
No Free Lunch Theorem and Black-Box Complexity Analysis for Adversarial Optimisation

Per Kristian Lehre
School of Computer Science
University of Birmingham
Birmingham, United Kingdom
p.k.lehre@bham.ac.uk

Shishen Lin*
School of Computer Science
University of Birmingham
Birmingham, United Kingdom
sx11242@student.bham.ac.uk

Abstract

Black-box optimisation is one of the important areas in optimisation. The original No Free Lunch (NFL) theorems highlight the limitations of traditional black-box optimisation and learning algorithms, serving as a theoretical foundation for traditional optimisation. No Free Lunch Analysis in adversarial (also called maximin) optimisation is a long-standing problem [45, 46]. This paper first rigorously proves a (NFL) Theorem for general black-box adversarial optimisation when considering Pure Strategy Nash Equilibrium (NE) as the solution concept. We emphasise the solution concept (i.e. define the optimality in adversarial optimisation) as the key in our NFL theorem. In particular, if Nash Equilibrium is considered as the solution concept and the cost of the algorithm is measured in terms of the number of columns and rows queried in the payoff matrix, then the average performance of all black-box adversarial optimisation algorithms is the same. Moreover, we first introduce black-box complexity to analyse the black-box adversarial optimisation algorithm. We employ Yao’s Principle and our new NFL Theorem to provide general lower bounds for the query complexity of finding a Nash Equilibrium in adversarial optimisation. Finally, we illustrate the practical ramifications of our results on simple two-player zero-sum games. More specifically, no black-box optimisation algorithm for finding the unique Nash equilibrium in two-player zero-sum games can exceed logarithmic complexity relative to search space size. Meanwhile, no black-box algorithm can solve any bimatrix game with unique NE with fewer than a linear number of queries in the size of the payoff matrix.

1 Introduction

1.1 Black-Box Optimisation and the No Free Lunch Theorem

Black-Box Optimisation (BBO) is crucial for optimising complex, unknown, or expensive-to-evaluate functions in real-world scenarios, such as aerodynamic design and hyperparameter tuning, where only input-output observations are available. Formally, BBO is the task of optimising objective functions from some function class \mathcal{F} where \mathcal{F} consists of $f : \mathcal{X} \rightarrow \mathbb{R}$, where the algorithm is limited to making queries to f [11, 14, 36]. In this case, the algorithm is only able to sample and query the function value $f(x)$ of search points $x \in \mathcal{X}$ from a “black-box” or “oracle” without access to any description of the objective functions f . A similar framework can be extended to game-theoretic BBO, namely adversarial BBO. Specifically, adversarial BBO is the task that optimising payoff functions from some payoff function class \mathcal{G} where \mathcal{G} consists of black-box functions $g : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, with a limited budget of function evaluations. Additionally, two players $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ form

*Authors are listed in alphabetical order.

a maximin optimisation (as presented in Figure 1). The original No Free Lunch Theorems for traditional optimisation can be summarised as follows:

“For all possible metrics, no search algorithm or supervised learning algorithm is better than another when its performance is averaged over all possible problem instances [45, 44].”

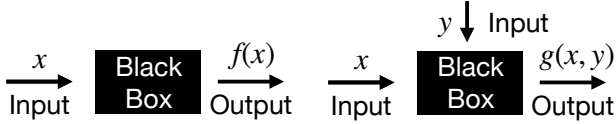


Figure 1: Comparison between traditional black-box optimisation and maximin black-box optimisation. Instead of querying at x in traditional optimisation, maximin optimisation queries at (x, y) include both strategy x and the best response y from the opponent, i.e. $\max_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} g(x, y)$. Their interaction is converted to the payoff $g(x, y)$ in the given black-box model.

Wolpert and Macready [45] and Wolpert [44] have revealed the underlying facts about the usefulness of traditional black-box optimisation algorithms, including various randomised search heuristics (such as evolutionary algorithms and simulated annealing) and machine learning algorithms (such as supervised learning). In particular, it shows that the performance of all black-box optimisation algorithms [45] and learning algorithms [44], when averaged over all problem instances, is the same for any maximisation or minimisation tasks. Droste et al. [11] has provided a generalised NFL theorem with more realistic scenarios by relaxing NFL theorem holds from all problem instances to problem instances closed under permutation. Another seminal work by Schaffer [37] also showed a conservation law for the generalised performance of learning algorithms in classification problems.

Adversarial optimisation tasks, such as maximin optimisation, are more complex and often counter-intuitive compared to traditional optimisation problems. In adversarial settings, defining solution concepts, or establishing what is meant by optimality, is essential. Common solution concepts include ‘Maximisation Over All Test Cases’, ‘maximin’, and Nash Equilibrium. A ‘free lunch’ in adversarial optimisation implies that, for a given solution concept, some algorithms consistently outperform others when averaged across all possible problem instances. This phenomenon was demonstrated by Wolpert and Macready [46] for adversarial optimisation with respect to the ‘maximin’ solution concept (Definition 10). Similarly, Service and Tauritz [39] established a ‘free lunch’ result for the ‘Maximisation Over All Test Cases’ concept (Definition 9).

Beyond ‘maximin’ and ‘Maximisation Over All Test Cases’, Nash Equilibrium is another widely studied solution concept in adversarial learning and maximin optimisation. This concept is central to various applications, including adversarial learning models such as GANs [15, 20] and spatial games [19, 26, 32]. While previous studies by Wolpert and Macready [46] and Service and Tauritz [39] have demonstrated the existence of ‘free lunches’ in adversarial optimisation, the following key questions remain open:

- (1) Does adversarial optimisation exhibit a ‘free lunch’ for all possible solution concepts?
- (2) If we use Nash Equilibrium as the solution concept, how can we characterise the difficulty of black-box adversarial optimisation problems for problem-independent, possibly randomised, search heuristics?

In this paper, we answer Question (1) in the negative by showing a No Free Lunch theorem for Nash Equilibrium solution concept and address Question (2) by introducing black-box complexity tools.

1.2 Challenges and Technical Overview of the NFL and BBC Results

There are several challenges in showing the NFL and BBC results. First, we highlight the challenges and our technical details in the derivation process compared to previous NFL work. The classical No Free Lunch theorem applies proof by induction with respect to the size of the function domain. A natural idea is to extend the previous proof-by-induction method in our NFL proof. However, one of the most challenging aspects of deriving NFL for Nash Equilibrium (NE) is that the adversarial

setting significantly increases the difficulty since the problem depends on the performances of both the player and the opponents. To address this, we introduce two technical lemmas from game theory (i.e. Lemma 3 and Lemma 4) and help to construct the isomorphism between two different problem classes (i.e. Corollary 1 and Lemma 2), which is an essential step in the proof of the original NFL theorem for traditional optimisation (see Section B for more details). Another significant challenge is that we start with a very weak assumption: we only allow access to the payoff function and make no assumptions about its properties, such as convexity, continuity, or differentiability. In fact, no gradients, continuity, or differentiability make the NFL and black box results easier to prove (i.e., the function class lacks a specific structure). On the other hand, it makes it more difficult for the algorithm. Consequently, we have limited analytical tools to proceed. To analyse the black-box complexity of adversarial optimisation, we introduce Yao’s minimax principle and apply our new NFL result. Please consult the proof of Theorem 4.2 for more details.

1.3 Contribution

We prove a new impossibility result on black-box adversarial optimisation. In a two-player zero-sum game setting, there is no free lunch with respect to an approximation of the real cost model when regarding the unique Nash Equilibrium as the solution concept. In other words, all black-box adversarial optimisation algorithms have the same expected performance over a uniform distribution of all possible problem instances with the unique Nash Equilibrium as the solution concept. It is the first step to resolve this long-standing open research problem about NFL in general black-box adversarial optimisation since [45, 46]. Our results highlight the significance of the choice of solution concepts and the limitation of general black-box adversarial optimisation. Additionally, we introduce the first general black-box model and the notion of black-box complexity for adversarial optimisation. Under this general black-box model, we provide the general lower bounds for query complexity of computing Nash Equilibrium in adversarial optimisation. Finally, we illustrate our results on examples of computing unique Nash Equilibrium in two-player zero-sum games.

2 Preliminaries

2.1 Notation

Given $n \in \mathbb{N}$, $[n] := \{1, 2, \dots, n\}$. Given a finite set \mathcal{X} , we denote the permutation group by $\mathfrak{S}(\mathcal{X}) := \{\sigma : \mathcal{X} \rightarrow \mathcal{X} \mid \sigma \text{ is a permutation of } \mathcal{X}\}$. For any $v \in \mathbb{R}^n$, let $\text{SUPP}(v) := \{i \in [n] \mid v_i \neq 0\}$. We refer to query complexity or runtime as the number of payoff function evaluations until the given algorithm finds the optimum. We consider the search space $\mathcal{X} \times \mathcal{Y}$, where \mathcal{X} and \mathcal{Y} are finite and denote the cardinality of a set by $|\mathcal{X}|$. For any bitstring z , $|z|_1$ denotes the number of 1-bits in z . $X \simeq Y$ denotes the isomorphism between X and Y . $X \simeq Y$ if there exists a one-to-one correspondence map from X to Y .

Let $f : X \rightarrow Y$ be a function from a set X to a set Y . If A is a subset of X , then the restriction of f to A is the function: $f|_A : A \rightarrow Y; x \mapsto f(x)$. Let $f : X \rightarrow Y$ be any function and A and B be sets such that $X \subseteq A$ and $Y \subseteq B$. An extension of f to A is a function $g : A \rightarrow B$ such that $f(x) = g(x)$ for all $x \in X$. Alternatively, g is an extension of f to A if f is a restriction of g to X .

In this paper, we assume that the black-box optimisation algorithms do not make the same query twice. This can be achieved by memorising the outcome of previous queries.

2.2 Solution Concepts

Solution concepts for classical function optimisation are not directly applicable to adversarial optimisation in general-sum or zero-sum game settings [33, 24, 35]. Each agent or player’s payoff depends on not only its action but also the response from its opponents, and thus, we need to introduce different solution concepts to specify what kind of optimum we look for.

Pure Strategy Nash equilibrium is considered as our solution concept in this work. We are interested in whether the black-box adversarial optimisation can efficiently find any given game’s Nash equilibrium. We use the formulation in [30] to define Nash equilibrium rigorously. In this paper, we only focus on Pure Strategy Nash Equilibrium (abbrev. NE).

Definition 1. (Nash) Consider a two-player game. Given strategy spaces $\mathcal{X} \times \mathcal{Y}$ and payoff functions $g, h : \mathcal{X} \times \mathcal{Y} \rightarrow O$, where O is an ordered set which determines the outcome of candidate solution \mathcal{X} on test \mathcal{Y} , (x^*, y^*) is a Nash equilibrium if for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, $g(x^*, y^*) \geq g(x, y^*)$ and $h(x^*, y^*) \geq h(x^*, y)$. In particular, if the two-player game is zero-sum (that is, if $g(x, y) + h(x, y) = 0$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$), then (x^*, y^*) is a Nash equilibrium if for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, $g(x, y^*) \leq g(x^*, y^*) \leq g(x^*, y)$. The solution concept is defined by

$$S = \{(x^*, y^*) \in \mathcal{X} \times \mathcal{Y} \mid \forall (x, y) \in \mathcal{X} \times \mathcal{Y}, g(x, y^*) \leq g(x^*, y^*) \leq g(x^*, y)\}.$$

We also defer other solution concepts for comparison in the appendix. We use the formulation in [46, 39].

3 No Free Lunch Theorem for Computing Nash Equilibrium in Adversarial Optimisation

In this section, we prove a No Free Lunch Theorem for black-box adversarial optimisation. As a first step toward the No Free Lunch Theorem, we consider two-player zero-sum games with the unique NE as the same setting in [34], but we relax the restriction from potential games to general two-player zero-sum games. The original NFL theorem for traditional optimisation assumed all problems instances [45], and later, sharpened work also proved that this holds for functions ‘closed under permutation’ [38, 11]. We now explain what ‘closed under permutation’ means and how it can be extended to games.

Permutation closure of a set of functions means that let \mathcal{X}, \mathcal{Y} be two finite sets and $f : \mathcal{X} \rightarrow \mathcal{Y}$ defined by $f(x_i) = y_i$. Let σ be a permutation $\sigma : \mathcal{X} \rightarrow \mathcal{X}$ and we can permute the function: $f(\sigma(x)) := \sigma \circ f(x)$. Schumacher et al. [38] and Droste et al. [11] defined ‘closed under permutation (c.u.p.)’ with respect to a single search space \mathcal{X} . A class of functions $\mathcal{F} = f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ is called c.u.p. if for all $f \in \mathcal{F}$ and all permutations $\sigma \in \mathfrak{S}(\mathcal{X})$, $f \circ \sigma \in \mathcal{F}$. For our adversarial setting, we need to extend this notion to $\mathcal{X} \times \mathcal{Y}$.

Definition 2. Given two-player zero-sum games, suppose \mathcal{F} as a subset of all the payoff functions in these games $g : \mathcal{X} \times \mathcal{Y} \rightarrow O$ where $O \subseteq \mathbb{R}$. We say \mathcal{F} is c.u.p if for any $g \in \mathcal{F}$ and any permutations on \mathcal{X}, \mathcal{Y} denoted by $\sigma \in \mathfrak{S}(\mathcal{X}), \tau \in \mathfrak{S}(\mathcal{Y})$, we have $(\sigma \otimes \tau) \circ g \in \mathcal{F}$ (or abbreviated $(\sigma\tau)g$) where $(\sigma \otimes \tau) \circ g(x, y) := g(\sigma(x), \tau(y))$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$.

If a set of two-player zero-sum games is c.u.p., we will also call it structure-free. In this paper, we restrict to the case that $\mathcal{X}, \mathcal{Y}, O$ are finite sets following the settings in [45, 46]. The proof of the No Free Lunch theorem for traditional optimisation by Droste et al. [12] is by induction, where one assumes that the statement is true for all “smaller problems”. Here, a smaller problem refers to the following definition of a sub-game or a sub-problem class. Next, we define a sub-problem class for two-player zero-sum games.

Definition 3. Given two-player zero-sum games, let $O \subseteq \mathbb{R}$ and \mathcal{F} be any subset of the set of payoff functions in these games $g : \mathcal{X} \times \mathcal{Y} \rightarrow O$ such that \mathcal{F} is closed under permutation. For any given $(x_1, y_1) \in \mathcal{X} \times \mathcal{Y}$, any function $b_1 : \mathcal{Y} \rightarrow O$, and any function $b_2 : \mathcal{X} \rightarrow O$, we define a sub-problem class $\mathcal{F}((x_1, y_1), (b_1, b_2))$ with respect to \mathcal{F} as follows: $f \in \mathcal{F}((x_1, y_1), (b_1, b_2))$ if and only if there exists a $g \in \mathcal{F}$ such that

- (1) $g(x_1, y) = b_1(y)$ for all $y \in \mathcal{Y}$;
- (2) $g(x, y_1) = b_2(x)$ for all $x \in \mathcal{X}$;
- (3) f is a restriction of g on $(\mathcal{X} \setminus \{x_1\}) \times (\mathcal{Y} \setminus \{y_1\})$.

Next, before we show that a sub-problem class $\mathcal{F}((x_1, y_1), (b_1, b_2))$ is c.u.p., we need a lemma to guarantee that after any permutation, the permuted payoff function still contains the same unique Nash Equilibrium². We provide a corollary to Lemma 4. We defer all the proofs to the appendix due to the page limit.

Corollary 1. Let $|\mathcal{X}| = |\mathcal{Y}|$ and $\mathcal{F} \subseteq \{g : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}\}$ be any set of two-player zero-sum games with a unique NE. Assume furthermore that \mathcal{F} is c.u.p.. Then, for any game $g \in \mathcal{F}$, let

²The further explanation for sub-problem classes can be found in Section E.

$(x^*, y^*) \in \mathcal{X} \times \mathcal{Y}$ denote the unique Nash Equilibrium of g . For any permutations over \mathcal{X}, \mathcal{Y} denoted by $\sigma \in \mathfrak{S}(\mathcal{X}), \tau \in \mathfrak{S}(\mathcal{Y}), (\sigma \otimes \tau) \circ g \in \mathcal{F}$ exhibits the same unique Nash Equilibrium (x^*, y^*) . Moreover, if for any $x_0 \neq x^*$ and $y_0 \neq y^*$, the restriction of g on $(\mathcal{X} \setminus \{x_0\}) \times (\mathcal{Y} \setminus \{y_0\})$, denoted by $g|_{(\mathcal{X} \setminus \{x_0\}) \times (\mathcal{Y} \setminus \{y_0\})}$, exhibits the same unique Nash Equilibrium (x^*, y^*) .

This permutation essentially swaps the rows and columns in the payoff matrix. Corollary 1 shows that the permutation of rows and columns does not change the optimum (Nash Equilibrium in any given payoff function). Moreover, if we remove the row and column in the given payoff function, which do not contain the unique Nash Equilibrium, then the unique Nash Equilibrium remains the same in the restricted sub-problem.

Next, we prove the sub-problem class is closed under permutation.

Lemma 1. If \mathcal{F} is c.u.p., then $\mathcal{F}((x_1, y_1), (b_1, b_2))$ is also c.u.p..

Then, we want to know if we choose different (x, y) in the sub-problem class, are they still isomorphic (i.e., essentially the same problem for any black-box optimisation algorithm)?

Lemma 2. For all $(x_1, y_1), (x_2, y_2) \in \mathcal{X} \times \mathcal{Y}$ and $b_1 : \mathcal{Y} \rightarrow \mathbb{R}, b_2 : \mathcal{X} \rightarrow \mathbb{R}$, we have the isomorphism³:

$$\mathcal{F}((x_1, y_1), (b_1, b_2)) \simeq \mathcal{F}((x_2, y_2), (b_1, b_2)).$$

Given (x_t, y_t) is the search point by Algorithm H on payoff function g at iteration $t \in \mathbb{N}$ and (x^*, y^*) is the unique NE in a two-player zero-sum game defined by $g : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, Theorem 3.1 considers the following query complexity: assume the cost C_t is the unique queries made by Algorithm H ,

$$T_{LB}(H, g) := \inf\{C_t > 0 \mid x_t = x^* \text{ or } y_t = y^*\}.$$

Now, we prove our main theorem.

Theorem 3.1. Given \mathcal{F} as a subset of all the payoff functions $g : \mathcal{X} \times \mathcal{Y} \rightarrow O$ with a unique Nash Equilibrium where $O \subseteq \mathbb{R}$ and $|\mathcal{X}| = |\mathcal{Y}|$. Let H be an arbitrary (randomised or deterministic) black-box adversarial optimisation algorithm for any $g \in \mathcal{F}$ where \mathcal{F} is closed under permutations. Let $r(H, \mathcal{F})$ be the average (under the uniform distribution on \mathcal{F}) of the expected query complexity of H on $g \in \mathcal{F}$ (i.e. $E(T_{LB}(H, g))$). Then $r(H, \mathcal{F}) = r(H', \mathcal{F})$ for all algorithms H, H' .

Proof Sketch. The proof of the main theorem is deferred to the appendix. Briefly, we use a proof by induction on the size of the search space. During the inductive hypothesis, problem class is reduced from \mathcal{F} to $\mathcal{F}((x_1, y_1), (b_1, b_2))$, decreasing the search space from size $N \times N$ to size $(N - 1) \times (N - 1)$, where $|\mathcal{X}| := N$. Then using Lemma 2 and inductive step, it follows that $r(H, \mathcal{F}) = r(H', \mathcal{F})$ for any two deterministic algorithms H, H' , and the claim for randomised algorithms quickly follows from the fact that any randomised algorithm can be viewed as a probability distribution among all deterministic algorithms. \square

Theorem 3.1 reveals an important underlying result: All black-box adversarial algorithms can exhibit the same average runtime $r(H, \mathcal{F})$ of all possible problem instances (or problem instances c.u.p.) with a unique Nash Equilibrium in a two-player zero-sum game setting. It is a reasonable result since Theorem 3.1 tells us that if a class of payoff functions with a unique Nash Equilibrium does not change by any permutation on the input space, there is no structure provided for any search heuristic or any optimisation algorithm to use and it cannot help to find the Nash Equilibrium. This is also the reason why the original No Free Lunch theorem for traditional optimisation holds [45, 44]. As concluded by Ho and Pepyne [23], “if anything is possible and occurs with the same probability, then nothing can be expected.

This result is also surprising since Wolpert and Macready [46] and Service and Tauritz [39] both show there exists free lunch with respect to the two solution concepts in Definition 9 and Definition 10. However, our result does contradict the previous result. We consider the performance measure $T(H, g)$ and a different query model considered by the previous work. In particular, Definition 9 and Definition 10 only take the player x into account, while the opponent optimum and different query model and performance measures can make a difference, resulting in either FL or NFL results. In summary, our new NFL theorem highlights the significance of solution concepts and also reveals that adversarial optimisation can exhibit “no-free-lunch”, in particular for NE solution concept.

³Here we define isomorphism following [11] rather than defining it as a usual group or ring isomorphism since we do not require any group or ring structure of \mathcal{F} in the proof.

4 Black-Box Complexity of Black-Box Adversarial Optimisation

As shown in the previous NFL theorem, no better universal algorithms exist on structure-free problems (i.e., only assume the payoff function consists of the unique NE). In order for an algorithm to guarantee good performance, it is necessary to restrict the algorithm to classes of games that possess some structure. To compute the Nash equilibrium of certain classes of problems, including Nash equilibrium in a black-box setting, there are many works on analysing the computational complexity for black-box algorithms, and researchers aim to minimise the query complexity and provide more efficient algorithms to compute Nash equilibrium in two-player zero-sum game settings [27, 26, 3, 21, 22]. The following questions remain under-explored: How does the performance measure of an algorithm (like query complexity) depend on the size of the search space, and is there any limitation that these algorithms will reach regardless of the problem g ? We answer these questions here using black-box complexity.

4.1 The Unrestricted Black-Box Model and the Black-Box Complexity

This section focuses on adversarial black-box optimisation and study the query complexity of learning Nash equilibrium in a two-player zero-sum-game setting. We refer to [13, 8, 10] as a more detailed introduction to the black-box complexity theory on traditional black-box optimisation. To prove a lower bound that holds for all algorithms, it is necessary first formally to define what constitutes an algorithm. Next, we construct an unrestricted black-box model of adversarial optimisation.

Algorithm 1 An Unrestricted Black-Box Model with Unique Query History

Require: Search spaces \mathcal{X}, \mathcal{Y} .

Require: Payoff functions $g : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}, h : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$;

- 1: Initialise (x_0, y_0) based on P_\emptyset ; Initialise $H_0 = \emptyset$ and $C_0 = 0$.
 - 2: **for** $t = 1, 2, \dots$ until the termination criterion met **do**
 - 3: Choose some probability distribution $P_{I(t)}$, depending only on $I(t)$ where
 $I(t) := \prod_{j=1}^{t-1} (x_j, y_j, g(x_j, y_j), h(x_j, y_j)) \in (\mathcal{X} \times \mathcal{Y} \times \mathbb{R} \times \mathbb{R})^{t-1}$
 - 4: Produce a random search point (x_t, y_t) based on $P_{I(t)}$.
 - 5: Query the payoffs $g(x_t, y_t), h(x_t, y_t)$
 - 6: **if** $(x_t, y_t) \notin H_{t-1}$ **then** $C_t = C_{t-1} + 1; H_t = H_{t-1} \cup \{(x_t, y_t)\}$.
 - 7: **else** $C_t = C_{t-1}; H_t = H_{t-1}$.
-

Algorithm 1 defines a class of algorithms subject to various probability distributions and samples the new strategy pair based on previous pairs and their payoffs. The initial search point (x_0, y_0) is independent of the problem, so we can choose any probability distribution P_\emptyset to initialise the algorithm. Subsequent strategy pairs are obtained by asking the oracle to apply a given variation operator to all previously queried search points subject to sample probability distribution $P_{I(t)}$. By specifying different sample probability distribution $P_{I(t)}$, Algorithm 1 represents various black-box optimisation algorithms, including several adversarial search (also called competitive coevolutionary) algorithms [46, 32] and randomised algorithms FINDPSNE (designed to learn the NE in bimatrix games) [27]. Note that the model only considers the cost of unique queries made by the algorithm (i.e. we check the search point in the previous query history H_t in line 6). We assume that an algorithm which makes the same query twice is only charged for the cost of one of the queries.

Now, we define the query complexity (or runtime) of black-box adversarial optimisation algorithms by extending the idea of the traditional single-objective black-box optimisation algorithm in [13, 43] and assuming $h = -g$ (i.e. zero-sum game) in this case.

Definition 4. Given any unrestricted black-box algorithm with unique query history A and the payoff function $g : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, $T(A, g)$ is the query complexity of A with respect to g and (x_t, y_t) is the search point generated by A if

$$T(A, g) := \inf \{ dC_t \in \mathbb{N} \mid (x_t, y_t) \in \arg \max_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} g(x, y) \}$$

where $d \in \{1, 2\}$. If the game is zero-sum, then $d = 1$. Otherwise, $d = 2$

Note that $T(A, g) \in \mathbb{R} \cup \{\infty\}$ is the number of payoff evaluations until A queries for the first time some $(x^*, y^*) \in \arg \max_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} g(x, y)$. Now, we can define what black-box complexity means with respect to a given class of adversarial optimisation algorithms and problem classes.

Definition 5. For a class \mathcal{G} of payoff functions $g : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, the A -black-box complexity of \mathcal{G} is defined as $T(A, \mathcal{G}) := \sup_{g \in \mathcal{G}} T(A, g)$, the runtime of A under the worst-case scenario on \mathcal{G} . Then, the \mathcal{A} -black-box complexity of \mathcal{G} is $T(\mathcal{A}, \mathcal{G}) := \inf_{A \in \mathcal{A}} T(A, \mathcal{G}) = \inf_{A \in \mathcal{A}} \sup_{g \in \mathcal{G}} T(A, g)$; the best or minimum complexity among all $A \in \mathcal{A}$ with respect to \mathcal{G} . If \mathcal{A} is the whole class of all black-box algorithms, we denote $T(\mathcal{A}, \mathcal{G})$ the unrestricted black-box complexity of \mathcal{G} .

We want to point out the difference between these two definitions. Definition 4 considers the query complexity for a particular algorithm and problem instance, and Definition 5 considers the black-box complexity for the best possible query complexity of all possible given algorithms under the worst-case scenario. The traditional black-box complexity theory characterises the difficulty of a certain class of problems and explores the limitations of the given black-box optimisation algorithms. We expect our extension to adversarial optimisation can provide similar insights as well.

4.2 A General Lower Bound for Black-Box Adversarial Optimisation

We first provide a lower bound of query complexity for a general class of black-box adversarial optimisation problems.

Theorem 4.1. *Let \mathcal{X} and \mathcal{Y} be any finite sets. Assume that $B \subset \mathbb{R}$ with $k := |B| \geq 2$. Consider any class of two-player zero-sum games $\mathcal{G} \subset \{g : \mathcal{X} \times \mathcal{Y} \rightarrow B\}$ such that for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, there exists a game $g_{x,y} \in \mathcal{G}$ which has (x, y) as unique, pure Nash Equilibrium. Then, the class \mathcal{G} has black box complexity at least $\lceil \log_k |\mathcal{X} \times \mathcal{Y}| \rceil - 1$.*

4.3 A General Lower Bound for Two-player Zero-Sum Bimatrix Games

In this subsection, we employ Yao's Principle and No Free Lunch Theorem to provide a general lower bound for query complexity of searching Nash Equilibrium in zero-sum bimatrix games, and this leads to a sharper lower bound compared to the previous result in [27].

Theorem 4.2. *Let \mathcal{A} be the set of all randomised algorithms defined by Algorithm 1 and $T(\mathcal{A}, P)$ ⁴ denote the query complexity of \mathcal{A} with respect to the input payoff matrix P for a two-player zero-sum game. Then, there exists an input matrix $P \in \mathbb{R}^{n \times n}$ with a unique pure Nash equilibrium (x^*, y^*) such that $\mathbb{E}(T(\mathcal{A}, P)) \geq (n + 1)/2$. Thus, the black-box complexity with respect to \mathcal{A} of the problem class consisting of all bimatrix games with a unique Nash Equilibrium is at least $(n + 1)/2$.*

Theorem 4.2 provides a sharper lower bound by a multiplicative factor 4 compared with the current best bound by [27]. This result also demonstrates that in a two-player zero-sum bimatrix game (with an $n \times n$ payoff matrix), there are complex instances where no randomised algorithm can achieve better than $O(n)$ query complexity unless additional problem structure is provided.

4.4 Applications on Two-Player Zero-Sum Games

4.4.1 Introduction to Binary Voting Games

This section provides some example applications of our black-box complexity results. Voting games are popular games studied in game theory, computational social choice theory [5, 4, 16] and Boolean games [18]. Voting is considered a fundamental tool for analysing multi-agent systems [16, 1]. We start with simple binary voting games in which the outcome or payoff is 0 or 1 (or -1 and 1). These games also play a role in the analysis of Boolean functions [31].

Convergence to NE in plurality voting has been studied from the perspective of social choice theory and researchers specify certain conditions to guarantee the voting games to converge to NEs [28]. Some natural question arises: are there any randomised algorithms that can find these NEs in voting games efficiently? What are the limitations of these black-box optimisation algorithms? Using the black-box complexity analysis, we can answer the questions about the efficiency/inefficiency of black-box optimisation algorithms on binary voting games.

⁴Note that $T(\mathcal{A}, P) = T(\mathcal{A}, g)$ where $g(x, y) := e_y^T P e_x$ with e_x, e_y denote the elementary probability distribution over probability simplex $\Delta_{\{0,1\}^n}$

We formulate binary voting games in the context of adversarial optimisation as follows. Consider two parties represented by vectors $x, y \in \{0, 1\}^n$ where $n \in \mathbb{N}$. Each group has n members that either “in favour” (encoded by 1) or “against” (encoded by 0) a particular proposal or decision. One group seeks a strategy x^* that maximises its minimum gains against any strategy of the other group, while the other group seeks a strategy y^* such that its choice minimises the maximum gains of the first group. It essentially forms a two-player zero-sum game.

Definition 6. For $\mathcal{X} = \mathcal{Y} = \{0, 1\}^n$, the payoff function $\text{DIAGONAL} : \mathcal{X} \times \mathcal{Y} \rightarrow \{-1, 1\}$ is

$$\text{DIAGONAL}(x, y) := \begin{cases} 1 & |y|_1 \leq |x|_1 \\ -1 & \text{otherwise.} \end{cases}$$

In Definition 6, we present the votes of both groups by binary bitstrings and the payoff g can be viewed as a binary voting game where the payoff only depends on which group has the stronger majority “influence”. If one group has a stronger ‘in favour’ “influence” in the sense of the number of the support votes (i.e. the number of 1 in the encoding binary bitstring), then we get a payoff 1 and -1 otherwise. We are interested in computing NE in these two-player zero-sum games, i.e. solving $(x^*, y^*) \in \arg \max_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} g(x, y)$. Notice that in DIAGONAL , $(x_n, y_n) = (1^n, 1^n)$ is the unique NE optimum. In this optimum, neither of the two groups is willing to deviate from affecting their payoff $g(x_n, y_n)$ anymore. This exactly coincides with the definition of NE.

Next, we consider a different binary voting game, denoted by PLATEAU . To make the binary voting game more challenging, we introduce some plateaus in games.

Definition 7. For $\mathcal{X} = \mathcal{Y} = \{0, 1\}^n$, a constant $\delta \in (0, 1)$, the payoff function $\text{PLATEAU} : \mathcal{X} \times \mathcal{Y} \rightarrow \{-1, 1\}$ is defined as

$$\text{PLATEAU}(x, y) := \begin{cases} f(y) & \text{if } ||x|_1 - \frac{n}{2}| < \frac{\delta n}{2} \\ g(x, y) & \text{otherwise} \end{cases}$$

where $f : \mathcal{Y} \rightarrow \{-1, 1\}$ and $g : \mathcal{X} \times \mathcal{Y} \rightarrow \{-1, 1\}$ are any functions such that the NE of PLATEAU is $(x^*, y^*) \notin \{(x, y) \mid ||x|_1 - \frac{n}{2}| < \frac{\delta n}{2}\}$.

Definition 7 introduces a plateau when comparing the “influence” between two groups and defines a general class of pseudo-Boolean benchmarks with a plateau. Imagine a committee deciding on a new policy where there are two groups with equal voting power. If the votes from group \mathcal{X} are balanced or nearly balanced (within the plateau), then the votes from the second group (\mathcal{Y}) come into play. It is like their votes are the tiebreaker. If the second group votes in favour, then the policy passes; if they vote against it, then it fails. If the votes from group \mathcal{X} are outside the plateau, then the payoff is not restricted to be determined by $y \in \mathcal{Y}$.

Finally, we define game instances generated by (u, v) where $u, v \in \{0, 1\}^n$.

Definition 8 ((u, v) -game instance). For all $u, v \in \{0, 1\}^n$, given $f : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \mathbb{R}$, we define the (u, v) -instance of f , denoted by $f_{(u,v)}$, as $f_{(u,v)}(x, y) := f(u \oplus x, v \oplus y)$.

We can see that for any $u, v \in \{0, 1\}^n$, $f_{(u,v)}$ generates the same payoff landscape as f . In this paper, f will be either DIAGONAL or PLATEAU . We defer more details to Section G⁵.

4.4.2 Black-Box Complexity of Learning Nash Equilibrium in Binary Voting Games

First, let us illustrate how Theorem 4.1 and Theorem 4.2 work on these simple examples. If we consider a general class of black-box optimisation algorithms defined by Algorithm 1 on binary voting games with a unique Nash Equilibrium, then we provide a general lower bound of black-box complexity as follows.

Theorem 4.3. *The black-box complexity with respect to Algorithm 1 of the binary voting games with problem size $n \in \mathbb{N}$ and a unique Nash Equilibrium is $e^{\Omega(n)}$.*

Theorem 4.3 means that there exist no universal good black-box optimisation algorithms defined by Algorithm 1 that can solve all binary voting games with unique Nash Equilibrium efficiently, (i.e. with polynomial query complexity of the problem size). To yield a better performance of black-box

⁵We defer the definition of xor \oplus to Section G in the supplementary material.

optimisation algorithms on binary voting games, we need to specify the problem class we work on. Next, we consider `DIAGONAL` and explore the black-box complexity with respect to the class of black-box optimisation algorithms \mathcal{A} defined by Algorithm 1 of `DIAGONAL`. This reveals that `DIAGONAL` is a feasible benchmark problem for black-box adversarial optimisation algorithms.

Theorem 4.4. *Given the game class*

$$\text{DIAGONAL}_n := \{\text{DIAGONAL}_{(u,v)} \mid (u, v) \in \{0, 1\}^n \times \{0, 1\}^n\},$$

the black-box complexity with respect to Algorithm 1 of `DIAGONAL` is $\Theta(n)$.

Theorem 4.4 implies that `DIAGONAL`_{*n*} is a sensible maximin benchmark for testing black-box optimisation algorithms. It means that if we restrict the problem class to a certain class with a specific structure, then it is possible to solve them in polynomial query complexity.

We have seen the black-box complexity results on `DIAGONAL`_{*n*}. Next, we start to consider more challenging binary voting games, `PLATEAU`. We are interested in whether there is any efficient black-box adversarial optimisation that can solve `PLATEAU`_{*n*} in polynomial query complexity (i.e. $O(n)$). To answer this question, we need to compute its black-box complexity.

Theorem 4.5. *Given the game class,*

$$\text{PLATEAU}_n := \{\text{PLATEAU}_{(u,v)} \mid (u, v) \in \{0, 1\}^n \times \{0, 1\}^n\},$$

the black-box complexity with respect to the class of algorithms defined by Algorithm 1 of `PLATEAU` is $e^{\Omega(n)}$.

Theorem 4.5 implies that all black-box adversarial optimisation algorithms defined by unrestricted model (i.e. Algorithm 1) have exponential runtime on `PLATEAU`_{*n*}. They need at least an exponentially large query complexity with respect to the problem size *n*. It is evident that `PLATEAU` is too challenging that it may not be a proper benchmark for black-box adversarial optimisation algorithms.

4.4.3 Summary

Our study introduces the concept of black-box complexity in binary voting games, providing insights into the challenges faced by general black-box adversarial optimisation algorithms. These examples illustrate two kinds of problems within the general class of binary voting games with unique NE: the polynomial-solvable class (i.e., there exists an algorithm that can solve all problem instances of this class in polynomial runtime) and the non-polynomial-solvable class (i.e., there exists no algorithm that can solve all problem instances of this class in polynomial runtime).

Theorem 4.3 rigorously show that no universal algorithm can efficiently learn the unique Nash Equilibrium (NE) in these games due to their structure-free nature, where efficiency means small query complexity. However, when assuming specific problem structures, such as `DIAGONAL`_{*n*}, it is possible to come up with a better algorithm which achieves better polynomial query complexity as shown in Theorem 4.4. `DIAGONAL`_{*n*} can be a promising benchmark for evaluating black-box algorithms. Additional assumptions on the payoff function, like those in `PLATEAU`_{*n*} problems, proved in Theorem 4.5, do not lower the difficulty of the problems. This emphasises the need for a more careful selection of benchmarks. Black-box complexity emerges as a valuable tool for distinguishing between potentially easy and hard problem instances, guiding the design of black-box algorithms.

5 Further Related Work

5.1 Co-evolutionary Search Heuristics

Next, we provide some practical examples of adversarial optimisation. There are various adversarial search heuristics, including competitive co-evolutionary algorithms (CoEAs) [35, 26]. CoEAs are a class of algorithms applied in game-theoretic and strategic adversarial optimisation scenarios. For example, CoEAs are used to solve maximin optimisation in a cybersecurity context [32] and to enhance GANs by using a co-evolutionary approach for image translation [41]. Similarly, the neural architecture search system Lipizzaner employs co-evolutionary adversarial search to find suitable neural architectures for GANs [42].

Although several applications of adversarial (or co-evolutionary) search exist, there is limited theoretical literature on this topic. Lehre [26] demonstrated that the running time of the co-evolutionary

algorithm PDCoEA on instances of the discrete bilinear problem is polynomial. Later, Hevia Fajardo et al. [22] showed a weakness of RLS-PD and the sufficiency of a simple archive to prevent evolutionary forgetting. On the other hand, regarding the general black-box optimisation framework, Wolpert and Macready [46] showed there exists a “free lunch” in such adversarial (or co-evolutionary) optimisation setting – there exist some algorithms have better performance than others averaged across all possible problem instances in adversarial optimisation with respect to the maximin solution concept (see Definition 10).

5.2 Query Complexity of Learning in Games

The query complexity of various solution concepts in zero-sum and general-sum multi-player games has been well studied (for an overview, see [2, 27]). It is well-known that converging to an exact (mixed strategy) Nash Equilibrium in general-sum matrix games is PPAD-complete [7]. Chen and Deng [6] provide a simplified proof of the computational complexity of Nash Equilibrium in two-player games. Significant focus has also been placed on how the game dynamics converge to the Nash equilibrium or approximate it [27, 25, 17, 34].

Panageas et al. [34] considered the complexity of fictitious play in two-player potential games with a unique Nash Equilibrium (NE). They proved the existence of a hard instance which requires query complexity $\Omega\left(4^n ((n/2 - 2)!)^4\right)$ where n refers to the number of actions. They showed that fictitious play can take exponential time (in the number of strategies) to reach a unique Nash Equilibrium, even when the game is restricted to two agents and arbitrary tie-breaking rules, by constructing a two-player coordination game with a unique Nash Equilibrium. Unlike previous work, our paper focuses on a more general class of games: the entire class of two-player zero-sum games with a unique NE rather than just potential games. We show that all black-box adversarial algorithms, including fictitious play, exhibit the same average performance across all problem instances in terms of query complexity.

6 Conclusion and Discussion

Utilising the tools from game theory and Yao’s principle, we rigorously prove the impossibility results for a universally effective adversarial algorithm applicable across various problem classes in black-box adversarial optimisation. We emphasise the impact of solution concept selection on the feasibility of a “free lunch” in adversarial optimisation. Additionally, we introduce the notion of black-box complexity in black-box adversarial optimisation and characterise the difficulty of learning the unique optimum in adversarial optimisation and solving two-player zero-sum games.

The results from this paper build up a foundation for future studies on the strengths and limitations of adversarial optimisation. More specifically, it highlights the need for more comprehensive benchmarks and careful selections of solution concepts when using any black-box adversarial optimisation algorithms. Moreover, no black-box optimisation algorithm for learning the Nash equilibrium in two-player zero-sum games can exceed the logarithmic complexity relative to search space size. Meanwhile, no algorithm can solve any bimatrix game with unique NE faster than the linear query complexity in terms of the size of input payoff matrices.

Although our work makes a first step towards the new NFL and BBC results on black-box adversarial optimisation, we want to point out some limitations of our current work and list them as our future work. Firstly, our theoretical results build on discrete exponential large search spaces rather than countably infinite (e.g. \mathbb{N}) or uncountable infinite (e.g. \mathbb{R}) sets. To generalise our results to infinite sets, we might require further assumptions on our search spaces. Secondly, our NFL focuses on two-player zero-sum games with unique NE.

Future direction of our work includes extending Theorem 3.1 to other solution concepts like mixed strategy Nash Equilibrium or exploring different possible solution concepts which may exhibit free lunch or not. Additionally, it is interesting to generalise NFL and BBC analysis to zero-sum games with multiple NEs and more general search spaces. Finally, it is interesting to analyse other black-box models, such as unbiased black-box complexity models, to characterise the difficulty of adversarial problems that different classes of search heuristics can solve.

Acknowledgements

We would like to thank Dr Alistair Benford and Dr Mario Alejandro Hevia Fajardo for the fruitful discussion and comments on an earlier draft of this paper. This work was supported by a Turing AI Fellowship (EPSRC grant ref EP/V025562/1).

References

- [1] Haris Aziz, Felix Brandt, Edith Elkind, and Piotr Skowron. Computational social choice: The first ten years and beyond. *Computing and Software Science: State of the Art and Perspectives*, pages 48–65, 2019.
- [2] Yakov Babichenko. Informational Bounds on Equilibria (a survey). *ACM SIGecom Exchanges*, 17(2):25–45, 2020.
- [3] Alistair Benford and Per Kristian Lehre. Runtime Analysis of Coevolutionary Algorithms on a Class of Symmetric Zero-Sum Games. In *GECCO’24: Genetic and Evolutionary Computation Conference*. Association for Computing Machinery (ACM), 2024.
- [4] Felix Brandt, Vincent Conitzer, and Ulle Endriss. Computational social choice. *Multiagent systems*, 2:213–284, 2012.
- [5] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. *Handbook of computational social choice*. Cambridge University Press, 2016.
- [6] Xi Chen and Xiaotie Deng. Settling the complexity of two-player Nash equilibrium. In *FOCS*, volume 6, pages 261–272, 2006.
- [7] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a Nash equilibrium. *Communications of the ACM*, 52(2):89–97, 2009.
- [8] Benjamin Doerr, Timo Kötzing, Johannes Lengler, and Carola Winzen. Black-box complexities of combinatorial problems. *Theoretical Computer Science*, 471:84–106, 2013.
- [9] Benjamin Doerr, Carola Doerr, and Franziska Ebel. From black-box complexity to designing new genetic algorithms. *Theoretical Computer Science*, 567:87–104, 2015.
- [10] Carola Doerr. Complexity Theory for Discrete Black-Box Optimization Heuristics. *Theory of Evolutionary Computation: Recent Developments in Discrete Optimization*, pages 133–212, 2020.
- [11] Stefan Droste, Thomas Jansen, and Ingo Wegener. Optimization with randomized search heuristics—the (A) NFL theorem, realistic scenarios, and difficult functions. *Theoretical Computer Science*, 287(1):131–144, 2002.
- [12] Stefan Droste, Thomas Jansen, and Ingo Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276(1-2):51–81, April 2002.
- [13] Stefan Droste, Thomas Jansen, and Ingo Wegener. Upper and lower bounds for randomized search heuristics in black-box optimization. *Theory of computing systems*, 39(4):525–544, 2006.
- [14] Daniel Golovin, Benjamin Solnik, Subhodeep Moitra, Greg Kochanski, John Karro, and David Sculley. Google vizier: A service for black-box optimization. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1487–1495, 2017.
- [15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. *Advances in Neural Information Processing systems*, 27, 2014.
- [16] Umberto Grandi, Davide Grossi, and Paolo Turrini. Equilibrium refinement through negotiation in binary voting. In *Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI’15*, page 540–546. AAAI Press, 2015.

- [17] Hedi Hadiji, Sarah Sachs, Tim van Erven, and Wouter M Koolen. Towards Characterizing the First-order Query Complexity of Learning (Approximate) Nash Equilibria in Zero-sum Matrix Games. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=IPNg84RF1k>.
- [18] Paul Harrenstein, Wiebe van der Hoek, John-Jules Meyer, and Cees Witteveen. Boolean games. In *Proceedings of the 8th Conference on Theoretical Aspects of Rationality and Knowledge*, TARK '01, page 287–298, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
- [19] Erik Hemberg, Jamal Toutouh, Abdullah Al-Dujaili, Tom Schmiedlechner, and Una-May O'Reilly. Spatial coevolution for generative adversarial network training. *ACM Transactions on Evolutionary Learning and Optimization*, 1(2):1–28, 2021.
- [20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-scale Update Rule Converge to a Local Nash Equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017.
- [21] Mario Alejandro Hevia Fajardo and Per Kristian Lehre. How fitness aggregation methods affect the performance of competitive coeas on bilinear problems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '23, page 1593–1601, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701191.
- [22] Mario Alejandro Hevia Fajardo, Per Kristian Lehre, and Shishen Lin. Runtime analysis of a co-evolutionary algorithm: Overcoming negative drift in maximin-optimisation. In *Proceedings of the 17th ACM/SIGEVO Conference on Foundations of Genetic Algorithms*, FOGA '23, page 73–83, New York, NY, USA, 2023. Association for Computing Machinery.
- [23] Y.C. Ho and D.L. Pepyne. Simple Explanation of the No-Free-Lunch Theorem and Its Implications. *Journal of Optimization Theory and Applications*, 115(3):549–570, December 2002.
- [24] Junling Hu, Michael P Wellman, et al. Multiagent reinforcement learning: theoretical framework and an algorithm. In *ICML*, volume 98, pages 242–250, 1998.
- [25] Fivos Kalogiannis and Ioannis Panageas. Zero-sum Polymatrix Markov Games: Equilibrium Collapse and Efficient Computation of Nash Equilibria. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=NGiq8qCQNk>.
- [26] Per Kristian Lehre. Runtime Analysis of Competitive co-Evolutionary Algorithms for Maximin Optimisation of a Bilinear Function. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '22, page 1408–1416, 2022.
- [27] Arnab Maiti, Ross Boczar, Kevin Jamieson, and Lillian J Ratliff. Query-Efficient Algorithms to Find the Unique Nash Equilibrium in a Two-Player Zero-Sum Matrix Game. *arXiv preprint arXiv:2310.16236*, 2023.
- [28] Reshef Meir, Maria Polukarov, Jeffrey S. Rosenschein, and Nicholas R. Jennings. Convergence to equilibria in plurality voting. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, page 823–828. AAAI Press, 2010.
- [29] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge university press, 1995.
- [30] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vazirani Vijay V. *Algorithmic Game Theory*. Cambridge University Press, 2007.
- [31] Ryan O'Donnell. Some topics in analysis of Boolean functions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing (STOC)*, pages 569–578, 2008.
- [32] Una-May O'Reilly, Per Kristian Lehre, Mario Hevia Fajardo, Jamal Toutouh, and Erik Hemberg. Analysis of a pairwise dominance coevolutionary algorithm and defendit. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '23, page 1027–1035, New York, NY, USA, 2023. Association for Computing Machinery.

- [33] Martin J Osborne and Ariel Rubinstein. *A course in game theory*. MIT press, 1994.
- [34] Ioannis Panageas, Nikolas Patris, Stratis Skoulakis, and Volkan Cevher. Exponential Lower Bounds for Fictitious Play in Potential Games. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=tkenkPYkxj>.
- [35] Elena Popovici, Anthony Bucci, R. Paul Wiegand, and Edwin D. De Jong. Coevolutionary Principles. In Grzegorz Rozenberg, Thomas Bäck, and Joost N. Kok, editors, *Handbook of Natural Computing*, pages 987–1033. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [36] Elad Sarafian, Mor Sinay, Yoram Louzoun, Noa Agmon, and Sarit Kraus. Explicit gradient learning for black-box optimization. In *ICML*, pages 8480–8490, 2020.
- [37] Cullen Schaffer. A Conservation Law for Generalization Performance. In *Proceedings of the Eleventh International Conference on International Conference on Machine Learning, ICML'94*, page 259–265, San Francisco, CA, USA, 1994.
- [38] C. Schumacher, M. D. Vose, and L. D. Whitley. The No Free Lunch and Problem Description Length. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, GECCO'01*, page 565–570, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
- [39] Travis C. Service and Daniel R. Tauritz. A No-Free-Lunch Framework for Coevolution. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO '08*, page 371–378, New York, NY, USA, 2008. Association for Computing Machinery.
- [40] Lloyd S Shapley, Samuel Karlin, and HF Bohnenblust. Solutions of discrete, two-person games. *Contributions to the Theory of Games*, 1950.
- [41] Han Shu, Yunhe Wang, Xu Jia, Kai Han, Hanting Chen, Chunjing Xu, Qi Tian, and Chang Xu. Co-evolutionary compression for unpaired image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3235–3244, 2019.
- [42] Jamal Toutouh, Erik Hemberg, and Una-May O'Reilly. Spatial evolutionary generative adversarial networks. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '19*, page 472–480, New York, NY, USA, 2019. Association for Computing Machinery.
- [43] Carola Winzen. Toward a complexity theory for randomized search heuristics : black-box models. *Adviser: Mehlhorn, Kurt. PhD thesis. Germany, Germany: Max-Planck-Institut für Informatik*, 2011.
- [44] David H Wolpert. The Supervised Learning No-Free-Lunch Theorems. *Soft computing and industry: Recent applications*, pages 25–42, 2002.
- [45] David H Wolpert and William G Macready. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.
- [46] David H Wolpert and William G Macready. Coevolutionary Free Lunches. *IEEE Transactions on Evolutionary Computation*, 9(6):721–735, December 2005.
- [47] Andrew Chi-Chin Yao. Probabilistic computations: Toward a unified measure of complexity. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 222–227, October 1977.

Appendix / Supplementary material

Contents

1	Introduction	1
1.1	Black-Box Optimisation and the No Free Lunch Theorem	1
1.2	Challenges and Technical Overview of the NFL and BBC Results	2
1.3	Contribution	3
2	Preliminaries	3
2.1	Notation	3
2.2	Solution Concepts	3
3	No Free Lunch Theorem for Computing Nash Equilibrium in Adversarial Optimisation	4
4	Black-Box Complexity of Black-Box Adversarial Optimisation	6
4.1	The Unrestricted Black-Box Model and the Black-Box Complexity	6
4.2	A General Lower Bound for Black-Box Adversarial Optimisation	7
4.3	A General Lower Bound for Two-player Zero-Sum Bimatrix Games	7
4.4	Applications on Two-Player Zero-Sum Games	7
4.4.1	Introduction to Binary Voting Games	7
4.4.2	Black-Box Complexity of Learning Nash Equilibrium in Binary Voting Games	8
4.4.3	Summary	9
5	Further Related Work	9
5.1	Co-evolutionary Search Heuristics	9
5.2	Query Complexity of Learning in Games	10
6	Conclusion and Discussion	10
A	Two Other Solution Concepts	15
B	Technical Lemmas for Games with Unique Nash Equilibrium	15
C	Analytic Tools from Probability Theory and Analysis of Randomised Algorithms	16
C.1	Yao’s Principle	16
C.2	Markov’s inequality and Chernoff’s bound	16
D	Omitted Proofs	16
E	Further Explanations for Sub-Problem Class	21
F	Further Explanations for the role of Corollary 1 and Lemma 1 in Theorem 3.1	22
G	Further Explanations for bitwise exclusive or in Black-Box Complexity Analysis	22

A Two Other Solution Concepts

Definition 9 (Maximisation Over All Test Cases [39]). Suppose a set of candidate solutions, \mathcal{X} , and a set of test cases, \mathcal{Y} . Given an interaction function or payoff function, $g : \mathcal{X} \times \mathcal{Y} \rightarrow O$, where O is an ordered set which determines the outcome of candidate solution \mathcal{X} on test \mathcal{Y} , the solution concept is defined by: $S = \{x^* \in \mathcal{X} \mid \forall x \in \mathcal{X}, \forall y \in \mathcal{Y}, g(x^*, y) \geq g(x, y)\}$.

Next, we define the Maximin solutions are those solution configurations which maximise the minimum outcome over all test cases.

Definition 10 (Maximin [46]). Suppose a set of candidate solutions, \mathcal{X} , and a set of test cases, \mathcal{Y} . Given an interaction function or payoff function, $g : \mathcal{X} \times \mathcal{Y} \rightarrow O$, where O is an ordered set which determines the outcome of candidate solution \mathcal{X} on test \mathcal{Y} , the solution concept is defined by:

$$S = \{x^* \in \mathcal{X} \mid \forall x \in \mathcal{X}, \min_{y \in \mathcal{Y}} g(x^*, y) \geq \min_{y \in \mathcal{Y}} g(x, y)\}.$$

Notice that in Definition 9, this solution concept considers the candidate solution x with respect to all test cases y . Definition 10 considers the candidate solution with respect to the worst-case scenario. If \mathcal{Y} is compact, then these two solution concepts are essentially the same. Compared with Nash Equilibrium in Definition 1, these two solution concepts only focus on finding the best candidate solution and the algorithm is not required also to produce an opposing ‘‘optimal’’ solution y^* .

B Technical Lemmas for Games with Unique Nash Equilibrium

We adopt the notation in [27].

Definition 11. Given a two-player zero-sum game on a payoff matrix $P \in \mathbb{R}^{n \times n}$. Let Δ_n denote the n -dimensional probability simplex (i.e. for $p \in \Delta_n$, $\sum_{i=1}^n p_i = 1$ and $p_i \geq 0$). A pair (p^*, q^*) is a (mixed strategy) Nash Equilibrium of an input matrix $P \in \mathbb{R}^{n \times n}$ if and only if the following holds for all $(p, q) \in \Delta_n \times \Delta_n$:

$$\langle p, Pq^* \rangle \leq \langle p^*, Pq^* \rangle \leq \langle p^*, Pq \rangle.$$

In this paper, we are only interested in Pure Strategy Nash Equilibrium.

Definition 12. For any payoff matrix $P \in \mathbb{R}^{n \times n}$, let $V_P^* := \max_{p \in \Delta_n} \min_{q \in \Delta_n} \langle p, Pq \rangle$. In particular, we say an entry $(i^*, j^*) \in [n] \times [n]$ is a unique pure strategy Nash Equilibrium if $(e_{i^*}, e_{j^*}) \in \Delta_n \times \Delta_n$ is a unique Nash Equilibrium of P where e_{i^*}, e_{j^*} are the unit probability vector.

We refer to the following lemma from [40].

Lemma 3 (Lemma 1 in [27]). Consider a matrix $A \in \mathbb{R}^{n \times n}$ with a unique Nash equilibrium (p^*, q^*) . The following conditions hold:

1. The support sizes of p^* and q^* are equal (i.e., $|\text{supp}(p^*)| = |\text{supp}(q^*)|$).
2. $V_A^* = \langle e_i, Aq^* \rangle$ for all $i \in \text{supp}(p^*)$, and $V_A^* > \langle e_i, Aq^* \rangle$ for all $i \notin \text{supp}(p^*)$.
3. $V_A^* = \langle p^*, Ae_j \rangle$ for all $j \in \text{supp}(q^*)$, and $V_A^* < \langle p^*, Ae_j \rangle$ for all $j \notin \text{supp}(q^*)$.

Lemma 3 allows us to derive the following result, which is proved in Appendix B.1 in [27].

Lemma 4 (Lemma 2 in [27]). Consider a matrix $P \in \mathbb{R}^{n \times n}$ with a unique Nash equilibrium (p^*, q^*) . Consider a submatrix M of P with row index set I_X and column index set I_Y . If $\text{SUPP}(p^*) \subseteq I_X$ and $\text{SUPP}(q^*) \subseteq I_Y$, then (\hat{p}, \hat{q}) is the unique Nash equilibrium of M where $(\hat{p})_i = (p^*)_i$ for all $i \in I_X$ and $(\hat{q})_j = (q^*)_j$ for all $j \in I_Y$. Moreover, $V_M^* = V_P^*$.

C Analytic Tools from Probability Theory and Analysis of Randomised Algorithms

C.1 Yao's Principle

In this section, we provide the mathematical tool that we use in the rest of the analysis. Yao's Principle [47] can be used to give lower bounds of a class of randomised algorithms.

Theorem C.1 (Yao's Principle [47]). *Let Π be a problem with a finite set \mathcal{I} of input instances (of a fixed size) permitting a finite set \mathcal{A} of deterministic algorithms. Let p be a probability distribution over \mathcal{I} and q be a probability distribution over \mathcal{A} . Then,*

$$\min_{A \in \mathcal{A}} \mathbb{E}(T(I_p, A)) \leq \max_{I \in \mathcal{I}} \mathbb{E}(T(I, A_q))$$

where I_p denotes a random input chosen from \mathcal{I} according to p , A_q a random algorithm chosen from \mathcal{A} according to q and $T(I, A)$ denote the runtime of algorithm A on input I .

C.2 Markov's inequality and Chernoff's bound

Theorem C.2 (Markov's inequality). *Given a non-negative random variable X and a $a > 0$, $\Pr(X \geq a) \leq \frac{\mathbb{E}(X)}{a}$.*

Theorem C.3 (Chernoff's bound). *Given $X := \sum_{i=1}^n X_i$ where $X_i \sim \text{Ber}(p_i)$ where $\text{Ber}(p_i)$ is a Bernoulli random variable with probability p_i and all X_i are independent. Let $\mu := \mathbb{E}(X) = \sum_{i=1}^n p_i$. Then, for $\delta \in (0, 1)$,*

$$\Pr(|X - \mu| \geq \delta\mu) \leq 2e^{-\mu\delta^2/3}.$$

D Omitted Proofs

Corollary 2. Let $|\mathcal{X}| = |\mathcal{Y}|$ and $\mathcal{F} \subseteq \{g : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}\}$ be any set of two-player zero-sum games with a unique NE. Assume furthermore that \mathcal{F} is c.u.p.. Then, for any game $g \in \mathcal{F}$, let $(x^*, y^*) \in \mathcal{X} \times \mathcal{Y}$ denote the unique Nash Equilibrium of g . For any permutations over \mathcal{X}, \mathcal{Y} denoted by $\sigma \in \mathfrak{S}(\mathcal{X}), \tau \in \mathfrak{S}(\mathcal{Y})$, $(\sigma \otimes \tau) \circ g \in \mathcal{F}$ exhibits the same unique Nash Equilibrium (x^*, y^*) . Moreover, if for any $x_0 \neq x^*$ and $y_0 \neq y^*$, the restriction of g on $(\mathcal{X} \setminus \{x_0\}) \times (\mathcal{Y} \setminus \{y_0\})$, denoted by $g|_{(\mathcal{X} \setminus \{x_0\}) \times (\mathcal{Y} \setminus \{y_0\})}$, exhibits the same unique Nash Equilibrium (x^*, y^*) .

Proof of Lemma 1. We employ the result in [27] (See Lemma 4). For any permutations on \mathcal{X}, \mathcal{Y} denoted by $\sigma \in \mathfrak{S}(\mathcal{X}), \tau \in \mathfrak{S}(\mathcal{Y})$, $(\sigma \otimes \tau) \circ g \in \mathcal{F}$ is defined as for any $x \in \mathcal{X}, y \in \mathcal{Y}$,

$$(\sigma \otimes \tau) \circ g(x, y) = g(\sigma(x), \tau(y)).$$

We consider the payoff matrix defined by $P = (p_{i,j})$ where $p_{i,j} = g(x_i, y_j)$ with $x_i \in \mathcal{X}, y_j \in \mathcal{Y}$. We denote row index set by $I_{\mathcal{X}} \subseteq \mathbb{N}$ and column index set by $I_{\mathcal{Y}} \subseteq \mathbb{N}$. Notice that $|I_{\mathcal{X}}| = |\mathcal{X}| = |\mathcal{Y}| = |I_{\mathcal{Y}}|$. We denote the row index by $i \in I_{\mathcal{X}}$ and the column index by $j \in I_{\mathcal{Y}}$. $P \in \mathbb{R}^{|I_{\mathcal{X}}| \times |I_{\mathcal{Y}}|}$ is the payoff matrix of two-player zero-sum game g with a unique Nash Equilibrium (x^*, y^*) . Let us denote the row index of x^* in the payoff matrix P by i_* and the column index of y^* by j_* in the payoff matrix P . So $(e_{i_*}, e_{j_*}) \in \Delta_n \times \Delta_n$ is the unique Nash Equilibrium in the form of probability vector.

Now, given any permutation σ, τ , we consider a submatrix P' such that $P' = (p_{\sigma(i), \tau(j)})$ with a new row index set $I'_{\mathcal{X}} \subseteq \mathbb{N}$ and column index set $I'_{\mathcal{Y}} \subseteq \mathbb{N}$. Since σ only shuffles around the row indices in $I_{\mathcal{X}}$, $I'_{\mathcal{X}} \simeq I_{\mathcal{X}}$ with the isomorphism σ . Similarly, we have $I'_{\mathcal{Y}} \simeq I_{\mathcal{Y}}$ with the isomorphism τ . Now, we can see that $i^* = \text{SUPP}(e_{j_*}) \in I'_{\mathcal{X}}$ and $j^* = \text{SUPP}(e_{i_*}) \in I'_{\mathcal{Y}}$, so $(\hat{p}, \hat{q}) \in \Delta_n \times \Delta_n$ is the unique Nash Equilibrium of P' (in the form of a probability vector) where $(\hat{p})_i = (e_{i_*})_i$ for all $i \in I'_{\mathcal{X}}$ and $(\hat{q})_j = (e_{j_*})_j$ for all $j \in I'_{\mathcal{Y}}$. In other words, $(\hat{p}, \hat{q}) = (e_{i_*}, e_{j_*})$. So by using Lemma 4, we can conclude that $(x^*, y^*) \in \mathcal{X} \times \mathcal{Y}$ is the unique Nash Equilibrium of $(\sigma \otimes \tau) \circ g$.

For the second claim, we construct a submatrix Q such that $Q = (q_{i,j})$ where $q_{i,j} = g(x_i, y_j)$ with $x_i \in \mathcal{X} \setminus \{x_0\}, y_j \in \mathcal{Y} \setminus \{y_0\}$. We denote the row index set $I_{\mathcal{X} \setminus \{x_0\}}$ and the column index set $I_{\mathcal{Y} \setminus \{y_0\}}$. Since $x_0 \neq x^*$ and $y_0 \neq y^*$, $i^* = \text{SUPP}(e_{i_*}) \in I_{\mathcal{X} \setminus \{x_0\}}$ and $j^* = \text{SUPP}(e_{j_*}) \in I_{\mathcal{Y} \setminus \{y_0\}}$, by using

Lemma 4 again, we can conclude that (x^*, y^*) is the unique Nash Equilibrium of $g|_{(\mathcal{X} \setminus \{x_0\}) \times (\mathcal{Y} \setminus \{y_0\})}$. We complete the proof. \square

Lemma 1. If \mathcal{F} is c.u.p., then $\mathcal{F}((x_1, y_1), (b_1, b_2))$ is also c.u.p..

Proof of Lemma 1. For any $\sigma' \in \mathfrak{S}(\mathcal{X} \setminus \{x_1\})$, $\tau' \in \mathfrak{S}(\mathcal{Y} \setminus \{y_1\})$ and any $g' \in \mathcal{F}((x_1, y_1), (b_1, b_2))$, we want to show $(\sigma'\tau')g' \in \mathcal{F}((x_1, y_1), (b_1, b_2))$. We consider the extensions of each permutation. We define $\sigma : \mathcal{X} \rightarrow \mathcal{X}$ by its restriction on $\mathcal{X} \setminus \{x_1\}$ as

$$\sigma|_{\mathcal{X} \setminus \{x_1\}} = \sigma' \text{ and } \sigma(x_1) = x_1.$$

We define $\tau : \mathcal{Y} \rightarrow \mathcal{Y}$ by its restriction on $\mathcal{Y} \setminus \{y_1\}$ is

$$\tau|_{\mathcal{Y} \setminus \{y_1\}} = \tau' \text{ and } \tau(y_1) = y_1.$$

Now, let g be the extension of g' which satisfies (1), (2) and (3) of Definition 3 (such an extension g exists since we take $g' \in \mathcal{F}((x_1, y_1), (b_1, b_2))$ and it follows from the definition of sub-problem class). As $g \in \mathcal{F}$, so are $(\sigma\tau)g \in \mathcal{F}$. Thus, we construct $(\sigma\tau)g$ as an extension of $(\sigma'\tau')g'$ and $(\sigma\tau)g$ satisfies (1), (2) and (3) of Definition 3. So $(\sigma'\tau')g' \in \mathcal{F}((x_1, y_1), (b_1, b_2))$. \square

Lemma 2. For all $(x_1, y_1), (x_2, y_2) \in \mathcal{X} \times \mathcal{Y}$ and $b_1 : \mathcal{Y} \rightarrow \mathbb{R}$, $b_2 : \mathcal{X} \rightarrow \mathbb{R}$, we have the isomorphism⁶:

$$\mathcal{F}((x_1, y_1), (b_1, b_2)) \simeq \mathcal{F}((x_2, y_2), (b_1, b_2)).$$

Proof of Lemma 2. To prove the isomorphism, we need to find a bijection between these two sets. We consider the following map between sets with , defined by:

$$\phi : \mathcal{F}((x_1, y_1), (b_1, b_2)) \rightarrow \mathcal{F}((x_2, y_2), (b_1, b_2)) \text{ where } \phi(g') = h' \text{ such that}$$

for $(x, y) \in (\mathcal{X} \setminus \{x_1, x_2\}) \times (\mathcal{Y} \setminus \{y_1, y_2\})$,

$$h'(x, y) = \phi(g')(x, y) = g'(x, y).$$

For $y \in \mathcal{Y} \setminus \{y_2\}$, $h'(x_1, y) = \phi(g')(x_2, y) = g'(x_2, y)$.

For $x \in \mathcal{X} \setminus \{x_2\}$, $h'(x, y_1) = \phi(g')(x, y_2) = g'(x, y_2)$ Let h be the extension of h' defined by

$$h(x_2, y) = b_1(y), h(x, y_2) = b_2(x)$$

For all $(x, y) \in (\mathcal{X} \setminus \{x_2\}) \times (\mathcal{Y} \setminus \{y_2\})$,

$$h(x, y) = h'(x, y)$$

Now, we notice that $h = (\pi_x \pi_y)g$ where π_x, π_y are two transpositions in $\mathfrak{S}(\mathcal{X})$, $\mathfrak{S}(\mathcal{Y})$ defined by

$$\pi_x(x_1) = x_2, \pi_x(x_2) = x_1$$

$$\pi_y(y_1) = y_2, \pi_y(y_2) = y_1$$

and for all $x' \in \mathcal{X} \setminus \{x_1, x_2\}$ and for all $y' \in \mathcal{Y} \setminus \{y_1, y_2\}$,

$$\pi_x(x') = x', \pi_y(y') = y'$$

Since $g \in \mathcal{F}$, so is $h = (\pi_x \pi_y)g \in \mathcal{F}$. So h' has the extension h satisfies (1), (2) in Definition 3. Also, $h' = \phi(g')$ is uniquely determined by g' and the argument would also hold when swapping h', g' with the map ϕ^{-1} . So we prove there exists an isomorphism between these two sub-problem classes. \square

Theorem 3.1. Given \mathcal{F} as a subset of all the payoff functions $g : \mathcal{X} \times \mathcal{Y} \rightarrow O$ with a unique Nash Equilibrium where $O \subseteq \mathbb{R}$ and $|\mathcal{X}| = |\mathcal{Y}|$. Let H be an arbitrary (randomised or deterministic) black-box adversarial optimisation algorithm for any $g \in \mathcal{F}$ where \mathcal{F} is closed under permutations. Let $r(H, \mathcal{F})$ be the average (under the uniform distribution on \mathcal{F}) of the expected query complexity of H on $g \in \mathcal{F}$ (i.e. $\mathbb{E}(T_{LB}(H, g))$). Then $r(H, \mathcal{F}) = r(H', \mathcal{F})$ for all algorithms H, H' .

⁶Here we define isomorphism following [11] rather than defining it as a usual group or ring isomorphism since we do not require any group or ring structure of \mathcal{F} in the proof.

Proof of Theorem 3.1. Recall that the query complexity of H on g is $T_{\text{LB}}(H, g) := \inf\{t > 0 \mid x_t = x^* \text{ or } y_t = y^*\}$ where (x_t, y_t) is the search point by Algorithm H on payoff function g and (x^*, y^*) is the unique Nash Equilibrium in a two-player zero-sum game defined by $g : \mathcal{X} \times \mathcal{Y} \rightarrow O$. We denote the set of all functions from a set \mathcal{X} to a set O by $\mathcal{H}(\mathcal{X}, O)$ and the set of all well-defined functions from a set \mathcal{Y} to a set O by $\mathcal{H}(\mathcal{Y}, O)$.

Let \mathcal{F} be a class of games with all the payoff functions $g : \mathcal{X} \times \mathcal{Y} \rightarrow O$ with a unique NE, and assume that \mathcal{F} is closed under permutation. For all $(x_0, y_0) \in \mathcal{X} \times \mathcal{Y}$, define

$$\begin{aligned} B_{x_0}^{(1)} &:= \{b_1 \in \mathcal{H}(\mathcal{Y}, O) \mid \text{there exists } g \in \mathcal{F} \text{ s.t. } b_1(y) = g(x_0, y) \text{ for all } y \in \mathcal{Y}\}; \\ B_{y_0}^{(2)} &:= \{b_2 \in \mathcal{H}(\mathcal{X}, O) \mid \text{there exists } g \in \mathcal{F} \text{ s.t. } b_2(x) = g(x, y_0) \text{ for all } x \in \mathcal{X}\}. \end{aligned}$$

Let $(u, v) \in \mathcal{X} \times \mathcal{Y}$ be the first query point that Algorithm 1 makes, we consider $b_1 \in B_u^{(1)}$ and $b_2 \in B_v^{(2)}$.

We first prove the claim holds for deterministic heuristics by induction. This is done by induction on the size of search space $N = |\mathcal{X}|$. Suppose for any two deterministic algorithms A, B on \mathcal{F} .

If $N = 1$, then it means that $\mathcal{X} \times \mathcal{Y}$ only consists of one unique NE and \mathcal{F} consists of one payoff function g . So for any two deterministic algorithms A, B , $r(A, \mathcal{F}) = r(B, \mathcal{F}) = 1$.

Next, assume $N \geq 2$ and $r(A, \mathcal{G}) = r(B, \mathcal{G})$ where \mathcal{G} is any set of payoff functions with unique NE and search space in payoff functions of size $N - 1$ and \mathcal{G} is closed under permutation. Moreover, we assume \mathcal{F} now consists of payoff functions with unique NE and search space in payoff functions of size N . We expand out the average of $r(A, \mathcal{F})$.

$$r(A, \mathcal{F}) = \sum_{g \in \mathcal{F}} \Pr(g \text{ is selected from } \mathcal{F}) \cdot T(A, g)$$

If we are lucky, then the first point $(u, v) \in \mathcal{X} \times \mathcal{Y}$ we query is the optimum (opt.) and (x^*, y^*) is the NE of the selected g .

$$\begin{aligned} &= 1 \cdot \Pr((u, v) \text{ is the opt. s.t. } u = x^* \text{ or } v = y^*) \\ &\quad + \Pr((u, v) \text{ is not the opt. s.t. } u = x^* \text{ or } v = y^*) \\ &\quad \times (1 + r(A, \mathcal{F}(u, v), (b_1, b_2))) \end{aligned}$$

Notice that after we query (u, v) , we can reduce the whole problem class \mathcal{F} to $\mathcal{F}((u, v), (b_1, b_2))$ with case $N - 1$ where b_1, b_2 are defined above.

Lemma 1 shows that if \mathcal{F} is closed under permutation, then $\mathcal{F}((u, v), (b_1, b_2))$ is closed under permutation. Corollary 1 shows that if we restrict $g \in \mathcal{F}((u, v), (b_1, b_2))$, then all the restriction of g on $(\mathcal{X} \setminus \{u\}) \times (\mathcal{Y} \setminus \{v\})$, denoted by $g|_{(\mathcal{X} \setminus \{u\}) \times (\mathcal{Y} \setminus \{v\})}$, exhibits the same unique Nash Equilibrium (x^*, y^*) as g . So $\mathcal{F}((u, v), (b_1, b_2))$ is a set of payoff functions with unique NE (x^*, y^*) and the search space of size $N - 1$ and closed under permutation for all (u, v) in which (u, v) is not the opt. s.t. $u = x^*$ or $v = y^*$.

With the help of Corollary 1 and Lemma 1, we apply the inductive hypothesis to this sub-problem class $\mathcal{F}((u, v), (b_1, b_2))$ ⁷. Also note that we consider the uniform distribution on problem class \mathcal{F} and thus for any $(u', v') \neq (u, v)$, we have

$$\Pr((u, v) \text{ is the opt. s.t. } u = x^* \text{ or } v = y^*) = \Pr((u', v') \text{ is the opt. s.t. } u = x^* \text{ or } v = y^*).$$

Using the inductive hypothesis on the case $N - 1$ with Lemma 2 gives since two sub-problem classes are essentially isomorphic and thus have the same average query complexity from the inductive hypothesis step.

$$r(A, \mathcal{F}((u, v), (b_1, b_2))) = r(B, \mathcal{F}((u', v'), (b_1, b_2))).$$

Thus, we can conclude that $r(A, \mathcal{F}) = r(B, \mathcal{F})$ for any deterministic algorithms A, B .

⁷We defer further explanations of the role of Corollary 1 and Lemma 1 to the supplementary material Section D.

Now, we generalise the result to randomised algorithms. Let m denote the number of different deterministic search heuristics. We consider a randomised search strategy to be a probability distribution $p = (p_1, \dots, p_m)$ and choose the i -th deterministic search strategy with probability p_i . It is well-known (see detail in [29]) that the expected cost of a randomised search heuristic is the weighted average of the cost of the deterministic search heuristics. Since all deterministic search heuristics have the same cost, this should also hold for all randomised search strategies. \square

Theorem 4.1. *Let \mathcal{X} and \mathcal{Y} be any finite sets. Assume that $B \subset \mathbb{R}$ with $k := |B| \geq 2$. Consider any class of two-player zero-sum games $\mathcal{G} \subset \{g : \mathcal{X} \times \mathcal{Y} \rightarrow B\}$ such that for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, there exists a game $g_{x,y} \in \mathcal{G}$ which has (x, y) as unique, pure Nash Equilibrium. Then, the class \mathcal{G} has black box complexity at least $\lceil \log_k |\mathcal{X} \times \mathcal{Y}| \rceil - 1$.*

Proof of Theorem 4.1. The proof which uses Yao's minimax principle is analogous to the proof of Theorem 2 in [13]. We need to construct a suitable probability distribution p over the set of games. By assumption, for each $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we can associate one game $g_{x,y}$ which has (x, y) as the unique pure NE. We let p be the uniform distribution over the set of the games $\{g_{x,y} \mid (x, y) \in \mathcal{X} \times \mathcal{Y}\} \subseteq \mathcal{G}$.

We now consider the runtime of any deterministic black-box algorithm $A \in \mathcal{A}_{\text{det}}$ with respect to a random game g_{x^*, y^*} which is sampled according to distribution p . The algorithm is a decision tree, where each node is a pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ corresponding to a query made by algorithm A , and each edge corresponds to one of at most k possible outcomes $g(x, y)$ of this query. The runtime of algorithm A on the random game g_{x^*, y^*} corresponds to the depth of (x^*, y^*) in this decision tree. The expected depth is therefore lower bounded by $\lceil \log_k (|\mathcal{X} \times \mathcal{Y}|) \rceil - 1$.

Hence, we have for all deterministic algorithms $A \in \mathcal{A}_{\text{det}}$,

$$\mathbb{E}(T(I_p, A)) \geq \log_k (|\mathcal{X} \times \mathcal{Y}|) - 1.$$

This implies that

$$\min_{A \in \mathcal{A}_{\text{det}}} \mathbb{E}(T(I_p, A)) \geq \log_k (|\mathcal{X} \times \mathcal{Y}|) - 1.$$

By Yao's Principle, for any distribution q over the set of deterministic algorithms \mathcal{A}_{det} ,

$$\begin{aligned} \max_{I \in \mathcal{I}} \mathbb{E}(T(I, A_q)) &\geq \min_{A \in \mathcal{A}_{\text{det}}} \mathbb{E}(T(I_p, A)) \\ &\geq \log_k (|\mathcal{X} \times \mathcal{Y}|) - 1. \end{aligned}$$

The proof now follows by noting that any randomised algorithm can be described as sampling a deterministic algorithm according some distribution q , and applying this algorithm. \square

Theorem 4.2. *Let \mathcal{A} be the set of all randomised algorithms defined by Algorithm 1 and $T(\mathcal{A}, P)$ ⁸ denote the query complexity of \mathcal{A} with respect to the input payoff matrix P for a two-player zero-sum game. Then, there exists an input matrix $P \in \mathbb{R}^{n \times n}$ with a unique pure Nash equilibrium (x^*, y^*) such that $\mathbb{E}(T(\mathcal{A}, P)) \geq (n + 1)/2$. Thus, the black-box complexity with respect to \mathcal{A} of the problem class consisting of all bimatrix games with a unique Nash Equilibrium is at least $(n + 1)/2$.*

Proof of Theorem 4.2. Given payoff matrix P , recall that for any $A \in \mathcal{A}$,

$$\begin{aligned} T_{\text{LB}}(A, P) &:= \inf\{C_t > 0 \mid x_t = x^* \text{ or } y_t = y^*\}; \\ T(A, P) &:= \inf\{C_t > 0 \mid x_t = x^* \text{ and } y_t = y^*\}; \\ \mathcal{M} &:= \{P \in \mathbb{R}^{n \times n} \text{ with a unique PSNE}\}. \end{aligned}$$

Clearly, $T(A, P) \geq T_{\text{LB}}(A, P)$. We denote \mathcal{M} as the set of input instances. Now, we estimate the lower bound by using Yao's minimax principle (we denote the query complexity of a specified algorithm namely ALG searching the PSNE of P by $T(\text{ALG}, P)$ and the set of all deterministic black-box adversarial optimisation algorithms by \mathcal{A}_{det}): for any randomised algorithm $A \in \mathcal{A}$,

$$\max_{P \in \mathcal{M}} \mathbb{E}(T(A, P)) \geq \min_{\text{ALG} \in \mathcal{A}_{\text{det}}} \mathbb{E}_{P \sim \text{Unif}(\mathcal{M})} (T(\text{ALG}, P))$$

⁸Note that $T(A, P) = T(A, g)$ where $g(x, y) := e_y^T P e_x$ with e_x, e_y denote the elementary probability distribution over probability simplex $\Delta_{\{0,1\}^n}$

Using the definition of $T(\text{ALG}, P)$ and $T_{\text{LB}}(\text{ALG}, P)$ gives

$$\geq \min_{\text{ALG} \in \mathcal{A}_{\text{det}}} \mathbb{E}_{P \sim \text{Unif}(\mathcal{M})} (T_{\text{LB}}(\text{ALG}, P))$$

Using Theorem 3.1, we know the expected performance of all algorithms on any all problem instance in \mathcal{M} is the same with respect to the unique Nash Equilibrium solution concept. So, we define a new algorithm ALG^* such that it is a deterministic algorithm starting from $i = j = 1$, and it makes one query in each iteration. It continues to query $P_{1,1}, P_{2,2}, \dots, P_{n,n}$ (i.e. query the entries in the diagonal of the payoff matrix P). ALG^* continues the processes until it reaches either the column or the row of the position of the unique Nash Equilibrium. So, we derive

$$= \mathbb{E}_{P \sim \text{Unif}(\mathcal{M})} (T_{\text{LB}}(\text{ALG}^*, P))$$

Note that we consider the uniform distribution on \mathcal{M} . It means that the probability that (x^*, y^*) lies in the j -th column of payoff matrix P is $1/n$ for all $j \in [n]$. Then, the total expected query complexity is $\sum_{j=1}^n j \frac{1}{n} = \frac{n+1}{2}$. So, we derive

$$\geq \frac{n+1}{2}.$$

This completes the proof. \square

Theorem 4.3. *The black-box complexity with respect to Algorithm 1 of the binary voting games with problem size $n \in \mathbb{N}$ and a unique Nash Equilibrium is $e^{\Omega(n)}$.*

Proof of Theorem 4.3. Given an arbitrary binary voting game with problem size n defined by a payoff function $g : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \mathbb{R}$, we consider the corresponding payoff matrix P . For each party, there are 2^n possible strategies encoded by a binary bitstring of length n . So $P \in \mathbb{R}^{2^n \times 2^n}$. Using Theorem 4.2 with arbitrary algorithms in the class defined by Algorithm 1 and the payoff matrix P gives us the black-box complexity is at least $\frac{2^n+1}{2} = e^{\Omega(n)}$. \square

Theorem 4.4. *Given the game class*

$$\text{DIAGONAL}_n := \{\text{DIAGONAL}_{(u,v)} \mid (u, v) \in \{0, 1\}^n \times \{0, 1\}^n\},$$

the black-box complexity with respect to Algorithm 1 of DIAGONAL_n is $\Theta(n)$.

Proof of Theorem 4.4. We consider an algorithm $A \in \mathcal{A}$, which operates in two phases. In each iteration of Phase 1, the algorithm samples x and y uniformly at random, then flips one bit uniformly at random in x to obtain x' . It then compares the values of $\text{DIAGONAL}_{u,v}(x, y)$ and $\text{DIAGONAL}_{u,v}(x', y)$. Phase 1 ends when $\text{DIAGONAL}_{u,v}(x, y) \neq \text{DIAGONAL}_{u,v}(x', y)$. Once Phase 1 ends, the algorithm knows that $|u \oplus x|_1 = |v \oplus y|_1$.

In Phase 2, Algorithm A sample a bitstring y' by flipping a single bit in y . If $\text{DIAGONAL}_{u,v}(x, y') = -1$, we set $y = y'$, otherwise we repeatedly sampling a new y' . Once we get payoff -1 , we accept the new search point and then sample x' by flipping a single bit in x until repeat the process until we get payoff 1. This procedure continues until the termination criterion is met.

We now consider how long it takes for Phase 1 to finish (i.e. the search point arrives at the diagonal). Since we are using uniformly at random initialisation, so $Z_1 := |u \oplus x|_1$ and $Z_2 := |v \oplus x|_1$ are subject to two independent binomial random variables $\text{Bin}(n, 1/2)$. We first estimate $\Pr(Z_1 = Z_2)$. Notice that let $Y := n - Z_2$ and Y is also subject to $\text{Bin}(n, 1 - 1/2) = \text{Bin}(n, 1/2)$. So the event $\{Z_1 = Z_2\}$ is equivalent to $\{Z_1 = Y\} = \{Z_1 + Z_2 = n\}$. Thus, we can derive the following estimate by using Stirling's approximation (i.e. $n! \sim \sqrt{2\pi n} (\frac{n}{e})^n$),

$$\Pr(Z_1 = Z_2) = \binom{2n}{n} 2^{-2n} \sim \frac{1}{\sqrt{\pi n}} 2^{2n} \cdot 2^{-2n} = \frac{1}{\sqrt{\pi n}}.$$

So the expected runtime for finishing Phase 1 is $O(\sqrt{n})$.

Now, we consider how long it takes for Phase 2 to reach the optimum. Notice that for each bit, we need 1 query of the payoff function to determine whether it is the correct bit to flip. We need to flip the correct bits for both x and y . Since the maximum Hamming distance to the optimum of DIAGONAL is bounded by $2n$, then the overall runtime is bounded above by $2n = O(n)$. Adding the expected runtime for both Phases gives $O(n) + O(\sqrt{n}) = O(n)$. Together with Theorem 4.1, we can conclude that the black-box complexity of DIAGONAL_n is $\Theta(n)$. \square

Theorem 4.5. *Given the game class,*

$$\text{PLATEAU}_n := \{\text{PLATEAU}_{(u,v)} \mid (u, v) \in \{0, 1\}^n \times \{0, 1\}^n\},$$

the black-box complexity with respect to the class of algorithms defined by Algorithm 1 of PLATEAU_n is $e^{\Omega(n)}$.

Proof of Theorem 4.5. Recall the definition of $\text{PLATEAU}_{u,v}$ for arbitrary u, v . We call the set $\{(x, y) \mid ||u \oplus x|_1 - \frac{n}{2}| \leq \frac{\varepsilon n}{2}\}$ x -independent, and we call any query to this set x -independent. Note that for any x -independent query has payoff $\text{PLATEAU}_{u,v}(x, y) = f(v \oplus y)$, i.e., it is independent of x . Furthermore, note that the Nash Equilibrium is not x -independent.

We will apply Yao's Principle with respect to the distribution p over instances where u is sampled uniformly at random among all bitstrings of length n , and $v = 0^n$. We consider the average case runtime of deterministic algorithms with respect to distribution p . Such algorithms can be modelled as binary decision trees: in each node, the algorithm makes a query (x, y) , and for each outgoing edge, the algorithm obtains one of two possible payoff values $\text{PLATEAU}_{u,v}(x, y) \in \{-1, 1\}$. We call the x -independent path in the decision tree the longest path $(x_1, y_1), \dots, (x_t, y_t)$ such that (x_1, y_1) is the root node and all nodes (x_i, y_i) for $i \in [t]$ are x -independent. We let t denote the length of the x -independent path. Since the outcome of an x -independent query only depends on $v \oplus y$ which is deterministic, the x -independent path of a decision tree is unique. Furthermore, the length of the x -independent path is a lower bound on the runtime of the algorithm since the Nash Equilibrium is not x -independent.

We now lower bound the length of the x -independent path. Let (x_i, y_i) be any fixed query in the decision tree, and F_i the event that (x_i, y_i) is x -independent. We sample u by picking each bit independently and uniformly at random in $\{0, 1\}$. This implies that the number of 1 bits for each bit is subject to a Bernoulli random variable $\text{Bin}(1/2)$. We therefore have $|u \oplus x_i|_1$ is binomially distributed $\text{Bin}(n, 1/2)$. Applying Chernoff's bound, we get

$$\Pr(F_i) = \Pr\left(|u \oplus x_i|_1 - \frac{n}{2} > \varepsilon \frac{n}{2}\right) \leq 2e^{-\varepsilon^2 n/6}. \quad (1)$$

Let E_1 be the complement of all the failure events, i.e., $E_1 = \overline{\cup_{i=1}^t F_i}$. Therefore, by a union bound, the x -independent path has length $t = 2e^{\varepsilon^2 n/12}$ with probability

$$\Pr(E_1) = 1 - \Pr\left(\cup_{i=1}^t F_i\right) \geq 1 - t2e^{-\varepsilon^2 n/6} = 1 - e^{-\Omega(n)}.$$

We then have by the law of total probability

$$\begin{aligned} \min_{A \in \mathcal{A}} \mathbb{E}(T(I_p, A)) &\geq \min_{A \in \mathcal{A}} \Pr(E_1) \mathbb{E}(T(I_p, A) \mid E_1) \\ &\geq (1 - e^{-\Omega(n)}) 2e^{\varepsilon^2 n/12} \\ &= e^{\Omega(n)}. \end{aligned}$$

The proof now follows by Yao's minimax Principle. The expected worst-case runtime of any randomised algorithm is lower bounded by the average case runtime of deterministic algorithms.

$$\max_{I \in \mathcal{I}} \mathbb{E}(T(I, A_q)) \geq \min_{A \in \mathcal{A}} \mathbb{E}(T(I_p, A)) = e^{\Omega(n)}.$$

□

E Further Explanations for Sub-Problem Class

We provide an example to illustrate the concept of the sub-problem class (see Definition 3).

Figure 2 is an example of a sub-problem class. When querying (x_1, y_1) , we also check the corresponding row and column to verify whether the unique Nash Equilibrium lies on this blue cross or not. If the answer is negative, then we restrict to the sub-matrix by removing the blue entries in P .

$$P = \begin{pmatrix} \text{---} & \text{---} \\ \text{---} & \text{---} \end{pmatrix} \begin{matrix} y_1 \\ y^* \end{matrix}$$

$x_1 \quad x^*$

Figure 2: Example of a sub-problem class. Given a payoff matrix P , the blue entries mean that the actual values are already assigned to $g(x_1, \cdot)$ and $g(\cdot, y_1)$. (x^*, y^*) is the Nash Equilibrium we search for. The sub-problem class here means that the problem consists of different payoff matrices with given blue entries.

F Further Explanations for the role of Corollary 1 and Lemma 1 in Theorem 3.1

In the main proof of Theorem 3.1, we rely on proof by induction: assume the theorem holds for a search space of size N , we want to show it also holds for $N + 1$. Corollary 1 and Lemma 1 are crucial since they ensure that for every payoff function $g \in \mathcal{F}$, there is a corresponding function g' where the NE (x^*, y^*) is switched to $(\sigma(x^*), \tau(y^*))$ (i.e. σ, τ are two permutations on \mathcal{X}, \mathcal{Y} respectively) and the NE is still unique in the game defined by g' . This means that the inductive hypothesis can be applied to g' due to the closure under permutation. In short, Corollary 1 and Lemma 1 establish a property of the problems or payoff functions under consideration that is preserved under permutations. This property helps to ensure that the induction step can be applied successfully.

G Further Explanations for bitwise exclusive or in Black-Box Complexity Analysis

We define \oplus as bitwise exclusive or. \oplus means that for any two binary bitstrings $x = x_1 \cdots x_n$ and $u = u_1 \cdots u_n$, $x \oplus u := (x_1 \oplus u_1, \cdots, x_n \oplus u_n)$ where $1 \oplus 1 = 0, 1 \oplus 0 = 1, 0 \oplus 1 = 1, 0 \oplus 0 = 0$. We introduce bitwise exclusive or here to generate problem instances with the same structure. For example, $\text{DIAGONAL}(x, y)$ is only one possible problem instance in binary voting games and to find its optimum, we only need to design the algorithm querying $(1^n, 1^n)$ to reach its optimum. And its black-box complexity is trivially $\Theta(1)$. Using bitwise exclusive or \oplus (i.e. $\text{DIAGONAL}(u \oplus x, v \oplus y)$ where $(u, v) \in \mathcal{U} \times \mathcal{V}$ is sampled uniformly at random), we can generate a set of problem instances which have the same structure as $\text{DIAGONAL}(x, y)$ and the goal of black-box algorithms is to find (x, y) such that $(u \oplus x, v \oplus y) = (1^n, 1^n)$. (Note that if $(u, v) = (0^n, 0^n)$, then $\text{DIAGONAL}(0^n \oplus x, 0^n \oplus y)$ is reduced to the vanilla $\text{DIAGONAL}(x, y)$.) This bitwise exclusive generator can avoid the trivial black-box complexity mentioned above and reflect the true difficulty of the class DIAGONAL . We refer to [13, 9] for more detail about black-box complexity theory.

NeurIPS Paper Checklist

1. Claims

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

Answer: [\[Yes\]](#)

Justification: Our main claims made in the abstract and introduction accurately reflect the paper's contributions and scope. The detailed contribution can be checked in Section 1.2 in the main paper. We also provide a content list as the roadmap of this paper at the beginning of the appendix.

Guidelines:

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

2. Limitations

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

Answer: [\[Yes\]](#)

Justification: The limitations of our results have been discussed in the conclusion (line 372-384). Moreover, the bounds for the black-box complexity of binary voting games derived in this paper mainly consider the asymptotic order, which is sufficient for benchmark dichotomy. There is room to improve these bounds with respect to the leading coefficients.

Guidelines:

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

3. Theory Assumptions and Proofs

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

Answer: [Yes]

Justification: All the theorems, formulas, and proofs in the paper are numbered and cross-referenced, and all assumptions are clearly stated or referenced in the statement of all theorems. Due to the page limit, we defer all the proofs in the supplementary material (see Section D). We also provide a short proof sketch for our main result: Theorem 3.1.

Guidelines:

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

4. Experimental Result Reproducibility

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

Answer: [NA]

Justification: Our paper is purely a theory paper, and it does not include experiments.

Guidelines:

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

5. Open access to data and code

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

Answer: [NA]

Justification: The paper does not include experiments requiring code.

Guidelines:

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

6. Experimental Setting/Details

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

Answer: [NA]

Justification: The paper does not include experiments.

Guidelines:

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

7. Experiment Statistical Significance

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

Answer: [NA]

Justification: The paper does not include experiments.

Guidelines:

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

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

8. Experiments Compute Resources

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

Answer: [NA]

Justification: The paper does not include experiments.

Guidelines:

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

9. Code Of Ethics

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

Answer: [Yes]

Justification: This work strictly follows the NeurIPS Code of Ethics.

10. Broader Impacts

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

Answer: [Yes]

Justification: This paper studies NFL analysis and BBC analysis of black-box adversarial optimisation algorithms. From a theoretical viewpoint, our work rigorously outlines the theoretical limitations of black-box adversarial optimisation algorithms for finding Nash Equilibrium and characterises the general difficulty of different class problems to be solved via the given black-box model. From a practical viewpoint, more careful benchmark selections are suggested for use in many black-box optimisation applications which solve maximin problems with complicated constraints, e.g. robust optimisation, neuroevolution, bimatrix games, two-player games etc. This work is of a theoretical nature, and there is no foreseeable negative ethical or societal implication.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point

out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

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

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

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

12. Licenses for existing assets

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

Answer: [NA]

Justification: The paper does not use existing assets.

Guidelines:

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

13. New Assets

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

Answer: [NA]

Justification: The paper does not release new assets.

14. **Crowdsourcing and Research with Human Subjects**

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

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

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

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

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