design ensures several structural properties, so that we can easily tune the parameters to expose multiple strategies simultaneously through equilibria (Section 6.1). Similar techniques are introduced when we perform nested binary search to learn relative parameters between two actions. There, simultaneous control of the actions' best response regions is required and achieved via a parameterized type (Appendix D.3).

- Third, to expose certain follower actions of interest through Stackelberg equilibria, we need to suppress the leader's maximum attainable utilities in other best response regions. To introduce a general approach to handling this, we introduce the concept *cover* (Definition 7), as a follower type that reduces the leader's utility on their maximin set to below the maximin value. This allows more strategy profiles associated with the actions of interest to become equilibria. We establish a necessary and sufficient condition for the existence of a cover (Lemma 6.3).

1.3 Related Work

Inverse Game Theory and Active Learning The problem we study is both an inverse game theory and active learning problem. Inverse game theory aims to recover the underlying game parameters from observed equilibrium behaviors. These tasks are often accomplished via active learning approaches, where equilibrium samples are obtained through interactions. A line of work in Stackelberg games explored how a leader can learn the follower's incentives by observing their responses and proposed active learning algorithms [Letchford et al., 2009, Balcan et al., 2015, Haghtalab et al., 2016, Blum et al., 2014b, Roth et al., 2016, Peng et al., 2019, Wu et al., 2022]. Specifically, Peng et al. [2019] analyzed the sample complexity of this approach. Haghtalab et al. [2016] and Wu et al. [2022] considered followers with bounded rationality under the quantal response model. Beyond this, Xu et al. [2023] studied how to learn players' coalition structure by designing games. The equilibrium oracle considered is similar to ours, requiring a candidate equilibrium to be provided as part of the input.

Imitative Deception in Stackelberg Games Motivated in part by the above work on learning to commit, the imitative deception problem explores how learning-based commitment strategies can be manipulated. Gan et al. [2019b] introduced this term and examined countermeasures, proving them NP-hard to approximate. In Stackelberg security games, Gan et al. [2019a] found that followers often benefit by steering the game into a zero-sum equilibrium, reducing the leader to their maximin payoff. While this does not hold in general bimatrix Stackelberg games, Birmpas et al. [2021] established a broader connection between payoff manipulations and the leader's maximin payoff. Nguyen and Xu [2019] studied similar manipulations with bounded payoff reporting. More recently, Chen et al. [2025] extended this to extensive-form games, while Cen et al. [2024] examined platform-user Stackelberg interactions.

Repeated Games and Stackelberg Equilibria In our model, the learner repeatedly queries the leader's optimal commitments. This may appear similar to a repeated Stackelberg game, a topic of growing interest. Prior work has studied such games where both players maximize long-term rewards [Haghtalab et al., 2022, Arunachaleswaran et al., 2024], as well as interactions with patient followers in auctions [Amin et al., 2013, 2014, Mohri and Medina, 2014, Liu et al., 2018] and principal-agent problems [Zhang and Conitzer, 2021a, Gan et al., 2022, 2023]. Beyond Stackelberg settings, repeated simultaneous-move games with information asymmetry have also been explored [Sorin, 1983, Aumann et al., 1995]. Connections between repeated games and Stackelberg equilibrium have been explored: Deng et al. [2019] and Mansour et al. [2022] compared optimizers' guaranteed values against no-regret learners to Stackelberg payoffs; Zrnic et al. [2021] showed that slower learners can attain Stackelberg values in strategic classification; and Ananthakrishnan et al. [2024] argued that players with information advantages can achieve Stackelberg values in meta-games with machine strategies. Unlike these works, our model assumes repeated Stackelberg play but differs fundamentally: the leader remains unaware that the followers are controlled by the same learner, treating each round as a new game against an independent follower.

Manipulation in Other Strategic Settings Kolumbus and Nisan [2022b,a] studied the Nash equilibrium of the meta-game where multiple players attempt to manipulate simultaneously. Earlier, Kash and Parkes [2010] examined imitative deception (termed "impersonation" in their work) in auction design, focusing on designing mechanisms resistant to such deception rather than learning optimal player types to imitate. More broadly, our work relates to private information manipulation and data robustness in learning-driven environments, which have raised growing interest in game theory and machine learning. The strategic misreporting of private preferences has been widely studied across various settings [e.g., Amanatidis et al., 2021, Tang and Zeng, 2018, Deng et al., 2020, Chen et al., 2022, Cheng et al., 2022, Wu et al., 2023]. Countermeasures have been explored through mechanism design, robust learning, and differential privacy [e.g., Hardt et al., 2016, Gan et al., 2019b, Zhang and Conitzer, 2021b, Wang et al., 2021, Bei et al., 2022, Haghtalab et al., 2022].

Conceptual Leader in Meta-Games Hierarchically, the learner in our model can be viewed as a "conceptual leader" in a meta-game: the learner first commits to a follower type; then the leader best responds by playing a Stackelberg equilibrium based on their true payoffs and the reported follower type. This perspective aligns with research on imitative deception [Gan et al., 2019b,a, Nguyen and Xu, 2019, Birmpas et al., 2021, Chen et al., 2025] and strategizing against learning players [Braverman et al., 2018, Deng et al., 2019, Mansour et al., 2022, Lin and Chen, 2025], where the player who can predict the opponent's behavior acts as the "conceptual leader". Unlike most of these works, which typically assume full knowledge of the opponent's payoff function, we operate in an incomplete information setting, where the learner must infer the leader's incentives. In the complete information case of imitative deception [e.g., Birmpas et al., 2021], to induce an equilibrium, the manipulator constructs a certain follower type to minimize the leader's utility beyond that equilibrium. In contrast, our learner must design various follower types to generate different best response partitions, in order to extract sufficient information about the leader's payoff structure.

2 Preliminaries

Stackelberg games are a standard framework for studying strategy commitment in game theory. In a Stackelberg game, a leader commits to a strategy and a follower best responds to this commitment. We consider general bi-matrix games in this paper. A bi-matrix game $\mathcal{G} = (u^L, u^F)$ is given by two matrices $u^L, u^F \in \mathbb{R}^{m \times n}$, which specify the leader's and the follower's payoffs, respectively. The leader has m actions (i.e., pure strategies) at their disposal, each corresponding to a row of the payoff matrices; the follower has n actions, each to a column. Entries $u^L(i,j)$ and $u^F(i,j)$ are the payoffs of the leader and the follower when a pair $(i,j) \in [m] \times [n]$ of actions is played.

In the mixed strategy setting, the leader can further commit to a mixed strategy, which is a distribution over the actions, i.e., $\mathbf{x} \in \Delta_m := \{\mathbf{x} \in \mathbb{R}^m_{\geq 0} : \sum_{i \in [m]} x_i = 1\}$. Slightly abusing notation, we denote by $u^L(\mathbf{x}, j) = \sum_{i \in [m]} x_i \cdot u^L(i,j)$ the leader's expected payoff for a strategy profile $(\mathbf{x}, j)$, and similarly denote by $u^F(\mathbf{x}, j) = \sum_{i \in [m]} x_i \cdot u^F(i,j)$ the follower's expected payoff. We will refer to a payoff matrix and the corresponding payoff function interchangeably throughout this paper.

2.1 Stackelberg Equilibrium

A (pure) best response of the follower to a mixed strategy $\mathbf{x}$ of the leader is given by $j \in \mathrm{BR}(\mathbf{x})$, where $\mathrm{BR}(\mathbf{x}) := \mathrm{argmax}_{j \in [n]} u^F(\mathbf{x}, j)$ is the follower's *best response set*, or a *BR-correspondence* $\mathrm{BR} : \Delta_m \to 2^{[n]}$. We consider only pure best responses as at least one optimal response is always pure. The inverse of BR will also be useful: for any $j \in [n]$, $\mathrm{BR}^{-1}(j) := \{\mathbf{x} \in \Delta_m : j \in \mathrm{BR}(\mathbf{x})\}$ is the set of leader strategies that incentivize the follower to respond j.

The leader's optimal commitment is captured by the following optimization, under the assumption that the follower breaks ties in favor of the leader when there are multiple actions in $\mathrm{BR}(\mathbf{x})$:

$$(\mathbf{x}, j) \in \underset{\mathbf{x}' \in \Delta_m,\ j' \in \mathrm{BR}(\mathbf{x}')}{\mathrm{argmax}} u^L(\mathbf{x}', j'). \tag{1}$$

$(\mathbf{x}, j)$ is called a *strong Stackelberg equilibrium* (SSE). Equivalently, using the inverse of BR, we have,

$$(\mathbf{x}, j) \in \underset{\mathbf{x}' \in \mathrm{BR}^{-1}(j'),\ j' \in [n]}{\mathrm{argmax}} u^L(\mathbf{x}', j'). \tag{2}$$

The SSE is the most widely used solution concept for Stackelberg games. The optimistic tie-breaking assumption it adopts is justified by the fact that the desired tie-breaking behavior can often be induced by an infinitesimal perturbation in the leader's strategy [von Stengel and Zamir, 2004].

Definition 1 (SSE)**.** A strategy profile $(\mathbf{x}, j) \in \Delta_m \times [n]$ is an SSE of a game $\mathcal{G} = (u^L, u^F)$ if Eq. (1) holds (or equivalently, Eq. (2) holds).

2.2 Inducible SSEs

We denote by $\mathrm{SSE}(u^L, u^F)$, or $\mathrm{SSE}(\mathcal{G})$, the set of SSEs of the game $\mathcal{G} = (u^L, u^F)$. Moreover, we denote by $\mathrm{SSE}(u^L) := \bigcup_{\tilde{u}^F \in \mathbb{R}^{m \times n}} \mathrm{SSE}(u^L, \tilde{u}^F)$ the set of *inducible* SSEs. For ease of description, we will refer to a payoff matrix $\tilde{u}^F \in \mathbb{R}^{m \times n}$ of the follower as a *follower type*.

Definition 2 (Inducibility)**.** A follower type $\tilde{u}^F \in \mathbb{R}^{m \times n}$ *induces* an SSE $(\mathbf{x}, j)$ (w.r.t. u^L) if $(\mathbf{x}, j) \in \mathrm{SSE}(u^L, \tilde{u}^F)$. A strategy profile $(\mathbf{x}, j)$ is said to be *inducible* (w.r.t. u^L) if $(\mathbf{x}, j) \in \mathrm{SSE}(u^L)$.

The following result by Birmpas et al. [2021] provides an elegant characterization of the set of inducible SSEs. The characterization establishes an interesting connection between inducible SSEs and the leader's maximin value $M_{[n]}$, where for every subset $S \subseteq [n]$ of the follower's actions,

$$M_S := \max_{\mathbf{x} \in \Delta_m} \min_{j \in S} u^L(\mathbf{x}, j) \quad \text{and} \quad \mathcal{M}_S := \underset{\mathbf{x} \in \Delta_m}{\mathrm{argmax}} \min_{j \in S} u^L(\mathbf{x}, j), \tag{3}$$

are the leader's maximin payoff and the set of maximin strategies when the follower's responses are restricted within S. Note that by definition $M_{S'} \geq M_S \geq M_{[n]}$ for any $S' \subseteq S \subseteq [n]$.

Theorem 2.1 (Birmpas et al. [2021]). *A strategy profile $(\mathbf{x}, j)$ is inducible if and only if $u^L(\mathbf{x}, j) \geq M_{[n]} := \max_{\mathbf{y} \in \Delta_m} \min_{k \in [n]} u^L(\mathbf{y}, k)$. Moreover, given u^L, a payoff matrix $\tilde{u}^F$ that induces $(\mathbf{x}, j)$ can be computed in polynomial time.*

Intuitively, the follower can behave as follows to induce $(\mathbf{x}, j)$: for any strategy $\mathbf{y} \neq \mathbf{x}$ of the leader, respond with a worst action $k \in \text{argmin}_{k' \in [n]} u^L(\mathbf{y}, k)$ for the leader; and respond with j if the leader commits to $\mathbf{x}$. Namely, the follower is completely adversarial to the leader unless she commits to $\mathbf{x}$. The leader obtains at most the maximin value $M_{[n]}$ for any strategy $\mathbf{y} \neq \mathbf{x}$ as a result. So, as long as $u^L(\mathbf{x}, j) \geq M_{[n]}$ holds, $\mathbf{x}$ becomes the leader's optimal commitment. Despite the simple intuition, to construct a payoff matrix that realizes this behavior is non-trivial and relies crucially on a known u^L as shown in Birmpas et al. [2021]. Theorem 2.1 will be useful for achieving inducibility-equivalence based on in- and cross-component equivalent functions (Section 5.3) and for solving the equilibrium inducing problem (Section 5.4).

3 Main Problem: Incentive Learning via the SSE Oracle

We consider the scenario where the leader's payoff matrix u^L is unknown to us (as the learner), and we want to *learn the leader's incentive*. Our only source of information is an *SSE oracle* $\mathcal{A}_{\text{SSE}}$, which decides whether a given strategy profile can be induced as an SSE by a given follower type.

Definition 3 (SSE oracle)**.** Given as input a follower type $\tilde{u}^F$ and a strategy profile $(\mathbf{x}, j) \in \Delta_m \times [n]$, the *SSE oracle* $\mathcal{A}_{\text{SSE}}$ outputs whether $(\mathbf{x}, j) \in \text{SSE}(u^L, \tilde{u}^F)$ or not.

Remark 1. A query to the SSE oracle creates a new follower of the given type $\tilde{u}^F$ to interact with the leader. As discussed in the previous sections, one can create a new user account to interact with an app or web service. Note that making multiple queries differs from playing a repeated game with the leader (e.g., Haghtalab et al. [2022]). In the latter, the leader knows that she is interacting with the *same* follower repeatedly. In contrast, in our model, each follower appears as a new player to the leader and it is impossible for the leader to know for a fact that they are all controlled by the same learner (e.g., an app with millions of users cannot easily identify a few accounts controlled by one user).

Remark 2. The oracle requires each query to specify a candidate SSE $(\mathbf{x}, j)$ in addition to the follower type $\tilde{u}^F$. This may appear weaker and harder to maneuver than one that only requires the latter. In fact, the SSE in the input is an additional lever for us to sidestep the equilibrium selection issue when multiple SSEs exist. We will extend our results to oracles that do not require this input in section 8, where the leader effectively selects one SSE for us, in the presence of multiple.

Remark 3. An alternative formulation is to assume that the leader adopts a policy that maps each follower type to a leader strategy, with the learning task being to infer this policy. However, without any structural assumptions on the policy, the learning task becomes intractable. Moreover, learning from equilibrium behaviors not only facilitates this process but also provides deeper insights into the nature of the solution concept itself.

3.1 Strategic Equivalences

With query access to $\mathcal{A}_{\text{SSE}}$, we aim to learn a payoff matrix, $\tilde{u}^L$, that is *strategically equivalent* to u^L in the strongest possible sense that can be achieved. We measure the time and query complexity w.r.t. the bit size of u^L, assuming that the entries of u^L are rational numbers. Moreover, we assume that an upper bound of the bit size of u^L is given.

Definition 4 (Strategic equivalence)**.** A payoff matrix $\tilde{u}^L \in \mathbb{R}^{m\times n}$ is *(strategically) equivalent* to u^L in a set $\mathcal{S} \subseteq \Delta_m \times \Delta_n$ if $\tilde{u}^L(s) > \tilde{u}^L(s') \iff u^L(s) > u^L(s')$ for all strategy profiles $s, s' \in \mathcal{S}$.

Ideally, we would like to find a matrix equivalent to u^L in the entire space $\Delta_m \times \Delta_n$ of strategy profiles. However, this is not possible because of cases indistinguishable under information exposed by $\mathcal{A}_{\text{SSE}}$. In general, the matrix $\tilde{u}^L$ our learning algorithm outputs ensures all of the following forms of equivalences (also see Fig. 1).

- **In-Component Equivalence:** $\tilde{u}^L$ is equivalent to u^L in $\Delta_m \times \Delta(C)$ for each *non-dominance component* $C \subseteq [n]$ (Definition 5).
- **Cross-Component Equivalence:** $\tilde{u}^L$ is equivalent to u^L in $\Delta_m \times ([n] \setminus \Lambda)$, where Λ is a "hard set" (defined below in Eqs. (4) and (5)).
- **Inducibility Equivalence:** $\tilde{u}^L$ is *inducibility-equivalent* to u^L (Definition 6).

The non-dominance components are connected components of the *non-dominance graph*. It is an undirected graph that describes, for each pair of the follower's actions, whether they *obviously dominate* each other (w.r.t. the leader's payoff). To define the graph, we exclude a hard set of follower actions, $\Lambda := \Lambda' \cup \Lambda''$, for which the SSE oracle reveals little information:

$$\Lambda' := \left\{ j \in [n] : M_{\{j\}} = M_{[n]} \right\}, \tag{4}$$

$$\text{and} \quad \Lambda'' := \left\{ j \in [n] \setminus \Lambda' : M_{\{j,k\}} = M_{[n]} \text{ for some } k \in [n] \right\}. \tag{5}$$

Namely, these actions achieve the leader's maximin value $M_{[n]}$ either alone or in a pair. Intuitively, since the leader's value in an SSE is always at least $M_{[n]}$, strategy profiles with values lower than this cannot never be an SSE. Therefore, the SSE oracle reveals almost no information about these profiles. The components of the graph and the hard sets Λ' and Λ'' form a partition of $[n]$.

Definition 5 (Non-dominance graph)**.** The *non-dominance graph* of u^L is an undirected graph $G = (E, [n] \setminus \Lambda)$ where E consists of edges $\{j, k\}$ in which j and k do not *obviously dominate* each other, i.e.,

$$\min_{i\in[m]} u^L(i,j) < \max_{i\in[m]} u^L(i,k) \quad \text{and} \quad \min_{i\in[m]} u^L(i,k) < \max_{i\in[m]} u^L(i,j).$$

The set of vertices in a connected component of G is called a *non-dominance component* of u^L.

Therefore, $\tilde{u}^L$ ensures the strongest form of equivalence within each non-dominance component. Across the components, it ensures a weaker form of equivalence defined over the follower's pure strategies, excluding those in Λ. One can verify that $|\Lambda'| > 1$ and $|\Lambda''| > 2$ only in *degenerate* cases, which appear with probability zero if every entry of u^L is perturbed independently by a small random noise. Hence, for non-degenerate instances, $\tilde{u}^L$ ensures an equivalence almost over the entirety of $\Delta_m \times [n]$. Across the components, this is the maximum degree of equivalence that can be achieved due to the obvious dominance between the components.[2] In

[2]E.g., in Fig. 1, if we simultaneously apply different affine transformations to the components, through the oracle $\mathcal{A}_{\text{SSE}}$ one cannot differentiate the original matrix from the outcome resulting from the transformations.

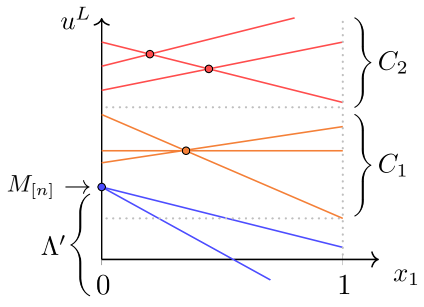

Figure 1: Non-dominance components C_1 and C_2 and the set Λ. The leader has two actions in the illustrated instance, so her mixed strategy can be represented by the probability x_1. Each line represents $u^L(x_1, j)$ for an action j of the follower. Note that actions in non-dominance components do not necessarily obviously dominate actions in Λ. In this instance, we have $\Lambda'' = \emptyset$ and $M_{[n]} = M_{\{j\}}$ for any $j \in \Lambda'$ (i.e., the maximum of $u^L(x_1, j)$). If we remove the two actions in Λ' from this example, the set C_1 will become Λ'' and Λ' will be empty, in which case $M_{[n]}$ is achieved at the intersection of the lines in Λ'' (C_1).

instances where $\Lambda = \emptyset$ and all the followers' actions are in the same non-dominance component,[3] the function $\tilde{u}^L$ we obtain will be equivalent to u^L in the entire strategy space $\Delta_m \times \Delta_n$.

In addition to the in- and cross-component equivalences, $\tilde{u}^L$ is also inducibility-equivalent to u^L.

Definition 6 (Inducibility-equivalence)**.** A payoff matrix $\tilde{u}^L \in \mathbb{R}^{m \times n}$ is *inducibility-equivalent* to u^L if $\mathrm{SSE}(\tilde{u}^L) = \mathrm{SSE}(u^L)$.

A notable application of inducibility-equivalence is the following problem proposed by Birmpas et al. [2021]: $\max_{\tilde{u}^F, (\mathbf{x}, j) \in \mathrm{SSE}(u^L, \tilde{u}^F)} u^F(\mathbf{x}, j)$, i.e., to find an inducible SSE that is optimal w.r.t. a follower's actual payoff. The inducibility-equivalence between $\tilde{u}^L$ and u^L enables us to solve the problem based on $\tilde{u}^L$, instead of u^L. When u^L is known, a polynomial-time algorithm based on the characterization in Theorem 2.1 is provided by Birmpas et al. [2021] to solve this problem.

4 Warm-up: Basic Results via the SSE Oracle

Before presenting our overall approach, we first describe several basic results that can be efficiently obtained by using the SSE oracle. The SSE oracle does not directly reveal any payoffs but provides hints about the relation of payoffs in SSEs. For example, if through the oracle we can confirm that two strategy profiles $(\mathbf{x}, j)$ and $(\mathbf{x}', j')$ are both SSEs, then we know that $u^L(\mathbf{x}, j) = u^L(\mathbf{x}', j')$. With this in mind, we can easily and efficiently identify the following useful information.

Observation 4.1. *With query access to $\mathcal{A}_{\mathrm{SSE}}$, for every follower action $j \in [n]$, the set of the* leader's *(pure) best responses to j, $I_j := \mathrm{argmax}_{i \in [m]}\, u^L(i, j)$, can be computed efficiently. Hence, $\mathcal{M}_{\{j\}} = \{\mathbf{x} \in \Delta_m : \sum_{i \in I_j} x_i = 1\}$, which is the convex hull of I_j, can be computed efficiently, too.*

Specifically, to decide whether $i \in I_j$, we can use a follower type $\tilde{u}^F$ where j is strictly dominant, i.e., $\tilde{u}^F(i, j) = 1$ and $\tilde{u}^F(i, j') = 0$ for all $j' \neq j$. Then $\widetilde{\mathrm{BR}}^{-1}(j) = \Delta_m$ is the only non-empty best response region. By definition, $(i, j) \in \mathrm{SSE}(u^L, \tilde{u}^F) \iff u^L(i, j) =$

[3] Having all the actions in the same non-dominance component is even milder than the condition where *no* action obviously dominates another. Also note that obvious dominance is a stronger notion than *strong dominance* (see Definition 9).

$\max_{\mathbf{x}\in\Delta_m} u^L(\mathbf{x}, j) \iff i \in I_j$. We can decide whether $i \in I_j$ by checking if $(i, j) \in \mathrm{SSE}(u^L, \tilde{u}^F)$ holds via querying $\mathcal{A}_{\mathrm{SSE}}$.

Observation 4.2. *With query access to $\mathcal{A}_{\mathrm{SSE}}$, for every $i \in [m]$ and $j, k \in [n]$, the relation (i.e., $>$, $=$, or $<$) between $M_{\{j\}}$ and $u^L(i, k)$ can be decided efficiently.*

To decide the above relation, we slightly modify $\tilde{u}^F$ defined above by letting $\tilde{u}^F(i, k) = 1$. This results in $\widetilde{\mathrm{BR}}^{-1}(k) = \{i\}$, while all other best response regions remain as before. Consequently, $u^L(i, k)$ and $M_{\{j\}}$ are the leader's maximum utilities in the respective regions. We can learn the relation by querying whether (i, k) and some (i', j) for some $i' \in I_j$ are in $\mathrm{SSE}(u^L, \tilde{u}^F)$.

5 Main Theorem and Approach Overview

Our main approach is to learn *affine transformations* of u^L. By writing u^L as

$$u^L(\mathbf{x}, j) \equiv \gamma_j \cdot \mathbf{a}_j \cdot \mathbf{x} + \beta_j \tag{6}$$

for some $\gamma_j \in \mathbb{R}_{>0}$, $\mathbf{a}_j \in \mathbb{R}^n$, and $\beta_j \in \mathbb{R}$, we address two tasks: 1) learn the gradient direction $\mathbf{a}_j$ for each $j \in [n] \setminus \Lambda'$, and 2) learn the relative parameters γ_j/γ_{j^*} and $(\beta_j - \beta_{j^*})/\gamma_{j^*}$ within each component. Specifically, details of learning $\mathbf{a}_j$ is presented in Section 6, where the set Λ' will also be identified as a byproduct. Based on the gradient directions, we proceed as follows.

5.1 In-Component Equivalence

The non-dominance relations between all pairs of actions can be determined efficiently via Observation 4.2, so we obtain the non-dominance graph. The non-dominance components can be identified by using standard algorithms for computing connected components of a graph. For each component C, we fix an action $j^* \in C$ and learn additional information to infer two quantities γ_j/γ_{j^*} and $(\beta_j - \beta_{j^*})/\gamma_{j^*}$ for every action $j \in C$. (The exact values of γ_j, γ_{j^*}, β_j, and β_{j^*} remain unknown.) With these quantities, we construct the following function:

$$\tilde{u}^L(\mathbf{x}, j) = \tfrac{\gamma_j}{\gamma_{j^*}} \cdot \mathbf{a}_j \cdot \mathbf{x} + \tfrac{\beta_j - \beta_{j^*}}{\gamma_{j^*}}. \tag{7}$$

Given Eq. (6), $\tilde{u}^L$ is then an affine transformation of u^L (for actions in C), such that $\tilde{u}^L = u^L/\gamma_{j^*} - \beta_{j^*}/\gamma_{j^*}$. The equivalence between $\tilde{u}^L$ and u^L in $\Delta_m \times \Delta(C)$ then follows immediately.[4]

The details of learning γ_j/γ_{j^*} and $(\beta_j - \beta_{j^*})/\gamma_{j^*}$ are shown in Section 7. Note that, we actually begin with a *supergraph* of the non-dominance graph, defined on a larger vertex set $[n] \setminus \Lambda'$ as Λ'' is initially unknown. However, actions in Λ'' will be detected during the learning process. Whenever we detect an action $\lambda \in \Lambda''$, we restart the algorithm with the smaller set where λ is excluded.

5.2 Cross-Component Equivalence

By definition, an action either dominates (w.r.t. the leader's payoff) all the actions in another component, or is dominated by all of them. Hence, we can order the components so that every action in the $(\ell + 1)$-th component $C_{\ell+1}$ dominates every action in the ℓ-th, C_ℓ (see Fig. 1). Formally,

$$\min_{i\in[m], j\in C_{\ell+1}} u^L(i, j) \geq \max_{i\in[m], j\in C_\ell} u^L(i, j) \tag{8}$$

[4]Since j^* is different for different components, $\tilde{u}^L$ is *not* an affine transformation of u^L w.r.t. all the columns. Therefore, we cannot conclude that $\tilde{u}^L$ is equivalent to u^L in the entire space $\Delta_m \times \Delta_n$.

for all $\ell = 1, \ldots, K-1$, where K denotes the number of components. To achieve the equivalence across the components in $\Delta_m \times ([n] \setminus \Lambda)$, we modify the matrix obtained in Section 5.1: we raise the values in each component by an appropriate amount to capture the dominance of each $C_{\ell+1}$ against C_ℓ. (Intuitively, we stack the components on top of each other according to their order in u^L, as in Fig. 1.) The resulting matrix preserves the equivalence in $\Delta_m \times \Delta(C)$ for each component C because such modification does not change the relative payoffs of the actions in each component.

Proposition 5.1. *Given a matrix $\tilde{u}^L$ that is equivalent to u^L in $\Delta_m \times C$ for every non-dominance component C, a matrix equivalent to u^L in $\Delta_m \times ([n] \setminus \Lambda)$ can be computed in time polynomial in the total bit size of $\tilde{u}^L$ and u^L.*

5.3 Inducibility-Equivalence

Finally, to achieve the inducibility-equivalence, we apply the following modification to $\tilde{u}^L$ obtained above (which already ensures the in- and cross-component equivalence). As we will show next, for a suitable choice of v, the matrix $\tilde{u}'_v$ will be inducibility-equivalent to u^L. For every $i \in [m], j \in [n]$:

$$\tilde{u}'_v(i,j) \coloneqq \begin{cases} \tilde{u}^L(i,j) & \text{if } j \notin \Lambda \\ v + a_{ji} - \mathbf{a}_j \cdot \mathbf{x}_j^* & \text{if } j \in \Lambda'' \\ v + \tilde{a}_{ji} - \tilde{\mathbf{a}}_j \cdot \mathbf{x}_j^* & \text{if } j \in \Lambda' \end{cases} \tag{9}$$

where $\mathbf{x}_j^*$ is an arbitrary strategy such that $u^L(\mathbf{x}_j^*, j) = M_{[n]}$, and each $\tilde{\mathbf{a}}_j$ is a vector such that $\tilde{a}_{ji} = 1$ if $i \in \mathcal{M}_{\{j\}}$, and $\tilde{a}_{ji} = 0$, otherwise. The strategies $\mathbf{x}_j^*$ will be obtained along with Λ' and Λ'' (see Propositions 6.9 and 7.3), while $\tilde{\mathbf{a}}_j$ can be obtained by using Observation 4.1.

Intuitively, with different v, $\tilde{u}'_v$ changes the values of actions in Λ relative to the values of the other actions. The key is to find a v that matches the values of strategy profiles that separate inducible and non-inducible strategy profiles. (The boundary condition is given by Theorem 2.1.) We show that the following value v^* of v is the desired one:

$$v^* \coloneqq \begin{cases} \phi^{-1}(M_{[n]}) & \text{if } u^L(s) = M_{[n]} \text{ for some } s \in \Delta_m \times ([n] \setminus \Lambda); \\ \min_{s \in \Delta_m \times ([n]\setminus\Lambda)} \tilde{u}^L(s) & \text{otherwise;} \end{cases} \tag{10}$$

where $\phi : \mathbb{R} \to \mathbb{R}$ is a function that converts the values of strategy profiles under $\tilde{u}^L$ to their values under u^L, i.e., $\tilde{u}^L(s) = v \iff u^L(s) = \phi(v)$ for every $s \in \Delta_m \times ([n] \setminus \Lambda)$. It is straightforward that if $\tilde{u}^L$ and u^L are equivalent in $\Delta_m \times ([n] \setminus \Lambda)$, then ϕ is well-defined and strictly increasing.

Lemma 5.2 proves the inducibility-equivalence between $\tilde{u}'_{v^*}$ and u^L. Moreover, it offers a way to decide whether $v \geq v^*$ or $v < v^*$ for any given v, so we can compute v^* via binary search (it is impossible to obtain v^* directly from Eq. (10) due to unknown parameters). Since Eq. (9) does not change the values of actions not in Λ, the matrix $\tilde{u}'_{v^*}$ preserves the equivalences achieved in Sections 5.1 and 5.2. Finally, in addition to deciding the inducibility, statement (i) in Lemma 5.2 allows us to efficiently compute a follower type that induces a given strategy profile. This gives Proposition 5.3.

Lemma 5.2. *Suppose that $\tilde{u}^L$ is equivalent to u^L in $\Delta_m \times ([n] \setminus \Lambda)$. Then $\tilde{u}'_{v^*}$ is inducibility-equivalent to u^L. Moreover, for all $s \in \Delta_m \times ([n] \setminus \Lambda)$ such that $\tilde{u}^L(s) = v$:*

(i) if $v \geq v^*$, then $s \in \mathrm{SSE}(\tilde{u}'_v)$ and, for all $\tilde{u}^F \in \mathbb{R}^{m \times n}$, $s \in \mathrm{SSE}(\tilde{u}'_v, \tilde{u}^F) \implies s \in \mathrm{SSE}(u^L, \tilde{u}^F)$;

(ii) if $v < v^*$, then $s \notin \mathrm{SSE}(u^L)$.

Proposition 5.3. *Given a matrix $\tilde{u}^L$ equivalent to u^L in $\Delta_m \times ([n] \setminus \Lambda)$, a matrix $\tilde{u}'$ inducibility-equivalent to u^L can be computed in polynomial time (in the total bit size of $\tilde{u}^L$ and u^L). Moreover, for every $s \in \mathrm{SSE}(u^L)$, a follower type $\tilde{u}^F$ such that $s \in \mathrm{SSE}(u^L, \tilde{u}^F)$ can be computed in polynomial time.*[5]

Specifically, to use Lemma 5.2 to decide the relation of v^* and v, we pick a strategy profile s with $\tilde{u}^L(s) = v$, and ask the oracle if $s \in SSE(\tilde{u}^L, \tilde{u}^F)$ for $\tilde{u}^F$ such that $s \in SSE(\tilde{u}'_v, \tilde{u}^F)$. The type $\tilde{u}^F$ can be computed efficiently via the algorithm by Birmpas et al. [2021]. Moreover, we can compute the exact value of v^* using the Stern-Brocot tree [Graham et al., 1989], given an upper bound on the bit size of v^*.[6]

5.4 Summary and Application to Equilibrium Inducing

Given the procedures outlined above, the next sections focus on learning the gradient direction $\mathbf{a}_j$ (Section 6), and the quantities γ_j/γ_{j^*} and $(\beta_j - \beta_{j^*})/\gamma_{j^*}$ (Section 7) for achieving the in- and cross-component equivalences. During these processes, we will also obtain Λ', Λ'', and $\mathbf{x}_j^*$ (for constructing $\tilde{u}'_v$ in Eq. (9)). The following theorem summarizes our main result.

Theorem 5.4. *With query access to $\mathcal{A}_{\mathrm{SSE}}$, in time polynomial in the bit size of u^L, we can compute a matrix $\tilde{u}^L$ that is simultaneously inducibility-equivalent to u^L, equivalent to u^L in $\Delta_m \times ([n] \setminus \Lambda)$, and equivalent to u^L in $\Delta_m \times \Delta(C)$ for every non-dominance component C. Moreover, for every $s \in \mathrm{SSE}(u^L)$, a follower type $\tilde{u}^F$ such that $s \in \mathrm{SSE}(u^L, \tilde{u}^F)$ can be computed in polynomial time.*

As a corollary of Theorem 5.4, we can efficiently solve the equilibrium inducing problem, without knowing u^L. Namely, we can compute a $\tilde{u}^F$ to induce an SSE s^* maximizing a given objective u^F. One can use $\tilde{u}^F$ to misinform the leader about their incentive and steer the game to a more profitable equilibrium s^*. This extends the results of Birmpas et al. [2021] to the incomplete information setting.

Corollary 5.5. *With query access to $\mathcal{A}_{\mathrm{SSE}}$, for any given payoff function $u^F \in \mathbb{R}^{m \times n}$, one can compute a strategy profile $s^* \in \operatorname{argmax}_{s \in \mathrm{SSE}(u^L)} u^F(s)$ and a matrix $\tilde{u}^F \in \mathbb{R}^{m \times n}$ such that $s^* \in \mathrm{SSE}(u^L, \tilde{u}^F)$ in time polynomial in the bit size of u^L.*

6 Learning Gradient Directions

We demonstrate how to learn a gradient direction $\mathbf{a}_j$ for every $j \notin \Lambda'$, i.e.,

$$\gamma_j \cdot \mathbf{a}_j = \nabla u^L(\cdot, j) \equiv \big(u^L(1, j), \ldots, u^L(m, j)\big)$$

for some positive factor γ_j (which need not be known). To simplify the notation, we will present an algorithm for learning $\mathbf{a}_n$. The other $\mathbf{a}_j$'s can be learned analogously.

To learn $\mathbf{a}_n$, intuitively, we construct a correspondence $\widetilde{\mathrm{BR}}$ where the boundary of $\widetilde{\mathrm{BR}}^{-1}(n)$ is perpendicular to $\nabla u^L(\cdot, n)$, so that we can infer $\mathbf{a}_n$ based on the normal vector of the boundary. Since the oracle $\mathcal{A}_{\mathrm{SSE}}$ only exposes information about SSEs, the construction needs to ensure

[5] Note that, however, $s \in \mathrm{SSE}(\tilde{u}', \tilde{u}^F)$ does not necessarily imply $s \in \mathrm{SSE}(u^L, \tilde{u}^F)$. Indeed, as shown in the proof, we obtain $\tilde{u}'_{v^*}$ as an inducibility-equivalent matrix to u^L, but use $\tilde{u}'_v$ to compute $\tilde{u}^F$, where $v = \tilde{u}^L(s)$.

[6] All the binary searches in this paper are based on the Stern-Brocot tree, so that we find the *exact* rational numbers. Notably, if we can distinguish whether $v > v^*$, $v = v^*$, or $v < v^*$ for any v, the search will terminate automatically when $v = v^*$, in which case there is no need to know the exact bit size bound of v^*, in order to determine when to terminate the search. As long as the bit size is bounded by a polynomial, the binary search terminates in polynomial time. Hence, only when we cannot distinguish the three possible relations of v and v^* (e.g, we can only distinguish $v > v^*$ or $v \leq v^*$), we derive an upper bound on the bit size to determine when to terminate the search. And this holds for most binary searches in this paper.

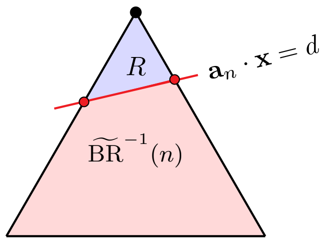

Figure 2: A BR-correspondence $\widetilde{\mathrm{BR}}$ constructed for learning $\mathbf{a}_n$. The triangle represents the strategy space Δ_m of the leader. The black dot at the top represents $\mathcal{M}_{\{n\}}$ in this example. The two red dots on the boundary of $\widetilde{\mathrm{BR}}^{-1}(n)$ are two critical points defining the boundary hyperplane.

that all the points on the boundary of $\widetilde{\mathrm{BR}}^{-1}(n)$ are SSE strategies. For example, in Fig. 2, we want the boundary to be at a position where both red points, say $\mathbf{x}^1$ and $\mathbf{x}^2$, are SSE strategies: $(\mathbf{x}^1, n), (\mathbf{x}^2, n) \in \mathrm{SSE}(u^L, \tilde{u}^F)$. This can be confirmed via the oracle $\mathcal{A}_{\mathrm{SSE}}$, and it implies that $u^L(\mathbf{x}^1, n) = u^L(\mathbf{x}^2, n)$, so the boundary defined by $\mathbf{x}^1$ and $\mathbf{x}^2$ is perpendicular to $\nabla u^L(\cdot, n)$.

For ease of description, we reorder the leader's actions $i \in [m]$ according to $u^L(i, n)$ so that, throughout this section, for some $m^* \in [m]$ and all $i \in [m_* - 1]$:

$$u^L(i, n) < u^L(m_*, n) = u^L(m_* + 1, n) = \cdots = u^L(m, n) = M_{\{n\}}. \tag{11}$$

The payoff information required for the reordering is known by Observation 4.1. W.l.o.g., we can also assume that $m_* > 1$: in the case where $m_* = 1$, the payoff $u^L(i, n)$ is the same for all $i \in [m]$. Trivially, this implies that the gradient of $u^L(\cdot, n)$ is $\mathbf{0}$, so $\mathbf{a}_n = \mathbf{0}$ is all we need.

Now we formalize our approach via Lemma 6.1, which reduces our task to finding a set of $m_* - 1$ points $\mathbf{x}^1, \ldots, \mathbf{x}^{m_*-1}$ that satisfy the properties stated in the lemma. In Fig. 2, these points are the red ones that define the boundary between R and $\widetilde{\mathrm{BR}}^{-1}(n)$; we have $m_* = m$ in this example. When $m_* < m$, the boundary may be a more complicated surface formed by multiple hyperplanes (e.g., Fig. 3). The points $\mathbf{x}^1, \ldots, \mathbf{x}^{m_*-1}$ define a face on this surface where the leader's payoffs are equal. $\nabla u^L(\cdot, n)$ is perpendicular both to this face and to a $(m + 1 - m_*)$-dimensional face of Δ_m.

Lemma 6.1. *Suppose that the following conditions hold for strategies $\mathbf{x}^1, \ldots, \mathbf{x}^{m_*-1} \in \Delta_m$:*

(i) $u^L(\mathbf{x}^1, n) = \cdots = u^L(\mathbf{x}^{m_*-1}, n)$;

(ii) $x_i^i > 0$ for all $i \in [m_* - 1]$; and

(iii) $1 - x_i^i = \sum_{k=m_*}^{m} x_k^i$ for all $i \in [m_* - 1]$.

Then $(-1/x_1^1, -1/x_2^2, \cdots, -1/x_{m_*-1}^{m_*-1}, 0, \cdots, 0) = \gamma \cdot \nabla u^L(\cdot, n)$ for some $\gamma > 0$.

For any given candidate points $\mathbf{x}^1, \ldots, \mathbf{x}^{m_*-1}$, we must verify that they indeed satisfy the above properties before using them to compute the gradient direction. While properties (ii) and (iii) can be verified directly based on the probability values, to verify whether (i) holds requires information in u^L. Since $\mathcal{A}_{\mathrm{SSE}}$ is the only source of this information, we use candidate points such that $(\mathbf{x}^i, n) \in \mathrm{SSE}(u^L, \tilde{u}^F)$—which is verifiable via $\mathcal{A}_{\mathrm{SSE}}$—to ensure that (i) holds. To achieve this, the correspondence $\widetilde{\mathrm{BR}}$ must assign sufficiently low payoffs to leader strategies in $R = \Delta_m \setminus \widetilde{\mathrm{BR}}^{-1}(n)$ (see Fig. 2), so that these strategies do not prevent $(\mathbf{x}^i, n)$ from being an SSE. In some cases, more than one follower response is needed to "cover" R under low payoffs. This motivates the notion of *maximin-cover*.

Definition 7 (Cover)**.** A follower payoff matrix μ is a *maximin-cover* (or *cover*) of $S \subseteq [n]$ if

$$\max_{\mathbf{x} \in \mathcal{M}_S} \max_{j \in \widetilde{\mathrm{BR}}(\mathbf{x})} u^L(\mathbf{x}, j) < M_S, \tag{12}$$

where $\widetilde{\mathrm{BR}}$ denotes the BR-correspondence defined by μ. It is a *proper cover* of S if it holds in addition that $S \cap \widetilde{\mathrm{BR}}(\mathbf{x}) = \emptyset$ for all $\mathbf{x} \in \Delta_m$.

Namely, we want to apply a cover of $\{n\}$ to the region R to limit the leader's maximum attainable payoff in this region. Intuitively, in Fig. 2, when we move the boundary between R and $\widetilde{\mathrm{BR}}^{-1}(n)$ towards the vertex at the top, the leader's maximum attainable payoffs in R and $\widetilde{\mathrm{BR}}^{-1}(n)$ will approach the left and right sides, respectively, of Eq. (12) (with $S = \{n\}$).

The additional requirement that a cover is *proper* is useful, because we do not want any strategies in R to induce n in order to make the boundary of $\widetilde{\mathrm{BR}}^{-1}(n)$ easy to manage. As it turns out in Lemma 6.2, a proper cover is computationally *not* more demanding than a cover. Therefore, in what follows, unless otherwise specified, we will simply refer to a proper cover as a cover. By Lemma 6.3, the existence of a cover is characterized by a positive gap between M_S and the maximin value $M_{[n]}$.

Lemma 6.2. *Given a cover of set $S \subseteq [n]$, a proper cover of S can be constructed in polynomial time.*

Lemma 6.3. *For any $S \subseteq [n]$, a cover of S exists if and only if $M_S > M_{[n]}$.*

Next, we first demonstrate in Section 6.1 that, given a cover of $\{n\}$, a set of strategies $\mathbf{x}^1, \dots, \mathbf{x}^{m_*-1}$ satisfying the properties in Lemma 6.1 can be computed efficiently; hence we obtain a desired $\mathbf{a}_n$. Section 6.2 then demonstrates how to efficiently find a cover, or decide that no cover exists. When no cover exists, we can conclude that $M_{\{n\}} = M_{[n]}$ and hence $n \in \Lambda'$. So, the attempt to learn the $\mathbf{a}_j$'s will also reveal the set Λ'. According to our agenda outlined in Section 5, there is no need to further learn $\mathbf{a}_n$ if $n \in \Lambda'$.

In summary, the approach to learning $\mathbf{a}_n$ is as follows.

1. Compute a cover μ of $\{n\}$ (see Section 6.2).
2. If no cover of $\{n\}$ exists, claim that $n \in \Lambda'$ and terminate.
3. Otherwise, use μ to compute $\mathbf{x}^1, \dots, \mathbf{x}^{m_*-1}$ satisfying the properties in Lemma 6.1 (see Section 6.1). Output $\mathbf{a}_n = (-1/x_1^1, -1/x_2^2, \cdots, -1/x_{m_*-1}^{m_*-1}, 0, \cdots, 0)$.

6.1 Computing $\mathbf{x}^1, \dots, \mathbf{x}^{m_*-1}$

We show how to compute $\mathbf{x}^1, \dots, \mathbf{x}^{m_*-1}$, given a cover μ of $\{n\}$. For each $i \in [m_* - 1]$, define

$$\Gamma_i := \left\{ \mathbf{x} \in \Delta_m : 1 - x_i = \sum\nolimits_{k=m_*}^{m} x_k \right\}, \tag{13}$$

which is a face of Δ_m consisting of leader strategies satisfying Lemma 6.1 (iii). We aim to construct a function $\tilde{u}^F$ that induces a strategy $\mathbf{x}^i \in \Gamma_i$ to form an SSE with n, for each $i \in [m_* - 1]$. This requires the leader's maximum attainable payoff under $\tilde{u}^F$ to be achievable at every Γ_i. If $\mathcal{M}_{\{n\}}$ is covered by only one follower action, say action i, this is relatively easy to achieve because the boundary separating $\widetilde{\mathrm{BR}}^{-1}(n)$ and $\widetilde{\mathrm{BR}}^{-1}(i)$ will be a hyperplane as in Fig. 2. However, if more than one action is used to cover $\mathcal{M}_{\{n\}}$, the shape of the separating boundary may become irregular, potentially with protruding vertices within the boundary's interior. We need a better construction to ensure that these vertices do not yield payoffs higher than the leader's best strategies within Γ_i.

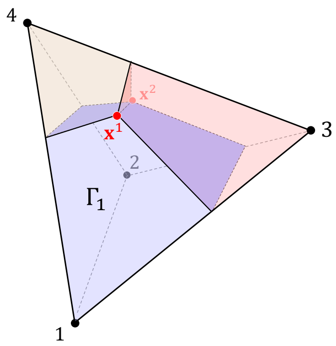

Figure 3: An illustration of $f_{\mathbf{g}}(n)$ where $m = 4$, $n = 3$, and $m_* = 3$. This means $u^L(i,3) < u^L(3,3) = u^L(4,3) = M_{\{3\}}$ for $i = 1, 2$ according to Eq. (11); hence, $\mathcal{M}_{\{3\}}$ is the edge connecting the vertices 3 and 4. The polytope represents Δ_4 and is divided into three best response regions $f_{\mathbf{g}}^{-1}(1)$, $f_{\mathbf{g}}^{-1}(2)$, and $f_{\mathbf{g}}^{-1}(3)$, where the light blue one is $f_{\mathbf{g}}^{-1}(3)$. The set Γ_1, as defined in Eq. (13), is the facet at the front, where $x_1 + x_3 + x_4 = 1$ for all $\mathbf{x}$.

To proceed, pick arbitrary $\mathbf{x}^1, \dots, \mathbf{x}^{m_*-1}$ such that

$$\mathbf{x}^i \in \operatorname*{argmin}_{\mathbf{x} \in \Gamma_i \cap f_{\mathbf{g}}^{-1}(n)} x_i, \tag{14}$$

where $f_{\mathbf{g}} : \Delta_m \to 2^{[n]}$ denotes the BR-correspondence of the following follower type $\tilde{u}^F_{\mathbf{g}}$, parameterized by $\mathbf{g} = (g_1, \dots, g_{m_*-1}) \in \mathbb{R}^{m_*-1}$:

$$\tilde{u}^F_{\mathbf{g}}(\mathbf{x}, j) \coloneqq \begin{cases} \sum_{i=m_*}^{m} x_i \cdot \mu(i,j) & \text{if } j < n; \\ \sum_{i=1}^{m_*-1} x_i \cdot g_i + W \cdot \sum_{i=m_*}^{m} x_i & \text{if } j = n. \end{cases} \tag{15}$$

where $W = \min_{i,j} \mu(i,j) - 1$.

Fig. 3 illustrates the above construction. Intuitively, each component g_i of $\mathbf{g}$ controls how far we want to push $\mathbf{x}^i$ towards $\mathcal{M}_{\{n\}}$. We show next how to find an appropriate $\mathbf{g}$ that results in $\mathbf{x}^1, \dots, \mathbf{x}^{m_*-1}$ defined above satisfying the conditions in Lemma 6.1. We first observe that $\mathbf{x}^i$ is well-defined—that is, $\Gamma_i \cap f_{\mathbf{g}}^{-1}(n) \neq \emptyset$—as long as $g_i > 0$. Moreover, $\mathbf{x}^1, \dots, \mathbf{x}^{m_*-1}$ satisfy (ii) and (iii) in Lemma 6.1 (where (iii) holds by Eq. (13)).

Observation 6.4. *If $g_i > 0$, then $\Gamma_i \cap f_{\mathbf{g}}^{-1}(n) \neq \emptyset$ and $x_i^i > 0$.*

It remains to achieve (i) in Lemma 6.1. We define the following set for every $\mathbf{g} \in \mathbb{R}^{m_*-1}_{>0}$:

$$I_{\mathbf{g}} \coloneqq \left\{ i \in [m_* - 1] : (\mathbf{x}^i, n) \in \mathrm{SSE}(u^L, \tilde{u}^F_{\mathbf{g}}) \right\}, \tag{16}$$

whereby the aim is to find $\mathbf{g}$ that makes $I_{\mathbf{g}} = [m_* - 1]$, so that (i) holds. Clearly, for any given $\mathbf{x}^1, \dots, \mathbf{x}^{m_*-1}$, $I_{\mathbf{g}}$ can be computed by using the oracle $\mathcal{A}_{\mathrm{SSE}}$. Moreover, for any $\mathbf{x} \in \Gamma_i$, the leader's utility $u^L(\mathbf{x}, n)$ depends only on x_i, as $x_i = 1 - \sum_{k=m_*}^{m} x_k$ and $u^L(m_*, n) = \cdots = u^L(m, n)$ (Observation 6.5). Hence, the definition of $I_{\mathbf{g}}$ is invariant w.r.t. the specific choice of $\mathbf{x}^i$ in Eq. (14).

Observation 6.5. *For all $\mathbf{x} \in \Gamma_i$, it holds that $u^L(\mathbf{x}, n) = c_i \cdot x_i + d_i$ for some constants $c_i < 0$ and d_i.*

We will gradually increase each component of $\mathbf{g}$ until $I_{\mathbf{g}} = [m_* - 1]$. According to Lemma 6.6, this can be done efficiently. Intuitively, increasing each g_i causes x_i^i and, in turn, $u^L(\mathbf{x}^i, n)$ to increase. So, ideally, when $u^L(\mathbf{x}^i, n)$ is sufficiently large, $(\mathbf{x}^i, n)$ yields as much utility as the other optimal strategy profiles and becomes an SSE. Notably, $\max_{\mathbf{x}\in f^{-1}(n)\setminus(\cup_{i\in[m_*-1]}\Gamma_i)} u^L(\mathbf{x}, n)$ may also increase with g_i. As we show in the proof of Lemma 6.6, our design ensures that no extreme point of $f_{\mathbf{g}}^{-1}(n)$ appears in the interior of Δ_m (also see Fig. 3). Therefore, points outside $\bigcup_{i\in[m_*-1]} \Gamma_i$ will never yield a higher utility than $(\mathbf{x}^i, n)$. $\mathbf{x}^1, \ldots, \mathbf{x}^{m_*-1}$ are always the leader's optimal strategies in $f_{\mathbf{g}}^{-1}(n)$.

Lemma 6.6. *A vector* $\mathbf{g}$ *such that* $I_{\mathbf{g}} = [m_* - 1]$ *can be computed in polynomial time.*

By definition, $\mathbf{x}^1, \ldots, \mathbf{x}^{m_*-1}$ defined with this $\mathbf{g}$ then satisfy all the conditions in Lemma 6.1.

Proposition 6.7. *Given a cover* μ *of* $\{n\}$*, a set of strategies* $\mathbf{x}^1, \ldots, \mathbf{x}^{m_*-1} \in \Delta_m$ *that satisfy the conditions in Lemma 6.1 can be computed in polynomial time.*

6.2 Computing a Cover of $\{n\}$

Now we describe how to compute a cover of $\{n\}$ to complete this section. We first deal with an easier case, where $n \notin S^* := \operatorname{argmin}_{j\in[n]} M_{\{j\}}$. In this case, any arbitrary $\ell \in S^*$ gives

$$u^L(\mathbf{x}, \ell) \le M_{\{\ell\}} < M_{\{n\}} \quad \text{for all } \mathbf{x} \in \Delta_m.$$

Thus, any payoff matrix where ℓ strictly dominates all other actions forms a cover of $\{n\}$. Using results in Section 6.1, we can then compute $\mathbf{a}_n$. So, in what follows, we assume $\mathbf{a}_j$ is given for all $j \in [n] \setminus S^*$. Note that we can efficiently decide whether $j \in S^*$ using $\mathcal{A}_{\text{SSE}}$, similarly to Observation 4.1.

Next, we consider the case where $n \in S^*$. We present a more general result, Proposition 6.8, which finds a cover for any $S \subseteq [n]$. This result will also be useful for arguments in the next sections, where we need to find covers for size-2 subsets of $[n]$. To apply this method requires a base function that restrict the leader's utility to at most M_S and the follower's best responses to actions in S (Definition 8). By this definition, a follower type where n strictly dominates all other actions is a straightforward base function for $\{n\}$.

Definition 8 (Base function)**.** A follower type $\tilde{u}^F$ with BR-correspondence $\widetilde{\text{BR}}$ is a *base function* of a set $S \subseteq [n]$ if: 1) $\widetilde{\text{BR}}(\mathbf{x}) \subseteq S$ for all $\mathbf{x} \in \Delta_m$, and 2) the leader's SSE payoff in $(u^L, \tilde{u}^F)$ is M_S.

Proposition 6.8. *Suppose that* $S \subseteq [n]$*,* $Q = \{j \in [n] : M_{\{j\}} = M_S\}$*, and the following are given: 1)* $\mathbf{a}_j$ *for all* $j \in [n] \setminus (S \cup Q)$*, 2) a base function* $\tilde{u}^F$ *of* S*, and 3)* $\mathcal{M}_S$ *(given by linear constraints). In polynomial time, we can either compute a cover of* S *or correctly decide that* S *does not admit a cover.*

Intuitively, a cover aims to bring down the leader's maximum attainable payoff in $\mathcal{M}_S$ to below M_S, so it needs to avoid responding actions j that would result in $u^L(\mathbf{x}, j) \ge M_S$. Ideally, we could just use $-u^L$ to achieve this, but since we do not know u^L, we learn a cover that functions similarly.

When no cover of $\{n\}$ exists, Lemma 6.3 implies that $M_{\{n\}} = M_{[n]}$, so $n \in \Lambda'$. Consequently, our attempt to obtain a cover for every action reveals Λ' as a byproduct. For every $j \in \Lambda'$, we can compute $\mathcal{M}_{\{j\}}$ according to Observation 4.1. Any arbitrary strategy $\mathbf{x} \in \mathcal{M}_{\{j\}}$ gives $u^L(\mathbf{x}, j) = M_{\{j\}} = M_{[n]}$ and serves as a necessary component for the construction in Eq. (9).

Proposition 6.9. Λ' *can be computed in polynomial time. Moreover, for every* $j \in \Lambda'$*, a strategy* $\mathbf{x}$ *such that* $u^L(\mathbf{x}, j) = M_{[n]}$ *can be computed in polynomial time.*

7 Constructing Affine Transformations

Recall that given the gradient information $\mathbf{a}_j$ we have obtained for each $j \notin \Lambda'$, we can write

$$u^L(\mathbf{x}, j) \equiv \gamma_j \cdot \mathbf{a}_j \cdot \mathbf{x} + \beta_j$$

for some constants $\gamma_j \in \mathbb{R}_{>0}$ and $\beta_j \in \mathbb{R}$ unknown to us. We now present how to learn the quantities γ_j/γ_{j^*} and $(\beta_j - \beta_{j^*})/\gamma_{j^*}$ for every $j \in C$ in each non-dominance component C, where $j^* \in C$ is a fixed action for every $j \in C$. As described in Section 5, these quantities will allow us to construct an in-component affine transformation $\tilde{u}^L = u^L/\gamma_{j^*} - \beta_{j^*}/\gamma_{j^*}$.

7.1 Overview

Recall that the set of vertices on non-dominance graph is $[n] \setminus \Lambda$. Since Λ'' (and hence Λ) is still unknown, we will start with a supergraph defined based on a larger vertex set $[n] \setminus \Lambda'$. We will gradually identify Λ'' during the process described in this section. Once any follower action $j \in \Lambda''$ is discovered, we exclude it from the graph and rerun our algorithm.

Given a non-dominance component C, we pick an arbitrary $j^* \in C$ and learn γ_j/γ_{j^*} and $(\beta_j - \beta_{j^*})/\gamma_{j^*}$ for every $j \in C$ as follows.

1. Find a path $(j_1, \ldots, j_{k+1})$ on the non-dominance graph such that: $j_1 = j^*$, $j_{k+1} = j$, and $\mathbf{a}_{j_\ell} \neq \mathbf{0}$ for all $\ell = 1, \ldots, k$.
2. Learn $c_\ell \coloneqq \gamma_{j_{\ell+1}}/\gamma_{j_\ell}$ and $b_\ell \coloneqq (\beta_{j_{\ell+1}} - \beta_{j_\ell})/\gamma_{j_\ell}$ for every $\ell = 1, \ldots, k$.
3. Output $\gamma_j/\gamma_{j^*} = \prod_{\ell=1}^{k} c_\ell$ and $(\beta_j - \beta_{j^*})/\gamma_{j^*} = \sum_{\ell=1}^{k} b_\ell \prod_{\ell'=1}^{\ell} c_{\ell'}$.

Namely, we find a path connecting j and j^* in the graph and learn the values c_ℓ and b_ℓ. The values of γ_j/γ_{j^*} and $(\beta_j - \beta_{j^*})/\gamma_{j^*}$ can then be calculated via the chains of products in Step 3, i.e.,

$$\gamma_j/\gamma_{j^*} = \prod_{\ell=1}^{k} c_\ell \quad \text{and} \quad (\beta_j - \beta_{j^*})/\gamma_{j^*} = \sum_{\ell=1}^{k} b_\ell \prod_{\ell'=1}^{\ell} c_{\ell'}.$$

We do not learn γ_j/γ_{j^*} and $(\beta_j - \beta_{j^*})/\gamma_{j^*}$ *directly* unless an edge exists between j and j^* (otherwise, the payoffs they yield do not overlap, so we cannot directly infer any relative information). Since j and j^* are in the same component, a path between them always exists and can be found efficiently via standard path-finding algorithms. Moreover, the requirement $\mathbf{a}_{j_\ell} \neq \mathbf{0}$ in Step 1 can always be satisfied: First, we can always find some j^* with $\mathbf{a}_{j^*} \neq \mathbf{0}$ to start with—otherwise, it must be that C contains only one action. Second, if $\mathbf{a}_{j_\ell} = \mathbf{0}$ for some $\ell > 1$, we can eliminate this node because that implies an edge exists between $j_{\ell-1}$ and $j_{\ell+1}$. While $\mathbf{a}_j$ (i.e., $\mathbf{a}_{j_{k+1}}$) may indeed be $\mathbf{0}$, in that case $\tilde{u}^L(\mathbf{x}, j) = (\beta_j - \beta_{j^*})/\gamma_{j^*}$, so we only need to learn $(\beta_j - \beta_{j^*})/\gamma_{j^*}$.

Our task then reduces to Step 2: learning c_ℓ and b_ℓ. We present how this can be done next.

7.2 Learning c_ℓ and b_ℓ: Preliminaries

For convenience, we assume $j_\ell = 1$ and $j_{\ell+1} = 2$, and show how to learn γ_2/γ_1 and $(\beta_2 - \beta_1)/\gamma_1$. The assumption implies the following:

- $1, 2 \notin \Lambda'$ (so $M_{\{1\}} > M_{[n]}$ and $M_{\{2\}} > M_{[n]}$).
- There is an edge between 1 and 2 in the non-dominance graph.
- $M_{\{1\}} \leq M_{\{2\}}$, and action 1 does not *strongly dominate* (Definition 9) action 2 w.r.t. u^L.

Specifically, the third condition holds w.l.o.g.: if $M_{\{1\}} > M_{\{2\}}$, we can swap the two actions and learn γ_1/γ_2 and $(\beta_1-\beta_2)/\gamma_2$, from which γ_2/γ_1 and $(\beta_2-\beta_1)/\gamma_1$ can be derived trivially. A similar approach applies if action 1 strongly dominates action 2. Note that the strong dominance is a weaker notion than the obvious dominance we used to define the non-dominance graph.

Definition 9 (Strong dominance)**.** An action $j \in [n]$ *strongly dominates* an action $k \in [n]$ w.r.t. u^L if : 1) $u^L(i,j) \geq u^L(i,k)$ for all $i \in [m]$; and 2) there exists $i' \in [m]$ such that $u^L(i',j) > u^L(i',k)$.[7]

7.2.1 Equilibrium Response Oracle

Additionally, when $\mathbf{a}_j$ is known, we can also assume access to the following oracle $\mathcal{A}_{\text{ER}}$ (Definition 10), which determines if a given action j can be induced as an *equilibrium response* (ER) by a given follower type $\tilde{u}^F$, i.e., whether it is in the following ER set:

$$\text{ER}(u^L, \tilde{u}^F) \coloneqq \{j \in [n] : \exists \mathbf{x}, \text{s.t.}, (\mathbf{x}, j) \in \text{SSE}(u^L, \tilde{u}^F)\}.$$

To determine this, it suffices to check whether $(\mathbf{x}, j)$ is an SSE (by using $\mathcal{A}_{\text{SSE}}$) for an arbitrary $\mathbf{x}$ in the set $\operatorname{argmax}_{\mathbf{x}' \in \widetilde{\text{BR}}^{-1}(j)} u^L(\mathbf{x}', j) \equiv \operatorname{argmax}_{\mathbf{x}' \in \widetilde{\text{BR}}^{-1}(j)} \mathbf{a}_j \cdot \mathbf{x}'$, where $\widetilde{\text{BR}}$ is the BR-correspondence defined by $\tilde{u}^F$ in the input to $\mathcal{A}_{\text{ER}}$. Such an $\mathbf{x}$ can be computed efficiently when $\mathbf{a}_j$ is known.

Definition 10 (Equilibrium response oracle)**.** Given a follower type $\tilde{u}^F$ and an action $j \in [n]$, the *equilibrium response oracle*, $\mathcal{A}_{\text{ER}}$, outputs whether or not $j \in \text{ER}(u^L, \tilde{u}^F)$. Moreover, when $j \in \text{ER}(u^L, \tilde{u}^F)$, the oracle also outputs a strategy $\mathbf{x} \in \Delta_m$ such that $(\mathbf{x}, j) \in \text{SSE}(u^L, \tilde{u}^F)$.

7.2.2 Reference Pairs

To learn γ_2/γ_1 and $(\beta_2-\beta_1)/\gamma_1$, the key is to find two different *reference pairs* $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}', \mathbf{y}')$ such that $u^L(\mathbf{x}, 1) = u^L(\mathbf{y}, 2)$ and $u^L(\mathbf{x}', 1) = u^L(\mathbf{y}', 2)$. With these pairs, we can establish the following system of linear equations to compute γ_2/γ_1 and $(\beta_2 - \beta_1)/\gamma_1$:

$$\begin{cases} \gamma_1 \cdot \mathbf{a}_1 \cdot \mathbf{x} \ + \beta_1 = \gamma_2 \cdot \mathbf{a}_2 \cdot \mathbf{y} \ + \beta_2 \\ \gamma_1 \cdot \mathbf{a}_1 \cdot \mathbf{x}' + \beta_1 = \gamma_2 \cdot \mathbf{a}_2 \cdot \mathbf{y}' + \beta_2, \end{cases} \tag{17}$$

which gives

$$\gamma_2/\gamma_1 = \frac{\mathbf{a}_1 \cdot \mathbf{x} - \mathbf{a}_1 \cdot \mathbf{x}'}{\mathbf{a}_2 \cdot \mathbf{y} - \mathbf{a}_2 \cdot \mathbf{y}'}, \quad \text{and} \quad (\beta_2 - \beta_1)/\gamma_1 = \mathbf{a}_1 \cdot \mathbf{x} - \frac{\mathbf{a}_1 \cdot \mathbf{x} - \mathbf{a}_1 \cdot \mathbf{x}'}{\mathbf{a}_2 \cdot \mathbf{y} - \mathbf{a}_2 \cdot \mathbf{y}'} \cdot \mathbf{a}_2 \cdot \mathbf{y}.$$

Specifically, we look for reference pairs such that $\mathbf{a}_1 \cdot \mathbf{x} \neq \mathbf{a}_1 \cdot \mathbf{x}'$ (i.e., $u^L(\mathbf{x}, 1) \neq u^L(\mathbf{x}', 1)$) so that the equation system is of full rank. We demonstrate how these pairs can be obtained next.

7.3 Obtaining the First Reference Pair

For the first reference pair, we aim to find $\mathbf{x}$ and $\mathbf{y}$ such that

$$u^L(\mathbf{x}, 1) = u^L(\mathbf{y}, 2) = M_{\{1,2\}}. \tag{$*$}$$

The value $M_{\{1,2\}}$ will be useful for ensuring linear independence between this and the second reference pair. We use the following payoff function $\tilde{u}^F_d$, parameterized by $d \in \mathbb{R}$, as a key tool.

[7] An equivalent statement is: $u^L(\mathbf{x}, j) \geq u^L(\mathbf{x}, k)$ for all $\mathbf{x} \in \Delta$ and there exists $\mathbf{x}' \in \Delta$ such that $u^L(\mathbf{x}', j) > u^L(\mathbf{x}', k)$.

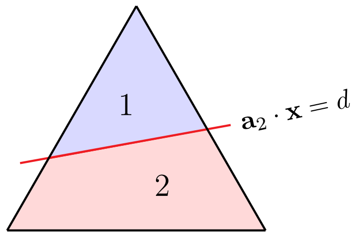

Figure 4: The BR-correspondence of $\tilde{u}_d^F$. The triangle represents the strategy space Δ_m of the leader. The regions labeled 1 and 2 are $f_d^{-1}(1)$ and $f_d^{-1}(2)$, respectively.

$$\tilde{u}_d^F(\mathbf{x}, j) := \begin{cases} -1, & \text{if } j \in [n] \setminus \{1,2\}; \\ 0, & \text{if } j = 1; \\ d - \mathbf{a}_2 \cdot \mathbf{x}, & \text{if } j = 2. \end{cases} \tag{18}$$

Let f_d be the BR-correspondence corresponding to $\tilde{u}_d^F$, and let $\mathcal{G}_d := (u^L, \tilde{u}_d^F)$ be the game $\tilde{u}_d^F$ induces. As illustrated in Fig. 4, the definition in Eq. (18) gives

$$\begin{aligned} & f_d^{-1}(1) = \{\mathbf{x} \in \Delta_m : \mathbf{a}_2 \cdot \mathbf{x} \geq d\}, \\ \text{and} \quad & f_d^{-1}(2) = \{\mathbf{x} \in \Delta_m : \mathbf{a}_2 \cdot \mathbf{x} \leq d\}. \end{aligned}$$

Based on the structure of f_d^{-1}, we search for a value of d that makes both actions 1 and 2 equilibrium responses, i.e., $\mathrm{ER}(\mathcal{G}_d) = \{1,2\}$. Intuitively, a sufficiently small d results in $f_d^{-1}(1)$ being the whole simplex Δ_m while $f_d^{-1}(2)$ being empty, whereby $\mathrm{ER}(\mathcal{G}_d) = \{1\}$. Conversely, a sufficiently large d results in $\mathrm{ER}(\mathcal{G}_d) = \{2\}$. Hence, the hope is that there exists a middle point d^* where $\mathrm{ER}(\mathcal{G}_{d^*}) = \{1,2\}$. As we demonstrate in Lemma 7.1, this is indeed the case with

$$d^* := (M_{\{1,2\}} - \beta_2)/\gamma_2. \tag{19}$$

Since the values of $M_{\{1,2\}}$, β_2, and γ_2 are unknown, d^* cannot be computed directly via the expression. We use binary search to pin down its value: by Lemma 7.1, $\mathcal{G}_d$ has different SSE responses when $d \leq d^*$ and $d \geq d^*$, so the two cases can be identified using $\mathcal{A}_{\mathrm{ER}}$. Let $V_d^L(j) := \max_{\mathbf{x} \in f_d^{-1}(j)} u^L(\mathbf{x}, j)$ be the leader's maximum attainable payoff for inducing best response j.

Lemma 7.1. $V_{d^*}^L(1) = V_{d^*}^L(2) = M_{\{1,2\}}$. *Moreover,* $1 \in \mathrm{ER}(\mathcal{G}_d)$ *if and only if* $d \leq d^*$, *and* $2 \in \mathrm{ER}(\mathcal{G}_d)$ *if and only if* $d \geq d^*$.

Since $\mathrm{ER}(\mathcal{G}_{d^*}) = \{1,2\}$, any $\mathbf{x}, \mathbf{y}$ such that $(\mathbf{x}, 1), (\mathbf{y}, 2) \in \mathrm{SSE}(\mathcal{G}_{d^*})$ ensure $u^L(\mathbf{x}, 1) = u^L(\mathbf{y}, 2)$. Lemma 7.1 further ensures $u^L(\mathbf{x}, 1) = u^L(\mathbf{y}, 2) = M_{\{1,2\}}$. Such $\mathbf{x}$ and $\mathbf{y}$ can be obtained directly from the ER oracle (see Definition 10). In summary, we present Proposition 7.2.

Proposition 7.2. $\mathbf{x}, \mathbf{y} \in \Delta_m$ *satisfying Eq. (∗) can be computed in polynomial time.*

7.4 Obtaining the Second Reference Pair

Now we search for the second reference pair. As mentioned previously, in the special case where $\mathbf{a}_1$ or $\mathbf{a}_2$ is $\mathbf{0}$, we only need to learn $(\beta_2 - \beta_1)/\gamma_1$ or $(\beta_1 - \beta_2)/\gamma_2$, so one reference pair suffices and there is no need for a second pair. Therefore, w.l.o.g., we assume $\mathbf{a}_1 \neq \mathbf{0}$ in what follows. We aim to find a pair $(\mathbf{x}, \mathbf{y})$ such that

$$u^L(\mathbf{x}, 1) = u^L(\mathbf{y}, 2) < M_{\{1,2\}}. \tag{$**$}$$

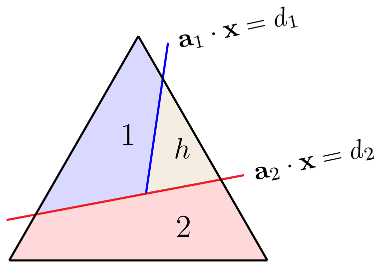

Figure 5: f_{d_1,d_2}: the BR-correspondence used for finding the second reference pair. The triangle represents the strategy space Δ_m of the leader. The regions labeled 1 and 2 are $f^{-1}_{d_1,d_2}(1) = P_1$ and $f^{-1}_{d_1,d_2}(2) = P_2$, respectively.

Given that the payoff of the first pair is exactly $M_{\{1,2\}}$, a second pair satisfying Eq. $(**)$ ensures that the equations in Eq. (17) yield a unique solution.

We aim to construct a BR-correspondence to induce two SSEs $(\mathbf{x}, 1)$ and $(\mathbf{y}, 2)$. As illustrated in Fig. 5, the idea is to pull back the boundaries of the best-response regions $f^{-1}_{d^*}(1)$ and $f^{-1}_{d^*}(2)$ (where we set a new boundary $\mathbf{a}_1 \cdot \mathbf{x} = d_1$). If we move the boundaries at the right pace, we can keep the leader's maximum attainable payoffs in the two regions equal and simultaneously strictly smaller than $M_{\{1,2\}}$. This will then yield a pair that satisfies Eq. $(**)$. Note that a blank region will appear when the boundaries move. Similar to our approach in Section 6, this region will need to be filled by a cover to keep the leader's attainable payoff there lower than the other two regions.

To better illustrate the approach, we temporarily allow the oracle $\mathcal{A}_{\mathrm{SSE}}$ to take as part of the input a BR-correspondence $\widetilde{\mathrm{BR}}$, instead of a follower type $\tilde{u}^F$. The oracle answers if the queried strategy profile is an SSE in the game $\widetilde{\mathcal{G}} = (u^L, \widetilde{\mathrm{BR}})$, as defined in Eq. (1). We present an algorithm that uses BR-correspondences in queries to $\mathcal{A}_{\mathrm{SSE}}$. The BR-correspondences may not be realizable with any payoff matrices, so we also design a stronger algorithm that only uses payoff matrices in the queries. This stronger algorithm is much more involved but can be better understood based on intuition conveyed by the weaker one. We defer the details of the stronger algorithm to Appendix D.3.

7.4.1 Querying by Using BR-correspondences

Define the following partition $P = (P_0, P_1, P_2)$ of Δ_m, which is parameterized by two numbers d_1 and d_2:

$$\begin{aligned}
P_0(d_1, d_2) &\coloneqq \{\mathbf{x} \in \Delta_m : \mathbf{a}_2 \cdot \mathbf{x} \ge d_2 \text{ and } \mathbf{a}_1 \cdot \mathbf{x} \ge d_1\},\\
P_1(d_1, d_2) &\coloneqq \{\mathbf{x} \in \Delta_m : \mathbf{a}_2 \cdot \mathbf{x} \ge d_2 \text{ and } \mathbf{a}_1 \cdot \mathbf{x} \le d_1\},\\
P_2(d_1, d_2) &\coloneqq \{\mathbf{x} \in \Delta_m : \mathbf{a}_2 \cdot \mathbf{x} \le d_2\}.
\end{aligned}$$

Based on the partition, define the BR-correspondence $f_{d_1,d_2} : \Delta_m \to 2^{[n]}$ such that for every $\mathbf{x} \in \Delta_m$:

$$f_{d_1,d_2}(\mathbf{x}) \supseteq \begin{cases} \{1\}, & \text{if} \quad \mathbf{x} \in P_1(d_1, d_2);\\ \{2\}, & \text{if} \quad \mathbf{x} \in P_2(d_1, d_2);\\ h(\mathbf{x}), & \text{if} \quad \mathbf{x} \in P_0(d_1, d_2); \end{cases} \tag{20}$$

where $h : \Delta_m \to 2^{[n]\setminus\{1,2\}}$ is the BR-correspondence of an arbitrary (proper) cover of $\{1, 2\}$. We will shortly discuss how to obtain h. According to the above construction, we have

- $f^{-1}_{d_1,d_2}(j) = P_j(d_1, d_2)$ for all $j \in \{1, 2\}$; and
- $f^{-1}_{d_1,d_2}(k) \subseteq P_0(d_1, d_2)$ for all $k \in [n] \setminus \{1, 2\}$.

Moreover, in comparison to the BR-correspondence f_d defined in Eq. (18), we have $P_2(d_1, d_2) = f_{d_2}^{-1}(2)$, and $P_1(d_1, d_2) \cup P_0(d_1, d_2) = f_{d_2}^{-1}(1)$. Fig. 5 presents an illustration of f_{d_1,d_2}. Our goal is to find two numbers d_1 and d_2 such that $\{1, 2\} \subseteq \mathrm{ER}(\mathcal{G}_{d_1,d_2})$ for $\mathcal{G}_{d_1,d_2} := (u^L, f_{d_1,d_2})$, so that the equilibrium strategies that form SSEs with actions 1 and 2 can be used as the second reference pair.

We define several useful notions. For each $j \in \{1, 2\}$, let $V^L_{d_1,d_2}(j) := \max_{\mathbf{x}\in f^{-1}_{d_1,d_2}(j)} u^L(\mathbf{x}, j)$ be the leader's maximum attainable payoff in P_j. Let

$$d_j^* = (M_{\{1,2\}} - \beta_j)/\gamma_j. \tag{21}$$

So, $d_2^* = d^*$ as in Eq. (19). We can prove that P coincides with f_{d^*} for $d_1 = d_1^*$ and $d_2 = d_2^*$. Moreover:

- $P_0(d_1^*, d_2^*) = \mathcal{M}_{\{1,2\}}$ and $P_j(d_1^*, d_2^*) = f_{d^*}^{-1}(j) \neq \emptyset$ for $j \in \{1, 2\}$; and
- for all $\odot \in \{>, <, =\}$ and $j \in \{1, 2\}$, $u^L(\mathbf{x}, j) \odot M_{\{1,2\}} \iff \mathbf{a}_j \cdot \mathbf{x} \odot d_j^*$.

We search for $d_1 < d_1^*$ and $d_2 < d_2^*$ to ensure that the payoffs of the second reference pair is strictly smaller than $M_{\{1,2\}}$. Before presenting the details, we first show how to learn h.

7.4.2 Computing h and Identifying Λ''

Now that $\mathbf{a}_j$ is known for all $j \in [n]\setminus\Lambda'$, to apply Proposition 6.8 to find a cover, we only need to provide a base function for $\{1, 2\}$ and the set $\mathcal{M}_{\{1,2\}}$. It can be verified that the function $\tilde{u}^F_{d^*}$ in Eq. (18) and the region $P_0(d_1^*, d_2^*)$, with d_1^* and d_2^* in Eq. (21), fulfill these demands. In the special case where $\{1, 2\}$ does not admit a cover, we conclude that $\{1, 2\} \subseteq \Lambda''$ according to Lemma 6.3, whereby the first reference pair $(\mathbf{x}, \mathbf{y})$ also gives $u^L(\mathbf{x}, 1) = u^L(\mathbf{y}, 2) = M_{\{1,2\}} = M_{[n]}$. Hence, by repeating this process for all edges in the non-dominance graph, we can identify Λ'', as well as a strategy $\mathbf{x}_j$ such that $u^L(\mathbf{x}_j, j) = M_{[n]}$ for every $j \in \Lambda''$. The latter is necessary for the construction in Eq. (9).

Proposition 7.3. *Λ'' can be computed in polynomial time. Moreover, for every $j \in \Lambda''$, a strategy $\mathbf{x}_j$ such that $u^L(\mathbf{x}_j, j) = M_{[n]}$ can be computed in polynomial time.*

7.4.3 Pinning Down d_1 and d_2

We use a nested binary search to find d_1 and d_2 such that $\{1, 2\} \subseteq \mathrm{ER}(\mathcal{G}_{d_1,d_2})$. The following lemma is key to this approach.

Lemma 7.4. *There exists $d_2' < d_2^*$ such that for all $d_2 \in [d_2', d_2^*]$, $\{1, 2\} \subseteq \mathrm{ER}(\mathcal{G}_{d_1,d_2})$ for some $d_1 < d_1^*$.*

Hence, the nested binary search works as follows. It searches for a value of d_2. Each time it checks a candidate value of d_2, it calls another binary search for a d_1 such that $\{1, 2\} \in \mathrm{ER}(G_{d_1,d_2})$. If no such d_1 exists, it moves d_2 to the middle point between d_2 and d_2^* and repeats, until a satisfying d_2 is found. Lemma 7.4 indicates that this will always work when d_2 is sufficiently close to d_2^*.

We then establish a weaker version of Proposition 7.5 (see Proposition D.4), assuming that $\mathcal{A}_{\mathrm{SSE}}$ can handle BR-correspondence queries. The proof that drops this assumption follows a similar approach but requires explicitly constructing payoff matrices equivalent to f_{d_1,d_2}.

Proposition 7.5. *$\mathbf{x}, \mathbf{y} \in \Delta_m$ satisfying Eq. $(**)$ can be computed in polynomial time.*

8 Learning under Other Oracles

Beyond the SSE oracle used so far, there can be other types of oracles. For example, some oracles do not allow querying specific strategy profiles. When multiple SSEs exists, the oracles output a random SSE. These alternative oracles model scenarios where the follower cannot resolve the equilibrium selection problem according to their own desire. Here, we introduce two such alternative oracles.

Definition 11 (Oracle $\mathcal{A}_{\mathrm{W1}}$)**.** Given a payoff matrix $\tilde{u}^F$, the oracle $\mathcal{A}_{\mathrm{W1}}$ outputs a basic SSE of the game $(u^L, \tilde{u}^F)$ uniformly at random. An SSE $(\mathbf{x}, j)$ is a basic SSE of a game $(u^L, \tilde{u}^F)$ if $\mathbf{x}$ is an extreme point of $\widetilde{\mathrm{BR}}^{-1}(j)$, where $\widetilde{\mathrm{BR}}^{-1}$ is the BR-correspondence induced by $\tilde{u}^F$.

Definition 12 (Oracle $\mathcal{A}_{\mathrm{W2}}$)**.** Given a payoff matrix $\tilde{u}^F$, the oracle $\mathcal{A}_{\mathrm{W2}}$ outputs each $s \in \mathrm{SSE}(u^L, \tilde{u}^F)$ uniformly at random.

The main difficulty of utilizing the alternative oracles arises when determining whether two strategy profiles are both SSEs induced by the same payoff function. This ability to check the equivalence of strategy profiles is crucial for, e.g., finding the whole set of $\mathcal{M}_{\{j\}}$. Nevertheless, we show that we can still learn an equivalent payoff matrix *with high probability* under the two oracles. As for the second alternative oracle, we introduce a condition on the leader's payoff matrix, called *non-max-degenerate*. That is, there does *not* exist $j \in [n]$ such that $|\operatorname{argmax}_{i\in[m]} u^L(i,j)| > 1$. The same condition is also assumed in Birmpas et al. [2021] to overcome an equilibrium selection issue.

Theorem 8.1. *Given oracle $\mathcal{A}_{\mathrm{W1}}$, in time polynomial in the bit size of u^L and $\delta > 0$, we can learn a matrix $\tilde{u}^L$ that achieves the same equivalences in Theorem 5.4 with probability at least $1-\delta$. The same holds for $\mathcal{A}_{\mathrm{W2}}$ if u^L is non-max-degenerate.*

9 Conclusion

We developed computationally and sample-efficient algorithms to learn the leader's incentives from active queries on their optimal commitments. Our approach learns a payoff function that is strategically equivalent to the true one, preserving both (1) the leader's preference order across strategy profiles and (2) their commitment behaviors in Stackelberg scenarios. The learned information allows a strategic follower to induce an equilibrium that maximizes their true utility.

For future work, we can explore alternative oracles, such as one that outputs a deterministic equilibrium based on a specific tie-breaking rule. Investigating other learning settings could also be interesting. For example, one can assume that the leader adopts a policy that maps each follower type to a strategy to play. To infer the leader's policy, however, it appears that additional structural assumptions on the policy are necessary, as the learning task may easily become intractable, otherwise.

Another interesting direction is to study counteractions when the leader notices one's learning attempts. Beyond the mechanism design approach [Gan et al., 2019b], Nash equilibrium may be a suitable concept when both players strategize. For example, consider such a payoff misreporting game: each player reports a payoff matrix, and gets their true payoff for an SSE of the induced game. Then, if we pick a follower type $\tilde{u}^F$ that induces the optimal Stackelberg equilibrium w.r.t. the leader's true payoff u^L, i.e., $\tilde{u}^F \in \operatorname{argmax}_{\tilde{u}^F, s\in\mathrm{SSE}(u^L,\tilde{u}^F)} u^F(s)$, (also see Corollary 5.5), then the strategy profile $(u^L, \tilde{u}^F)$ forms a Nash equilibrium in the game (as noted similarly by Chen et al. [2025]).

A Omitted Proofs in Section 5

A.1 Proof of Proposition 5.1

Proposition 5.1. *Given a matrix $\tilde{u}^L$ that is equivalent to u^L in $\Delta_m \times C$ for every non-dominance component C, a matrix equivalent to u^L in $\Delta_m \times ([n] \setminus \Lambda)$ can be computed in time polynomial in the total bit size of $\tilde{u}^L$ and u^L.*

We first prove the following lemma.

Lemma A.1. *Suppose that $\tilde{u}^L$ is equivalent to u^L in $\Delta_m \times \bigcup_{\ell=1}^{k} C_\ell$ and $\Delta_m \times C_{k+1}$. Then the following matrix $\tilde{u}'$ is equivalent to u^L in $\Delta_m \times \bigcup_{\ell=1}^{k+1} C_\ell$: for every $i \in [m]$ and $j \in [n]$,*

$$\tilde{u}'(i,j) = \begin{cases} \tilde{u}^L(i,j) & \text{if } j \in \bigcup_{\ell=1}^{k} C_\ell \\ \tilde{u}^L(i,j) - \check{U}_{k+1} + \hat{U}_k + \sigma_k & \text{if } j \in C_{k+1} \end{cases} \tag{22}$$

where $\hat{U}_k = \max_{i\in[m], j\in C_k} \tilde{u}^L(i,j)$ and $\check{U}_{k+1} = \min_{i\in[m], j\in C_{k+1}} \tilde{u}^L(i,j)$.

Proof. Suppose that Eq. (22) holds for all i, j. We show that $\tilde{u}'$ is equivalent to u^L in $\Delta_m \times \bigcup_{\ell=1}^{k+1} C_\ell$.

Since $\tilde{u}'$ is the same as $\tilde{u}^L$ for columns in $\bigcup_{\ell=1}^{k} C_\ell$, and it is $\tilde{u}^L$ with every entry in columns in C_{k+1} raised by a constant value, it preserves equivalences in $\Delta_m \times \bigcup_{\ell=1}^{k} C_\ell$ and $\Delta_m \times C_{k+1}$. It remains to verify the equivalence across these two sets of columns, i.e.,

$$\begin{aligned} & u^L(s) > u^L(s') \iff \tilde{u}'(s) > \tilde{u}'(s') \\ \text{and} \quad & u^L(s) < u^L(s') \iff \tilde{u}'(s) < \tilde{u}'(s') \end{aligned}$$

for any two arbitrary strategy profiles $s \in \Delta_m \times \bigcup_{\ell=1}^{k} C_\ell$ and $s' \in \Delta_m \times C_{k+1}$. Indeed, according to Eq. (8), it must be that $u^L(s) \le u^L(s')$ (note that $\max_{i\in[m], j\in\bigcup_{\ell=1}^{k} C_\ell} u^L(i,j) = \max_{i\in[m], j\in C_k} u^L(i,j)$). So, it suffices to argue that $u^L(s) = u^L(s') \iff \tilde{u}'(s) = \tilde{u}'(s')$.

By Eq. (8), $u^L(s) = u^L(s')$ implies

$$u^L(s) = \max_{i\in[m], j\in C_k} u^L(i,j) = \min_{i\in[m], j\in C_{k+1}} u^L(i,j) = u^L(s').$$

So $\sigma_k = 0$. Given the equivalence between $\tilde{u}^L$ and u^L in $\Delta_m \times C_k$ and $\Delta_m \times C_{k+1}$, the above equation further implies that $\tilde{u}^L(s) = \hat{U}_k$ and $\tilde{u}^L(s') = \check{U}_{k+1}$. According to the definition of $\tilde{u}'$, it then follows that $\tilde{u}'(s') = \tilde{u}^L(s') - \check{U}_{k+1} + \hat{U}_k = \tilde{u}'(s)$. Hence, $u^L(s) = u^L(s') \implies \tilde{u}'(s) = \tilde{u}'(s')$. The fact that $u^L(s) = u^L(s') \impliedby \tilde{u}'(s) = \tilde{u}'(s')$ can be verified analogously. $\square$

Proof of Proposition 5.1. Given Lemma A.1, an induction on $k = 1, \dots, K-1$ yields a matrix equivalent to u^L in $\Delta_m \times \bigcup_{\ell=1}^{K} C_\ell \equiv \Delta_m \times ([n] \setminus \Lambda)$. Specifically, for the base case where $k = 1$, the required equivalences in $\Delta_m \times C_1$ and $\Delta_m \times C_2$ are already achieved in Section 5.1. $\square$

A.2 Proof of Lemma 5.2

Lemma 5.2. *Suppose that $\tilde{u}^L$ is equivalent to u^L in $\Delta_m \times ([n] \setminus \Lambda)$. Then $\tilde{u}'_{v^*}$ is inducibility-equivalent to u^L. Moreover, for all $s \in \Delta_m \times ([n] \setminus \Lambda)$ such that $\tilde{u}^L(s) = v$:*

(i) if $v \ge v^*$, then $s \in \mathrm{SSE}(\tilde{u}'_v)$ and, for all $\tilde{u}^F \in \mathbb{R}^{m\times n}$, $s \in \mathrm{SSE}(\tilde{u}'_v, \tilde{u}^F) \implies s \in \mathrm{SSE}(u^L, \tilde{u}^F)$;

(ii) if $v < v^*$, then $s \notin \mathrm{SSE}(u^L)$.

We first prove the following useful result.

Lemma A.2. *For every* $(\mathbf{x}, j) \in \Delta_m \times [n]$, $v \in \mathbb{R}$, *and* $\odot \in \{>, =, <\}$, *the following statements hold:*

(i) $\tilde{u}'_{v^*}(\mathbf{x}, j) \odot v^* \iff u^L(\mathbf{x}, j) \odot M_{[n]}$*; and*

(ii) if $j \in \Lambda$, *then* $\tilde{u}'_v(\mathbf{x}, j) \odot v \iff u^L(\mathbf{x}, j) \odot M_{[n]}$.

(iii) if $\Lambda \neq \emptyset$, *then* $\min_{k\in[n]} \tilde{u}'_v(\mathbf{x}, k) \leq v$.

Proof. We first prove (ii). Suppose that $j \in \Lambda$. Note that the construction in Eq. (9) ensures

$$u^L(\mathbf{x}, j) \odot u^L(\mathbf{x}^*_j, j) \iff \tilde{u}'_v(\mathbf{x}, j) \odot \tilde{u}'_v(\mathbf{x}^*_j, j).$$

Indeed, if $j \in \Lambda''$, then $\tilde{u}'_v(\mathbf{x}, j)$ is an affine transformation of $u^L(\mathbf{x}, j)$ according to Eq. (9), so the above holds readily; whereas when $j \in \Lambda'$, the definition of $\tilde{\mathbf{a}}_j$ gives $\mathbf{x}^*_j \in \operatorname{argmax}_{\mathbf{x}\in\Delta_m} u^L(\mathbf{x}, j) = \operatorname{argmax}_{\mathbf{x}\in\Delta_m} \tilde{u}'_v(\mathbf{x}, j)$, so the above holds, too. Now that $u^L(\mathbf{x}^*_j, j) = M_{[n]}$, it follows that

$$u^L(\mathbf{x}, j) \odot M_{[n]} \iff u^L(\mathbf{x}, j) \odot u^L(\mathbf{x}^*_j, j) \iff \tilde{u}'_v(\mathbf{x}, j) \odot \tilde{u}'_v(\mathbf{x}^*_j, j).$$

By further noting that $\tilde{u}'_v(\mathbf{x}^*_\ell, \ell) = v$ for all $\ell \in \Lambda$ according to Eq. (9), we get the desired result.

Now consider (i). Given (ii), it suffices to consider the case where $j \notin \Lambda$.

- If $v^* = \phi^{-1}(M_{[n]})$, then the monotonicity of ϕ implies (i) by noting that

$$\phi\left(\tilde{u}'_{v^*}(\mathbf{x}, j)\right) = \phi\left(\tilde{u}^L(\mathbf{x}, j)\right) = u^L(\mathbf{x}, j)$$

  and $\phi(v^*) = M_{[n]}$, where $\tilde{u}'_{v^*}(\mathbf{x}, j) = \tilde{u}^L(\mathbf{x}, j)$ because $j \notin \Lambda$.

- If $\phi(v^*) \neq M_{[n]}$, according to the definition of v^*, we have $u^L(s) \neq M_{[n]}$ for all $s \in \Delta_m \times ([n] \setminus \Lambda)$. It must be that $u^L(s) > M_{[n]}$ because by definition $u^L(s) < M_{[n]}$ is not possible. Consequently, we have both $u^L(\mathbf{x}, j) \geq M_{[n]}$ and

$$\tilde{u}'_{v^*}(\mathbf{x}, j) \geq \min_{s\in\Delta_m\times([n]\setminus\Lambda)} \tilde{u}'_{v^*}(s) = v^*.$$

  So (i) follows.

Finally, consider (iii). If $\Lambda \neq \emptyset$, then according to the definitions of Λ' and Λ'', there must be an action $k \in \Lambda$ such that $u^L(\mathbf{x}, k) \leq M_{[n]}$. Applying (ii) gives $\tilde{u}'_v(\mathbf{x}, k) \leq v$, so (iii) follows. ☐

Proof of Lemma 5.2. If $\Lambda = \emptyset$, then $\tilde{u}'_v = \tilde{u}^L$ for any v, and $\tilde{u}^L$ is an affine transformation of u^L, in which case all the results stated in the lemma follow immediately. Hence, in what follows, we assume $\Lambda \neq \emptyset$.

Inducibility Equivalence We first prove the inducibility-equivalence between $\tilde{u}'_{v^*}$ and u^L. By Theorem 2.1, it suffices to prove that, for every $s \in \Delta_m \times [n]$,

$$u^L(s) \geq M_{[n]} \iff \tilde{u}'_{v^*}(s) \geq \widetilde{M}^*_{[n]}$$

where $\widetilde{M}^*_{[n]} := \max_{\mathbf{x}\in\Delta_m} \min_{j\in[n]} \tilde{u}'_{v^*}(\mathbf{x}, j)$ is the maximin value of $\tilde{u}'_{v^*}$. By Lemma A.2 (i), this holds if $\widetilde{M}^*_{[n]} = v^*$, or equivalently:

- $\min_{j\in[n]} \tilde{u}'_{v^*}(\mathbf{x}, j) \leq v^*$ for all $\mathbf{x} \in \Delta_m$, and
- $\min_{j\in[n]} \tilde{u}'_{v^*}(\mathbf{x}, j) \geq v^*$ for *some* $\mathbf{x} \in \Delta_m$.

For the first part follows by Lemma A.2 (iii). The second part is equivalent to: $\tilde{u}'_{v^*}(s) \geq v^*$ for some $s \in \Delta_m \times [n]$. Indeed, v^* is defined in a way such that $\tilde{u}'_{v^*}(s) = v^*$ always holds for some s, so this follows immediately.

Statement (i) Next, we prove (i). Suppose that $v \geq v^*$ and $\tilde{u}^L(s) = v$. To see that $s \in \mathrm{SSE}(\tilde{u}'_v)$, we invoke Theorem 2.1 and show that $\tilde{u}'_v(s) \geq \max_{\mathbf{x}\in\Delta_m} \min_{j\in[n]} \tilde{u}'_v(\mathbf{x}, j)$. Indeed, since $\tilde{u}'_v(s) = v$, this inequality follows immediately by Lemma A.2 (iii).

Now suppose that $s \in \mathrm{SSE}(\tilde{u}'_v, \tilde{u}^F)$. By definition, this means that

$$\tilde{u}'_v(s) \geq \max\nolimits_{j\in\widetilde{\mathrm{BR}}(\mathbf{x})} \tilde{u}'_v(\mathbf{x}, j) \quad \text{for all } \mathbf{x} \in \Delta_m,$$

where $\widetilde{\mathrm{BR}}$ denotes the BR-correspondence of $\tilde{u}^F$. We show that this implies

$$u^L(s) \geq \max_{j\in\widetilde{\mathrm{BR}}(\mathbf{x})} u^L(\mathbf{x}, j) \quad \text{for all } \mathbf{x} \in \Delta_m.$$

It suffices to show that for every $(\mathbf{x}, j) \in \Delta_m \times [n]$:

$$\tilde{u}'_v(s) \geq \tilde{u}'_v(\mathbf{x}, j) \implies u^L(s) \geq u^L(\mathbf{x}, j).$$

Consider the following two cases.

- If $j \notin \Lambda$, then

$$\begin{aligned}\tilde{u}'_v(s) \geq \tilde{u}'_v(\mathbf{x}, j) &\implies \phi\left(\tilde{u}'_v(s)\right) \geq \phi\left(\tilde{u}'_v(\mathbf{x}, j)\right) \\ &\implies \phi\left(\tilde{u}^L(s)\right) \geq \phi\left(\tilde{u}^L(\mathbf{x}, j)\right) \implies u^L(s) \geq u^L(\mathbf{x}, j),\end{aligned}$$

  which relies on the fact that $\tilde{u}'_v = \tilde{u}^L$ for both s and $(\mathbf{x}, j)$, as well as the monotonicity of ϕ.

- If $j \in \Lambda$, then Lemma A.2 (ii) means that:

$$v \geq \tilde{u}'_v(\mathbf{x}, j) \implies M_{[n]} \geq u^L(\mathbf{x}, j).$$

  By further noting that

$$\tilde{u}'_v(s) = \tilde{u}^L(s) = v$$

  and

$$u^L(s) = \phi(v) \geq \phi(v^*) \geq M_{[n]},$$

  the desired result then follows. Specifically, $\phi(v) \geq \phi(v^*)$ because $v \geq v^*$; and $\phi(v^*) \geq M_{[n]}$ because by definition:

  - either $v^* = \phi^{-1}(M_{[n]})$, which implies $\phi(v^*) = M_{[n]}$;
  - or $u^L(s) \neq M_{[n]}$ for all $s \in \Delta_m \times ([n] \setminus \Lambda)$, in which case it must be that $u^L(s) > M_{[n]}$ according to the definition of $M_{[n]}$, so $\phi(v^*) > M_{[n]}$.

Therefore, $u^L(s) \geq \max_{j\in\widetilde{\mathrm{BR}}(\mathbf{x})} u^L(\mathbf{x}, j)$ for all $\mathbf{x} \in \Delta_m$. By definition, $s \in \mathrm{SSE}(u^L, \tilde{u}^F)$.

Statement (ii) Finally, consider (ii). Suppose that $v < v^*$. So, $\tilde{u}^L(s) = v < v^*$. This means

$$v^* > \tilde{u}^L(s) \geq \min_{s'\in\Delta_m\times([n]\setminus\Lambda)} \tilde{u}^L(s'),$$

so according to the definition of v^* it must be that $v^* = \phi^{-1}(M_{[n]})$. Consequently, $v < \phi^{-1}(M_{[n]})$. Since ϕ is strictly increasing, it follows that

$$\phi(v) < \phi\left(\phi^{-1}(M_{[n]})\right) = M_{[n]}.$$

Moreover, since $\tilde{u}^L(s) = v$, by definition $u^L(s) = \phi(v)$, which then gives $u^L(s) < M_{[n]}$. By Theorem 2.1, $s \notin \mathrm{SSE}(u^L)$. □

A.3 Proof of Proposition 5.3

Proposition 5.3. *Given a matrix $\tilde{u}^L$ equivalent to u^L in $\Delta_m \times ([n] \setminus \Lambda)$, a matrix $\tilde{u}'$ inducibility-equivalent to u^L can be computed in polynomial time (in the total bit size of $\tilde{u}^L$ and u^L). Moreover, for every $s \in \mathrm{SSE}(u^L)$, a follower type $\tilde{u}^F$ such that $s \in \mathrm{SSE}(u^L, \tilde{u}^F)$ can be computed in polynomial time.*

Proof. If $\Lambda = \emptyset$, then $\tilde{u}^L$ is readily inducibility-equivalent to u^L. Hence, in what follows, we consider the case where $\Lambda \neq \emptyset$. By Lemma 5.2, it suffices to compute v^* to obtain an inducibility-equivalent matrix.

We use binary search to compute v^*. Indeed, according to the statements (i) and (ii) in Lemma 5.2, we can efficiently determine whether $v \geq v^*$ or $v < v^*$ for any given v as follows. We construct $\tilde{u}'_v$ according to Eq. (9) and compute a follower type $\tilde{u}^F$ that induces an arbitrarily chosen strategy profile s such that $\tilde{u}^L(s) = v$, i.e., $s \in \mathrm{SSE}(\tilde{u}'_v, \tilde{u}^F)$. Given $\tilde{u}'_v$, the follower type $\tilde{u}^F$ can be computed efficiently by using the algorithm proposed by Birmpas et al. [2021]. We then query $\mathcal{A}_{\mathrm{SSE}}$ to check if $s \in \mathrm{SSE}(u^L, \tilde{u}^F)$. By Lemma 5.2 (i) and (ii), the yes- and no-answers imply $v \geq v^*$ and $v < v^*$, respectively.

Now that we can determine whether $v \geq v^*$ or $v < v^*$, we then use binary search to find v^*. In particular, we can compute the exact value of v^* by using a binary search based on the Stern-Brocot tree [Graham et al., 1989]. Specifically, by Eq. (10), v^* must be a rational number as long as u^L and $\tilde{u}^L$ are rational. Moreover, it is straightforward that v^* admits an expression based on the solutions of LPs whose parameters are the entries of u^L and $\tilde{u}^L$ and some constants. Hence, the bit size of v^* can be bounded from above by a polynomial in the total bit size of $\tilde{u}^L$ and u^L. The bound can be obtained given the assumption that a bound on the bit size of u^L is known. So, even though we cannot further distinguish whether $v > v^*$ or $v = v^*$, we are able to determine when we should terminate the search on the Stern-Brocot tree, whereby the value we obtain is exactly v^*. (Intuitively, we terminate the search when other unexplored candidate values of v^* in the Stern-Brocot tree cannot be represented with fewer bits than the upper bound of the bit size of v^*.)

Finally, to compute a follower type to induce any $s \in \mathrm{SSE}(u^L)$, if it is the case where $s \in \Delta_m \times ([n] \setminus \Lambda)$, then by Lemma 5.2 (i) a follower type $\tilde{u}^F$ such that $s \in \mathrm{SSE}(\tilde{u}'_v, \tilde{u}^F)$ is readily one such that $s \in \mathrm{SSE}(u^L, \tilde{u}^F)$. Moreover, $\tilde{u}^F$ can be computed by using the algorithm proposed by Birmpas et al. [2021] as we mentioned above. If it is the case where $s \in \Delta_m \times \Lambda$, we observe that $\tilde{u}'_v$ is in fact equivalent to u^L in $\Delta_m \times \Lambda$. In this case, we can remove actions outside of Λ, compute a follower type to induce s in this smaller game, and augment the obtained type by setting the follower's payoffs for actions in $[n] \setminus \Lambda$ to a sufficiently small value (so that they are strictly dominated) for a type that induces s in the original game. □

A.4 Proof of Corollary 5.5

Corollary 5.5. *With query access to $\mathcal{A}_{\mathrm{SSE}}$, for any given payoff function $u^F \in \mathbb{R}^{m \times n}$, one can compute a strategy profile $s^* \in \operatorname{argmax}_{s \in \mathrm{SSE}(u^L)} u^F(s)$ and a matrix $\tilde{u}^F \in \mathbb{R}^{m \times n}$ such that $s^* \in \mathrm{SSE}(u^L, \tilde{u}^F)$ in time polynomial in the bit size of u^L.*

Proof. According to Theorem 5.4, $\mathrm{SSE}(u^L) = \mathrm{SSE}(\tilde{u}^L)$. Recall that Theorem 2.1 indicates that $\mathrm{SSE}(\tilde{u}^L)$ can be characterized by the linear constraint $u^L(s) \geq \widetilde{M}_{[n]}$ (where $\widetilde{M}_{[n]}$ is the maximin value of $\tilde{u}^L$). Hence, $\max_{s \in \mathrm{SSE}(u^L)} u^F(s)$ can be formulated as an LP and solved in polynomial time. Finally, Theorem 5.4 also indicates that, once an optimal solution s^* to the LP is known, a matrix $\tilde{u}^F$ that induces s^* can be obtained efficiently. □

B Omitted Proofs in Section 6

B.1 Proof of Lemma 6.1

Lemma 6.1. *Suppose that the following conditions hold for strategies $\mathbf{x}^1, \ldots, \mathbf{x}^{m_*-1} \in \Delta_m$:*

(i) $u^L(\mathbf{x}^1, n) = \cdots = u^L(\mathbf{x}^{m_*-1}, n)$*;*

(ii) $x_i^i > 0$ *for all* $i \in [m_* - 1]$*; and*

(iii) $1 - x_i^i = \sum_{k=m_*}^{m} x_k^i$ *for all* $i \in [m_* - 1]$*.*

Then $(-1/x_1^1, -1/x_2^2, \cdots, -1/x_{m_*-1}^{m_*-1}, 0, \cdots, 0) = \gamma \cdot \nabla u^L(\cdot, n)$ *for some* $\gamma > 0$*.*

Proof. Recall that by assumption $u^L(m_*, n) = \cdots = u^L(m, n)$. Hence,

$$u^L(\mathbf{x}, n) \equiv \sum_{i=1}^{m} u^L(i, n) \cdot x_i \equiv \sum_{i=1}^{m_*-1} \left(u^L(i, n) - u^L(m, n)\right) \cdot x_i + u^L(m, n).$$

Given (iii), for each $i \in [m_* - 1]$, we have $u^L(\mathbf{x}^i, n) = (u^L(i, n) - u^L(m, n)) \cdot x_i^i + u^L(m, n)$, so (i) implies that:

$$\left(u^L(1, n) - u^L(m, n)\right) \cdot x_1^1 = \cdots = \left(u^L(m_* - 1, n) - u^L(m, n)\right) \cdot x_{m_*-1}^{m_*-1} =: \gamma.$$

Clearly, $\gamma > 0$. Since $x_i^i > 0$ according to (ii), we get that $u^L(\mathbf{x}, n) \equiv \gamma \cdot \mathbf{a} \cdot \mathbf{x} + \beta$, where $\mathbf{a} = (-1/x_1^1, -1/x_2^2, \cdots, -1/x_{m_*-1}^{m_*-1}, 0, \cdots, 0)$ and $\beta = u^L(m, n)$. □

B.2 Proof of Lemma 6.2

Lemma 6.2. *Given a cover of set $S \subseteq [n]$, a proper cover of S can be constructed in polynomial time.*

Proof. Suppose that μ is a cover of S. Consider the following matrix μ':

$$\mu'(i, j) = \begin{cases} \mu(i, j) & \text{if } j \in [n] \setminus S \\ W & \text{if } j \in S \end{cases}$$

where $W := \min_{i,j \in [m] \times [n] \setminus S} \mu(i, j) - 1$. We argue that μ' is a proper cover of S. Indeed, it is proper since by the above construction no action in S can be a best response to any $\mathbf{x} \in \Delta_m$. It suffices to prove that μ' is a cover.

Let f and f' be the BR-correspondences of μ and μ', respectively. Note that for all $\mathbf{x} \in \mathcal{M}_S$, it must be that $S \cap f(\mathbf{x}) = \emptyset$. Indeed, if $k \in S \cap f(\mathbf{y})$ and $\mathbf{y} \in \mathcal{M}_S$, we would then have

$$\max_{\mathbf{x} \in \mathcal{M}_S} \max_{j \in f(\mathbf{x})} u^L(\mathbf{x}, j) \geq \max_{j \in f(\mathbf{y})} u^L(\mathbf{y}, j) \geq u^L(\mathbf{y}, k) \geq \min_{j \in S} u^L(\mathbf{y}, j) = M_S,$$

contradicting the assumption that μ is a cover.

By construction, $f'(\mathbf{x}) = f(\mathbf{x})$ if $S \cap f(\mathbf{x}) = \emptyset$. Hence, $f'(\mathbf{x}) = f(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{M}_S$. It follows that

$$\max_{\mathbf{x} \in \mathcal{M}_S} \max_{j \in f'(\mathbf{x})} u^L(\mathbf{x}, j) = \max_{\mathbf{x} \in \mathcal{M}_S} \max_{j \in f(\mathbf{x})} u^L(\mathbf{x}, j) < M_S.$$

Hence, μ' is a cover of S. □

B.3 Proof of Lemma 6.3

Lemma 6.3. *For any $S \subseteq [n]$, a cover of S exists if and only if $M_S > M_{[n]}$.*

Proof. We first prove the necessity. Suppose a cover μ of S exists. According to the definition of a cover, for any $\mathbf{x} \in \mathcal{M}_S$, there exists $j \in [n] \setminus S$, such that $u^L(\mathbf{x}, j) < M_S$. Thus, for all $\mathbf{x} \in \mathcal{M}_S$, we have

$$\min_{k \in [n]} u^L(\mathbf{x}, k) \le u^L(\mathbf{x}, j) < M_S.$$

Meanwhile, for all $\mathbf{x} \notin \mathcal{M}_S$,

$$\min_{k \in [n]} u^L(\mathbf{x}, k) \le \min_{k \in S} u^L(\mathbf{x}, k) < M_S.$$

Thus, $M_{[n]} = \max_{\mathbf{x} \in \Delta_m} \min_{j \in [n]} u^L(\mathbf{x}, j) < M_S$.

Next, consider the sufficiency. Suppose that $M_S > M_{[n]}$. We show that a function μ such that $\mu(i, j) = -u^L(i, j)$ is a cover of S. Let f be the BR-correspondence of μ. By construction, $\operatorname{argmax}_{k \in [n]} \mu(\mathbf{x}, k) = \operatorname{argmin}_{k \in [n]} u^L(\mathbf{x}, k)$ for any $\mathbf{x} \in \Delta_m$, so $j \in f(\mathbf{x})$ implies that $u^L(\mathbf{x}, j) = \min_{k \in [n]} u^L(\mathbf{x}, k)$. It follows that

$$u^L(\mathbf{x}, j) = \min_{k \in [n]} u^L(\mathbf{x}, k) \le M_{[n]} < M_S$$

for all $\mathbf{x} \in \Delta_m$. Thus, $\max_{\mathbf{x} \in \mathcal{M}_S} \min_{j \in f(\mathbf{x})} u^L(\mathbf{x}, j) < M_S$, which completes the proof. □

B.4 Proof of Lemma 6.6

Lemma 6.6. *A vector $\mathbf{g}$ such that $I_{\mathbf{g}} = [m_* - 1]$ can be computed in polynomial time.*

Proof. Our approach is to start with an arbitrary $\mathbf{g} \in \mathbb{R}_{>0}^{m_*-1}$, say $\mathbf{g} = (1, \ldots, 1)$, and increase g_i if $(\mathbf{x}^i, n)$ is not yet an SSE. Intuitively, increasing each g_i will cause $u^L(\mathbf{x}^i, n)$ to grow, so the hope is that when $u^L(\mathbf{x}^i, n)$ is sufficiently large, $(\mathbf{x}^i, n)$ yields as much utility as the other optimal strategy profiles, hence forming an SSE. Notably, the value $\max_{\mathbf{x} \in f^{-1}(n) \setminus (\cup_{i \in [m_*-1]} \Gamma_i)} u^L(\mathbf{x}, n)$ may also increase with g_i, so there is the possibility that some $(\mathbf{x}, n)$ outside of $\bigcup_{i \in [m_*-1]} \Gamma_i$ yields a higher utility for the leader than $(\mathbf{x}^i, n)$, hence preventing $(\mathbf{x}^i, n)$ from being an SSE. Thanks to the design of $\tilde{u}^F_{\mathbf{g}}$, this possibility can be eliminated: as we prove in Lemma B.1, $\mathbf{x}^1, \ldots, \mathbf{x}^{m_*-1}$ are always the leader's optimal strategies in $f_{\mathbf{g}}^{-1}(n)$.

We proceed as follows.

- If initially $I_{\mathbf{g}} = \emptyset$, then we increase all the components of $\mathbf{g}$ to a sufficiently large number N. We show in Lemma B.2 that there exists an N that results in at least one $(\mathbf{x}^i, n)$ being an SSE, and this N can be computed in polynomial time.
- Subsequently, if $I_{\mathbf{g}} \neq \emptyset$ but $I_{\mathbf{g}} \subsetneq [m_* - 1]$, we show in Lemma B.3 that, by increasing g_i for an arbitrary $i \notin I_{\mathbf{g}}$ to an appropriate value g_i' (while fixing all the other components of $\mathbf{g}$), the strategy profile $(\mathbf{x}^i, n)$ will become a new SSE in addition to the existing ones. Moreover, g_i' can be computed in polynomial time.

Hence, by performing the above steps by at most $m_* - 1$ times, we will find a $\mathbf{g}$ such that $I_{\mathbf{g}} = [m_* - 1]$.

We prove Lemmas B.1 to B.3 to complete the proof of Lemma 6.6.

Lemma B.1. *For any $\mathbf{g} \in \mathbb{R}_{>0}^{m_*-1}$, it holds that $\max_{\mathbf{x} \in f_{\mathbf{g}}^{-1}(n)} u^L(\mathbf{x}, n) = \max_{i \in [m_*-1]} u^L(\mathbf{x}^i, n)$.*

Proof. We prove that every vertex $\mathbf{v}$ of $f_{\mathbf{g}}^{-1}(n)$ lies in some Γ_i, $i \in [m_* - 1]$; in other words, $\mathbf{v} \in \Gamma_i \cap f_{\mathbf{g}}^{-1}(n)$. Recall that $\mathbf{x}^i \in \operatorname{argmin}_{\mathbf{x} \in \Gamma_i \cap f_{\mathbf{g}}^{-1}(n)} x_i$. Hence, $\mathbf{v} \in \Gamma_i \cap f_{\mathbf{g}}^{-1}(n)$ implies that $x_i^i \le v_i$, and in turn, $u^L(\mathbf{v}, n) \le u^L(\mathbf{x}^i, n)$ according to Observation 6.5. Since $\max_{\mathbf{x} \in f_{\mathbf{g}}^{-1}(n)} u^L(\mathbf{x}, n)$ is always attained at some vertex of $f_{\mathbf{g}}^{-1}(n)$, the stated result then follows.

Pick an arbitrary vertex $\mathbf{v}$ of $f_{\mathbf{g}}^{-1}(n)$. Since $\mathbf{v} \in \mathbb{R}^m$, it lies in m hyperplanes of $f_{\mathbf{g}}^{-1}(n)$ and is uniquely defined by them. By definition, $f_{\mathbf{g}}^{-1}(n)$ is defined by the following linear constraints, each corresponding to a hyperplane:

$$
\begin{aligned}
& x_1 + \cdots + x_m = 1 && && \text{(23a)} \\
& x_i \ge 0 && \text{for all } i \in [m] && \text{(23b)} \\
& \tilde{u}^F_{\mathbf{g}}(\mathbf{x}, n) \ge \tilde{u}^F_{\mathbf{g}}(\mathbf{x}, j) && \text{for all } j \in [n-1] && \text{(23c)}
\end{aligned}
$$

W.l.o.g., by rearranging the rows and columns of the matrix, we can assume that the m hyperplanes defining $\mathbf{v}$ correspond to the following coefficient matrix A, for some ℓ_0, ℓ_1, ℓ_2, and k, where we let $\tilde{\mu}(i,j) = W - \mu(i,j)$ for all i, j.

$$
\begin{array}{r}
\ell_0 \\ \\ \ell_1 \\ \\ \\ \ell_2 \\ \\ \\ k \\ \\
\end{array}
\left(\begin{array}{ccc|ccc}
1 & \cdots & 1 & 1 & \cdots & 1 \\ \hline
1 & & & & & \\
& \ddots & & & & \\
& & 1 & & & \\ \hline
& & & 1 & & \\
& & & & \ddots & \\
& & & & 1 & \\ \hline
g_1 & \cdots & g_{m_*-1} & \tilde{\mu}(m_*, 1) & \cdots & \tilde{\mu}(m, 1) \\
\vdots & & \vdots & \vdots & & \vdots \\
g_1 & \cdots & g_{m_*-1} & \tilde{\mu}(m_*, k) & \cdots & \tilde{\mu}(m, k)
\end{array}\right)
\begin{array}{l}
\textit{Eq. (23a)} \\ \\ \\ \\ \textit{Eq. (23b)} \\ \\ \\ \\ \textit{Eq. (23c)} \\ \\
\end{array}
$$

Note that ℓ_0 is either 1 or 0 depending on whether Eq. (23a) is one of the m hyperplane or not, so in general, we have $\ell_1 + \ell_2 + k \ge m - 1$. We have $A \cdot \mathbf{v} = \mathbf{b}$, where $\mathbf{b}$ is the vector consisting of the corresponding constants in the inequalities in Eq. (23). Since $\mathbf{v}$ is uniquely defined by A, we have $\operatorname{rank}(A) = m$. In what follows, we let $A_{i:j}$ denote the submatrix formed by the i-th to the j-th rows of A.

We next argue that $\ell_1 \ge m_* - 2$ to complete the proof. According to Eq. (23b), this indicates that for at least $m_* - 2$ actions $i \in [m_* - 1]$, we have that $v_i = 0$. In other words, $v_i > 0$ for at most one $i \in [m_* - 1]$, so by definition this means that $\mathbf{v} \in \Gamma_i$ for some $i \in [m_* - 1]$.

Suppose for the sake of contradiction that $\ell_1 \le m_* - 3$. Hence, $\ell_2 + k \ge m - 1 - \ell_1 \ge m - m_* + 2$. Consider the following submatrix of A:

$$
U = \begin{pmatrix}
\tilde{\mu}(m_* + \ell_2, 1) & \cdots & \tilde{\mu}(m, 1) \\
\vdots & & \vdots \\
\tilde{\mu}(m_* + \ell_2, k) & \cdots & \tilde{\mu}(m, k)
\end{pmatrix}
$$

and the sub-vector $\mathbf{v}' = (v_{m_* + \ell_2 - 1}, \ldots, v_m)$ of $\mathbf{v}$ corresponding to the same columns. Hence, U is a k-by-k' matrix, with $k' \le k - 1$.

By $A \cdot \mathbf{v} = \mathbf{b}$ we get that $U \cdot \mathbf{v}' = \mathbf{1} \cdot \lambda$, where $\lambda = \sum_{i=1}^{m_*-1} g_i \cdot v_i$. Note that $\mathbf{v}'$ is a size-k' vector, so it must be that $\operatorname{rank}\begin{pmatrix} U & \mathbf{1}^\top \end{pmatrix} \le k' \le k - 1$ (otherwise, the system of linear equations $U \cdot \mathbf{x} = \mathbf{1} \cdot \lambda$ would have no solution). Note that via linear transformation, the submatrix $A_{m-k+1:m}$ can be transformed into $\begin{pmatrix} O & \mathbf{1}^\top & U \end{pmatrix}$, where O denotes a k-by-$(m_* + \ell_2 - 2)$ matrix with all entries being 0. Hence, we get that

$$\operatorname{rank}\left(A_{m-k+1:m}\right) = \operatorname{rank}\begin{pmatrix} O & \mathbf{1}^\top & U \end{pmatrix} \leq k-1.$$

Consequently,

$$\operatorname{rank}(A) \leq \operatorname{rank}\left(A_{1:m-k}\right) + \operatorname{rank}\left(A_{m-k+1:m}\right) \leq (m-k) + (k-1) < m,$$

which contradicts the fact that $\operatorname{rank}(A) = m$. □

Lemma B.2. *There exists $N > 0$, such that $I_{\mathbf{g}} \neq \emptyset$ if $g_i \geq N$ for all $i \in [m_* - 1]$. Moreover, N can be computed in polynomial time.*

Proof. Given Lemma B.1, it suffices to find a value N and prove that action n is an SSE response of game $(u^L, \tilde{u}^F_{\mathbf{g}})$ if $g_i \geq N$ for all $i \in [m_* - 1]$.

Let $D = \max_{i,j,i',j'} |u^L(i,j) - u^L(i',j')|$. For every $\mathbf{x} \in \Delta_m$, we have

$$u^L(\mathbf{x}, n) = \sum_{i=1}^{m_*-1} x_i \cdot u^L(i,n) + \left(1 - \sum_{i=1}^{m_*-1} x_i\right) \cdot M_{\{n\}} \geq M_{\{n\}} - \left(\sum_{i=1}^{m_*-1} x_i\right) \cdot D,$$

where we used the fact that $u^L(i,n) \geq M_{\{n\}} - D$ for all i.

Moreover, let

$$U = \max_{i,j} u^L(i,j) \quad \text{and} \quad T = \max_{\mathbf{x} \in \mathcal{M}_{\{n\}}, j \in h(\mathbf{x})} u^L(\mathbf{x}, j),$$

where h is the BR-correspondence of μ. For every $j \in [n-1]$,

$$\begin{aligned} u^L(\mathbf{x}, j) &= \sum_{i=1}^{m_*-1} x_i \cdot u^L(i,j) + \sum_{i=m_*}^{m} x_i \cdot u^L(i,j) \\ &\leq \left(\sum_{i=1}^{m_*-1} x_i\right) \cdot U + \left(1 - \sum_{i=1}^{m_*-1} x_i\right) \cdot T = \left(\sum_{i=1}^{m_*-1} x_i\right) \cdot (U - T) + T. \end{aligned}$$

Recall that since μ is a cover of $\{n\}$, by definition $T < M_{\{n\}}$. So if it holds that $f_{\mathbf{g}}(\mathbf{x}) = \{n\}$ for every $\mathbf{x}$ such that

$$c \coloneqq \frac{M_{\{n\}} - T}{D + U - T} < \sum_{i=1}^{m_*-1} x_i,$$

then we know that n must be an SSE response of $(u^L, \tilde{u}^F_{\mathbf{g}})$; indeed, for all $\mathbf{y}$ such that $\sum_{i=1}^{m_*-1} y_i \leq c$, this leads to

$$u^L(\mathbf{y}, j) \leq T + c \cdot (U - T) = M_{\{n\}} - c \cdot D \leq \max_{\mathbf{x} \in f_{\mathbf{g}}^{-1}(n)} u^L(\mathbf{x}, n)$$

for all $j \in [n-1]$. It suffices to let

$$g_i \geq \overline{N} \coloneqq \frac{1-c}{c} \cdot \left(\max_{i,j} \mu(i,j) - W\right)$$

for every $i \in [m_* - 1]$ to ensure this, so any $N \geq \overline{N}$ satisfies the condition in the statement of this lemma.

To compute N, we can start from an arbitrary value $N > 0$ and $\mathbf{g} \coloneqq (N, \ldots, N)$. We query the oracle $\mathcal{A}_{\text{SSE}}$ to check if n is an SSE response of $(u^L, \tilde{u}^F_{\mathbf{g}})$. Double N and repeat the step if it is not. This way, a value $N \geq \overline{N}$ can be found in time $\log(\overline{N})$. □

Lemma B.3. *Suppose that $I_{\mathbf{g}} \neq \emptyset$ and $i \in [m_* - 1] \setminus I_{\mathbf{g}}$. There exists $g^* > g_i$, such that $I_{\mathbf{g}^*} = I_{\mathbf{g}} \cup \{i\}$, where $\mathbf{g}^* = (g_1, \dots, g_{i-1}, g^*, g_{i+1}, \dots, g_{m_*-1})$. Moreover, g^* can be computed in polynomial time.*

Proof. Let $V_{\mathbf{g}} := \max_{j \in [n]} \max_{\mathbf{x} \in f_{\mathbf{g}}^{-1}(j)} u^L(\mathbf{x}, j)$ denote the leader's SSE payoff in $(u^L, \tilde{u}_{\mathbf{g}}^F)$. We show that the following g^* satisfies the stated properties.

$$g^* = \frac{u^* - (1 - y_i^*) \cdot W}{y_i^*}, \tag{24}$$

where u^* and $\mathbf{y}^*$ are an optimal solution to the following LP.

$$\begin{aligned} \min_{u, \mathbf{y} \in \Delta_m} \quad & u && && (25) \\ \text{subject to} \quad & \mathbf{y} \in \Gamma_i && && (25a) \\ & \tilde{u}_{\mathbf{g}}^F(\mathbf{y}, j) \leq u && \text{for all } j \in [n-1] && (25b) \\ & u^L(\mathbf{y}, n) = V_{\mathbf{g}} && && (25c) \end{aligned}$$

We demonstrate the following to complete the proof.

- g^* is well-defined, i.e., LP (25) is feasible and has a bounded optimal value; moreover, $y_i^* > 0$ (Lemma B.4).
- $g^* > g_i$ (Lemma B.6), as stated in the lemma. This is important because it ensures that $g_\ell > 0$ for all $\ell \in [m_* - 1]$ throughout the process (as long as we initialize all the components of $\mathbf{g}$ to be positive).
- Finally, let $\mathbf{g}' := (g_1, \dots, g_{i-1}, g', g_{i+1}, \dots, g_{m_*-1})$. Then $g' = g^*$ is the only point where $I_{\mathbf{g}'} = I_{\mathbf{g}} \cup \{i\}$ (Lemma B.7). This means that we can use binary search to pin down g^*. In particular, since g^* is defined based on the solution to LP (25), its bit-size is bounded by a polynomial in the size of the LP. Hence, we can find g^* in polynomial time.

Note that we cannot solve LP (25) directly because it involves unknown parameters. Moreover, $\tilde{u}_{\mathbf{g}}^F(\cdot, j)$ in fact does not depend on $\mathbf{g}$ according to its definition in Eq. (15), and Lemmas B.6 and B.7 imply that $V_{\mathbf{g}^*} = V_{\mathbf{g}}$ remains unchanged after we update $\mathbf{g}$ to $\mathbf{g}^*$, so repeat application of Lemma B.3 as we do in the proof of Lemma 6.6 will not increase the bit size of LP (25). ☐

This completes the proof of Lemma 6.6. ☐

B.4.1 Lemmas Used in the Proof of Lemma B.3

In what follows, we let $\mathbf{x}'^k$ denote the points defined with $\mathbf{g}'$ instead of $\mathbf{g}$ in Eq. (14), $k \in [m_* - 1]$. Similarly, we denote by $\mathbf{x}''^k$ and $\mathbf{x}^{*k}$ the points defined with $\mathbf{g}''$ and $\mathbf{g}^*$, respectively.

Lemma B.4. *LP (25) is feasible and bounded, and $y_i^* > 0$.*

Proof. Notice that LP (25) essentially computes $\max_{j \in [n-1]} \tilde{u}_{\mathbf{g}}^F(\mathbf{y}, j)$ with $\mathbf{y}$ satisfying Eq. (25a) and Eq. (25c). Hence, its optimal value is bounded, and it is feasible if there exists $\mathbf{y} \in \Gamma_i$ such that $u^L(\mathbf{y}, n) = V_{\mathbf{g}}$. Note that for all $\mathbf{y} \in \Gamma_i$, we have

$$u^L(\mathbf{y}, n) = y_i \cdot u^L(i, n) + (1 - y_i) \cdot M_{\{n\}}.$$

Let $\mathbf{z}$ be an arbitrary point such that $\sum_{k=m_*}^m z_k = 1$ (hence $\mathbf{z} \in \Gamma_i$).

- When $\mathbf{y} = \mathbf{z}$, we have $u^L(\mathbf{y}, n) = M_{\{n\}}$.
- When $\mathbf{y} = \mathbf{x}^i$, we have $u^L(\mathbf{y}, n) = u^L(\mathbf{x}^i, n) < V_{\mathbf{g}}$ as $(\mathbf{x}^i, n)$ is not an SSE by assumption.

By definition (i.e., Eq. (14)), $\mathbf{x}^i \in \Gamma_i$. By assumption $I_{\mathbf{g}} \neq \emptyset$, so $(\mathbf{x}^k, n)$ is an SSE response for some $k \in [m_* - 1]$. Hence, by definition,

$$V_{\mathbf{g}} = u^L(\mathbf{x}^k, n) < \max_{\mathbf{x}\in\Gamma_i} u^L(\mathbf{x}, n) \leq M_{\{n\}},$$

where $u^L(\mathbf{x}^k, n) < \max_{\mathbf{x}\in\Gamma_k} u^L(\mathbf{x}, n)$ follows by the fact that $x_k^k > 0$ (Observation 6.4) and $u^L(\mathbf{x}, n)$ decreases with x_k when $\mathbf{x}$ is restricted in Γ_k (Observation 6.5). Therefore, by continuity, there must exist $\mathbf{y} \in \Gamma_i$ such that $u^L(\mathbf{y}, n) = V_{\mathbf{g}}$. Moreover, for all such $\mathbf{y}$, $y_i > 0$ as otherwise $u^L(\mathbf{y}, n) = M_{\{n\}} > V_{\mathbf{g}}$. □

Lemma B.5. *$u^L(\mathbf{x}^i, n)$ increases strictly with g_i.*

Proof. Pick two arbitrary values $g', g'' \in \mathbb{R}$ such that $g' < g''$. Let

$$\begin{aligned} &\mathbf{g}' = (g_1, \dots, g_{i-1}, g', g_{i+1}, g_{m_*-1}) \\ \text{and} \quad &\mathbf{g}'' = (g_1, \dots, g_{i-1}, g'', g_{i+1}, g_{m_*-1}). \end{aligned}$$

Suppose for the sake of contradiction that $u^L(\mathbf{x}'^i, n) \geq u^L(\mathbf{x}''^i, n)$.

By Observation 6.5, $u^L(\mathbf{x}'^i, n) \geq u^L(\mathbf{x}''^i, n)$ implies that $x_i'^i \leq x_i''^i$. Moreover,

$$\begin{aligned} \tilde{u}^F_{\mathbf{g}'}(\mathbf{x}'^i, n) &= g' \cdot x_i'^i + \sum_{k=m_*}^{m} x_k'^{\,i} \cdot W \\ &< g'' \cdot x_i'^i + \sum_{k=m_*}^{m} x_k'^{\,i} \cdot W = \tilde{u}^F_{\mathbf{g}''}(\mathbf{x}'^i, n), \end{aligned} \tag{26}$$

where we used the fact that $x_i'^i > 0$ (Observation 6.4).

By definition, $\mathbf{x}'^i \in f_{\mathbf{g}'}^{-1}(n)$. So, $\tilde{u}^F_{\mathbf{g}'}(\mathbf{x}'^i, n) \geq \tilde{u}^F_{\mathbf{g}'}(\mathbf{x}'^i, j)$ for all $j \in [n-1]$. Hence, using Eq. (26), we have

$$\tilde{u}^F_{\mathbf{g}''}(\mathbf{x}'^i, n) > \tilde{u}^F_{\mathbf{g}'}(\mathbf{x}'^i, n) \geq \tilde{u}^F_{\mathbf{g}'}(\mathbf{x}'^i, j) = \tilde{u}^F_{\mathbf{g}''}(\mathbf{x}'^i, j),$$

where the last equality holds since $\tilde{u}^F_{\mathbf{g}}(\cdot, j)$ does not depend on $\mathbf{g}$ by construction. This means that $\tilde{u}^F_{\mathbf{g}''}(\mathbf{x}, n) > \tilde{u}^F_{\mathbf{g}''}(\mathbf{x}, j)$ for $\mathbf{x}$ in a neighborhood $\mathcal{N}$ of $\mathbf{x}'^i$. Since $x_i'^i > 0$, $\mathcal{N} \cap \Gamma_i$ must contain a point $\mathbf{x}$ such that $x_i < x_i'^i$. We then establish the following contradictory transitions:

$$\min_{\mathbf{y}\in\Gamma_i\cap f_{\mathbf{g}'}^{-1}(n)} y_i \leq x_i < x_i'^i = \min_{\mathbf{y}\in\Gamma_i\cap f_{\mathbf{g}'}^{-1}(n)} y_i.$$

This completes the proof. □

Lemma B.6. *$u^L(\mathbf{x}^{*i}, n) = V_{\mathbf{g}}$ and $g^* > g_i$.*

Proof. We have $V_{\mathbf{g}} > u^L(\mathbf{x}^i, n)$ given that $i \notin I_{\mathbf{g}}$. Hence, once $u^L(\mathbf{x}^{*i}, n) = V_{\mathbf{g}}$ holds, we get that $u^L(\mathbf{x}^{*i}, n) = V_{\mathbf{g}} > u^L(\mathbf{x}^i, n)$, and in turn, $g^* > g_i$ according to the monotonicity shown in Lemma B.5. Hence, it suffices to prove $u^L(\mathbf{x}^{*i}, n) = V_{\mathbf{g}}$.

By Eq. (25a) $\mathbf{y}^* \in \Gamma_i$, so we have

$$\tilde{u}^F_{\mathbf{g}^*}(\mathbf{y}^*, n) = y_i^* \cdot g^* + (1 - y_i^*) \cdot W = u^*,$$

where the second transition follows by Eq. (24). Note that $\tilde{u}^F_{\mathbf{g}'}(\cdot, j)$ is not dependent on $\mathbf{g}'$ for all $j \in [n-1]$. Hence, since $\mathbf{y}^*$ and u^* satisfy Eq. (25b), we have

$$\tilde{u}^F_{\mathbf{g}^*}(\mathbf{y}^*, j) \leq u^*, \quad \text{for all } j \in [n-1].$$

As a result, $\tilde{u}^F_{\mathbf{g}^*}(\mathbf{y}^*, n) \ge \tilde{u}^F_{\mathbf{g}^*}(\mathbf{y}^*, j)$ for all $j \in [n]$, which means $\mathbf{y}^* \in f^{-1}_{\mathbf{g}^*}(n)$. Hence, by Eq. (13), $x^{*i}_i \le y^*_i$, so Observation 6.5 implies that $u^L(\mathbf{x}^{*i}, n) \ge u^L(\mathbf{y}^*, n) = V_{\mathbf{g}}$, where $u^L(\mathbf{y}^*, n) = V_{\mathbf{g}}$ follows by Eq. (25c).

To see that $u^L(\mathbf{x}^{*i}, n) \le V_{\mathbf{g}}$, suppose for the sake of contradiction that $u^L(\mathbf{x}^{*i}, n) > V_{\mathbf{g}}$. Since $u^L(\mathbf{y}^*, n) = V_{\mathbf{g}}$, using Observation 6.5 again, we have that $x^{*i}_i < y^*_i$. Consider a point

$$\mathbf{z} = \lambda \cdot \mathbf{x}^{*i} + (1 - \lambda) \cdot \mathbf{e}^i,$$

where $\lambda \in [0, 1]$, and $\mathbf{e}^i = (e^i_1, \ldots, e^i_m)$ such that $e^i_i = 1$ and $e^i_k = 0$ for all $k \neq i$. Since both $\mathbf{x}^{*i}$ and $\mathbf{e}^i$ are in Γ_i, we have $\mathbf{z} \in \Gamma_i$. Moreover, since $e^i_i = 1 \ge y^*_i$, there exists $\lambda \in [0, 1)$ that results in $z_i = y^*_i$ and, in turn,

$$\tilde{u}^F_{\mathbf{g}^*}(\mathbf{z}, n) = \tilde{u}^F_{\mathbf{g}^*}(\mathbf{y}^*, n) = u^*, \tag{27}$$

and

$$u^L(\mathbf{z}, n) = u^L(\mathbf{y}^*, n) = V_{\mathbf{g}}. \tag{28}$$

Meanwhile, note that for all $j \in [n-1]$, we have

$$\tilde{u}^F_{\mathbf{g}^*}(\mathbf{e}^i, n) = g^* > 0 = \tilde{u}^F_{\mathbf{g}^*}(\mathbf{e}^i, j),$$

In addition, since $\mathbf{x}^{*i} \in f^{-1}_{\mathbf{g}^*}(n)$, by definition, we have

$$\tilde{u}^F_{\mathbf{g}^*}(\mathbf{x}^{*i}, n) \ge \tilde{u}^F_{\mathbf{g}^*}(\mathbf{x}^{*i}, j).$$

Consequently, now that $\lambda < 1$, we must have

$$\tilde{u}^F_{\mathbf{g}^*}(\mathbf{z}, n) > \tilde{u}^F_{\mathbf{g}^*}(\mathbf{z}, j).$$

Plugging into Eq. (27) gives $\tilde{u}^F_{\mathbf{g}^*}(\mathbf{z}, j) < u^*$ for all $j \in [n-1]$. Pick $u \in [\max_{j \in [n-1]} \tilde{u}^F_{\mathbf{g}^*}(\mathbf{z}, j), u^*)$. The fact that $\mathbf{z} \in \Gamma_i$ and $u^L(\mathbf{z}, n) = V_{\mathbf{g}}$ (i.e., Eq. (28)) we argued above implies that u and $\mathbf{z}$ form a feasible solution to LP (25). The objective value of this solution, i.e., u, is strictly smaller than u^*, which is a contradiction. This completes the proof of Lemma B.6. □

Lemma B.7. *$I_{\mathbf{g}'} = I_{\mathbf{g}}$ if $g' \in [g_i, g^*)$, $I_{\mathbf{g}'} = \{i\}$ if $g' > g^*$, and $I_{\mathbf{g}'} = I_{\mathbf{g}} \cup \{i\}$ if $g' = g^*$.*

Proof. We first argue that the following results hold for any $g' \ge g_i$.

(i) $\max_{\mathbf{x} \in f^{-1}_{\mathbf{g}'}(j)} u^L(\mathbf{x}, j) \le \max_{\mathbf{x} \in f^{-1}_{\mathbf{g}}(j)} u^L(\mathbf{x}, j)$, for all $j \in [n-1]$; and

(ii) $u^L(\mathbf{x}'^k, n) = u^L(\mathbf{x}^k, n)$, for all $k \in [m_* - 1] \setminus \{i\}$.

In other words, (i) says that the leader's payoffs for inducing a best response $j \in [n-1]$ does not increase with g_i; and (ii) says that the leader's payoffs for $(\mathbf{x}^k, n)$, $k \neq i$, does not change with g_i.

To see (i), note that by construction, $\tilde{u}^F_{\mathbf{g}}(\mathbf{x}, j)$ does not depend on g_i for all $j \in [n-1]$. Moreover, now that $g' \ge g_i$, $\tilde{u}^F_{\mathbf{g}'}(\mathbf{x}, n) \ge \tilde{u}^F_{\mathbf{g}}(\mathbf{x}, n)$ for all $\mathbf{x} \in \Delta_m$. Therefore, any leader strategy $\mathbf{x}$ that does not induce j under $\tilde{u}^F_{\mathbf{g}}$ does not induce j under $\tilde{u}^F_{\mathbf{g}'}$, either. We have $f^{-1}_{\mathbf{g}'}(j) \subseteq f^{-1}_{\mathbf{g}}(j)$, so (i) follows immediately.

To see (ii), note that $\tilde{u}^F_{\mathbf{g}}(\mathbf{x}, n)$ does not depend on g_i when $\mathbf{x}$ is restricted in Γ_k, $k \neq i$. Hence, $\Gamma_k \cap f^{-1}_{\mathbf{g}'}(n) = \Gamma_k \cap f^{-1}_{\mathbf{g}}(n)$. By Observation 6.5, the statement then follows.

We then proceed as follows. By assumption, $I_{\mathbf{g}} \neq \emptyset$. W.l.o.g., let us assume that $1 \in I_{\mathbf{g}}$. Hence, $u^L(\mathbf{x}^1, n) = V_{\mathbf{g}}$. The statements (i) and (ii) then imply that

$$\begin{aligned}\max_{j\in[n-1]} \max_{\mathbf{x}\in f_{\mathbf{g}'}^{-1}(j)} u^L(\mathbf{x}, j) &\leq \max_{j\in[n-1]} \max_{\mathbf{x}\in f_{\mathbf{g}}^{-1}(j)} u^L(\mathbf{x}, j) \\ &\leq V_{\mathbf{g}} \\ &= u^L(\mathbf{x}^1, n) = u^L(\mathbf{x}'^1, n) \leq \max_{\mathbf{x}\in f_{\mathbf{g}'}^{-1}(n)} u^L(\mathbf{x}, n),\end{aligned}$$

where $u^L(\mathbf{x}'^1, n) \leq \max_{\mathbf{x}\in f_{\mathbf{g}'}^{-1}(n)} u^L(\mathbf{x}, n)$ because $\mathbf{x}'^1 \in f_{\mathbf{g}'}^{-1}(n)$ given that $1 \in I_{\mathbf{g}}$. This means that the following holds for the leader's SSE utility in $(u^L, \tilde{u}^F_{\mathbf{g}'})$:

$$V_{\mathbf{g}'} \equiv \max_{j\in[n]} \max_{\mathbf{x}\in f_{\mathbf{g}'}^{-1}(j)} u^L(\mathbf{x}, j) = \max_{\mathbf{x}\in f_{\mathbf{g}'}^{-1}(n)} u^L(\mathbf{x}, n).$$

Applying Lemma B.1 further gives

$$V_{\mathbf{g}'} = \max_{k\in[m_*-1]} u^L(\mathbf{x}'^k, n).$$

Recall that

- $u^L(\mathbf{x}'^k, n) = u^L(\mathbf{x}^k, n) \leq V_{\mathbf{g}}$, for all $k \in [m_* - 1] \setminus \{i\}$;
- $u^L(\mathbf{x}'^1, n) = u^L(\mathbf{x}^1, n) = V_{\mathbf{g}}$;
- $u^L(\mathbf{x}^{*i}, n) = V_{\mathbf{g}}$ (Lemma B.6) and $u^L(\mathbf{x}'^i, n)$ increases with g'_i (Lemma B.5).

As a result, for all $k \in I_{\mathbf{g}}$:

- when $g' \in [g_i, g^*)$, we have $V_{\mathbf{g}'} = u^L(\mathbf{x}'^k, n) > u^L(\mathbf{x}'^i, n)$;
- when $g' = g^*$, we have $V_{\mathbf{g}'} = u^L(\mathbf{x}'^k, n) = u^L(\mathbf{x}'^i, n)$; and
- when $g' > g^*$, we have $V_{\mathbf{g}'} = u^L(\mathbf{x}'^i, n) > u^L(\mathbf{x}'^k, n)$.

Lemma B.7 then follows. □

B.5 Proof of Proposition 6.8

Proposition 6.8. *Suppose that $S \subseteq [n]$, $Q = \{j \in [n] : M_{\{j\}} = M_S\}$, and the following are given: 1) $\mathbf{a}_j$ for all $j \in [n] \setminus (S \cup Q)$, 2) a base function $\tilde{u}^F$ of S, and 3) $\mathcal{M}_S$ (given by linear constraints). In polynomial time, we can either compute a cover of S or correctly decide that S does not admit a cover.*

Proof. We construct the following follower type. For every $\mathbf{x} \in \Delta_m$,

$$\mu(\mathbf{x}, j) := \begin{cases} 0, & \text{if } j \in S; \\ \mathbf{b}_j \cdot \mathbf{x} - c_j, & \text{otherwise.} \end{cases} \tag{29}$$

where for each $j \in [n] \setminus S$, $(\mathbf{b}_j, c_j)$ is a hyperplane such that for any $\mathbf{x} \in \Delta_m$:

$$u^L(\mathbf{x}, j) \geq M_S \quad \Longleftrightarrow \quad \mathbf{b}_j \cdot \mathbf{x} \leq c_j. \tag{30}$$

As we will show in Lemma B.8 below, whether μ is a cover of S can be decided efficiently, and if not, then S does not admit any other cover, either. Intuitively, μ aims to bring down the leader's maximum attainable payoff in $\mathcal{M}_S$ to below M_S, so it needs to avoid responding with actions j that make $u^L(\mathbf{x}, j) \geq M_S$. Ideally, we could just use $\mu = -u^L$ to achieve this, but since we do not know u^L we use μ defined above, which functions similarly. The hyperplanes $(\mathbf{b}_j, c_j)$ satisfying Eq. (30) can indeed be computed efficiently according to Lemma B.9.

We prove Lemmas B.8 and B.9 to complete the proof.

Lemma B.8. *It can be decided in polynomial time whether the payoff matrix μ defined in Eq. (29) is a cover of S or not. Moreover, if μ is not a cover of S, then S does not admit any other cover.*

Proof. To decide whether μ is a cover of S, we check the satisfiability of the following linear constraints (by assumption, $\mathcal{M}_S$ is given as a set of linear constraints):

$$\begin{aligned} &\mathbf{x} \in \mathcal{M}_S \\ &\mu(\mathbf{x}, j) \le 0 \qquad \text{for all } j \in [n] \end{aligned}$$

This can be done in polynomial time. Specifically:

- If the constraints are satisfiable, then there exists $\mathbf{y} \in \mathcal{M}_S$ such that $\mu(\mathbf{y}, j) \le 0 = \mu(\mathbf{y}, k)$ for all $j \in [n], k \in S$; hence, $k \in h(\mathbf{y})$, where h denotes the BR-correspondence of μ. It follows that
$$\max_{\mathbf{x} \in \mathcal{M}_S, j \in h(\mathbf{x})} u^L(\mathbf{x}, j) \ge u^L(\mathbf{y}, k) \ge \min_{j \in S} u^L(\mathbf{x}, j) = M_S.$$
By definition, this means that μ is not a cover of S.

- Conversely, suppose that the constraints are not satisfiable. Pick arbitrary $\mathbf{x} \in \mathcal{M}_S$ and $j \in h(\mathbf{x})$. Hence, $\mu(\mathbf{x}, j) > 0$, and according to the definition of μ, it must be that $j \notin S$. We then have $\mu(\mathbf{x}, j) = \mathbf{b}_j \cdot \mathbf{x} - c_j > 0$, which implies that $u^L(\mathbf{x}, j) < M_S$ by Eq. (30). Since the choice of $\mathbf{x}$ and k is arbitrary, we then have
$$\max_{\mathbf{x} \in \mathcal{M}_S, j \in h(\mathbf{x})} u^L(\mathbf{x}, j) < M_S,$$
so μ is a cover of S.

Next, consider the second part of the statement. Suppose that S admits a cover. We show that μ must be a cover of it. According to the necessary condition demonstrated in Lemma 6.3, we have $M_S > M_{[n]}$. Hence, for every $\mathbf{x} \in \mathcal{M}_S$, $\min_{j \in [n]} u^L(\mathbf{x}, j) \le M_{[n]} < M_S$, which means that there exists $j \in [n] \setminus S$ such that $u^L(\mathbf{x}, j) < M_S$ (note that $\min_{j \in S} u^L(\mathbf{x}, j) = M_S$ as $\mathbf{x} \in \mathcal{M}_S$). By Eq. (30), we then have $\mathbf{b}_j \cdot \mathbf{x} > c_j$, which gives

$$\mu(\mathbf{x}, j) = \mathbf{b}_j \cdot \mathbf{x} - c_j > 0.$$

This means that the linear constraints above are not satisfiable. Therefore, μ is a cover of S. □

Lemma B.9. *A hyperplane $(\mathbf{b}_j, c_j)$ satisfying Eq. (30) can be computed in polynomial time for every $j \in [n] \setminus S$.*

Proof. W.l.o.g., we show how to learn $\mathbf{b}_n$ and c_n, and we assume that $n \notin S$. We first define the following parameters:

$$\check{d} := \min_{\mathbf{x} \in \mathcal{M}_S} \mathbf{a}_j \cdot \mathbf{x}, \quad \hat{d} := \max_{\mathbf{x} \in \mathcal{M}_S} \mathbf{a}_j \cdot \mathbf{x}, \quad \text{and} \quad d^* := (M_S - \beta_n)/\gamma_n.$$

One can easily verify that $\mathbf{b}_n$ and c_n defined as follows satisfy Eq. (30):

- If $n \in Q$, we let $b_{n,i} = \begin{cases} 1, & \text{if } u^L(i, n) = M_{\{n\}} \\ 0, & \text{otherwise} \end{cases}$, and $c_n = \check{d}$.

- If $n \notin Q$, we let $\mathbf{b}_n = \mathbf{a}_n$ and $c_n = \begin{cases} \check{d} - 1, & \text{if } d^* < \check{d}; \\ d^*, & \text{if } d^* \in [\check{d}, \hat{d}]; \\ \hat{d}, & \text{if } d^* > \hat{d}. \end{cases}$

According to the assumption stated in Proposition 6.8, $\mathbf{a}_n$ is given if $n \notin Q$. Moreover, $\mathcal{M}_S$ is known, so $\check{d}$ and $\hat{d}$ can be computed directly. However, we cannot compute d^* yet since β_n, γ_n, and M_S remain unknown. To complete the proof, we show how to learn d^* next.

Recall that as stated in Proposition 6.8, a based function $\tilde{u}^F$ of S is given. Define the following payoff function parameterized by $d \in \mathbb{R}$. For every $\mathbf{x} \in \Delta_m$,

$$\tilde{u}^F_d(\mathbf{x}, j) \coloneqq \begin{cases} \tilde{u}^F(\mathbf{x}, j), & \text{if } j \in [n-1]; \\ \tilde{u}^F(\mathbf{x}, k) + (d - \mathbf{b}_n \cdot \mathbf{x}), & \text{if } j = n. \end{cases} \tag{31}$$

where we use an arbitrary $k \in \operatorname{argmax}_{\ell \in [n-1]} \tilde{u}^F(\mathbf{z}, \ell)$, defined with arbitrarily selected

$$\mathbf{z} \in Z(d) \coloneqq \{\mathbf{x} \in \mathcal{M}_S : \mathbf{b}_n \cdot \mathbf{x} = d\}.$$

$\mathbf{z}$ is well-defined if $d \in [\check{d}, \hat{d}]$.

We argue that n is the follower's response in an SSE in the game $(u^L, \tilde{u}^F_d)$ if and only if $d \geq d^*$; and it is the only possible response if and only if $d > d^*$. This will enable us to use binary search to compute d^* (or find out that $d^* < \check{d}$ or $d^* > \hat{d}$, in which case we only need $\check{d}$ and $\hat{d}$ to compute $\mathbf{b}_n$ and c_n as defined above).

Denote the BR-correspondences of $\tilde{u}^F_d$ and $\tilde{u}^F$ as f_d and f, respectively. Note the following facts:

(i) $f_d(\mathbf{z}) = f(\mathbf{z}) \cup \{n\}$.

(ii) $f_d^{-1}(j) \subseteq f^{-1}(j)$ for all $j \in [n-1]$.

(iii) $\max_{\mathbf{x} \in f_d^{-1}(n)} \mathbf{b}_n \cdot \mathbf{x} = d$ if $d \in [\check{d}, \hat{d}]$.

(iv) $u^L(\mathbf{x}, j) = M_S$ for all $\mathbf{x} \in \mathcal{M}_S$ and $j \in f(\mathbf{x})$.

Indeed, (i) can be verified by comparing $\tilde{u}^F_d(\mathbf{z}, j)$, $j \in [n-1]$, with $\tilde{u}^F(\mathbf{z}, n)$. (ii) is due to the fact that $f^{-1}(n) = \emptyset$ according to the property of $\tilde{u}^F$ as a base function.

To see (iii), note that $n \in f_d(\mathbf{z})$ implies that

$$\max_{\mathbf{x} \in f_d^{-1}(n)} \mathbf{b}_n \cdot \mathbf{x} \geq \mathbf{b}_n \cdot \mathbf{z} = d.$$

Moreover, for all $\mathbf{x} \in f_d^{-1}(n)$,

$$\tilde{u}^F_d(\mathbf{x}, n) \geq \max_{j \in [n-1]} \tilde{u}^F_d(\mathbf{x}, j) = \max_{j \in [n-1]} \tilde{u}^F(\mathbf{x}, j) \geq \tilde{u}^F(\mathbf{x}, k),$$

which implies $\mathbf{b}_n \cdot \mathbf{x} \leq d$ according to Eq. (31). Hence, $\max_{\mathbf{x} \in f_d^{-1}} \mathbf{b}_n \cdot \mathbf{x} = d$.

Finally, if (iv) did not hold, then $u^L(\mathbf{x}, j) \neq M_S$ for some $\mathbf{x} \in \mathcal{M}_S$ and $j \in f(\mathbf{x})$. Since $\tilde{u}^F$ is a base function, by definition, $j \in f(\mathbf{x}) \subseteq S$. Moreover, $\mathbf{x} \in \mathcal{M}_S$ means $\min_{k \in S} u^L(\mathbf{x}, k) = M_S$, so it must be that $u^L(\mathbf{x}, j) > M_S$, which contradicts the definition of a base function.

Hence, (i)–(iv) hold, and we proceed as follows. For each $j \in [n]$, define

$$V^L_d(j) \coloneqq \max_{\mathbf{x} \in f_d^{-1}(j)} u^L(\mathbf{x}, j),$$

which is the leader's maximum attainable payoff for inducing the follower to respond with j. According to (ii), for all $j \in [n-1]$, we have

$$V^L_d(j) \leq \max_{j' \in [n]} \max_{\mathbf{x} \in f^{-1}(j')} u^L(\mathbf{x}, j') = M_S,$$

where the second transition is due to $\tilde{u}^F$ being a base function. Moreover, pick an arbitrary $j \in f(\mathbf{z})$; by (i), (iv), and the fact that $\mathbf{z} \in \mathcal{M}_S$, we get that

$$\max_{j' \in [n-1]} V_d^L(j') \geq u^L(\mathbf{x}, j) = M_S.$$

Therefore, $\max_{j' \in [n-1]} V_d^L(j') = M_S$. By (iii), we also have $V_d^L(n) = \gamma_n \cdot d + \beta_n$ if $d \in [\check{d}, \hat{d}]$. So,

$$\Phi(d) \coloneqq V_d^L(n) - \max_{j \in [n-1]} V_d^L(j) = \gamma_n \cdot d + \beta_n - M_S$$

is continuous and strictly increasing w.r.t. d. We can then use binary search and oracle $\mathcal{A}_{\mathrm{SSE}}$ to pin down d^* (or decide if $d^* < \check{d}$ or $d^* > \hat{d}$): n is an SSE response of $(u^L, \tilde{u}_d^F)$ if and only if $d \geq d^*$; meanwhile, there exists an SSE response $j \in [n-1]$ if and only if $d \leq d^*$. $\square$

This completes the proof of Proposition 6.8. $\square$

C Omitted Proofs in Section 7.3

C.1 Proof of Lemma 7.1

Lemma 7.1. $V_{d^*}^L(1) = V_{d^*}^L(2) = M_{\{1,2\}}$. *Moreover,* $1 \in \mathrm{ER}(\mathcal{G}_d)$ *if and only if* $d \leq d^*$*, and* $2 \in \mathrm{ER}(\mathcal{G}_d)$ *if and only if* $d \geq d^*$*.*

Proof. By definition, we have $1 \in \mathrm{ER}(\mathcal{G}_d) \iff V_d^L(1) \geq V_d^L(2)$ and $2 \in \mathrm{ER}(\mathcal{G}_d) \iff V_d^L(2) \geq V_d^L(1)$. Moreover, it can be verified that $V_d^L(2)$ increases strictly with d (in the range where $\emptyset \neq f_d^{-1}(2) \subsetneq \Delta_m$), while $V_d^L(1)$ is non-increasing. So, it suffices to prove the first part of the lemma: $V_{d^*}^L(1) = V_{d^*}^L(2) = M_{\{1,2\}}$. Given Lemma C.1, we have $V_{d^*}^L(j) \leq u^L(s^*) = M_{\{1,2\}}$ for both $j \in \{1, 2\}$, so it further reduces to proving that $V_{d^*}^L(j) \geq M_{\{1,2\}}$.

Indeed, the existence of $\mathbf{y} \in \Delta_m$ such that $u^L(\mathbf{y}, 2) = M_{\{1,2\}}$ demonstrated in Lemma C.2 readily implies that $V_{d^*}^L(2) \geq M_{\{1,2\}}$. Now consider $V_{d^*}^L(1)$. Suppose for a contradiction that $V_{d^*}^L(1) < M_{\{1,2\}}$. Pick any $\mathbf{x} \in \Delta_m$ such that $\mathbf{a}_2 \cdot \mathbf{x} = d^*$. We would get $\mathbf{x} \in f_{d^*}^{-1}(1)$ and hence $u^L(\mathbf{x}, 1) < M_{\{1,2\}}$. Pick another strategy $\mathbf{y} \in \Delta_m$ such that $\mathbf{a}_2 \cdot \mathbf{y} < d^*$ (which exists as argued above), and consider $\mathbf{z} = \lambda \mathbf{y} + (1 - \lambda)\mathbf{x}$ for a number $\lambda \in (0, 1)$. It holds that $\mathbf{a}_2 \cdot \mathbf{z} < d^*$ for all $\lambda > 0$, in which case $u^L(\mathbf{x}, 2) < \gamma_2 \cdot d^* + \beta_2 = M_{\{1,2\}}$. Moreover, by continuity, when λ is sufficiently small, it holds that $u^L(\mathbf{z}, 1) < M_{\{1,2\}}$. As a result, $\max_{j \in \{1,2\}} u^L(\mathbf{z}, j) < M_{\{1,2\}}$, which contradicts the definition of $M_{\{1,2\}}$.

Now we prove Lemmas C.1 and C.2 to complete the proof.

Lemma C.1. $u^L(s^*) = M_{\{1,2\}}$ *for every* $s^* \in \mathrm{SSE}(\mathcal{G}_{d^*})$*.*

Proof. Consider the following two cases: 1) there exists $\mathbf{x} \in \Delta_m$ such that $\mathbf{a}_2 \cdot \mathbf{x} > d^*$, and 2) there does not exist such an $\mathbf{x}$. We show that in both cases

$$V_d^L(j) \leq M_{\{1,2\}} \tag{32}$$

for both $j \in \{1, 2\}$ to complete the proof. Indeed, this will imply that $M_{\{1,2\}}$ is an upper bound of the leader's SSE payoff: $u^L(s^*) \equiv \max_{j \in \{1,2\}} V_d^L(j) \leq M_{\{1,2\}}$ for any $s^* \in \mathrm{SSE}(\mathcal{G}_{d^*})$. Meanwhile, $M_{\{1,2\}}$ is also naturally a lower bound of the SSE payoff because:

$$M_{\{1,2\}} \equiv \max_{\mathbf{x} \in \Delta_m} \min_{j \in \{1,2\}} u^L(\mathbf{x}, j) \leq \max_{\mathbf{x} \in \Delta_m} \max_{j \in f_{d^*}(\mathbf{x})} u^L(\mathbf{x}, j) \equiv u^L(s^*).$$

It then follows that $u^L(s^*) = M_{\{1,2\}}$.

Note that Eq. (32) holds automatically for $j = 2$: by definition $\mathbf{a}_2 \cdot \mathbf{x} \leq d^*$ for all $\mathbf{x} \in f_{d^*}^{-1}(2)$, which means that $u^L(\mathbf{x}, 2) \leq \gamma_2 \cdot d^* + \beta_2 = M_{\{1,2\}}$. Hence, it remains to argue that Eq. (32) holds for $j = 1$.

- In Case 1, it must be that $u^L(\mathbf{x},1) \le M_{\{1,2\}}$ for all $\mathbf{x} \in \Delta_m$ such that $\mathbf{a}_2 \cdot \mathbf{x} > d^*$: otherwise, we would find an $\mathbf{x} \in \Delta_m$ such that $u^L(\mathbf{x},1) > M_{\{1,2\}}$ and $u^L(\mathbf{x},2) > M_{\{1,2\}}$ (with the latter implied by $\mathbf{a}_2 \cdot \mathbf{x} > d^*$), This leads to the following contradiction:

$$M_{\{1,2\}} \equiv \max_{\mathbf{x}' \in \Delta_m} \min_{j \in \{1,2\}} u^L(\mathbf{x}', j) \ge \min_{j \in \{1,2\}} u^L(\mathbf{x}, j) > M_{\{1,2\}}.$$

  Since $f_d^{-1}(1)$ is the closure of $\{\mathbf{x} \in \Delta_m : \mathbf{a}_2 \cdot \mathbf{x} > d^*\}$, by continuity, we get that $u^L(\mathbf{x},1) \le M_{\{1,2\}}$ for all $\mathbf{x} \in f_d^{-1}(1)$. This implies that $u^L(s^*) \le M_{\{1,2\}}$ for every $s^* \in \mathrm{SSE}(\mathcal{G}_{d^*})$.

- In Case 2, suppose for a contradiction that Eq. (32) does not hold for $j = 1$. The following can be established:

$$M_{\{1\}} \equiv \max_{\mathbf{x} \in \Delta_m} u^L(\mathbf{x},1) \ge V_d^L(1) > M_{\{1,2\}} \ge V_d^L(2) = \max_{\mathbf{x} \in \Delta_m} u^L(\mathbf{x},2) \equiv M_{\{2\}},$$

  where we used the fact $V_d^L(2) \le M_{\{1,2\}}$ as we argued above and the fact that $f_d^{-1}(2) = \Delta_m$ in Case 2. This contradicts the assumption we made in this section that $M_{\{1\}} \le M_{\{2\}}$. □

Lemma C.2. *There exist $\mathbf{x}, \mathbf{y} \in \Delta_m$ such that $\mathbf{a}_2 \cdot \mathbf{x} < d^*$ and $\mathbf{a}_2 \cdot \mathbf{y} = d^*$ (equivalently, $u^L(\mathbf{x},2) < M_{\{1,2\}}$ and $u^L(\mathbf{y},2) = M_{\{1,2\}}$).*

Proof. First, we analyze the existence of $\mathbf{x}$ in the statement. Suppose for a contradiction that $\mathbf{a}_2 \cdot \mathbf{x} \ge d^*$ for all $\mathbf{x} \in \Delta_m$. Hence, $f_{d^*}^{-1}(1) = \Delta_m$ and $V_{d^*}^L(1) = \max_{\mathbf{x} \in \Delta_m} u^L(\mathbf{x},1)$. By definition, $V_{d^*}^L(1)$ is at most the leader's SSE payoff, so it follows by Lemma C.1 that

$$\max_{\mathbf{x} \in \Delta_m} u^L(\mathbf{x},1) \le M_{\{1,2\}}. \tag{33}$$

Note that $\mathbf{a}_2 \cdot \mathbf{x} \ge d^*$ implies that $u^L(\mathbf{x},2) \ge \gamma_2 \cdot d^* + \beta_2 = M_{\{1,2\}}$. Now that $\mathbf{a}_2 \cdot \mathbf{x} \ge d^*$ holds for all $\mathbf{x} \in \Delta_m$, so does $u^L(\mathbf{x},2) \ge M_{\{1,2\}}$; hence, $\min_{\mathbf{x} \in \Delta_m} u^L(\mathbf{x},2) \ge M_{\{1,2\}}$. Together with Eq. (33), this gives $\max_{\mathbf{x} \in \Delta_m} u^L(\mathbf{x},1) \le \min_{\mathbf{x} \in \Delta_m} u^L(\mathbf{x},2)$. If this were true, then, contrary to our assumption, there would be no edge between actions 1 and 2 in the non-dominance graph.

Next, we analyze the existence of $\mathbf{y}$. Indeed, given the existence of $\mathbf{x} \in \Delta_m$ with $\mathbf{a}_2 \cdot \mathbf{x} < d^*$, if it were the case where $\mathbf{a}_2 \cdot \mathbf{y} \ne d^*$ for all $\mathbf{y} \in \Delta_m$, then we would have $\mathbf{a}_2 \cdot \mathbf{x} < d^*$ for all $\mathbf{x} \in \Delta_m$. This would imply that $f_{d^*}^{-1}(2) = \emptyset$ and hence the following contradiction:

$$M_{\{1,2\}} \equiv \max_{\mathbf{x} \in \Delta_m} \min_{j \in \{1,2\}} u^L(\mathbf{x}, j) \le \max_{\mathbf{x} \in \Delta_m} u^L(\mathbf{x},2) = \max_{\mathbf{x} \in f_{d^*}^{-1}(1)} u^L(\mathbf{x},2) < \gamma_2 \cdot d^* + \beta_2 = M_{\{1,2\}}. \square$$

□

D Omitted Proofs in Section 7.4

To simplify the notation, we sometimes omit the dependencies on d_1 and d_2, and just write P_0, P_1, and P_2 when the parameters are clear from the context.) We first prove the following lemmas that are used throughout the section.

Lemma D.1. *Suppose that $d_1 \le d_1^*$, and $d_2 \le d_2^*$. For each $j \in \{1,2\}$, if $P_j(d_1,d_2) \ne \emptyset$, then $V_{d_1,d_2}^L(j) = \gamma_j \cdot d_j + \beta_j$.*

Proof. We prove the equivalent statement: $\max_{\mathbf{x} \in P_j(d_1,d_2)} \mathbf{a}_j \cdot \mathbf{x} = d_j$ if $P_j(d_1,d_2) \ne \emptyset$. We present the proof with $j = 1$; the case where $j = 2$ can be replicated analogously. By definition, $\mathbf{a}_1 \cdot \mathbf{x} \le d_1$ for all $\mathbf{x} \in P_1(d_1,d_2)$, so it suffices to show $\max_{\mathbf{x} \in P_1(d_1,d_2)} \mathbf{a}_1 \cdot \mathbf{x} \ge d_1$.

Choose arbitrary $\mathbf{x} \in P_1(d_1,d_2)$. By definition,

$$\mathbf{a}_1 \cdot \mathbf{x} \le d_1, \quad \text{and} \quad \mathbf{a}_2 \cdot \mathbf{x} \ge d_2.$$

By Lemma 7.1, there exists $\mathbf{y} \in \Delta_m$ such that

$$\mathbf{a}_1 \cdot \mathbf{y} = d_1^* \geq d_1, \quad \text{and} \quad \mathbf{a}_2 \cdot \mathbf{y} \geq d_2^* \geq d_2.$$

Hence, there exists a convex combination $\mathbf{z}$ of $\mathbf{x}$ and $\mathbf{y}$, such that $\mathbf{a}_1 \cdot \mathbf{z} = d_1$ and $\mathbf{a}_2 \cdot \mathbf{z} \geq d_2$, which means $\mathbf{z} \in P_1(d_1, d_2)$. Hence, $\max_{\mathbf{x} \in P_1(d_1,d_2)} \mathbf{a}_1 \cdot \mathbf{x} \geq \mathbf{a}_1 \cdot \mathbf{z} = d_1$. □

Lemma D.2. *There exist $\mathbf{x}_1, \mathbf{x}_2 \in \Delta_m$ such that $u^L(\mathbf{x}_1, 1) = u^L(\mathbf{x}_2, 2) < M_{\{1,2\}}$.*

Proof. We first show that there exists $\mathbf{x}_1 \in \Delta_m$ such that $u^L(\mathbf{x}_1, 1) < M_{\{1,2\}}$. Suppose for a contradiction that

$$u^L(\mathbf{x}, 1) \geq M_{\{1,2\}} \quad \text{for all } \mathbf{x} \in \Delta_m \tag{34}$$

Consider the following two cases: 1) $u^L(\mathbf{x}, 1) \geq u^L(\mathbf{x}, 2)$ for all $\mathbf{x} \in \Delta_m$, and 2) $u^L(\mathbf{x}', 1) < u^L(\mathbf{x}', 2)$ for some $\mathbf{x}' \in \Delta_m$.

In Case 1, by definition we get that $M_{\{1,2\}} = \max_{\mathbf{x} \in \Delta_m} u^L(\mathbf{x}, 2)$. Hence, Eq. (34) implies that $\min_{\mathbf{x} \in \Delta_m} u^L(\mathbf{x}, 1) \geq M_{\{1,2\}} = \max_{\mathbf{x} \in \Delta_m} u^L(\mathbf{x}, 2)$. So, action 1 obviously dominates action 2, but this contradicts the fact that there is an edge between 1 and 2 in the non-dominance graph.

Now consider Case 2. By definition, $M_{\{1,2\}} \geq \min_{j \in \{1,2\}} u^L(\mathbf{x}', j) = u^L(\mathbf{x}', 1)$. Given Eq. (34), it must be that $u^L(\mathbf{x}', 1) = M_{\{1,2\}}$. Recall that $\mathbf{a}_1 \neq \mathbf{0}$, so there exists $\mathbf{y}' \in \Delta_m$ such that $u^L(\mathbf{y}', 1) > u^L(\mathbf{x}', 1) = M_{\{1,2\}}$ (note that $u^L(\mathbf{y}', 1) < u^L(\mathbf{x}', 1)$ is not possible because of Eq. (34)).

Consider a convex combination of $\mathbf{x}'$ and $\mathbf{y}'$, that is, $\mathbf{z} = \lambda \mathbf{x}' + (1 - \lambda)\mathbf{y}'$ for $\lambda \in (0, 1)$. Now that $u^L(\mathbf{y}', 1) > M_{\{1,2\}}$ and $u^L(\mathbf{x}', 1) = M_{\{1,2\}}$, it holds automatically that

$$u^L(\mathbf{z}, 1) > M_{\{1,2\}}.$$

Moreover, by assumption we have $u^L(\mathbf{x}', 1) < u^L(\mathbf{x}', 2)$, so by continuity when λ is sufficiently close to 1, the same must also hold for $\mathbf{z}$, i.e.,

$$u^L(\mathbf{z}, 1) < u^L(\mathbf{z}, 2).$$

Consequently, the following contradiction arises: $M_{\{1,2\}} \geq \min_{j \in \{1,2\}} u^L(\mathbf{z}, j) = u^L(\mathbf{z}, 1) > M_{\{1,2\}}$. We then conclude that there exists $\mathbf{x}_1 \in \Delta_m$ such that $u^L(\mathbf{x}_1, 1) < M_{\{1,2\}}$.

Replicating the same argument, we can prove that there also exists $\mathbf{x}_2 \in \Delta_m$ such that $u^L(\mathbf{x}_2, 1) < M_{\{1,2\}}$ (note that the argument only relies on conditions that hold for both actions 1 and 2). So it remains to prove that $u^L(\mathbf{x}_1, 1) = u^L(\mathbf{x}_2, 2)$ can also be ensured.

Indeed, recall that it holds for the first reference pair we found in Section 7.3, say $\mathbf{x}_1^*$ and $\mathbf{x}_2^*$, that $u^L(\mathbf{x}_1^*, 1) = u^L(\mathbf{x}_2^*, 2) = M_{\{1,2\}}$. Suppose w.l.o.g. that $u^L(\mathbf{x}_1, 1) \leq u^L(\mathbf{x}_2, 2)$. Then by continuity there must be a point $\mathbf{x}_1' \in \Delta_m$ in the line segment between $\mathbf{x}_1$ and $\mathbf{x}_1^*$ such that $u^L(\mathbf{x}_1', 1) = u^L(\mathbf{x}_2, 2)$. This completes the proof. □

D.1 Proof of Lemma 7.4

Lemma 7.4. *There exists $d_2' < d_2^*$ such that for all $d_2 \in [d_2', d_2^*]$, $\{1, 2\} \subseteq \mathrm{ER}(\mathcal{G}_{d_1,d_2})$ for some $d_1 < d_1^*$.*

We first prove the following result, and we will argue that it is sufficient for proving Lemma 7.4.

Lemma D.3. *There exist $\mathbf{x}_1, \mathbf{x}_2 \in \Delta_m$ such that $u^L(\mathbf{x}_1, 1) = u^L(\mathbf{x}_2, 2) < M_{\{1,2\}}$ and $\mathbf{a}_2 \cdot \mathbf{x}_1 \geq \mathbf{a}_2 \cdot \mathbf{x}_2$.*

Proof. Pick two points $\mathbf{x}_1, \mathbf{x}_2 \in \Delta_m$ such that $u^L(\mathbf{x}_1, 1) = u^L(\mathbf{x}_2, 2) < M_{\{1,2\}}$, which always exist according to Lemma D.2. If it holds in addition that $\mathbf{a}_2 \cdot \mathbf{x}_1 \geq \mathbf{a}_2 \cdot \mathbf{x}_2$ then we are done. So in what follows we assume that $\mathbf{a}_2 \cdot \mathbf{x}_1 < \mathbf{a}_2 \cdot \mathbf{x}_2$. Equivalently, this means $u^L(\mathbf{x}_1, 2) < u^L(\mathbf{x}_2, 2)$, so we can establish

$$u^L(\mathbf{x}_1, 2) < u^L(\mathbf{x}_2, 2) = u^L(\mathbf{x}_1, 1).$$

By assumption, action 1 does not strongly dominate action 2. Now that $u^L(\mathbf{x}_1, 2) < u^L(\mathbf{x}_1, 1)$, according to the definition of the strong dominance there must be a point $\mathbf{y} \in \Delta_m$ such that $u^L(\mathbf{y}, 1) < u^L(\mathbf{y}, 2)$. This further means $u^L(\mathbf{y}, 1) \leq M_{\{1,2\}}$ (otherwise, the following contradiction would arise $M_{\{1,2\}} \geq \min_{j \in \{1,2\}} u^L(\mathbf{y}, j) = u^L(\mathbf{y}, 1) > M_{\{1,2\}}$).

In summary, we have argued that the following conditions hold:

$$\begin{aligned} u^L(\mathbf{y},\ 1) < u^L(\mathbf{y},\ 2) \quad &\text{and} \quad u^L(\mathbf{y},\ 1) \leq M_{\{1,2\}}; \\ u^L(\mathbf{x}_1, 1) > u^L(\mathbf{x}_1, 2) \quad &\text{and} \quad u^L(\mathbf{x}_1, 1) < M_{\{1,2\}}. \end{aligned}$$

So, by continuity, there must be a point $\mathbf{z} \in \Delta_m$ that lies between $\mathbf{x}_1$ and $\mathbf{y}$ such that

$$u^L(\mathbf{z}, 1) = u^L(\mathbf{z}, 2) < M_{\{1,2\}}.$$

Trivially, $\mathbf{a}_2 \cdot \mathbf{z} \geq \mathbf{a}_2 \cdot \mathbf{z}$, so the pair $(\mathbf{z}, \mathbf{z})$ satisfy the condition stated in the lemma. □

Proof of Lemma 7.4. By Lemma D.3, there exist $\mathbf{x}_1, \mathbf{x}_2 \in \Delta_m$ such that $u^L(\mathbf{x}_1, 1) = u^L(\mathbf{x}_2, 2) < M_{\{1,2\}}$ and $\mathbf{a}_2 \cdot \mathbf{x}_1 \geq \mathbf{a}_2 \cdot \mathbf{x}_2$. Letting $d_2' = \mathbf{a}_2 \cdot \mathbf{x}_2$ gives $d_2' < d_2^*$ as we analyzed in Section 7.4.1. Hence, for any arbitrary $d_2 \in [d_2', d_2^*]$, there is a unique number $\xi \in [0, 1]$ such that $d_2 = \xi \cdot d_2' + (1 - \xi) \cdot d_2^*$. Let

$$d_1 = \xi \cdot d_1' + (1 - \xi) \cdot d_1^* \quad \text{and} \quad \mathbf{y}_j = \xi \cdot \mathbf{x}_j + (1 - \xi) \cdot \mathbf{x}_j^*,$$

where we let $\mathbf{x}_1^*$ and $\mathbf{x}_2^*$ be the first reference pair we found in Section 7.3. By noting that $(\mathbf{y}_1, \mathbf{y}_2)$ is a convex combination of $(\mathbf{x}_1, \mathbf{x}_2)$ and $(\mathbf{x}_1^*, \mathbf{x}_2^*)$, one can verify that $\mathbf{y}_j \in P_j(d_1, d_2)$ for all $j \in \{1, 2\}$. Moreover, according to Lemma D.1, we have $V^L_{d_1,d_2}(1) = V^L_{d_1,d_2}(2)$. If these two values also outperform the maximum attainable payoffs for inducing other responses of the follower, i.e., $V^L_{d_1,d_2}(1) \geq V^L_{d_1,d_2}(k)$ for all $k \in [n] \setminus \{1, 2\}$, then it will follow immediately that $\{1, 2\} \subseteq \mathrm{ER}(\mathcal{G}_{d_1,d_2})$ and Lemma 7.4 will hold given the arbitrary choice of d_2.

Indeed, this can be ensured as follows. By construction, $V^L_{d_1,d_2}(1) \geq V^L_{d_1',d_2'}(1)$ and $V^L_{d_1,d_2}(k) \leq V^L_{d_1',d_2'}(k)$, so it suffices to ensure that $V^L_{d_1',d_2'}(1) \geq \max_{k \in [n] \setminus \{1,2\}} V^L_{d_1',d_2'}(k)$. This holds strictly if d_1' and d_2' are sufficiently close to d_1^* and d_2^* because the values on the two sides of the inequality change continuously with d_1' and d_2' and, given that h is a cover of $\{1, 2\}$, we have

$$V^L_{d_1^*,d_2^*}(1) = M_{\{1,2\}} > \max_{\mathbf{x} \in \mathcal{M}_{\{1,2\}}} \max_{k \in h^{-1}(\mathbf{x})} u^L(\mathbf{x}, k) = \max_{k \in [n] \setminus \{1,2\}} V^L_{d_1^*,d_2^*}(k).$$

Therefore, if $V^L_{d_1',d_2'}(1) \geq \max_{k \in [n] \setminus \{1,2\}} V^L_{d_1',d_2'}(k)$ does not hold, we can substitute each $\mathbf{x}_j$ with $\xi' \cdot \mathbf{x}_j + (1 - \xi') \cdot \mathbf{x}_j^*$ for some ξ' sufficiently close to 1 and repeat the argument above, whereby all the initial conditions still hold and, additionally, $V^L_{d_1',d_2'}(1) \geq \max_{k \in [n] \setminus \{1,2\}} V^L_{d_1',d_2'}(k)$. The proof is then complete. □

D.2 Weaker Version of Proposition 7.5

Proposition D.4. *Assume that $\mathcal{A}_{\mathrm{SSE}}$ can handle BR-correspondence queries. A pair of strategies $\mathbf{x}, \mathbf{y} \in \Delta_m$ such that $u^L(\mathbf{x}, 1) = u^L(\mathbf{y}, 2) < M_{\{1,2\}}$ can be computed in polynomial time.*

Proof. Given Lemma 7.4, we use a nested binary search to find two values $d_1 < d_1^*$ and $d_2 < d_2^*$ such that $\{1, 2\} \subseteq \mathrm{ER}(\mathcal{G}_{d_1,d_2})$. Once these values are found, any arbitrary $\mathbf{x} \in \operatorname{argmax}_{\mathbf{x}' \in P_1(d_1,d_2)} \mathbf{a}_1 \cdot \mathbf{x}'$ and $\mathbf{y} \in \operatorname{argmax}_{\mathbf{y}' \in P_2(d_1,d_2)} \mathbf{a}_1 \cdot \mathbf{y}'$ then form a reference pair such that $u^L(\mathbf{x}, 1) = u^L(\mathbf{y}, 2)$. Moreover, $u^L(\mathbf{x}, 1) < M_{\{1,2\}}$ because $d_1 < d_1^*$.

We start by letting $d_2 = \min_{\mathbf{x} \in \Delta_m} \mathbf{a}_2 \cdot \mathbf{x}$, which is the smallest possible value of d_2 (any value smaller than this would lead to an empty P_2). Using binary search, we aim to find in the range $(-\infty, d_1^*)$ a value of d_1 such that $\{1, 2\} \subseteq \mathrm{ER}(\mathcal{G}_{d_1,d_2})$. For every value of d_1 we try, we call the ER oracle $\mathcal{A}_{\mathrm{ER}}$ to check if actions 1 and 2 are in $\mathrm{ER}(\mathcal{G}_{d_1,d_2})$. Recall that, by construction, $V^L_{d_1,d_2}(2)$ do not change with d_1, whereas $V^L_{d_1,d_2}(1)$ increases with d_1 (when $P_1(d_1, d_2) \neq \emptyset$) and $V^L_{d_1,d_2}(k)$ is non-increasing w.r.t. d_1. Hence, for any d_2, if there is either no or a unique satisfying d_1 (that makes $\{1, 2\} \subseteq \mathrm{ER}(\mathcal{G}_{d_1,d_2})$). The unique point is also the smallest one that makes $1 \in \mathrm{ER}(\mathcal{G}_{d_1,d_2})$.

Consequently, if $1 \notin \mathrm{ER}(\mathcal{G}_{d_1,d_2})$ (resp. $1 \in \mathrm{ER}(\mathcal{G}_{d_1,d_2})$), we know that if a satisfying value of d_1 exists, it must be larger (resp. smaller) than the current value. If the smallest d_1 with $1 \in \mathrm{ER}(\mathcal{G}_{d_1,d_2})$ also gives $2 \in \mathrm{ER}(\mathcal{G}_{d_1,d_2})$, then we are done. Otherwise, we can conclude that no satisfying d_1 exists, in which case we decrease d_2 by moving it to $(d_2 + d_2^*)/2$. By Lemma 7.4, when d_2 is sufficiently close to d_2^*, a satisfying d_1 always exists. It then remains to argue that the nested binary search process always terminates in polynomial time.

Time Complexity We show that there exists a d_2 that satisfies the condition stated in Lemma 7.4 and its bit size is bounded by a polynomial in the bit size of u^L. This will then indicate that after trying at most polynomially many values of d_2, the binary search will find a satisfying value (note that any value). We will later also show a similar argument for d_1 to complete the analysis.

We first characterize a value of d_2 that satisfies $V^L_{d_1,d_2}(1) = V^L_{d_1,d_2}(2)$ for some d_1, as a necessary condition of $\{1, 2\} \subseteq \mathrm{ER}(\mathcal{G}_{d_1,d_2})$. This value is characterized by the following LP:

$$
\begin{aligned}
\min_{d_1,d_2,\mathbf{x}_1,\mathbf{x}_2} \quad & d_2 && && (35)\\
\text{subject to} \quad & \mathbf{x}_j \in P_j(d_1, d_2) && \text{for } j \in \{1, 2\} && (35a)\\
& d_j \le d_j^* && \text{for } j \in \{1, 2\} && (35b)\\
& \gamma_1 \cdot d_1 + \beta_1 = \gamma_2 \cdot d_2 + \beta_2 && && (35c)
\end{aligned}
$$

Namely, Eq. (35a) ensures that $P_j(d_1, d_2) \neq \emptyset$, so by Lemma D.1, Eqs. (35b) and (35c) then ensure that $V^L_{d_1,d_2}(1) = V^L_{d_1,d_2}(2)$. According to standard results on linear programming, the bit size of the optimal value of the LP is at most a polynomial in the bit size of its parameters. The bit size of each parameter is also bounded by polynomials in the bit size of u^L (in particular, the bit sizes of γ_j and β_j are bounded given that the bit size of $\mathbf{a}_j$ is bounded). Let the optimal value be $\bar{d}_2$. It must be that $\bar{d}_2 < d_2^*$ according to Lemma 7.4. Moreover, one can verify by using a similar argument to Lemma 7.4 that for any $d_2 \in [\bar{d}_2, d_2^*]$ there exists $d_1 < d_1^*$ such that $V^L_{d_1,d_2}(1) = V^L_{d_1,d_2}(2)$ (though this is not a direct corollary of Lemma 7.4).

Based on $\bar{d}_2$, we further characterize a value of d_2 that satisfy $\{1, 2\} \subseteq \mathrm{ER}(\mathcal{G}_{d_1,d_2})$ for some d_1, by using the following set of LPs. Each of these LPs corresponds to an action $k \in [n] \setminus \{1, 2\}$ and imposes constraints Eqs. (36b) to (36d) in addition to those in LP (35).

$$
\begin{aligned}
\max_{d_1,d_2,\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_k} \quad & d_2 && (36)\\
\text{subject to} \quad & \text{Eqs. (35a) to (35c)} && (36a)\\
& d_2 \ge \bar{d}_2 && (36b)\\
& \mathbf{x}_k \in P_0(d_1, d_2) \cap h^{-1}(k) && (36c)\\
& \gamma_1 \cdot d_1 + \beta_1 \le u^L(\mathbf{x}_k, k) && (36d)
\end{aligned}
$$

It shall be clear shortly how these additional constraints contribute to our analysis. Let d_2^k be the optimal value of the LP corresponding to action k (let $d_2^{(k)} = \min_{\mathbf{x}\in\Delta_m} \mathbf{a}_2 \cdot \mathbf{x}$ if the LP is infeasible). Moreover, let $d_2' = \left(\max_{k\in[n]\setminus\{1,2\}} d_2^k + d_2^*\right)/2$. It must be that $d_2^k < d_2^*$ because one can verify that: 1) any $d_2 > d_2^*$ would lead to $d_1 > d_1^*$ because of Eq. (35c), which further leads to $P_0(d_1, d_2) = \emptyset$; and 2) $d_2 = d_2^*$ gives $\gamma_1 \cdot d_1 + \beta_1 = V^L_{d_1,d_2}(2) > u^L(\mathbf{x}_k, k)$ because h is a cover. Consequently, $d_2' < d_2^*$.

According to the same argument for $\bar{d}_2$, the bit size of d_2' is bounded by a polynomial. We show that d_2' satisfies the condition stated in Lemma 7.4 to finish the proof. Consider any arbitrary value $d_2 > d_2'$. Since $d_2' > d_2^k$, we have $d_2 > d_2^k$, so by definition d_2 is strictly larger than the optimal value of LP (36). This means that if we fix the variable d_2 in LP (36) to this value, no values of the other variables can make the constraints of the LP hold simultaneously. Since $d_2 \in [\bar{d}_2, d_2^*]$, according to our analysis above, Eqs. (36a) and (36b) are satisfiable with some values of d_1, $\mathbf{x}_1$, and $\mathbf{x}_2$. So it must be that Eqs. (36c) and (36d) are not satisfiable. In other words, $\gamma_1 \cdot d_1 + \beta_1 > u^L(\mathbf{x}_k, k)$ for any $\mathbf{x}_k \in P_0(d_1, d_2) \cap h^{-1}(k)$ and any d_1 that satisfies Eqs. (36a) and (36b). Hence, any such d_1 gives $V^L_{d_1,d_2}(1) = V^L_{d_1,d_2}(2) > V^L_{d_1,d_2}(k)$ for all $k \in [n] \setminus \{1, 2\}$. □

D.3 Querying by Using Payoff Matrices

We show how to design payoff matrices to replace the BR-correspondences we used to prove Proposition D.4, thereby proving the stronger result Proposition 7.5. We define the following payoff matrix, which is parameterized by two numbers $d_1 < d_1^*$ and $d_2 < d_2^*$, too. We will show that the BR-correspondence of this payoff matrix functions the same way as f_{d_1,d_2} defined in Eq. (20).

For every $\mathbf{x} \in \Delta_m$ and $j \in [n]$, let

$$\tilde{u}^F_{d_1,d_2}(\mathbf{x}, j) := \begin{cases} \mathbf{b}_j \cdot (\mathbf{x} - \mathbf{z}_j) + c_j, & \text{if } j \in \{1,2\}; \\ \mu(\mathbf{x}, j), & \text{if } j \in [n] \setminus \{1,2\}. \end{cases} \tag{37}$$

where μ is an arbitrary proper cover of $\{1, 2\}$, and $\mathbf{b}_j, c_j \in \mathbb{R}$ and $\mathbf{z}_j \in \Delta_m$ are defined as follows based on d_1 and d_2.[8]

- First, for each $j \in \{1, 2\}$, using $\bar{d}_j := (d_j + d_j^*)/2$, we define:

$$Z_1 := \left\{\mathbf{x} \in \Delta_m : \mathbf{a}_1 \cdot \mathbf{x} = \bar{d}_1 \text{ and } \mathbf{a}_2 \cdot \mathbf{x} \geq \bar{d}_2\right\}, \tag{38}$$

$$\text{and} \quad Z_2 := \left\{\mathbf{x} \in \Delta_m : \mathbf{a}_2 \cdot \mathbf{x} = \bar{d}_2\right\}, \tag{39}$$

- Let $K_1 = [n]\setminus\{1, 2\}$ and $K_2 = [n]\setminus\{2\}$. For each $j \in \{1, 2\}$, we pick arbitrary $\mathbf{z}_j \in Z_j$ and

$$k_j \in \operatorname*{argmax}_{k\in K_j} \tilde{u}^F_{d_1,d_2}(\mathbf{z}_j, k), \tag{40}$$

  and let $c_j = \tilde{u}^F_{d_1,d_2}(\mathbf{z}_j, k_j)$. Note that according to Eq. (37), $\tilde{u}^F_{d_1,d_2}(\mathbf{z}_1, k) = \mu(\mathbf{z}_1, k)$ for all $k \in K_1$, so k_j is well-defined even though the complete definition of $\tilde{u}^F_{d_1,d_2}$ relies on k_j.

- Next, let

$$\mathbf{b}_1 := \mathbf{g}_{k_1} - \frac{6 \cdot W_1}{d_1^* - d_1} \cdot \mathbf{a}_1, \tag{41}$$

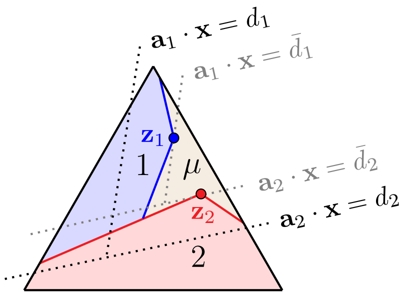

Figure 6: Illustration of $\tilde{f}_{d_1,d_2}$. The region labeled μ is $\bigcup_{k\in[n]\setminus\{1,2\}} \tilde{f}^{-1}_{d_1,d_2}(k)$. The dotted lines are the boundaries of $f^{-1}_{d_1,d_2}(j)$ and $f^{-1}_{\bar{d}_1,\bar{d}_2}(j)$. Recall that $f^{-1}_{d_1,d_2}(j) = P_j(d_1,d_2)$ and $f^{-1}_{\bar{d}_1,\bar{d}_2}(j) = P_j(\bar{d}_1,\bar{d}_2)$.

where we use $W_j := 1 + \max_{i\in[m],k\in K_j} \tilde{u}^F_{d_1,d_2}(i,k) - \min_{i\in[m],k\in K_j} \tilde{u}^F_{d_1,d_2}(i,k)$ for $j \in \{1,2\}$, which is a value larger than the maximum gap between the payoff parameters; and $\mathbf{g}_k$ is the gradient of $\tilde{u}^F_{d_1,d_2}(\cdot,k)$ for each $k \in [n]$:

$$\mathbf{g}_k := \left(\tilde{u}^F_{d_1,d_2}(1,k),\ldots,\tilde{u}^F_{d_1,d_2}(m,k)\right).$$

We extend the notation in Section 7.4.1 to the above defined utility function: let $\tilde{f}_{d_1,d_2}$ be the BR-correspondence of $\tilde{u}^F_{d_1,d_2}$, let $\widetilde{\mathcal{G}}_{d_1,d_2} := (u^L, \tilde{u}^F_{d_1,d_2})$, and let $\widetilde{V}^L_{d_1,d_2}(j) := \max_{\mathbf{x}\in\tilde{f}^{-1}_{d_1,d_2}(j)} u^L(\mathbf{x},j)$. We make several observations about $\tilde{f}_{d_1,d_2}$ in the following lemma, where we also compare it with the BR-correspondence f_{d_1,d_2} defined in Section 7.4.1. Fig. 6 provides an illustration of these observations and the general structure of $\tilde{f}_{d_1,d_2}$. Note that $\bar{d}_j$, Z_j, $\mathbf{z}_j$, and k_j are all functions of d_1 and d_2, but for notational simplicity, we will often omit their dependencies on d_1 and d_2; the exact values of d_1 and d_2 that define them should be clear from the context.

Lemma D.5. *Suppose that $d_1 \le d_1^*$ and $d_2 \le d_2^*$. The following statements hold for every $j \in \{1,2\}$ and $k \in [n]\setminus\{1,2\}$:*

(i) $\bigcup_{j\in\{1,2\}} f^{-1}_{d_1,d_2}(j) \subseteq \bigcup_{j\in\{1,2\}} \tilde{f}^{-1}_{d_1,d_2}(j)$;

(ii) $\tilde{f}^{-1}_{d_1,d_2}(k) \subseteq f^{-1}_{d_1,d_2}(k)$; *and*

(iii) if $Z_j \ne \emptyset$, *then* $\mathbf{z}_j \in \tilde{f}^{-1}_{d_1,d_2}(j)$ *and* $\widetilde{V}^L_{d_1,d_2}(j) = \gamma_j \cdot \bar{d}_j + \beta_j$.

Proof. We first prove the following additional statement:

$$\mathbf{a}_j \cdot \mathbf{x} \le \bar{d}_j \quad \text{for all } \mathbf{x} \in \tilde{f}^{-1}_{d_1,d_2}(j). \tag{42}$$

Suppose for the sake of contradiction that $\mathbf{a}_j \cdot \mathbf{x} > \bar{d}_j$. We argue that $\mathbf{x} \notin \tilde{f}^{-1}_{d_1,d_2}(j)$. Indeed, since $\mathbf{z}_j \in Z_j$, by definition $\mathbf{a}_j \cdot \mathbf{z}_j = \bar{d}_j$, so $\mathbf{a}_j \cdot (\mathbf{x} - \mathbf{z}_j) > 0$. It follows that

$$\begin{aligned}
\tilde{u}^F_{d_1,d_2}(\mathbf{x},j) &= \mathbf{b}_j \cdot (\mathbf{x} - \mathbf{z}_j) + c_j \\
&= \mathbf{g}_{k_j} \cdot (\mathbf{x} - \mathbf{z}_j) - \frac{6 \cdot W_j}{d_j^* - d_j} \cdot \mathbf{a}_j \cdot (\mathbf{x} - \mathbf{z}_j) + c_j \\
&< \mathbf{g}_{k_j} \cdot (\mathbf{x} - \mathbf{z}_j) + c_j \\
&= \tilde{u}^F_{d_1,d_2}(\mathbf{x},k_j) - \tilde{u}^F_{d_1,d_2}(\mathbf{z}_j,k_j) + c_j \\
&= \tilde{u}^F_{d_1,d_2}(\mathbf{x},k_j),
\end{aligned} \tag{43}$$

where we used the definition that $\tilde{u}^F_{d_1,d_2}(\mathbf{z}_j,k_j) = c_j$. Hence, $\mathbf{x} \notin \tilde{f}^{-1}_{d_1,d_2}(j)$, and Eq. (42) follows.

[8] We omit the dependencies of these parameters on d_1 and d_2 in the notation to simplify the presentation.

(i) Pick arbitrary $j \in \{1,2\}$ and $\mathbf{x} \in f_{d_1,d_2}(j) \equiv P_j(d_1,d_2)$. It suffices to argue that $\tilde{u}^F_{d_1,d_2}(\mathbf{x},j) > \tilde{u}^F_{d_1,d_2}(\mathbf{x},k)$ for all $k \in [n] \setminus \{1,2\}$. Indeed, since $\mathbf{x} \in P_j(d_1,d_2)$, by definition $\mathbf{a}_j \cdot \mathbf{x} \le d_j$. Hence,

$$\begin{aligned}\tilde{u}^F_{d_1,d_2}(\mathbf{x},j) &= \mathbf{b}_j \cdot (\mathbf{x} - \mathbf{z}_j) + c_j \\ &= \mathbf{g}_{k_j} \cdot (\mathbf{x} - \mathbf{z}_j) - \frac{4 \cdot W_j}{d^*_j - d_j} \cdot \mathbf{a}_j \cdot (\mathbf{x} - \mathbf{z}_j) + c_j \\ &> -2 \cdot W_j - \frac{6 \cdot W_j}{d^*_j - d_j} \cdot (d_j - \bar{d}_j) + c_j \\ &= W_j + c_j \\ &\ge \tilde{u}^F_{d_1,d_2}(\mathbf{x},k), \end{aligned} \tag{44}$$

where we used the fact that

$$\mathbf{g}_{k_j} \cdot (\mathbf{x} - \mathbf{z}_j) \ge \min_{\mathbf{x}' \in \Delta_m} \mathbf{g}_{k_j} \cdot \mathbf{x}' - \max_{\mathbf{x}' \in \Delta_m} \mathbf{g}_{k_j} \cdot \mathbf{x}' > -W_j.$$

(ii) Consider the BR-correspondence $h : \Delta_m \to 2^{[n]\setminus\{1,2\}}$ of μ. By construction,

$$f^{-1}_{d_1,d_2}(k) = h^{-1}(k) \cap P_0(d_1,d_2) \quad \text{and} \quad \tilde{f}^{-1}_{d_1,d_2}(k) = h^{-1}(k) \cap \widetilde{P}_0(d_1,d_2),$$

where

$$\widetilde{P}_0(d_1,d_2) \coloneqq \left\{ \mathbf{x} \in \Delta_m : \max_{k \in [n]\setminus\{1,2\}} \tilde{u}^F_{d_1,d_2}(\mathbf{x},k) \ge \max_{j \in \{1,2\}} \tilde{u}^F_{d_1,d_2}(\mathbf{x},j) \right\}.$$

We argue that $\widetilde{P}_0(d_1,d_2) \subseteq P_0(d_1,d_2)$ to complete the proof. Indeed,

$$P_0(d_1,d_2) = \mathrm{cl}\left(\Delta_m \setminus \bigcup_{j \in \{1,2\}} P_j(d_1,d_2) \right),$$

where $\mathrm{cl}(\cdot)$ denotes the closure of a set. Since Eqs. (43) and (44) are strictly satisfied, We also have

$$\widetilde{P}_0(d_1,d_2) = \mathrm{cl}\left(\Delta_m \setminus \bigcup_{j \in \{1,2\}} \tilde{f}^{-1}_{d_1,d_2}(j) \right).$$

According to (i) we proved above, $\bigcup_{j\in\{1,2\}} P_j(d_1,d_2) \subseteq \bigcup_{j\in\{1,2\}} \tilde{f}^{-1}_{d_1,d_2}(j)$ (note that $P_j(d_1,d_2) = f_{d_1,d_2}(j)$). Hence, $\widetilde{P}_0(d_1,d_2) \subseteq P_0(d_1,d_2)$, and (ii) follows.

(iii) It suffices to prove that $\mathbf{z}_j \in \tilde{f}^{-1}_{d_1,d_2}(j)$ for the following reasons. First, since $\mathbf{a}_j \cdot \mathbf{z}_j = \bar{d}_j$, $\mathbf{z}_j \in \tilde{f}^{-1}_{d_1,d_2}(j)$ implies that $\widetilde{V}^L_{d_1,d_2}(j) \ge u^L(\mathbf{z}_j,j) = \gamma_j \cdot \bar{d}_j + \beta_j$. Second, according to Eq. (42) we proved above, $\widetilde{V}^L_{d_1,d_2}(j) \le \gamma_j \cdot \bar{d}_j + \beta_j$. Hence, $\widetilde{V}^L_{d_1,d_2}(j) = \gamma_j \cdot \bar{d}_j + \beta_j$.

Next, we prove $\mathbf{z}_j \in \tilde{f}^{-1}_{d_1,d_2}(j)$ to complete the proof. According to Eq. (37),

$$\tilde{u}^F_{d_1,d_2}(\mathbf{z}_j,j) = c_j = \tilde{u}^F_{d_1,d_2}(\mathbf{z}_j,k_j) = \max_{k \in K_j} \tilde{u}^F_{d_1,d_2}(\mathbf{z}_j,k). \tag{45}$$

Since $K_2 = [n] \setminus \{1\}$, this immediately implies that $\tilde{u}^F_{d_1,d_2}(\mathbf{z}_2,2) \ge \tilde{u}^F_{d_1,d_2}(\mathbf{z}_2,k)$ for all $k \in [n]$. So, $\mathbf{z}_2 \in \tilde{f}^{-1}_{d_1,d_2}(2)$.

To argue that $\mathbf{z}_1 \in \tilde{f}^{-1}_{d_1,d_2}(1)$, since $K_1 = [n] \setminus \{1,2\}$, we need to prove in addition that $\tilde{u}^F_{d_1,d_2}(\mathbf{z}_1,1) \ge \tilde{u}^F_{d_1,d_2}(\mathbf{z}_1,2)$. Indeed, since $\mathbf{z}_2 \in Z_2$, by definition $\mathbf{a}_2 \cdot \mathbf{z}_1 \ge \bar{d}_2$. So, applying Eq. (43) gives $\tilde{u}^F_{d_1,d_2}(\mathbf{z}_1,2) \le \tilde{u}^F_{d_1,d_2}(\mathbf{z}_1,k_2)$. Since $k_2 \in K_1 \cup \{1\}$, Eq. (45) then implies

$$\tilde{u}^F_{d_1,d_2}(\mathbf{z}_1,1) \ge \tilde{u}^F_{d_1,d_2}(\mathbf{z}_1,k_2) \ge \tilde{u}^F_{d_1,d_2}(\mathbf{z}_1,2)$$

Consequently, $\tilde{u}^F_{d_1,d_2}(\mathbf{z}_1,1) \ge \tilde{u}^F_{d_1,d_2}(\mathbf{z}_1,k)$ for all $k \in [n]$, so $\mathbf{z}_1 \in \tilde{f}^{-1}_{d_1,d_2}(1)$. □

With the above observations, it becomes clear that the values of d_1 and d_2 that result in $\{1,2\} \subseteq \mathrm{ER}(\mathcal{G}_{d_1,d_2})$ (when we derive the weaker result) also lead to $\{1,2\} \subseteq \mathrm{ER}(\widetilde{\mathcal{G}}_{d_1,d_2})$. We can then easily generalize the proof of Proposition D.4 to prove Proposition 7.5.

Proposition 7.5. $\mathbf{x}, \mathbf{y} \in \Delta_m$ *satisfying Eq. (**) can be computed in polynomial time.*

Proof Sketch. Similarly to the approach to proving Proposition D.4, we use a nested binary search to find two values $d_1 < d_1^*$ and $d_2 < d_2^*$ such that $\{1,2\} \subseteq \mathrm{ER}(\widetilde{\mathcal{G}}_{d_1,d_2})$. This will yield a reference pair such that $u^L(\mathbf{x}, 1) = u^L(\mathbf{y}, 2) < M_{\{1,2\}}$.

The key is to ensure that there indeed exist the desired d_1 and d_2, and their bit size is bounded polynomially. We argue that $\{1,2\} \subseteq \mathrm{ER}(\mathcal{G}_{d_1,d_2}) \implies \{1,2\} \subseteq \mathrm{ER}(\widetilde{\mathcal{G}}_{d_1,d_2})$, so Lemma 7.4 readily implies the desired result.

Suppose that $\{1,2\} \subseteq \mathrm{ER}(\mathcal{G}_{d_1,d_2})$. Lemma D.5 (ii) implies that the best response regions of all actions $k \notin \{1,2\}$ are smaller compared with $f^{-1}_{d_1,d_2}(k)$, which means $\widetilde{V}^L_{d_1,d_2}(k) \le V^L_{d_1,d_2}(k)$. We then argue the following to complete the proof:

$$\widetilde{V}^L_{d_1,d_2}(1) = \widetilde{V}^L_{d_1,d_2}(2) \ge V^L_{d_1,d_2}(1).$$

Indeed, now that $1 \in \mathrm{ER}(\mathcal{G}_{d_1,d_2})$, it must be that $V^L_{d_1,d_2}(1) \ge V^L_{d_1,d_2}(k)$ for all $k \in [n]$. So once the above condition holds, we can establish $\widetilde{V}^L_{d_1,d_2}(1) = \widetilde{V}^L_{d_1,d_2}(2) \ge \widetilde{V}^L_{d_1,d_2}(k)$ for all $k \in [n]$, which means $\{1,2\} \subseteq \mathrm{ER}(\widetilde{\mathcal{G}}_{d_1,d_2})$.

Indeed, applying Lemma 7.4 and Lemma D.5 (i), one can verify that $\{1,2\} \subseteq \mathrm{ER}(\mathcal{G}_{\bar{d}_1,\bar{d}_2})$. So, $P_j(\bar{d}_1, \bar{d}_2) \ne \emptyset$ for $j \in \{1,2\}$. By further noting that $Z_j(d_1,d_2) = \mathrm{argmax}_{\mathbf{x} \in P_j(\bar{d}_1,\bar{d}_2)} \mathbf{a}_j \cdot \mathbf{x}$, we get that $Z_j(d_1,d_2) \ne \emptyset$. So, Lemma D.5 (iii) implies that $\widetilde{V}^L_{d_1,d_2}(j) = \gamma_j \cdot \bar{d}_j + \beta_j$. It follows that

$$\widetilde{V}^L_{d_1,d_2}(j) = \gamma_j \cdot \bar{d}_j + \beta_j \ge \gamma_j \cdot d_j + \beta_j = V^L_{d_1,d_2}(j), \tag{46}$$

where $\gamma_j \cdot d_j + \beta_j = V^L_{d_1,d_2}(j)$ follows by Lemma D.1. Moreover, since $\bar{d}_j = (d_j + d_j^*)/2$, we have

$$\widetilde{V}^L_{d_1,d_2}(j) = \frac{(\gamma_j \cdot d_j + \beta_j) + (\gamma_j \cdot d_j^* + \beta_j)}{2} = \frac{1}{2} \cdot \left(V^L_{d_1,d_2}(j) + V^L_{d_1^*,d_2^*}(j) \right). \tag{47}$$

We have $V^L_{d_1,d_2}(1) = V^L_{d_1,d_2}(2)$, as implied by $\{1,2\} \subseteq \mathrm{ER}(\mathcal{G}_{d_1,d_2})$. Moreover, $V^L_{d_1^*,d_2^*}(j) = V^L_{d^*}(j) = M_{\{1,2\}}$ for $j \in \{1,2\}$ as we proved in Lemma 7.1. Hence, Eq. (47) implies $\widetilde{V}^L_{d_1,d_2}(1) = \widetilde{V}^L_{d_1,d_2}(2)$. Using Eq. (46), we then establish $\widetilde{V}^L_{d_1,d_2}(1) = \widetilde{V}^L_{d_1,d_2}(2) \ge V^L_{d_1,d_2}(1)$, as desired. This completes the proof. $\square$

E Omitted Proofs in Section 8

E.1 Proof of Theorem 8.1

Theorem 8.1. *Given oracle $\mathcal{A}_{\mathrm{W1}}$, in time polynomial in the bit size of u^L and $\delta > 0$, we can learn a matrix $\tilde{u}^L$ that achieves the same equivalences in Theorem 5.4 with probability at least $1 - \delta$. The same holds for $\mathcal{A}_{\mathrm{W2}}$ if u^L is non-max-degenerate.*

Proof. Notice that the only part of the approach where the ability of specifying an equilibrium is needed is to prove Observation 4.1, Observation 4.2, and decide the termination of the binary searches. With an upper bound of the input's bit size given, we can efficiently terminate the binary searches. It now suffices to show that every query to $\mathcal{A}_{\mathrm{SSE}}$ used in proving Observation 4.1 and 4.2 can be replaced by a polynomial number of queries to oracles $\mathcal{A}_{\mathrm{W1}}$ and $\mathcal{A}_{\mathrm{W2}}$, whereby the information we need can be learned with high probability. Once this is proven, the result stated in the theorem follows readily.

Specifically, consider oracle $\mathcal{A}_{\mathrm{W1}}$, Assume that a strategy profile $(\mathbf{x}, j)$ is a basic SSE of game $(u^L, \tilde{u}^F)$. We now compute the number of queries needed to make $\mathcal{A}_{\mathrm{W1}}$ output $(\mathbf{x}, j)$ at least once with probability at least $1-\delta$. Denote the number of basic SSEs of game $(u^L, \tilde{u}^F)$ to be N. Actually, it suffices to query the oracle $\log(1/\delta)/\log(N/N-1) = O(N\log(1/\delta))$ times. Indeed, suppose K is the number of queries made. We have

$$1-(1-1/N)^K \geq 1-\delta \qquad \Longleftrightarrow \qquad K \geq \log\left(\frac{1}{\delta}\right) / \log\left(\frac{N}{N-1}\right). \tag{48}$$

In the induced games in proving Observation 4.1 and 4.2, N is bounded from above by the number of leader's pure actions, which is, respectively, m and $m+1$ in these two observations.

As for oracle $\mathcal{A}_{\mathrm{W2}}$, Observation 4.1 and 4.2 can be easily achieved as I_j only has one element for each follower action j in the case of non-max-degeneracy. $\square$

References

Georgios Amanatidis, Georgios Birmpas, Federico Fusco, Philip Lazos, Stefano Leonardi, and Rebecca Reiffenhäuser. Allocating indivisible goods to strategic agents: Pure Nash equilibria and fairness. In *International Conference on Web and Internet Economics*, pages 149–166. Springer, 2021.

Kareem Amin, Afshin Rostamizadeh, and Umar Syed. Learning prices for repeated auctions with strategic buyers. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1169–1177, 2013.

Kareem Amin, Afshin Rostamizadeh, and Umar Syed. Repeated contextual auctions with strategic buyers. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 622–630, 2014.

Nivasini Ananthakrishnan, Nika Haghtalab, Chara Podimata, and Kunhe Yang. Is knowledge power? on the (im)possibility of learning from strategic interaction, 2024. URL https://arxiv.org/abs/2408.08272.

Eshwar Ram Arunachaleswaran, Natalie Collina, and Jon Schneider. Pareto-optimal algorithms for learning in games, 2024. URL https://arxiv.org/abs/2402.09549.

Robert J Aumann, Michael Maschler, and Richard E Stearns. *Repeated games with incomplete information*. MIT press, 1995.

Maria-Florina Balcan, Avrim Blum, Nika Haghtalab, and Ariel D. Procaccia. Commitment without regrets: Online learning in Stackelberg security games. In *Proceedings of the 16th ACM Conference on Economics and Computation (EC)*, pages 61–78, 2015.

Xiaohui Bei, Xinhang Lu, and Warut Suksompong. Truthful cake sharing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4809–4817, 2022.

Georgios Birmpas, Jiarui Gan, Alexandros Hollender, Francisco J Marmolejo-Cossío, Ninad Rajgopal, and Alexandros A Voudouris. Optimally deceiving a learning leader in stackelberg games. *Journal of Artificial Intelligence Research*, 72:507–531, 2021.

Avrim Blum, Nika Haghtalab, and Ariel D Procaccia. Learning optimal commitment to overcome insecurity. *Advances in Neural Information Processing Systems*, 27, 2014a.

Avrim Blum, Nika Haghtalab, and Ariel D. Procaccia. Learning optimal commitment to overcome insecurity. In *Proceedings of the 28th Conference on Neural Information Processing Systems (NIPS)*, pages 1826–1834, 2014b.

Toni Böhnlein, Oliver Schaudt, and Joachim Schauer. Stackelberg packing games. *Theoretical computer science*, 943:16–35, 2023.

Mark Braverman, Jieming Mao, Jon Schneider, and Matt Weinberg. Selling to a no-regret buyer. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, EC '18, page 523–538, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450358293. doi: 10.1145/3219166.3219233. URL https://doi.org/10.1145/3219166.3219233.

Sarah H. Cen, Andrew Ilyas, and Aleksander Madry. User strategization and trustworthy algorithms. In *Proceedings of the 25th ACM Conference on Economics and Computation*, EC '24, page 202, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400707049. doi: 10.1145/3670865.3673545. URL https://doi.org/10.1145/3670865.3673545.

Yurong Chen, Xiaotie Deng, and Yuhao Li. Optimal private payoff manipulation against commitment in extensive-form games. *Games and Economic Behavior*, 149:118–147, 2025. ISSN 0899-8256. doi: https://doi.org/10.1016/j.geb.2024.11.008. URL https://www.sciencedirect.com/science/article/pii/S0899825624001647.

Zhaohua Chen, Xiaotie Deng, Jicheng Li, Chang Wang, Mingwei Yang, Zheng Cai, Yukun Ren, and Zhihua Zhu. Budget-constrained auctions with unassured priors: Strategic equivalence and structural properties, 2022. URL https://arxiv.org/abs/2203.16816.

Yukun Cheng, Xiaotie Deng, Yuhao Li, and Xiang Yan. Tight incentive analysis on sybil attacks to market equilibrium of resource exchange over general networks. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, pages 792–793, 2022.

Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, page 191–198, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340359. doi: 10.1145/2959100.2959190. URL https://doi.org/10.1145/2959100.2959190.

Xiaotie Deng, Tao Lin, and Tao Xiao. Private data manipulation in optimal sponsored search auction. In Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen, editors, *Proceedings of The Web Conference 2020*, WWW '20, pages 2676–2682, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370233. doi: 10.1145/3366423.3380023. URL https://doi.org/10.1145/3366423.3380023.

Yuan Deng, Jon Schneider, and Balasubramanian Sivan. Strategizing against no-regret learners. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/8b6dd7db9af49e67306feb59a8bdc52c-Paper.pdf.

Shaddin Dughmi. Algorithmic information structure design: a survey. *ACM SIGecom Exchanges*, 15(2):2–24, 2017.

P. Dütting, T. Roughgarden, and I. Talgam-Cohen. The complexity of contracts. *SIAM Journal on Computing*, 50(1):211–254, 2021.

Jiarui Gan, Qingyu Guo, Long Tran-Thanh, Bo An, and Michael Wooldridge. Manipulating a learning defender and ways to counteract. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8274–8283, 2019a.

Jiarui Gan, Haifeng Xu, Qingyu Guo, Long Tran-Thanh, Zinovi Rabinovich, and Michael Wooldridge. Imitative follower deception in Stackelberg games. In *Proceedings of the 2019 ACM Conference on Economics and Computation (EC)*, page 639–657, 2019b.

Jiarui Gan, Rupak Majumdar, Goran Radanovic, and Adish Singla. Bayesian persuasion in sequential decision-making. In *AAAI '22*, pages 5025–5033, 2022.

Jiarui Gan, Rupak Majumdar, Debmalya Mandal, and Goran Radanovic. Sequential principal-agent problems with communication: Efficient computation and learning, 2023. URL https://arxiv.org/abs/2306.03832v2.

Ronald L Graham, Donald E Knuth, Oren Patashnik, and Stanley Liu. Concrete mathematics: a foundation for computer science. *Computers in Physics*, 3(5):106–107, 1989.

Nika Haghtalab, Fei Fang, Thanh H. Nguyen, Arunesh Sinha, Ariel D. Procaccia, and Milind Tambe. Three strategies to success: learning adversary models in security games. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 308–314. AAAI Press, 2016. ISBN 9781577357704.

Nika Haghtalab, Thodoris Lykouris, Sloan Nietert, and Alexander Wei. Learning in stackelberg games with non-myopic agents. In *Proceedings of the 23rd ACM Conference on Economics and Computation (EC)*, page 917–918, 2022.

Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, ITCS '16, page 111–122, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340571. doi: 10.1145/2840728.2840730. URL https://doi.org/10.1145/2840728.2840730.

Ian A Kash and David C Parkes. Impersonation strategies in auctions. In *Internet and Network Economics: 6th International Workshop, WINE 2010, Stanford, CA, USA, December 13-17, 2010. Proceedings 6*, pages 492–495. Springer, 2010.

Yoav Kolumbus and Noam Nisan. Auctions between regret-minimizing agents. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 100–111, New York, NY, USA, 2022a. Association for Computing Machinery. ISBN 9781450390965.

Yoav Kolumbus and Noam Nisan. How and why to manipulate your own agent: On the incentives of users of learning agents. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022b.

Adriano Koshiyama, Emre Kazim, Philip Treleaven, Pete Rai, Lukasz Szpruch, Giles Pavey, Ghazi Ahamat, Franziska Leutner, Randy Goebel, Andrew Knight, Janet Adams, Christina Hitrova, Jeremy Barnett, Parashkev Nachev, David Barber, Tomas Chamorro-Premuzic, Konstantin Klemmer, Miro Gregorovic, Shakeel Khan, Elizabeth Lomas, Airlie Hilliard, and Siddhant Chatterjee. Towards algorithm auditing: managing legal, ethical and technological risks of ai, ml and associated algorithms. *Royal Society Open Science*, 11(5):230859, 2024. doi: 10.1098/rsos.230859. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsos.230859.

Joshua Letchford, Vincent Conitzer, and Kamesh Munagala. Learning and approximating the optimal strategy to commit to. In *International Symposium on Algorithmic Game Theory (SAGT)*, pages 250–262, 2009.

Yuqian Li and Vincent Conitzer. Game-theoretic question selection for tests. *Journal of Artificial Intelligence Research*, 59:437–462, 2017.

Tao Lin and Yiling Chen. Generalized principal-agent problem with a learning agent. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LqTz13JS2P.

Jinyan Liu, Zhiyi Huang, and Xiangning Wang. Learning optimal reserve price against non-myopic bidders. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2042–2052, 2018.

Yishay Mansour, Mehryar Mohri, Jon Schneider, and Balasubramanian Sivan. Strategizing against learners in bayesian games. In Po-Ling Loh and Maxim Raginsky, editors, *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings* *of Machine Learning Research*, pages 5221–5252. PMLR, 02–05 Jul 2022. URL https://proceedings.mlr.press/v178/mansour22a.html.

Anna-Katharina Meßmer and Martin Degeling. Auditing recommender systems – putting the dsa into practice with a risk-scenario-based approach, 2023. URL https://arxiv.org/abs/2302.04556.

Mehryar Mohri and Andres Muñoz Medina. Optimal regret minimization in posted-price auctions with strategic buyers. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1871–1879, 2014.

Thanh H. Nguyen and Haifeng Xu. Imitative attacker deception in Stackelberg security games. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 528–534, 2019.

Binghui Peng, Weiran Shen, Pingzhong Tang, and Song Zuo. Learning optimal strategies to commit to. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 2149–2156, 2019.

Aaron Roth, Jonathan Ullman, and Zhiwei Steven Wu. Watch and learn: Optimizing from revealed preferences feedback. In *Proceedings of the 48th annual ACM symposium on Theory of Computing (STOC)*, pages 949–962, 2016.

S. Sorin. Some results on the existence of Nash equilibria for non-zero-sum games with incomplete information. *Internat. J. Game Theory*, 12(4):193–205, 1983. ISSN 0020-7276. doi: 10.1007/BF01769090. URL https://doi.org/10.1007/BF01769090.

Kateřina Staňková and Alexander Boudewijn. Stackelberg and inverse stackelberg road pricing games: State of the art and future research. *Game theoretic analysis of congestion, safety and security: Traffic and transportation theory*, pages 191–209, 2015.

Milind Tambe. *Security and Game theory: Algorithms, Deployed Systems, Lessons Learned.* Cambridge University Press, 2011.

Pingzhong Tang and Yulong Zeng. The price of prior dependence in auctions. In *Proceedings of the 2018 ACM Conference on Economics and Computation (EC)*, pages 485–502, 2018.

Heinrich von Stackelberg. *Marktform und Gleichgewicht.* J. Springer, 1934.

Bernhard von Stengel and Shmuel Zamir. Leadership with commitment to mixed strategies. *CDAM Research Report*, LSE-CDAM-2004-01, 2004.

Bernhard Von Stengel and Shmuel Zamir. Leadership games with convex strategy sets. *Games and Economic Behavior*, 69(2):446–457, 2010.

Yunjuan Wang, Poorya Mianjy, and Raman Arora. Robust learning for data poisoning attacks. In *International Conference on Machine Learning*, pages 10859–10869. PMLR, 2021.

Jibang Wu, Weiran Shen, Fei Fang, and Haifeng Xu. Inverse game theory for stackelberg games: the blessing of bounded rationality. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 32186–32198. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/cfce833814505906445f8df2f65ab548-Paper-Conference.pdf.

Young Wu, Jeremy McMahan, Xiaojin Zhu, and Qiaomin Xie. On faking a Nash equilibrium. *arXiv preprint arXiv:2306.08041*, 2023.

Yixuan Even Xu, Chun Kai Ling, and Fei Fang. Learning coalition structures with games. *arXiv preprint arXiv:2312.09058*, 2023.

Boling Yang, Liyuan Zheng, Lillian J Ratliff, Byron Boots, and Joshua R Smith. Stackelberg games for learning emergent behaviors during competitive autocurricula. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5501–5507. IEEE, 2023.

Hanrui Zhang and Vincent Conitzer. Automated dynamic mechanism design. *NeurIPS '21*, 2021a.

Hanrui Zhang and Vincent Conitzer. Incentive-aware PAC learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5797–5804, 2021b.

Tijana Zrnic, Eric Mazumdar, S. Shankar Sastry, and Michael I. Jordan. Who leads and who follows in strategic classification? In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 15257–15269, 2021.