Min-Max Optimization Requires Exponentially Many Queries

Martino Bernasconi
Bocconi University
martino.bernasconi@unibocconi.it

Matteo Castiglioni
Politecnico di Milano
matteo.castiglioni@polimi.it

Andrea Celli
Bocconi University
andrea.celli2@unibocconi.it

Alexandros Hollender
Technical University of Munich
alexandros.hollender@tum.de

Abstract

We study the query complexity of min-max optimization of a nonconvex-nonconcave function f over the hypercube $[0,1]^d \times [0,1]^d$. We show that, given oracle access to f and to its gradient ∇f, any algorithm that finds an ε-approximate stationary point must make a number of queries that is exponential in $1/\varepsilon$ or d.

1 Introduction

In this paper, we consider the following constrained min-max optimization problem

$$\min_{x \in X} \max_{y \in Y} f(x,y) \tag{1}$$

where $X, Y \subset \mathbb{R}^d$ are compact convex sets, and $f : X \times Y \to \mathbb{R}$ is continuously differentiable with Lipschitz-continuous gradient ∇f.[1] This problem has wide practical relevance, as it is a fundamental component in optimization and machine learning applications such as generative adversarial networks [Goodfellow et al., 2014], robustness to adversarial attacks [Madry et al., 2018, Razaviyayn et al., 2020], and, most recently, large language model [Swamy et al., 2024, Liu et al., 2024, Munos et al., 2024, Sun et al., 2025, Paulus et al., 2025] (see also Daskalakis [2022] for a broader discussion on the importance of this problem in optimization and machine learning).

The problem can be solved efficiently in the case where f is *convex-concave*, i.e., when $x \mapsto f(x,y)$ is convex for all $y \in Y$, and $y \mapsto f(x,y)$ is concave for all $x \in X$. Namely, an ε-approximate (global) optimal solution of (1) can be identified using at most $\text{poly}(d, \log(1/\varepsilon))$ queries to f and ∇f [Rodomanov and Nesterov, 2023, Anagnostides et al., 2025]. However, as soon as we let the function f be only *nonconvex-concave* (or, analogously, *convex-nonconcave*), the problem already becomes harder to solve. Indeed, it is easy to see that it is at least as hard as a constrained nonconvex minimization problem of the form

$$\min_{x \in X} g(x) \tag{2}$$

where g is continuously differentiable with Lipschitz-continuous gradient ∇g. This is immediate by setting $f(x,y) := g(x)$. Importantly, since g can be nonconvex, it is known that we cannot hope to efficiently locate the approximate position of a global minimum, or even a point with an objective function value close to that of a global minimum [Vavasis, 1995].

[1] Here, and in the rest of this discussion, we assume that both f and ∇f are 1-Lipschitz continuous. This is without loss of generality, as it can be achieved by a simple rescaling of the function. This will allow us to express bounds solely in terms of the approximation error and the dimension d.

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

Instead, a more tractable solution concept for the optimization problem (2) is an approximate *local* minimum, or, more precisely, an approximate constrained first-order stationary point. This is also called an approximate Karush-Kuhn-Tucker (KKT) point, and corresponds to an ε-approximate fixed point of the projected gradient descent operator. It is well known that such a point can be found with at most $O(1/\varepsilon^2)$ first-order queries, for example, by projected gradient descent. Moreover, this dependence on ε is unavoidable in the worst case: no algorithm can find an ε-approximate KKT point of (2) with fewer than $\text{poly}(1/\varepsilon)$ queries [Vavasis, 1993, Carmon et al., 2020].

A similar picture holds for the nonconvex-concave case of the min-max problem (1). In this setting, one can still compute an appropriate local solution using at most $\text{poly}(d, 1/\varepsilon)$ first-order queries [Nouiehed et al., 2019, Lin et al., 2020, Ostrovskii et al., 2021]. Thus, in both nonconvex minimization and nonconvex-concave min-max optimization, local solution concepts remain algorithmically tractable, although the dependence on the accuracy is necessarily polynomial.

Unfortunately, in most practical applications the function $f(x, y)$ is neither convex in x, nor concave in y. Also in this setting, the focus shifts toward the less ambitious goal of finding local solutions to problem (1). Many notions of local solutions have been proposed for the problem (1), see, e.g., Jin, Netrapalli, and Jordan [2020] for an overview. In this work, we focus on ε-approximate (constrained) stationary points, namely a tuple $(x, y) \in X \times Y$ such that

$$-\nabla f_x(x, y)^\top (x' - x) \leq \varepsilon \quad \text{and} \quad \nabla f_y(x, y)^\top (y' - y) \leq \varepsilon \quad \forall (x', y') \in X \times Y.$$

This is a weak first-order condition: it merely requires that neither player has a feasible first-order improving direction larger than ε. This notion is very weak in the sense that it is necessary for the standard local optimality notions considered in the min-max literature, including local min-max points and local saddle points.[2] This is favorable in our case, since we will provide lower bounds, thereby strengthening our results. First-order stationary points can also be viewed as fixed points of the projected gradient descent-ascent map, just as KKT points of a minimization problem are fixed points of projected gradient descent. However, in contrast to minimization problems, it is well known that gradient descent-ascent dynamics exhibit cycling behavior and thus do not yield an algorithm guaranteed to find such solutions [Mertikopoulos et al., 2018, Daskalakis and Panageas, 2018].

This contrast naturally leads to the following question:

> When f is nonconvex-nonconcave, can an ε-approximate first-order stationary point of the min-max problem (1) be found using at most $\text{poly}(d, 1/\varepsilon)$ queries?

In this paper, we answer this question in the negative. Namely, we show that any algorithm finding an ε-approximate solution must make a number of queries that is exponential in d or in $1/\varepsilon$. This shows that finding approximate stationary points of (1) is significantly harder than in the corresponding minimization problem (2). In particular, our result implies that nonconvex-nonconcave min-max optimization does not admit an efficient gradient-descent-type algorithm.

A negative result of this form was already known for the more general case of *joint* constraints, i.e.,

$$\begin{aligned} &\min_{x \in \mathbb{R}^d} \max_{y \in \mathbb{R}^d} f(x, y) \\ &\text{s.t.} \quad (x, y) \in P \end{aligned} \tag{3}$$

where P is a convex polytope. Daskalakis, Skoulakis, and Zampetakis [2021] proved that any algorithm that finds an ε-approximate solution of (3) must make a number of queries that is exponential in d or in $1/\varepsilon$. Note that problem (3) is more general than (1), because it allows *joint* constraints between x and y, meaning that P cannot be decomposed as a product $P = X \times Y$. As discussed by Bernasconi, Castiglioni, Celli, and Farina [2024], the joint constraints are used in a crucial way in the work of Daskalakis et al. [2021].[3] Therefore, their result does not apply to the more natural setting of problem (1), i.e., when we have *product* constraints (meaning that the feasible choices of x and y are independent at the level of the constraint set). Determining the query complexity under product constraints (i.e., our main open question stated above) remains a major open problem, as recently discussed in a column in the SIAM Activity Group on Optimization newsletter [Diakonikolas, 2025].

[2] Unfortunately, the terminology for these solution concepts is not fully standardized across the literature.

[3] In fact, the lower bound proved by Daskalakis et al. [2021] for joint constraints even applies to the setting where f is nonconvex-concave. Since such problems can be solved using $\text{poly}(d, 1/\varepsilon)$ queries under product constraints, this shows that the hardness result in [Daskalakis et al., 2021] fundamentally relies on joint constraints.

A very recent result of Bernasconi and Castiglioni [2026] established that finding approximate first-order stationary points of problem (1), i.e., nonconvex-nonconcave min-max optimization with product constraints, is PPAD-hard. This means that when the function f and its gradient ∇f are provided as arithmetic circuits or Turing machines, then the problem cannot be solved in polynomial time, unless PPAD = P. Importantly, as mentioned in Bernasconi and Castiglioni [2026, Section 8], their result does not imply a query lower bound for the problem, because their reduction is not *black-box*: it reduces from a purely white-box problem called PURECIRCUIT [Deligkas et al., 2024]. They ask the question of whether it is possible to improve their reduction so as to also obtain a query lower bound for the problem. In this work, we show that this is indeed possible. Our improved reduction resolves the main open question stated above.

Our Contribution. We consider the min-max optimization problem (1) with the simple domain $X = Y = [0, 1]^d$ and show that any algorithm that finds an ε-approximate first-order stationary point must make a number of queries to f and ∇f that is exponential in $1/\varepsilon$ or d. To be more precise, we prove the following theorem.[4]

Theorem 1.1 (Informal version of Theorem 6.1)**.** *Any algorithm that outputs an ε-approximate stationary point for problem (1), must make at least a number of queries to f or ∇f that is exponential in d, even when ε is inversely polynomial in d.*

This provides an *unconditional* lower bound for the problem, as opposed to the conditional time-complexity lower bound of Bernasconi and Castiglioni [2026], which only holds if PPAD $\neq$ P. Furthermore, our lower bound even rules out algorithms that are allowed to perform unbounded computation between queries. Chen, Li, and Yannakakis [2024] recently gave such a query-efficient but time-inefficient algorithm for the problem of computing a fixed point of a contraction map. Theorem 1.1 shows that no such algorithm exists for nonconvex-nonconcave min-max optimization.

Our Techniques. As mentioned above, the main obstacle to obtaining a query lower bound from the construction of Bernasconi and Castiglioni [2026] is the usage of the PURECIRCUIT problem in their reduction. The PURECIRCUIT problem was introduced by Deligkas, Fearnley, Hollender, and Melissourgos [2024] as a tool for proving very strong inapproximability results in the context of PPAD. Importantly, it is a purely *white-box* problem, meaning there is no oracle in the problem definition and the algorithm has full knowledge of the instance. Thus, it cannot be used to prove a query lower bound for our min-max problem.

The natural thing to do is to try to replace the PURECIRCUIT problem in the reduction by some other Brouwer-like problem. The issue with this is that the reduction of Bernasconi and Castiglioni [2026] heavily relies on the fact that the gates of the PURECIRCUIT problem are very flexible and allow for a lot of error without breaking. This fact is also precisely why PURECIRCUIT has been so useful in proving PPAD-hardness for various approximation problems in game theory and beyond. Unfortunately, black-box Brouwer-like problems, for which query lower bounds are known, do not offer this flexibility.

We resolve this issue by introducing a black-box version of the PURECIRCUIT problem, which we call ORACLEPURECIRCUIT. The definition of this new problem is identical to PURECIRCUIT, except that the problem has been augmented with one additional type of gate, which we call an oracle gate. The ORACLEPURECIRCUIT problem allows us to achieve our two desiderata: (i) keeping the simplicity and flexibility of PURECIRCUIT, while also (ii) having a problem that has an exponential query lower bound. Indeed, the query lower bound for ORACLEPURECIRCUIT follows relatively easily by adapting the existing PPAD-hardness proof for PURECIRCUIT from [Deligkas et al., 2024], as we show in Section 3.

What remains then is modifying the reduction of Bernasconi and Castiglioni [2026] so that it uses ORACLEPURECIRCUIT instead of PURECIRCUIT. In particular, this entails implementing the newly introduced oracle gate. Another part of the reduction of Bernasconi and Castiglioni [2026] breaks when we try to use it to prove query lower bounds. Indeed, they reduce from two different problems: one of which is the PURECIRCUIT discussed above, and the other one is a problem related to linear variational inequalities, which is also not a query-hard problem; however, the modification of this part of the reduction is more standard, and we solve it by defining a smooth version of Brouwer's

[4] We use the term "exponential in x" in the weak sense, i.e., to mean $2^{\Omega(x^c)}$ for some absolute $c > 0$, as opposed to $2^{\Omega(x)}$.

fixed point theorem, as detailed in Sections 4 and 5. Finally, the proof culminates in Section 6, where we put everything together to prove Theorem 1.1.

2 Preliminaries

Mathematical Notation. For a function $F : [0,1]^d \to \mathbb{R}^m$, we say that F is G-Lipschitz if $\|F(x) - F(x')\| \leq G\|x - x'\|$, and L-smooth if $\|J_F(x) - J_F(x')\|_2 \leq L\|x - x'\|_2$, where $J_F(x) \in \mathbb{R}^{m\times d}$ is the Jacobian and $\|\cdot\|$ denotes the Euclidean norm for vectors and the spectral norm for matrices. For a matrix $A \in \mathbb{R}^{m\times n}$, let $\|A\|_{\max} := \max_{i,j} |A_{ij}|$. For $x \in \mathbb{R}^d$ and $r \geq 0$, let $B_r^\infty(x) := \{y \in \mathbb{R}^d : \|x - y\|_\infty \leq r\}$. For $f : \mathbb{R}^n \to \mathbb{R}$, we denote by $D^2 f(x) \in \mathbb{R}^{n\times n}$ its Hessian. For $g : \mathbb{R} \to \mathbb{R}$, let $\|g\|_\infty := \sup_{x\in\mathbb{R}} |g(x)|$. Finally, $[n] := \{1, \ldots, n\}$, and $\mathcal{C}^k$ denotes the class of functions whose partial derivatives up to order k are continuous.

The main result of the paper is a query lower bound for the following problem, which concerns finding approximate fixed points of gradient descent-ascent dynamics (GDA).

Problem 1 (GDA-FIXEDPOINT)**.** Given $\varepsilon, L, G, B \in \mathbb{R}_+$, two oracles implementing a G-Lipschitz and L-smooth function $f : [0,1]^d \times [0,1]^d \to [-B, B]$ and its gradient $\nabla f : [0,1]^d \times [0,1]^d \to \mathbb{R}^{2d}$, find $(x^\star, y^\star) \in [0,1]^d \times [0,1]^d$ such that for all $i \in [d], x_i \in [0,1]$ and $y_i \in [0,1]$

$$-\partial_{x_i} f(x^\star, y^\star)(x_i - x_i^\star) \leq \varepsilon \quad \text{and} \quad \partial_{y_i} f(x^\star, y^\star)(y_i - y_i^\star) \leq \varepsilon.$$

As also mentioned in the introduction, the naming of the problem is motivated by the problem being computationally equivalent to finding an approximate fixed-point of the GDA map $(x, y) \mapsto (\Pi_{[0,1]^d}(x - \nabla_x f(x, y)), \Pi_{[0,1]^d}(y + \nabla_y f(x, y)))$, as shown by Daskalakis et al. [2021].

Our query lower bound ultimately comes from known lower bounds for finding a Brouwer fixed point. Specifically, we will use the following discrete Brouwer-like problem, called STRONGSPERNER.[5]

Problem 2 (STRONGSPERNER)**.** Given integers M, d, and oracle access to a labeling $\lambda : [M]^d \to \{-1, +1\}^d$ satisfying the following boundary conditions for every $i \in [d]$:

$$x_i = 1 \implies [\lambda(x)]_i = +1 \quad \text{and} \quad x_i = M \implies [\lambda(x)]_i = -1,$$

output $x^{(1)}, \ldots, x^{(d)} \in [M]^d$ such that $\max_{i,j\in[d]} \|x^{(i)} - x^{(j)}\|_\infty \leq 1$, and such that all labels are covered, i.e., for all $i \in [d]$ and $\ell \in \{-1, +1\}$ there exists $j \in [d]$ with $[\lambda(x^{(j)})]_i = \ell$.

The following query lower bound for STRONGSPERNER easily follows from the seminal query lower bound of Hirsch, Papadimitriou, and Vavasis [1989] for finding Brouwer fixed points.

Theorem 2.1. *There exists a sufficiently large constant integer M such that any algorithm that outputs a solution to* STRONGSPERNER *with parameters M and d must make $2^{\Omega(d)}$ many queries to the labeling λ.*

Daskalakis et al. [2021, Corollary 9.2] prove a weaker version of Theorem 2.1 where M is only assumed to be $O(d)$, instead of constant. However, it is not hard to see that the work of Hirsch et al. [1989] in fact implies that M can be fixed to a constant. For completeness, we provide a proof of this in section A.

3 The ORACLEPURECIRCUIT Problem

In this section, we introduce the ORACLEPURECIRCUIT problem, which is a black-box version of the PURECIRCUIT problem of Deligkas et al. [2024], and prove a query lower bound for it.

Problem 3 (ORACLEPURECIRCUIT)**.** An instance of ORACLEPURECIRCUIT is given by a natural number N, a set of nodes V, three sets of gates $\mathcal{G}_{\text{NOR}}$, $\mathcal{G}_{\text{PURIFY}}$, and $\mathcal{G}_{\text{ORACLE}}$, and oracle access to a function $\mathcal{L} : \{0,1\}^N \to \{0,1\}$ (with $N \leq |V|$). Each gate is of the form $(u_1, u_2, \ldots)$ where u_i are distinct nodes in V with the following interpretation:

- if $(u, v, w) \in \mathcal{G}_{\text{NOR}}$, then u and v are the inputs of the gate, and w is its output.

[5] This problem was first defined by Daskalakis et al. [2021], where it was called HIGHD-BISPERNER. Later, it was used by Deligkas et al. [2024] under the name STRONGSPERNER to prove the PPAD-hardness of PURECIRCUIT.

- if $(u, v, w) \in \mathcal{G}_{\text{PURIFY}}$, then u is the input of the gate, and v and w are its outputs.
- if $(u_1, \ldots, u_N, v) \in \mathcal{G}_{\text{ORACLE}}$, then $u_1, \ldots, u_N$ are the inputs of the gate and v its output.

Each node is the output of exactly one gate. A solution to an instance of ORACLEPURECIRCUIT is an assignment $b : V \to \{0, 1, \perp\}$ that satisfies all the gates, i.e., for each gate we have:

- if $(u, v, w) \in \mathcal{G}_{\text{NOR}}$, then b satisfies:

$$b(u) = b(v) = 0 \implies b(w) = 1$$
$$(b(u) = 1) \text{ or } (b(v) = 1) \implies b(w) = 0,$$

- if $(u, v, w) \in \mathcal{G}_{\text{PURIFY}}$, then b satisfies:

$$\{b(v), b(w)\} \cap \{0, 1\} \neq \emptyset$$
$$b(u) \in \{0, 1\} \implies b(v) = b(w) = b(u),$$

- if $(u_1, \ldots, u_N, v) \in \mathcal{G}_{\text{ORACLE}}$, then b satisfies:

$$(b(u_1), \ldots, b(u_N)) \in \{0, 1\}^N \implies b(v) = \mathcal{L}(b(u_1), \ldots, b(u_N)).$$

The PURECIRCUIT problem originally defined by Deligkas et al. [2024] is the same as our ORACLEPURECIRCUIT, but without the oracle gates $\mathcal{G}_{\text{ORACLE}}$. In ORACLEPURECIRCUIT we have introduced oracle gates in the least constraining way: a gate of type $\mathcal{G}_{\text{ORACLE}}$ is only required to be correct when all its inputs are actual bits (i.e., no $\perp$). Just like PURECIRCUIT, our version is also always guaranteed to have a solution. This can be proved directly, and it also follows from our reduction to GDA-FIXEDPOINT, which is known to always have a solution [Daskalakis et al., 2021]. As mentioned in the introduction, PURECIRCUIT is PPAD-complete, but does not admit a query lower bound, because it is a purely white-box problem. In contrast, for ORACLEPURECIRCUIT we can show the following.

Theorem 3.1. *Given an instance of* ORACLEPURECIRCUIT *with* $|V|$ *nodes, any algorithm that has black-box oracle access to* $\mathcal{L}$ *and outputs a solution to* ORACLEPURECIRCUIT *requires* $2^{\Omega(|V|^{1/3})}$ *queries to* $\mathcal{L}$.

Proof sketch. The query lower bound is obtained by constructing a reduction from STRONGSPERNER with dimension d, for which a $2^{\Omega(d)}$ query lower bound is known by Theorem 2.1. To do this, we modify the existing reduction from the white-box version of STRONGSPERNER to PURECIRCUIT, provided by Deligkas et al. [2024]. Namely, whenever the standard PURECIRCUIT gates are used to implement the Boolean circuit of the white-box version of STRONGSPERNER, we instead use the oracle gate to implement the oracle of the black-box version of STRONGSPERNER. Using the fact that the parameter M of STRONGSPERNER can be assumed to be constant (by Theorem 2.1), we argue that the number of gates in the ORACLEPURECIRCUIT instance is at most $O(d^3)$, which yields a query lower bound of $2^{\Omega(|V|^{1/3})}$. For the details of the proof, we refer to Section B. □

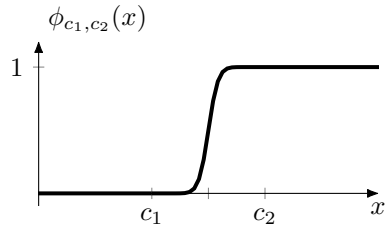
4 Smooth Interpolation of a Boolean-Valued Function

To bridge the gap between discrete Boolean circuits and continuous optimization, we require a mechanism for extending Boolean functions to the continuous domain. Specifically, we seek a C^∞ interpolation over $[0, 1]^N$ of a Boolean function defined only on $\{0, 1\}^N$. In pursuit of this gadget, we first consider a smooth-step function. Namely, given two values c_1, c_2, we use a standard $\mathcal{C}^\infty$ non-analytic function ϕ_{c_1,c_2}, which implements a smooth step-function that interpolates the step function that is 0 below c_1 and 1 above c_2.

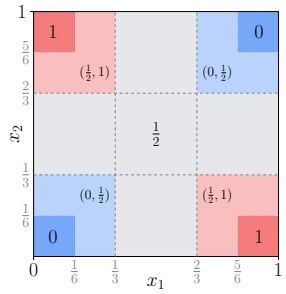
Definition 1 (Smooth step function)**.** Let $c_2 > c_1 \geq 0$. The function $\phi_{c_1,c_2} : \mathbb{R} \to [0, 1]$ is defined as

$$\phi_{c_1,c_2}(x) = \frac{\eta(x - c_1)}{\eta(x - c_1) + \eta(c_2 - x)},$$

where $\eta : \mathbb{R} \to [0, 1]$ is defined as $\eta(x) = \mathbb{I}(x > 0) \exp(-1/x)$.



(a) The smooth step function of Lemma 1.



(b) Schematic illustration of the interpolation presented in Lemma 2 for $N = 2$ and $\hat{h}(x_1, x_2) = \text{XOR}(x_1, x_2)$.

Figure 1

The main properties of ϕ_{c_1,c_2} are summarized in the next lemma. Basically, it guarantees that ϕ_{c_1,c_2} implements a smooth step from c_1 to c_2 and that its derivative is bounded by a constant (depending on c_1 and c_2). An illustration of such a function can be found in Figure 1a.

Lemma 1. *Given two $c_1 < c_2$, the function ϕ_{c_1,c_2} is in $\mathcal{C}^\infty$ and satisfies:*

$$\phi_{c_1,c_2}(x) \in \begin{cases} \{0\} & \text{If } x \leq c_1 \\ (0,1) & \text{if } x \in (c_1, c_2) \\ \{1\} & \text{if } x \geq c_2 \end{cases}.$$

Moreover, $\|\phi'_{c_1,c_2}\|_\infty \leq e^{\frac{2}{c_2-c_1}}$, and $\|\phi''_{c_1,c_2}\|_\infty \leq 12e^{\frac{4}{c_2-c_1}}$.

The main use of the smooth step function in Definition 1 is to extend a Boolean function $\hat{h}$ defined on the hypercube vertices $\{0,1\}^N$ to a continuous function h over the domain $[0,1]^N$. Besides being consistent on the vertices of the hypercube, we require it to be robust: the function $h(x) = \hat{h}(v)$ whenever x belongs to a "large" neighborhood of the vertex v. Moreover, another critical requirement for this interpolation is query efficiency: the value of the interpolated function at any point $x \in [0,1]^N$ must be computable using only a polynomial number of queries to the original function.

Lemma 2. *Let $\hat{h} : \{0,1\}^N \to \{0,1\}$ be a Boolean function. There exists a smooth $\mathcal{C}^\infty$ function $h = \mathcal{H}(\cdot, \hat{h}) : [0,1]^N \to [0,1]$ that satisfies*

(i) $\hat{h}(y) = h(x)$ for all $y \in \{0,1\}^N$ and all $x \in B^\infty_{1/6}(y) \cap [0,1]^N$;

(ii) $h(x) \in [0,1]$ for all $x \in [0,1]^N$;

(iii) $h(x)$ and $\nabla h(x)$ are computable with at most one call to $\hat{h}$ for all $x \in [0,1]^N$;

(iv) $\|\nabla h(x)\|_\infty \leq \frac{e^{12}}{2}$, and $\|D^2 h(x)\|_{\max} \leq 6e^{24}$ for all $x \in [0,1]^N$.

The construction of the interpolating function $\mathcal{H}(\cdot, \hat{h})$ is explicit and can be found in the proof of Lemma 2. In Figure 1b, we provide an illustrative example. The reason why we need a robust version of the interpolation will be discussed in the next section (and in particular in Remark 2).

5 An Oracle-Efficient Reduction From ORACLEPURECIRCUIT to SMOOTHBROUWER

As already mentioned, we will also rely on the following intermediate problem, which concerns the approximation of fixed points of a continuous function. This problem will be used in Section 6, in combination with ORACLEPURECIRCUIT, to construct a query-hard instance for GDA-FIXEDPOINT.

Problem 4 (SMOOTHBROUWER)**.** Given an approximation $\varepsilon > 0$ and an oracle implementing a Lipschitz and $\mathcal{C}^1$ function $F : [0,1]^d \to [0,1]^d$ and its Jacobian $J_F : [0,1]^d \to \mathbb{R}^{d \times d}$, find a $z \in [0,1]^d$ such that $\|F(z) - z\|_\infty \leq \varepsilon$.

The following theorem provides an oracle-efficient reduction from ORACLEPURECIRCUIT to SMOOTHBROUWER. In this construction, we rely heavily on the robustness of our interpolation as described in Lemma 2, and on the smooth step function introduced in Definition 1.

Theorem 5.1. *There is a reduction from an instance of* ORACLEPURECIRCUIT *(with oracle $\mathcal{L}$) to* SMOOTHBROUWER *with approximation $\varepsilon = \frac{1}{12}$, such that:*

- *$\mathcal{C}^\infty \ni F : [0,1]^d \to [0,1]^d$ where $d = |V|$ is the number of nodes of* ORACLEPURECIRCUIT*;*
- *any query to F or J_F can be simulated with at most $d = |V|$ queries to $\mathcal{L}$;*
- *we can efficiently recover a* ORACLEPURECIRCUIT *solution from any* SMOOTHBROUWER *one;*
- *the function F is $(e^{12}d)$-Lipschitz and $(12e^{24}d^2)$-smooth, and $|\partial_{z_j} F_i(z)| \le e^{12}$ for all $i, j \in [d]$.*

As a corollary, since the oracles of SMOOTHBROUWER can be simulated with $\text{poly}(d)$ queries to the oracle $\mathcal{L}$ of ORACLEPURECIRCUIT, and we can recover a solution of ORACLEPURECIRCUIT from any solution of SMOOTHBROUWER, we obtain an exponential lower bound on the number of queries required by SMOOTHBROUWER, recovering a similar result to that of Hirsch et al. [1989].

Corollary 1 (Exponential query-lower bound for SMOOTHBROUWER)**.** *For $\varepsilon = \frac{1}{12}$,* SMOOTHBROUWER *requires at least $2^{\Omega(d^c)}$ queries to the F and J_F oracles for some constant $c > 0$.*

It is important to make two comments regarding this result.

Remark 1. We need such a stronger result on the query complexity of SMOOTHBROUWER, since the historical version of Hirsch, Papadimitriou, and Vavasis [1989] only produces functions F that are Lipschitz but not $\mathcal{C}^1$. Indeed, in our final GDA-FIXEDPOINT construction, the oracle implementing the gradient ∇f is simulated using J_F, so J_F must be continuous for f to be smooth.

Remark 2. The robustness of the interpolation of Lemma 2 is essential in the reduction from ORACLEPURECIRCUIT to SMOOTHBROUWER since approximate fixed points only recover inputs that are close to Boolean vertices rather than exactly Boolean.

6 Exponential Lower Bound for GDA-FIXEDPOINT

In this section, we combine ORACLEPURECIRCUIT and SMOOTHBROUWER to provide a query-hard instance of GDA-FIXEDPOINT. Starting from an instance of ORACLEPURECIRCUIT with oracle $\mathcal{L}$ on $m = |V|$ nodes, we first construct, via Theorem 5.1, a SMOOTHBROUWER instance $F : [0,1]^m \to [0,1]^m$, where $m = |V|$. We then build a GDA-FIXEDPOINT instance in the spirit of Bernasconi and Castiglioni [2026], using many copies of SMOOTHBROUWER placed inside nodes of the ORACLEPURECIRCUIT instance. The major difference with respect to the proof of Bernasconi and Castiglioni [2026] is that we replaced the linear variational inequality used there with the SMOOTHBROUWER map F as the inner problem.

Let m be the size of the SMOOTHBROUWER map, i.e. $F : [0,1]^m \to [0,1]^m$, and let $\rho = \frac{1}{12}$ the approximation error. Recall that by Theorem 5.1 it holds $m = |V|$.

We construct the GDA-FIXEDPOINT instance based on the following parameters:

$$\delta = \frac{\rho^4}{400m^2e^{26}}, \quad n = \left\lceil \frac{2^{13}e^{13}m^4}{\delta^3} \right\rceil, \quad \text{and} \quad \varepsilon = \min\left(\frac{\delta}{n}, \frac{\delta^2}{m^4 2^4 e^{14}}\right).$$

For each $u \in V$, $i \in [n]$, $j \in [m]$, our instance includes variables $x^u_{i,j}$ and $y^u_{i,j}$, so the dimension of each player is $d = |V|nm = nm^2 = O(m^{12})$. Moreover we abbreviate $x^v_i = (x^v_{i,j})_{j\in[m]} \in \mathbb{R}^m$ and $x^v := (x^v_{i,j})_{i\in[n],j\in[m]} \in \mathbb{R}^{mn}$ (and similarly with $y^v_i \in \mathbb{R}^m$ and $y^v \in \mathbb{R}^{mn}$).

Before defining the objective f, it is useful to define, using F, the gadget:

$$H_v(x,y) = \sum_{i\in[n]} \left\langle F\left(\frac{x^v_i + y^v_i}{2}\right) - \frac{x^v_i + y^v_i}{2}, y^v_i - x^v_i \right\rangle,$$

and the "thresholded energy function":

$$\mathcal{E}_v(x,y) = \phi_{3m,3m+1}(\|x^v - y^v\|^2),$$

that is a smooth step function that is 0 if $\|x^v - y^v\|^2$ is less that $3m$ and 1 if it is bigger than $3m+1$.

Moreover, for each $w \in V$, we define a "signal" function $s_w(x,y)$ depending on the (unique) gate type w is output of. To do so, we use specific functions depending of the gate type. Formally, we set

$$s_w(x,y) = \begin{cases} g(\mathcal{E}_u(x,y) + \mathcal{E}_v(x,y)) & \text{if} \quad (u,v,w) \in \mathcal{G}_{\text{NOR}} \\ \ell(\mathcal{E}_u(x,y) - 1/4) & \text{if} \quad (u,w,v) \in \mathcal{G}_{\text{PURIFY}} \\ \ell(\mathcal{E}_u(x,y) + 1/4) & \text{if} \quad (u,v,w) \in \mathcal{G}_{\text{PURIFY}} \\ \mathcal{H}((\mathcal{E}_{u_i}(x,y))_{i\in[N]}, \mathcal{L}) & \text{if} \quad (u_1,\dots,u_N,w) \in \mathcal{G}_{\text{ORACLE}} \end{cases}$$

where $g(z) = \phi_{1/3,2/3}(1-z)$ and $\ell(z) = \phi_{5/12,7/12}(z)$. Finally, we let the objective

$$f(x,y) = \sum_{w\in V} s_w(x,y) \cdot H_w(x,y) + \varphi(x,y), \tag{4}$$

where $\varphi(x,y) = \sum_{w\in V}\sum_{i\in[n]} M_i \|x_i^w - y_i^w\|_2^2$ and $M_i := \delta(i - n/2)$.

Here, it becomes apparent the necessity of a smooth version of a query-hard Brouwer function F, since it is evident that the gradient of f will depend on J_F through the derivatives of H_w. Moreover, it is easy to see that, since all components of f have bounded first and second order derivatives, thanks to Lemmas 1 and 2, then $\|f\|_\infty, \|\nabla f\|_\infty, \|D^2 f\|_\infty = \text{poly}(d)$ and thus also $G, L, B = \text{poly}(d)$.

Thus, to complete the proof, we just need to show that:

1. **Correctness:** given a solution to GDA-FIXEDPOINT, we can recover a solution to either the ORACLEPURECIRCUIT or the SMOOTHBROUWER instance, and hence of the ORACLEPURECIRCUIT one by Theorem 5.1.
2. **Oracle Query Complexity:** any call to f or ∇f requires at most a polynomial number of calls to the ORACLEPURECIRCUIT oracle $\mathcal{L}$. Notice that these also include calls to the SMOOTHBROUWER oracles F and J_F, which in turn are implemented through queries to the ORACLEPURECIRCUIT oracle $\mathcal{L}$ by Theorem 5.1.

Indeed, if an algorithm solves GDA-FIXEDPOINT in $T(d)$ oracle queries to $(f, \nabla f)$, then this would give an algorithm for ORACLEPURECIRCUIT with $|V|$ nodes that runs in $T(\text{poly}(|V|))\text{poly}(|V|)$ queries, but the ORACLEPURECIRCUIT lower bound of Theorem 3.1 forces this quantity to be exponential in $|V|$ and in turn implies $T(d) = 2^{\Omega(d^c)}$ for some constant $c > 0$.

From any solution (x,y) of GDA-FIXEDPOINT and define an assignment $b : V \to \{0,1,\perp\}$ as

$$b(v) = \begin{cases} \mathcal{E}_v(x,y) & \text{if} \quad \mathcal{E}_v(x,y) \in \{0,1\} \\ \perp & \text{otherwise} \end{cases}. \tag{5}$$

Then, we can show the following dichotomy in the same spirit of Bernasconi and Castiglioni [2026]: either the decoded Boolean signals $b : V \to \{0,1,\perp\}$ are consistent with all gate outputs (and thus satisfy the ORACLEPURECIRCUIT constraints), or one of the replicated blocks $(v,i) \in V \times [n]$ already yields an approximate fixed point of the SMOOTHBROUWER instance. Formally:

Lemma 3 (Dichotomy)**.** *Let (x,y) be a solution to* GDA-FIXEDPOINT*. Then either there exists $i \in [n], v \in V$ such that $(x_i^v + y_i^v)/2$ is a solution to* SMOOTHBROUWER*, or $b : V \to \{0,1,\perp\}$, as defined in Equation (5), is a solution to* ORACLEPURECIRCUIT*.*

With this lemma, we can easily prove our main result.

Theorem 6.1. *Any algorithm that has oracle access to $f : [0,1]^d \times [0,1]^d \to [-1,1]$ and to its gradient ∇f where f and ∇f are 1-Lipschitz, and outputs a ε-approximate stationary point with $\varepsilon \le \text{poly}(1/d)$, requires at least $2^{d^{\Omega(1)}}$ queries to f or ∇f in the worst case.*

Proof. We divide the proof into two steps.

Correctness. Lemma 3 almost immediately proves the correctness of the reduction, as it shows that we either trivially have a solution to SMOOTHBROUWER or the ORACLEPURECIRCUIT constraints are satisfied by b, and ultimately a solution to ORACLEPURECIRCUIT thanks to Theorem 5.1.

Oracle Query Complexity. Now we show that the oracles for f and ∇f can be simulated with only polynomially many queries to the ORACLEPURECIRCUIT oracle $\mathcal{L}$. First, observe that evaluating $f(x, y)$ for any $x, y \in [0, 1]^d$ requires only polynomially many queries to the oracle $\mathcal{L}$. Specifically, computing $s_w(x, y)$ for each $w \in V$ requires at most $|V| = \text{poly}(d)$ queries to $\mathcal{L}$ thanks to Lemma 2. Moreover, the evaluation of the H_w's terms requires at most $m^2 n = \text{poly}(d)$ queries to $F_w(z)$. Thanks to Theorem 5.1, each evaluation requires at most m queries to $\mathcal{L}$. A similar analysis holds for ∇f, which also requires the evaluation of J_F and $\nabla s_w(x, y)$, for which similar bounds hold. □

Compared with the proof of Bernasconi and Castiglioni [2026], the key new ingredient is the treatment of the large-signal regime in the dichotomy argument, *i.e.*, that $s_v(x, y) = 1$ implies that $b(v) = 1$ or we can find a solution to SMOOTHBROUWER. The small-signal case follows the previous template, while the large-signal case now has to certify the existence of an approximate SMOOTHBROUWER fixed point, rather than a solution to the inner problem of Bernasconi and Castiglioni [2026]—a simpler problem related to linear variational inequalities (see Section E.2 for more details). We modified the inner gadget because the one based on variational inequalities was reduced from polymatrix games [Rubinstein, 2015]. While polymatrix games are computationally intractable from a complexity perspective, they can be solved using polynomially many queries.

Remark 3. In contrast to Section 5 (see Remark 2), the decoded gate signals in this reduction are already thresholded to Boolean values. For correctness, we therefore only require that the interpolation matches the function on the Boolean vertices, and we do not need the robust interpolation in Lemma 2. We nevertheless use the same robust interpolation throughout for the sake of uniformity.

Remark 4. The construction of our function f only guarantees that $G, L, B, 1/\varepsilon = \text{poly}(d)$, however, we can easily normalize f by $\max(G, L, B)$, by only needing to decrease ε by a polynomial factor.

6.1 Extension to High-Order Oracles

Our main theorem is stated for first-order algorithms, which have access to oracles f and ∇f. However, many algorithms that use second- (or higher-)order information have been proposed for min-max optimization [Letcher et al., 2018, Balduzzi et al., 2018, Zhang et al., 2020, Ha and Kim, 2022, Vyas and Bullins, 2023, Chinchilla et al., 2024]. However, all either assume an additional special structure or offer weaker guarantees (such as local convergence). Our argument can be extended even to rule out query-efficient algorithms of this sort. Indeed, fix any integer $p = O(1)$ and suppose that the algorithm can query $(f, \nabla f, D^2 f, \ldots, D^p f)$. Then, our lower bound still applies.

The main point is that each oracle can be simulated with polynomially many queries to the underlying ORACLEPURECIRCUIT oracle $\mathcal{L}$. The objective function f is $\mathcal{C}^\infty$ due to its construction from algebraic compositions of $\mathcal{C}^\infty$ components. Moreover, Lemma 2 can be strengthened to show that for any fixed p, the derivatives of the interpolation $h = \mathcal{H}(\cdot, \mathcal{L})$ can be computed with just one query to $\mathcal{L}$, as h depends on only one vertex of the hypercube. Computing the derivative tensor $D^r f$ for any $r \leq p$ also requires polynomially many queries to $\mathcal{L}$ because f is constructed from smooth step functions and interpolation gadgets through sums, products, and compositions, leading to polynomially many terms in the derivatives.

Therefore, if a p-th order algorithm solved the constructed GDA-FIXEDPOINT instance using T oracle queries, then we would obtain an algorithm for ORACLEPURECIRCUIT using $T \cdot \text{poly}(d)$ queries to $\mathcal{L}$. By the query lower bound of Theorem 3.1, we must have that T also grows exponentially.

7 Open Problems

Although our result resolves the query complexity of the problem in the regime where $\varepsilon = 1/\text{poly}(d)$, the question remains open for constant values of $\varepsilon > 0$. Indeed, our lower bound does not preclude the existence of a $\text{poly}(d)$-query algorithm for every fixed value of $\varepsilon > 0$, e.g., an algorithm making $d^{O(1/\varepsilon)}$ many queries. Proving or refuting such a PTAS-type query guarantee remains an interesting open problem. A second question concerns simpler instances. For min-max problems with a degree-2 objective over simplex domains, known enumeration-based techniques (e.g., the one of Lipton, Markakis, and Mehta [2003]) can be adapted to establish the existence of quasi-polynomial-time algorithms. It remains open whether this is tight: can one prove quasi-polynomial lower bounds already for degree-2 objectives over simplex domains, analogously to the phenomena of Nash equilibria shown by Rubinstein [2017]?

Acknowledgments and Disclosure of Funding

Martino Bernasconi and Andrea Celli were supported by an ERC grant (Project 101165466 — PLA-STEER). Matteo Castiglioni was supported by the EU Horizon project ELIAS (European Lighthouse of AI for Sustainability, No. 101120237).

This work was partially funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

Ioannis Anagnostides, Gabriele Farina, Tuomas Sandholm, and Brian Hu Zhang. A polynomial-time algorithm for variational inequalities under the Minty condition. *arXiv preprint arXiv:2504.03432*, 2025.

David Balduzzi, Sebastien Racaniere, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel. The mechanics of n-player differentiable games. In *International Conference on Machine Learning*, pages 354–363. PMLR, 2018.

Martino Bernasconi and Matteo Castiglioni. The complexity of min-max optimization with product constraints. In *STOC (to appear)*, 2026.

Martino Bernasconi, Matteo Castiglioni, Andrea Celli, and Gabriele Farina. On the role of constraints in the complexity of min-max optimization. *arXiv preprint arXiv:2411.03248*, 2024.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points I. *Mathematical Programming*, 184(1):71–120, 2020.

Xi Chen, Yuhao Li, and Mihalis Yannakakis. Computing a fixed point of contraction maps in polynomial queries. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1364–1373, 2024.

Raphael Chinchilla, Guosong Yang, and Joao P Hespanha. Newton and interior-point methods for (constrained) nonconvex–nonconcave minmax optimization with stability and instability guarantees. *Mathematics of Control, Signals, and Systems*, 36(2):381–421, 2024.

Constantinos Daskalakis. Non-concave games: A challenge for game theory's next 100 years. *Cowles Preprints*, page 4, 2022.

Constantinos Daskalakis and Ioannis Panageas. The limit points of (optimistic) gradient descent in min-max optimization. *Advances in neural information processing systems*, 31, 2018.

Constantinos Daskalakis, Stratis Skoulakis, and Manolis Zampetakis. The complexity of constrained min-max optimization. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1466–1478, 2021.

Argyrios Deligkas, John Fearnley, Alexandros Hollender, and Themistoklis Melissourgos. Pure-circuit: Tight inapproximability for PPAD. *Journal of the ACM*, 71(5):31:1–31:48, 2024.

Jelena Diakonikolas. Open problems in optimization. *SIAG on Optimization Views and News*, 33(1):1–12, 2025. URL `https://siagoptimization.github.io/assets/views/ViewsAndNews-33-1.pdf`.

Ian J 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.

Junsoo Ha and Gunhee Kim. On convergence of lookahead in smooth games. In *International Conference on Artificial Intelligence and Statistics*, pages 4659–4684. PMLR, 2022.

Michael D Hirsch, Christos H Papadimitriou, and Stephen A Vavasis. Exponential lower bounds for finding Brouwer fixed points. *Journal of Complexity*, 5(4):379–416, 1989.

Chi Jin, Praneeth Netrapalli, and Michael Jordan. What is local optimality in nonconvex-nonconcave minimax optimization? In *International conference on machine learning*, pages 4880–4889. PMLR, 2020.

Alistair Letcher, Jakob Foerster, David Balduzzi, Tim Rocktäschel, and Shimon Whiteson. Stable opponent shaping in differentiable games. *arXiv preprint arXiv:1811.08469*, 2018.

Tianyi Lin, Chi Jin, and Michael Jordan. On gradient descent ascent for nonconvex-concave minimax problems. In *International conference on machine learning*, pages 6083–6093. PMLR, 2020.

Richard J Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, pages 36–41, 2003.

Yixin Liu, Argyris Oikonomou, Weiqiang Zheng, Yang Cai, and Arman Cohan. COMAL: A convergent meta-algorithm for aligning LLMs with general preferences. *arXiv preprint arXiv:2410.23223*, 2024.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Panayotis Mertikopoulos, Christos Papadimitriou, and Georgios Piliouras. Cycles in adversarial regularized learning. In *Proceedings of the twenty-ninth annual ACM-SIAM symposium on discrete algorithms*, pages 2703–2717. SIAM, 2018.

Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Côme Fiegel, et al. Nash learning from human feedback. In *Forty-first International Conference on Machine Learning*, 2024.

Maher Nouiehed, Maziar Sanjabi, Tianjian Huang, Jason D Lee, and Meisam Razaviyayn. Solving a class of non-convex min-max games using iterative first order methods. *Advances in Neural Information Processing Systems*, 32, 2019.

Dmitrii M Ostrovskii, Andrew Lowy, and Meisam Razaviyayn. Efficient search of first-order Nash equilibria in nonconvex-concave smooth min-max problems. *SIAM Journal on Optimization*, 31 (4):2508–2538, 2021.

Anselm Paulus, Ilia Kulikov, Brandon Amos, Rémi Munos, Ivan Evtimov, Kamalika Chaudhuri, and Arman Zharmagambetov. Safety alignment of LMs via non-cooperative games. *arXiv preprint arXiv:2512.20806*, 2025.

Meisam Razaviyayn, Tianjian Huang, Songtao Lu, Maher Nouiehed, Maziar Sanjabi, and Mingyi Hong. Nonconvex min-max optimization: Applications, challenges, and recent theoretical advances. *IEEE Signal Processing Magazine*, 37(5):55–66, 2020.

Anton Rodomanov and Yurii Nesterov. Subgradient ellipsoid method for nonsmooth convex problems. *Mathematical Programming*, 199(1):305–341, 2023.

Aviad Rubinstein. Inapproximability of Nash equilibrium. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 409–418, 2015.

Aviad Rubinstein. Settling the complexity of computing approximate two-player Nash equilibria. *ACM SIGecom Exchanges*, 15(2):45–49, 2017.

Haoran Sun, Yusen Wu, Yukun Cheng, and Xu Chu. Game theory meets large language models: a systematic survey. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 10669–10677, 2025.

Gokul Swamy, Christoph Dann, Rahul Kidambi, Steven Wu, and Alekh Agarwal. A minimaximalist approach to reinforcement learning from human feedback. In *International Conference on Machine Learning*, pages 47345–47377. PMLR, 2024.

Stephen A Vavasis. Black-box complexity of local minimization. *SIAM Journal on Optimization*, 3 (1):60–80, 1993.

Stephen A Vavasis. Complexity issues in global optimization: a survey. In *Handbook of global optimization*, pages 27–41. Springer, 1995.

Abhijeet Vyas and Brian Bullins. Beyond first-order methods for non-convex non-concave min-max optimization. *arXiv preprint arXiv:2304.08389*, 2023.

Guojun Zhang, Kaiwen Wu, Pascal Poupart, and Yaoliang Yu. Newton-type methods for minimax optimization. *arXiv preprint arXiv:2006.14592*, 2020.

A Proof Omitted from Section 2

Theorem 2.1. *There exists a sufficiently large constant integer M such that any algorithm that outputs a solution to* StrongSperner *with parameters M and d must make $2^{\Omega(d)}$ many queries to the labeling λ.*

In order to prove this, we will use the following lower bound for finding a Brouwer fixed point.

Theorem A.1 (Hirsch et al. [1989])**.** *There exists a sufficiently small constant $\varepsilon > 0$ such that any algorithm that has black-box oracle access to a 2-Lipschitz (w.r.t. the ℓ_∞ norm) function $F : [0,1]^d \to [0,1]^d$ and outputs a point $x \in [0,1]^d$ such that $\|F(x) - x\|_\infty \leq \varepsilon$ requires $2^{\Omega(d)}$ queries.*

Proof of Theorem 2.1. Let ε be a sufficiently small constant for Theorem A.1. Consider any 2-Lipschitz function $F : [0,1]^d \to [0,1]^d$. We show how to reduce the problem of finding an ε-approximate fixed point of F to a StrongSperner instance.

Without loss of generality, we can assume that for all $i \in [d]$, $F_i(x) > 0$ when $x_i = 0$, and $F_i(x) < 1$ when $x_i = 1$. Indeed, if this is not the case, then we can consider the function $(1 - \varepsilon/2)F + (\varepsilon/2)v$ instead of the function F, where $v = (1/2, \ldots, 1/2) \in [0,1]^d$, and look for an $(\varepsilon/2)$-approximate fixed point.

Let $M := \lceil 1 + 3/\varepsilon \rceil$ and define the function $\phi : [M] \to [0,1], t \mapsto (t-1)/(M-1)$. For a vector $p \in [M]^d$, define by a slight abuse of notation $\phi(p) := (\phi(p_1), \ldots, \phi(p_d))$. Note that $\phi_i(p) = \phi(p_i)$. Now define the StrongSperner labeling $\lambda : [M]^d \to \{-1, +1\}^d$ as follows. For any $p \in [M]^d$

$$\lambda(p) := \begin{cases} +1 & \text{if } F_i(\phi(p)) > \phi_i(p) \\ -1 & \text{if } F_i(\phi(p)) \leq \phi_i(p) \end{cases}$$

Note that λ satisfies the StrongSperner boundary conditions. Furthermore, any query to λ can be answered by performing at most one query to F.

It remains to show that any StrongSperner solution $p^{(1)}, \ldots, p^{(d)} \in [M]^d$ to λ yields an ε-approximate fixed point of F. Let $q := p^{(1)}$. We will show that $\phi(q)$ is an approximate fixed point. Consider any $i \in [d]$. Since $p^{(1)}, \ldots, p^{(d)}$ form a StrongSperner solution, there exists q' with $\|q - q'\|_\infty \leq 1$ such that $\lambda_i(q') = -1$. This implies that $F_i(\phi(q')) \leq \phi_i(q')$. Now we can write

$$F_i(\phi(q)) \leq F_i(\phi(q')) + |F_i(\phi(q)) - F_i(\phi(q'))| \leq \phi_i(q') + 2\varepsilon/3 \leq \phi_i(q) + \varepsilon$$

where we used the fact that F is 2-Lipschitz and ϕ is $(\varepsilon/3)$-Lipschitz. Using the fact that there exists q' with $\|q - q'\|_\infty \leq 1$ such that $\lambda_i(q') = +1$, a similar argument also proves that $F_i(\phi(q)) \geq \phi_i(q) - \varepsilon$. Since this holds for any $i \in [d]$, we have shown that $\phi(q)$ is an ε-approximate fixed point of F. □

B Proof Omitted from Section 3

Theorem 3.1. *Given an instance of* OraclePureCircuit *with $|V|$ nodes, any algorithm that has black-box oracle access to $\mathcal{L}$ and outputs a solution to* OraclePureCircuit *requires $2^{\Omega(|V|^{1/3})}$ queries to $\mathcal{L}$.*

We will follow the reduction from Deligkas et al. [2024, Section 3.2] very closely. They provide a polynomial-time reduction from the white-box version of StrongSperner to PureCircuit. In what follows, we describe the modifications that need to be made to their reduction in order to obtain a query-efficient reduction from the black-box version of StrongSperner to OraclePureCircuit.

Let $\lambda : [M]^d \to \{-1, +1\}^d$ be a StrongSperner labeling satisfying the boundary conditions, as defined in Problem 2. We use the same notation as in Deligkas et al. [2024, Section 3.2], except that we have used d to denote the dimension of the StrongSperner instance, whereas they use N. We assume that the width M of the instance is a sufficiently large constant such that the query lower bound of $2^{\Omega(d)}$ from Theorem 2.1 applies.

Construction. The construction of the ORACLEPURECIRCUIT instance is completely identical to the construction of the PURECIRCUIT instance in Deligkas et al. [2024, Section 3.2], except for the "circuit stage". In that part of the construction, the goal is to simulate the evaluation of the STRONGSPERNER labeling λ on some input. In fact, this needs to be done on K separate occasions, where $K := 3dM^2$. Fix some $k \in [K]$. We are given nodes $(u_{i,j}^{(k)})_{(i,j)\in[d]\times[M]}$ representing a point in $[M]^d$. Namely, for each $i \in [d]$, $u_{i,1}^{(k)}, \ldots, u_{i,M}^{(k)}$ represents an element in $[M]$ in unary. Our task, or "contract", is to ensure that some designated output nodes, denoted $v_1^{(k)}, \ldots, v_d^{(k)}$, encode the output of λ on the input encoded by $(u_{i,j}^{(k)})_{(i,j)\in[d]\times[M]}$. By this we mean that if the ith output of λ is $+1$, then $v_i^{(k)} = 1$, and if the ith output of λ is -1, then $v_i^{(k)} = 0$. Importantly, for any given $k \in [K]$, the contract only requires us to ensure this correct output when *all* inputs $(u_{i,j}^{(k)})_{(i,j)\in[d]\times[M]}$ are correct bits, i.e., lie in $\{0, 1\}$.

In the original reduction by Deligkas et al. [2024], the contract is enforced by using the PURECIRCUIT gates to simulate the execution of the Boolean circuit computing λ. In our case, λ is a black-box and we are not given a circuit computing it. We will thus need to use the oracle gates to enforce the contract. We let $N := Md + d$ and define the oracle $\mathcal{L} : \{0, 1\}^N \to \{0, 1\}$ as follows. On input $p \in \{0, 1\}^N$, decomposed as $(z, t) \in \{0, 1\}^{Md} \times \{0, 1\}^d$:

1. If $t \in \{0, 1\}^d$ contains a 1 in exactly one position, and all other entries are 0, then let i denote the entry such that $t_i = 1$, and proceed to the next step. If this is not the case, then the oracle $\mathcal{L}$ outputs something arbitrary, say 0.
2. Interpret $z \in \{0, 1\}^{Md}$ as representing a corresponding point $\overline{z} \in [M]^d$, where each entry is given in unary. Evaluate $\lambda_i(\overline{z})$ by performing one query to λ. (Recall that i is the index such that $t_i = 1$ from the first step.)
3. If $\lambda_i(\overline{z}) = +1$, the oracle $\mathcal{L}$ outputs 1. If $\lambda_i(\overline{z}) = -1$, the oracle $\mathcal{L}$ outputs 0.

Coming back to the contract we have to enforce, we can now use the oracle gate (with oracle $\mathcal{L}$). Namely, for each $k \in [K]$ and $i \in [d]$, we ensure that $v_i^{(k)}$ has the correct value by introducing an oracle gate that has $v_i^{(k)}$ as output, and takes as input $(u_{\ell,j}^{(k)})_{(\ell,j)\in[d]\times[M]}$ for the first Md bits. The remaining d input bits are hardcoded to be $(0, \ldots, 0, 1, 0, \ldots, 0)$, where the 1 appears in the ith position. This indicates to the oracle $\mathcal{L}$ that we want the ith output of λ. In order to create these hardcoded bits, it suffices to create d nodes that are guaranteed to be 0 in any solution and d nodes that are guaranteed to be 1 in any solution. Then, each time we use an oracle gate, we can pick the corresponding hardcoded bits as input. In Figure 2 we show a constant-size gadget which ensures that a particular node always has value 0. We include a proof of this in Lemma 4 at the end of this section. From this node that always has value 0, a node that always has value 1 can easily be constructed using one PURIFY and one NOR gate.

Correctness. It is now easy to see that the contract is correctly enforced. Indeed, whenever the nodes $(u_{\ell,j}^{(k)})_{(\ell,j)\in[d]\times[M]}$ all have values in $\{0, 1\}$, the fact that the oracle gate has to be satisfied will imply that the nodes $v_1^{(k)}, \ldots, v_d^{(k)}$ encode the correct output values. When the nodes $(u_{\ell,j}^{(k)})_{(\ell,j)\in[d]\times[M]}$ do not all have values in $\{0, 1\}$, then the contract does not require us to enforce anything.

As we have fulfilled the "circuit stage" contract, i.e., Lemma 3.3 of Deligkas et al. [2024], and the rest of the construction is identical, the correctness of the reduction follows by the exact same arguments as in Deligkas et al. [2024, Section 3.2]. Namely, it holds that any solution of the constructed ORACLEPURECIRCUIT instance yields a solution of the STRONGSPERNER instance.

Oracle Query Complexity. First, let us note that in order to answer a query to $\mathcal{L}$ we only need to make at most one query to λ. Thus, by Theorem 2.1, we know that at least $2^{\Omega(d)}$ queries to $\mathcal{L}$ are required to solve the ORACLEPURECIRCUIT instance.

In order to show that this corresponds to a lower bound of $2^{\Omega(|V|^{1/3})}$, where $|V|$ is the number of nodes in the ORACLEPURECIRCUIT instance, it remains to argue that $|V| = O(d^3)$. By inspection of the construction in Section 3.2 of Deligkas et al. [2024], we have:

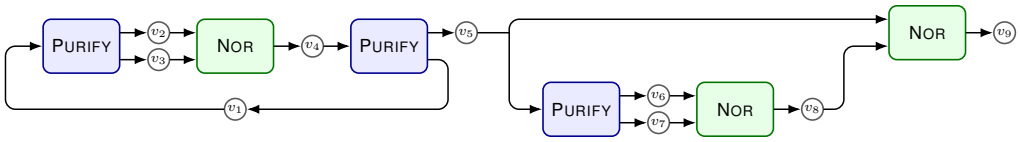


Figure 2: The gadget implementing the constant node $b(v_9) = 0$. Assuming that this gadget actually forces $b(v_9)$ (proven in Lemma 4), we can easily implement a similar one that forces an assignment of 1 to a specific node. Indeed, we could further apply a PURIFY gate to v_9, with outputs v_{10} and v_{11}, and then a further NOR gate with inputs v_{10} and v_{11} and output v_{12}. If $b(v_9) = 0$ it is clear that only $b(v_{12}) = 1$ satisfies the additional gates.

- The "purification stage" requires $O(MdK) = O(d^2M^3)$ nodes, where we used the fact that $K = 3dM^2$.
- The "circuit stage", which is the only stage that we modified, requires $dK = O(d^2M^2)$ for the oracle output nodes, as well as $O(d)$ nodes to create the hardcoded 0 and 1 bits.
- The "sorting stage" requires a sorting network over K elements for each $i \in [d]$. By using a simple sorting network of size $O(K^2)$, we thus obtain an upper bound of $O(dK^2) = O(d^3M^4)$ nodes.
- The "selection stage" does not introduce any additional nodes.

Putting everything together, we have $|V| = O(d^3M^4) = O(d^3)$, where we used the fact that M is a constant. A subtle point is that the reduction in Section 3.2 of Deligkas et al. [2024] also uses gates AND, OR, NOT, and COPY. However, it is easy to see that each of those gates can be simulated by a constant-size gadget that uses only NOR and PURIFY gates. Thus, we still have $|V| = O(d^3)$. This completes the proof of Theorem 3.1. Below, we include a lemma which proves the correctness of the gadget that we used to create the hardcoded bits.

Lemma 4. *In the gadget described in Figure 2, for every assignment $b : V \to \{0, 1, \perp\}$ that satisfies all gates, it must hold that $b(v_9) = 0$.*

Proof. Let $b : V \to \{0, 1, \perp\}$ be any assignment that satisfies all gates. First, we will argue that $b(v_1) = \perp$. Assume by contradiction that $b(v_1) = 1$, then we would also have $b(v_2) = b(v_3) = 1$ by definition of the PURIFY gate. Then $b(v_4) = 0$ by definition of the NOR constraints. Now the PURIFY gate, with v_4 as input, has a node with a pure bit input, and thus its outputs are both pure and assigned with the same bit, which is $0 = b(v_4) = b(v_5) = b(v_1)$. Contradicting the fact that $b(v_1)$ was assumed to be 1. A similar, symmetric situation will occur if we assume $b(v_1) = 0$, thus leaving only $b(v_1) = \perp$ as an option.

Now we claim that $b(v_5) \in \{0, 1\}$. Indeed, the PURIFY gate with input v_4 has as outputs v_1 and v_5, but the PURIFY gate constraints force at least one of the two output bits to be pure, and, since v_1 cannot be pure, then v_5 must be.

Now observe that if $b(v_5) \in \{0, 1\}$, then the statement follows. Indeed, if $b(v_5) = 0$, then $b(v_6) = b(v_7) = b(v_5) = 0$ by the constraints of the PURIFY gate, and $b(v_8) = 1$ by the ones on the NOR gate. Then, if $b(v_5) = 0$ and $b(v_8) = 1$, then the last NOR gate forces $b(v_9) = 0$.

Symmetrically, if $b(v_5) = 1$ then it leads to $b(v_8) = 0$ and similarly to before, to $b(v_9) = 0$, by the last NOR gate constraints. □

C Proofs Omitted from Section 4

Lemma 1. *Given two $c_1 < c_2$, the function ϕ_{c_1,c_2} is in $\mathcal{C}^\infty$ and satisfies:*

$$\phi_{c_1,c_2}(x) \in \begin{cases} \{0\} & \text{If } x \le c_1 \\ (0,1) & \text{if } x \in (c_1, c_2) \\ \{1\} & \text{if } x \ge c_2 \end{cases}.$$

Moreover, $\|\phi'_{c_1,c_2}\|_\infty \le e^{\frac{2}{c_2-c_1}}$, and $\|\phi''_{c_1,c_2}\|_\infty \le 12e^{\frac{4}{c_2-c_1}}$.

Proof. We only show the bounds on the derivatives. Define $a := x - c_1$ and $b := c_2 - x$, and also $A = e^{-1/a}$, $B = e^{-1/b}$, $S = A + B$ and $\Delta = c_2 - c_1$. Then we for all $x \in (c_1, c_2)$ we can write $\phi_{c_1,c_2}(x) = A/S$. We are only interested in the behavior of ϕ_{c_1,c_2} in $x \in (c_1, c_2)$ since outside this interval $\phi'_{c_1,c_2} = \phi''_{c_1,c_2} = 0$. We can observe that $A' = \frac{A}{a^2}$, $B' = \frac{B}{b^2}$, and thus $|A'|, |B'| \leq 1$ since $\frac{e^{-1/t}}{t^2} \leq 1$ for $t > 0$. Moreover, (since at least one between a and b will be at least $\Delta/2$) at least one between A and B is at least $e^{-2/\Delta}$ so $S \geq e^{-2/\Delta}$.

Thus, we can bound the first derivative by observing that $\phi'_{c_1,c_2}(x) = \frac{A'(A+B)-A(A'+B')}{S^2} = \frac{A'B-AB'}{S^2}$, which implies that

$$\begin{aligned}
|\phi'_{c_1,c_2}(x)| &\leq \frac{|A'|B + A|B'|}{S^2} \\
&\leq \frac{A+B}{S^2} \\
&\leq \frac{1}{S} \leq e^{2/\Delta}.
\end{aligned}$$

For the second derivative, we can observe that

$$\begin{aligned}
\phi''_{c_1,c_2}(x) &= \frac{(A'B - AB')'S^2 - (S^2)'(A'B - AB')}{S^4} \\
&= \frac{(A''B - AB'')S^2 - 2S(A' + B')(A'B - AB')}{S^4} \\
&= \frac{A''B - AB''}{S^2} - 2\frac{(A' + B')(A'B - AB')}{S^3},
\end{aligned}$$

Moreover, we can observe that, for $t > 0$ we have $\frac{e^{-1/t}}{t^4} \leq 5$ and $\frac{e^{-1/t}}{t^3} \leq \frac{3}{2}$, so that $|A''| \leq e^{-1/a}\left(\frac{1}{a^4} + \frac{2}{a^3}\right) \leq 8$ and the same holds for $|B''|$. Plugging these bound in $|\phi''_{c_1,c_2}(x)|$ we obtain

$$\begin{aligned}
|\phi''_{c_1,c_2}(x)| &\leq \frac{|A''|B + A|B''|}{S^2} + 2\frac{(|A'| + |B'|)(|A'|B + A|B'|)}{S^3} \\
&\leq \frac{8}{S} + \frac{4}{S^2} \\
&\leq 8e^{-2/\Delta} + 4e^{-4/\Delta} \\
&\leq 12e^{-4/\Delta},
\end{aligned}$$

concluding the proof. □

Lemma 2. *Let $\hat{h} : \{0,1\}^N \to \{0,1\}$ be a Boolean function. There exists a smooth $\mathcal{C}^\infty$ function $h = \mathcal{H}(\cdot, \hat{h}) : [0,1]^N \to [0,1]$ that satisfies*

(i) $\hat{h}(y) = h(x)$ for all $y \in \{0,1\}^N$ and all $x \in B^\infty_{1/6}(y) \cap [0,1]^N$;

(ii) $h(x) \in [0,1]$ for all $x \in [0,1]^N$;

(iii) $h(x)$ and $\nabla h(x)$ are computable with at most one call to $\hat{h}$ for all $x \in [0,1]^N$;

(iv) $\|\nabla h(x)\|_\infty \leq \frac{e^{12}}{2}$, and $\|D^2 h(x)\|_{\max} \leq 6e^{24}$ for all $x \in [0,1]^N$.

Proof. Let $\alpha(t) = 1 - \phi_{\frac{1}{6},\frac{1}{3}}(t)$. and for any vertex $y \in \{0,1\}^N$ define

$$\Phi_y(x) = \prod_{i \in [N]} \alpha(y_i + (1 - 2y_i)x_i).$$

Crucially, by Lemma 1, for a vertex $y \in \{0,1\}^N$, we have that $\Phi_y(x) = 1$ for all $x \in B^\infty_{1/6}(y)$; that $\Phi_y(x) = 0$ for all $x \notin B^\infty_{1/3}(y)$; and that $\Phi_y(x) \in (0,1)$ for all $x \in B^\infty_{1/3}(y) \setminus B^\infty_{1/6}(y)$.

For any $x \in [0,1]^N$ we call any vertex $y \in \{0,1\}^N$ *active* if $x \in B^\infty_{1/3}(y)$. Note that, since ℓ_∞-balls of radius $1/3$ centered at the vertices of the hypercube are disjoint, at most 1 vertex is active for all $x \in [0,1]^N$. So if there is any coordinate i such that $x_i \in (\frac{1}{3}, \frac{2}{3})$, then there is no active vertex. Otherwise, we can identify the active vertex y^* by rounding every $x_i \leq 1/3$ to $y^*_i = 0$ and every $x_i \geq 2/3$ to $y^*_i = 1$.

Then we define

$$h(x) = \frac{1}{2} + \sum_{y \in \{0,1\}^N} \Phi_y(x)(\hat{h}(y) - \tfrac{1}{2})$$

for all $x \in [0,1]^N$. Since there is at most one active node for all $x \in [0,1]^N$, we have that either $\Phi_y(x) = 0$ for all $y \in \{0,1\}^N$ or that there is a single active one y^*. In the first case, we have $h(x) = \frac{1}{2}$, in the second one we have $h(x) = \frac{1}{2} + \Phi_{y^*}(x)(\hat{h}(y^*) - 1/2)$. Moreover if $x \in B^\infty_{1/6}(y^*)$ we have that $\Phi_{y^*}(x) = 1$ and thus $h(x) = \hat{h}(y^*)$. This proves the first and second items of the statement. Note that since we can know what the unique active node (if any) is, we need only one evaluation of $\hat{h}$.

By direct calculations we have that for all $y \in \{0,1\}^N$ we have:

- $\partial_{x_j} \Phi_y(x) = (1-2y_j)\alpha'(y_j + (1-2y_j)x_j) \prod_{i \neq j} \alpha(y_i + (1-2y_i)x_i)$;
- $\partial_{x_j,x_k} \Phi_y(x) = (1-2y_j)\alpha'(y_j + (1-2y_j)x_j)(1-2y_k)\alpha'(y_k + (1-2y_k)x_k) \prod_{i \neq j,k} \alpha(y_i + (1-2y_i)x_i)$ if $k \neq j$;
- $\partial_{x_j,x_k} \Phi_y(x) = \alpha''(y_j + (1-2y_j)x_j) \prod_{i \neq j} \alpha(y_i + (1-2y_i)x_i)$;

which in turn, thanks to Lemma 1 and $\alpha(x) = \phi_{\frac{1}{6},\frac{1}{3}}(\frac{1}{2} - x)$, implies that for all $x \in [0,1]^N$:

- $|\partial_{x_j} \Phi_y(x)| \leq \|\alpha'\|_\infty \leq e^{12}$;
- $|\partial_{x_j,x_k} \Phi_y(x)| \leq \|\alpha'\|^2_\infty \leq e^{24}$ if $k \neq j$;
- $|\partial_{x_j,x_k} \Phi_y(x)| \leq \|\alpha''\|_\infty \leq 12e^{24}$.

Thus, thanks to the observations above, we can also bound the derivative. A direct calculation shows that

$$\partial_{x_j} h(x) = \begin{cases} 0 & \text{if no vertex is active,} \\ (\hat{h}(y) - \frac{1}{2})\partial_{x_j} \Phi_y(x) & \text{if } y \text{ is active.} \end{cases}$$

and thus,

$$|\partial_{x_j} h(x)| \leq \frac{1}{2}|\partial_{x_j} \Phi_y(x)| \leq \frac{1}{2}\|\alpha'\|_\infty \leq e^{12}/2.$$

Similarly, we can note that

$$\partial_{x_j,x_k} h(x) = \begin{cases} 0 & \text{if no vertex is active,} \\ (\hat{h}(y) - \frac{1}{2})\partial_{x_j,x_k} \Phi_y(x) & \text{if } y \text{ is active.} \end{cases}$$

and thus, thanks to the previous observations, $|\partial_{x_j,x_k} h(x)| \leq 6e^{24}$. □

D Proof Omitted from Section 5

Theorem 5.1. *There is a reduction from an instance of* OraclePureCircuit *(with oracle $\mathcal{L}$) to* SmoothBrouwer *with approximation $\varepsilon = \frac{1}{12}$, such that:*

- *$\mathcal{C}^\infty \ni F : [0,1]^d \to [0,1]^d$ where $d = |V|$ is the number of nodes of* OraclePureCircuit*;*
- *any query to F or J_F can be simulated with at most $d = |V|$ queries to $\mathcal{L}$;*
- *we can efficiently recover a* OraclePureCircuit *solution from any* SmoothBrouwer *one;*

- *the function F is $(e^{12}d)$-Lipschitz and $(12e^{24}d^2)$-smooth, and $|\partial_{z_j} F_i(z)| \le e^{12}$ for all $i, j \in [d]$.*

Proof. Given an instance of ORACLEPURECIRCUIT with vertex set V and oracle $\mathcal{L}$, we build a smooth function $F : [0,1]^d \to [0,1]^d$, where as promised $d = |V|$. Let $\varepsilon = 1/12$. For every $w \in V$, the w-th component $F_w : [0,1]^d \to [0,1]$ is defined based on the gate type for which w is an output:

- If $(u, v, w) \in \mathcal{G}_{\text{NOR}}$, we define $F_w(z) = g(z_u + z_v)$, where
$$g(z) = 1 - \phi_{\frac{1}{3},\frac{2}{3}}(z).$$
- If $(u, v, w) \in \mathcal{G}_{\text{PURIFY}}$, we define $F_v(z) = \ell(z_u + \frac{1}{4})$ and $F_w(z) = \ell(z_u - \frac{1}{4})$, where
$$\ell(z) = \phi_{\frac{5}{12},\frac{7}{12}}(z).$$
- If $(u_1, \ldots, u_N, v) \in \mathcal{G}_{\text{ORACLE}}$, we define $F_v(z) = \mathcal{H}((z_{u_1}, \ldots, z_{u_N}), \mathcal{L})$ according to Lemma 2.

Correctness. Let $z \in [0,1]^d$ be any point satisfying $\|F(z) - z\|_\infty \le \varepsilon$. We define the assignment $b : V \to \{0, 1, \perp\}$ as follows:
$$b(v) = \begin{cases} 0 & \text{if} \quad z_v \le \frac{1}{6} \\ \perp & \text{if} \quad z_v \in (\frac{1}{6}, \frac{5}{6}) \\ 1 & \text{if} \quad z_v \ge \frac{5}{6} \end{cases}.$$

Recall that by Lemma 1, it holds $g(z) = 0$ if $z \ge \frac{2}{3}$, $g(z) \in (0,1)$ if $z \in (\frac{1}{3}, \frac{2}{3})$, and $g(z) = 1$ if $z \le \frac{1}{3}$. Moreover, $\ell(z) = 0$ if $z \le \frac{5}{7}$, $\ell(z) \in (0,1)$ if $z \in (\frac{5}{12}, \frac{7}{12})$, and $\ell(z) = 1$ if $z \ge \frac{7}{12}$.

We prove that b is a valid assignment to ORACLEPURECIRCUIT since it satisfies all the gate constraints posed by ORACLEPURECIRCUIT:

- Gate $(u, v, w) \in \mathcal{G}_{\text{NOR}}$. If $b(v) = b(u) = 0$, then by definition $z_u, z_v \le 1/6$ and $z_u + z_v \le 1/3$ and thus $F_w(z) = g(z_u + z_v) = 1$. Since z is a solution to SMOOTHBROUWER it holds that $z_w \ge 1 - \varepsilon > 5/6$ and thus $b(w) = 1$ as required. Similarly, if $b(u) = 1$ or $b(v) = 1$, then we have that $z_u + z_v \ge 5/6$ and thus $F_w(z) = g(z_u + z_v) = 0$, which implies that $z_w \le \varepsilon < 1/6$ and thus $b(w) = 0$. In all other cases ORACLEPURECIRCUIT does not impose any constraints on the NOR gate.
- Gate $(u, v, w) \in \mathcal{G}_{\text{PURIFY}}$. If $b(u) = 0$ then $z_u \le 1/6$. Thus, $F_v(z) = \ell(z_u + 1/4) = 0$. Moreover, $z_u - 1/4 \le 5/12$ and thus also $F_w(z) = 0$. Thus, since z is a solution to SMOOTHBROUWER, $z_v, z_w \le \varepsilon < 1/6$ and $b(v) = b(w) = 0$. Similarly, if $b(u) = 1$ then $z_u \ge 5/6$, $z_u + 1/4, z_u - 1/4 \ge 7/12$ and $F_v(z) = F_w(z) = 1$. It follows that $z_v, z_w \ge 1 - \varepsilon > 5/6$ and $b(v) = b(w) = 1$. Finally, if $b(u) = \perp$ then $z_u \in (1/6, 5/6)$ and it is easy to see that either $z_u + 1/4 \ge 7/12$ or $z_u - 1/4 \le 5/12$. In the first case, we have $F_v(z) = 1$ (and thus $z_v \ge 1 - \varepsilon > 5/6$ and $b(v) = 1$), while in the second case we have $F_w(z) = 0$ (and thus $z_w \le \varepsilon < 1/6$ and $b(w) = 0$). This shows that the PURIFY constraints are also satisfied.
- Gate $(u_1, \ldots, u_N, v) \in \mathcal{G}_{ORACLE}$. Assume $(b(u_1), \ldots, b(u_N)) \in \{0,1\}^N$, otherwise there are no constraints. Define $z' = (b(u_1), \ldots, b(u_N))$. Note that if $z'_i = b(u_i) = 1$ then $z_{u_i} \ge 5/6$ and $|z'_i - z_{u_i}| \le 1/6$. Similarly, if $z'_i = b(u_i) = 0$, then $z_{u_i} \le 1/6$ and $|z'_i - z_{u_i}| \le 1/6$. Therefore, $(z_{u_1}, \ldots, z_{u_N}) \in B^\infty_{1/6}(z')$ and from Lemma 2 this implies that $F_v(z) = \mathcal{L}(z')$ and thus $|z_v - F_v(z)| \le \varepsilon$. If $\mathcal{L}(z') = 1$ then $z_v \ge 1 - \varepsilon > 5/6$ and $b(v) = 1$. If $\mathcal{L}(z') = 0$ then $z_v \le \varepsilon < 1/6$ and $b(v) = 0$. In either case $\mathcal{L}(z') = b(v)$. This shows that the constraints on the ORACLE gates are also satisfied.

Oracle Query Complexity. We show that any evaluation of F or its Jacobian J_F requires at most $|V|$ queries to the oracle $\mathcal{L}$. Only coordinates corresponding to outputs of ORACLE gates require access to $\mathcal{L}$. Since there are at most $|V|$ such components, we can compute the value and the Jacobian in at most $|V|$ oracle calls to $\mathcal{L}$, since Lemma 2 shows that each component value and derivative can be obtained via at most one query to $\mathcal{L}$.

Bounding the Lipschitzness and smoothness. Each coordinate $w \in V$ of F_w, is either the output of $g(z_u + z_v), \ell(z_u \pm 1/4)$ or $\mathcal{H}((z_{u_1}, \ldots, z_{u_N}), \mathcal{L})$. Furthermore, by and so by Lemma 1 we can bound $\|\partial_{z_u} g\|_\infty \leq e^6, \|\partial_{z_u} \ell\|_\infty \leq e^{12}$ for all $u \in V$, while Lemma 2 ensures that $\|\partial_{z_u} \mathcal{H}((z_{u_1}, \ldots, z_{u_N}), \mathcal{L})\|_\infty \leq e^{12}/2$ for all $u \in V$. In all cases, we have $|\partial_{z_u} F_w(z)| \leq e^{12}$, and thus every component is $e^{12}\sqrt{d}$-Lipschitz making the function $(e^{12}d)$-Lipschitz.

Similarly, we can show that F is polynomially smooth. Indeed, by the same reasoning above, we have that $\|\partial_{z_u, z_{u'}} g\|_\infty \leq 12e^{12}, \|\partial_{z_u, z_{u'}} \ell\|_\infty \leq 12e^{24}$ for all $u, u' \in V$ by Lemma 1, while $\|\partial_{z_u, z_{u'}} \mathcal{H}((z_{u_1}, \ldots, z_{u_N}), \mathcal{L})\|_\infty \leq 6e^{24}$ for all $u, u' \in V$ by Lemma 2. This lets us conclude that $|\partial_{z_u, z_{u'}} F_w(z)| \leq 12e^{24}$, and thus, that F is $(12e^{24}d^2)$-smooth. □

E Proof of the Dichotomy Lemma

We start by computing the partial derivatives of f. To do so, we first need some additional notation. We denote with $\text{In}(w) \subseteq V$ the set of nodes that are inputs to the gate to which w is an output, and $\text{Out}(q) = \{w : q \in \text{In}(w)\}$ is the set of nodes that are outputs of a gate to which q is an input. Moreover, we define the displacement function $G(z) = F(z) - z$, where F is the function defining the SMOOTHBROUWER instance.

Lemma 5. *The partial derivatives of f (defined as per Equation (4)) are*

$$\partial_{x^q_{i,j}} f(x, y) = s_q(x, y) \partial_{x^q_{i,j}} H_q(x, y) + 2(M_i + \Delta_q(x, y))(x^q_{i,j} - y^q_{i,j}) \tag{6a}$$

$$\partial_{y^q_{i,j}} f(x, y) = s_q(x, y) \partial_{y^q_{i,j}} H_q(x, y) - 2(M_i + \Delta_q(x, y))(x^q_{i,j} - y^q_{i,j}), \tag{6b}$$

where

$$\Delta_q(x, y) = \sum_{w \in \text{Out}(q)} H_w(x, y) \phi'_{3m,3m+1}(\|x^q - y^q\|^2) \partial_{\mathcal{E}_q(x,y)} s_w(x, y). \tag{7}$$

Proof. We recall the definition of $f(x, y)$ which is:

$$f(x, y) = \sum_{w \in V} s_w(x, y) \cdot H_w(x, y) + \varphi(x, y).$$

Since H_w is a local term (i.e., it depends only on the variables x^w, y^w), and s_w depends only on the nodes that are inputs of w (and in particular $q \notin \text{In}(q)$), we have that:

$$\partial_{x^q_{i,j}} \left(\sum_{w \in V} s_w(x, y) \cdot H_w(x, y) \right) = s_q(x, y) \partial_{x^q_{i,j}} H_q(x, y) + \sum_{w \in \text{Out}(q)} H_w(x, y) \cdot \partial_{x^q_{i,j}} s_w(x, y). \tag{8}$$

Then, by chain rule, for every $w \in \text{Out}(q)$, we have $\partial_{x^q_{i,j}} s_w(x, y) = \partial_{\mathcal{E}_q(x,y)} s_w(x, y) \partial_{x^q_{i,j}} \mathcal{E}_q(x, y)$,while for every $w \notin \text{Out}(q)$ we have $\partial_{x^q_{i,j}} s_w(x, y) = 0$.

Note that by definition, $\mathcal{E}_q(x, y)$ is a function of only $\|x^q - y^q\|^2$ and in particular $\mathcal{E}_q(x, y) = \phi_{3m,3m+1}(\|x^q - y^q\|^2)$. Thus, for all $w \in \text{Out}(q)$:

$$\partial_{x^q_{i,j}} s_w(x, y) = 2\partial_{\mathcal{E}_q(x,y)} s_w(x, y) \phi'_{3m,3m+1}(\|x^q - y^q\|^2)(x^q_{i,j} - y^q_{i,j}). \tag{9}$$

Moreover, it is easy to check that:

$$\partial_{x^q_{i,j}} \varphi(x, y) = 2M_i(x^q_{i,j} - y^q_{i,j}). \tag{10}$$

Combining Equations (8) to (10) with the definition of $\Delta_q(x, y)$ of Equation (7), we can deduce that:

$$\partial_{x^q_{i,j}} f(x, y) = s_q(x, y) \partial_{x^q_{i,j}} H_q(x, y) + 2 \left(\Delta_q(x, y) + M_i\right) (x^q_{i,j} - y^q_{i,j}).$$

Similarly, we get that

$$\partial_{y^q_{i,j}} f(x, y) = s_q(x, y) \partial_{y^q_{i,j}} H_q(x, y) - 2 \left(\Delta_q(x, y) + M_i\right) (x^q_{i,j} - y^q_{i,j}),$$

where the only difference comes from the derivative of $\|x^q - y^q\|^2$, which gives $\partial_{x^q_{i,j}} \|x^q - y^q\|^2 = -\partial_{y^q_{i,j}} \|x^q - y^q\|^2 = 2(x^q_{i,j} - y^q_{i,j})$. □

Then, we prove the following technical lemma that will help us compute an upper bound on the partial derivatives of H_q.

Lemma 6. *For every $(i,j,q) \in [n] \times [m] \times V$ we have*

$$\partial_{x_{i,j}^q} H_q(x,y) = -G_j\left(\frac{x_i^q + y_i^q}{2}\right) + R_{i,j,q}(x,y), \tag{11a}$$

$$\partial_{y_{i,j}^q} H_q(x,y) = G_j\left(\frac{x_i^q + y_i^q}{2}\right) + R_{i,j,q}(x,y), \tag{11b}$$

where $R_{i,j,q}(x,y) := \frac{1}{2}\sum_{k\in[m]}(y_{i,k}^q - x_{i,k}^q)\cdot \partial_{z_j} G_k\left(\frac{x_i^q+y_i^q}{2}\right)$ and Moreover, it holds that $\max_{i,j,q}|R_{i,j,q}(x,y)| \le e^{12}\|x_i^q - y_i^q\|_1$ and $\max_{i,j,q}\{|\partial_{x_{i,j}^q} H_q(x,y)|, |\partial_{y_{i,j}^q} H_q(x,y)|\} \le me^{13}$.

Proof. Recall that we have

$$H_q(x,y) = \sum_{\ell\in[n]} \left\langle G\left(\frac{x_\ell^q + y_\ell^q}{2}\right), y_\ell^q - x_\ell^q \right\rangle.$$

Since $x_{i,j}^q$ appears only in the $\ell = i$ summand we get

$$\begin{aligned}
\partial_{x_{i,j}^q} H_q(x,y) &= -G_j\left(\frac{x_i^q + y_i^q}{2}\right) + \sum_{k\in[m]} (y_{i,k}^q - x_{i,k}^q)\partial_{x_{i,j}^q} G_k\left(\frac{x_i^q + y_i^q}{2}\right) \\
&= -G_j\left(\frac{x_i^q + y_i^q}{2}\right) + \frac{1}{2}\sum_{k\in[m]} (y_{i,k}^q - x_{i,k}^q)\partial_{z_j} G_k\left(\frac{x_i^q + y_i^q}{2}\right) \\
&= -G_j\left(\frac{x_i^q + y_i^q}{2}\right) + R_{i,j,q}(x,y).
\end{aligned}$$

Similarly, we can conclude that

$$\partial_{y_{i,j}^q} H_q(x,y) = G_j\left(\frac{x_i^q + y_i^q}{2}\right) + R_{i,j,q}(x,y).$$

From the bounds of Theorem 5.1 it is clear that

$$\begin{aligned}
\max_{i,j,q}|R_{i,j,q}(x,y)| &\le \frac{1}{2}\max_{k\in[m]}\left|\partial_{z_j} G_k\left(\frac{x_i^q + y_i^q}{2}\right)\right| \cdot \|x_i^q - y_i^q\|_1 \\
&= \frac{1}{2}\max_{k\in[m]}\left|\partial_{z_j} F_k\left(\frac{x_i^q + y_i^q}{2}\right) - \mathbb{I}(j=k)\right| \cdot \|x_i^q - y_i^q\|_1 \\
&\le e^{12}\|x_i^q - y_i^q\|_1.
\end{aligned}$$

This easily implies also that

$$\begin{aligned}
\max_{i,j,q}\{|\partial_{x_{i,j}^q} H_q(x,y)|, |\partial_{y_{i,j}^q} H_q(x,y)|\} &\le \left|G_j\left(\frac{x_i^q + y_i^q}{2}\right)\right| + |R_{i,j,q}(x,y)| \\
&\le 1 + e^{12}\|x_i^q - y_i^q\|_1 \\
&\le me^{13},
\end{aligned}$$

concluding the proof. □

Now we show that when the regularizer M_i does not guess correctly the noise term $\Delta_q(x,y)$, the players are forced to play close to each other, namely:

Lemma 7. *Consider any solution (x,y) to* GDA-FIXEDPOINT *and any $q \in V$, $i \in [n]$, $j \in [m]$. If $M_i + \Delta_q(x,y) \neq 0$, it holds that:*

$$|x_{i,j}^q - y_{i,j}^q| \le me^{13}\frac{s_q(x,y)}{|M_i + \Delta_q(x,y)|} + \sqrt{\frac{\varepsilon}{|M_i + \Delta_q(x,y)|}}.$$

Proof. Consider the optimality conditions of the x-player w.r.t. $y^q_{i,j}$ and viceversa, i.e.

$$-\partial_{x^q_{i,j}} f(x,y)(y^q_{i,j} - x^q_{i,j}) \leq \varepsilon \tag{12a}$$

$$\partial_{y^q_{i,j}} f(x,y)(x^q_{i,j} - y^q_{i,j}) \leq \varepsilon. \tag{12b}$$

Define $A := M_i + \Delta_q(x,y)$. If $A > 0$, substituting the expression of $\partial_{x^q_{i,j}} f(x,y)$ of Equation (6a) in Equation (12a), we obtain

$$-\left(s_q(x,y)\partial_{x^q_{i,j}} H_q(x,y) + 2(M_i + \Delta_q(x,y))(x^q_{i,j} - y^q_{i,j})\right)(y^q_{i,j} - x^q_{i,j}) \leq \varepsilon.$$

On the other hand, if $A < 0$, substituting the expression of $\partial_{y^q_{i,j}} f(x,y)$ of Equation (6b) in Equation (12b) we obtain

$$\left(s_q(x,y)\partial_{y^q_{i,j}} H_q(x,y) - 2(M_i + \Delta_q(x,y))(x^q_{i,j} - y^q_{i,j})\right)(x^q_{i,j} - y^q_{i,j}) \leq \varepsilon.$$

In both cases, rearranging, implies the following quadratic expression in $|\kappa| := |x^q_{i,j} - y^q_{i,j}|$:

$$2\kappa^2|A| \leq \varepsilon + |\kappa| C s_q(x,y), \tag{13}$$

where by Lemma 6 we have $C := \max_{i,j,q}\{|\partial_{x^q_{i,j}} H_q(x,y)|, |\partial_{y^q_{i,j}} H_q(x,y)|\} \leq me^{13}$. Solving Equation (13) implies that

$$\begin{aligned}
|\kappa| &\leq \frac{C s_q(x,y) + \sqrt{C^2 s_q^2(x,y) + 8|A|\varepsilon}}{4|A|} \\
&\leq me^{13}\frac{s_q(x,y)}{|M_i + \Delta_q(x,y)|} + \sqrt{\frac{\varepsilon}{|M_i + \Delta_q(x,y)|}},
\end{aligned}$$

where in the last inequality we also used that $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for $a, b \geq 0$. □

The following lemma shows that, for components i, where the regularizer M_i does not guess correctly, we can upper bound the growth of the inverse errors geometrically.

Lemma 8. *Define* $B_\tau(x) := \{i \in [n] : |M_i + x| < \tau\}$*, then for all* $x \in \mathbb{R}$

$$\sum_{i \in [n] \setminus B_\tau(x)} \frac{1}{|M_i + x|} \leq \frac{2}{\tau} + \frac{2}{\delta}\log\left(1 + \frac{n\delta}{\tau}\right).$$

Proof. Define $a_i := M_i + x$ and observe that a_i is an arithmetic progression with difference δ. Let $B^+ := \{i \in [n] : a_i \geq \tau\}$ and $B^- := \{i \in [n] : a_i \leq -\tau\}$, and observe that $[n] \setminus B_\tau(x) = B^+ \cup B^-$, and that $B^+ \cap B^- = \emptyset$. Thus

$$\sum_{i \in [n] \setminus B_\tau(x)} \frac{1}{|M_i + x|} = \sum_{i \in B^+} \frac{1}{a_i} + \sum_{i \in B^-} \frac{1}{-a_i}.$$

Then, we can re-index the elements of $\{a_i\}_{i \in B^+}$ to a set $\{a_i^+\}_{i \in [m]}$ (with $m := |B^+|$), such that $\tau \leq a_1^+ < a_2^+ < \ldots < a_m^+$, and $a_k^+ \geq \tau + (k-1)\delta$, thus

$$\sum_{i \in B^+} \frac{1}{a_i} \leq \sum_{k=1}^{m} \frac{1}{a_k^+} \leq \sum_{k=1}^{m} \frac{1}{\tau + (k-1)\delta} \leq \sum_{k=0}^{n-1} \frac{1}{\tau + k\delta}.$$

Similarly, for $i \in B^-$ (now $m = |B^-|$), we can reorder the a_i's into a_i^- in reverse order $-\tau \geq a_1^- \geq a_2^- \geq \ldots \geq a_m^-$ and $a_k^- \leq -\tau - (k-1)\delta$ so that

$$\sum_{i \in B^-} \frac{1}{-a_i} \leq \sum_{k=1}^{m} \frac{1}{-a_i^-} \leq \sum_{k=0}^{n-1} \frac{1}{\tau + k\delta}.$$

Finally, we can combine the two to obtain

$$\sum_{i\in[n]\setminus B_\tau(x)} \frac{1}{|M_i+x|} \le 2\sum_{k=0}^{n-1} \frac{1}{\tau+k\delta} \le \frac{2}{\tau} + 2\int_0^{n-1} \frac{1}{\tau+\delta t}dt \le \frac{2}{\tau} + \frac{2}{\delta}\log\left(1+\frac{\delta n}{\tau}\right),$$

which is the desired result. □

Lemma 9. *Assume $\delta \le \tau$ and $\tau \le n\delta/8$. Then*

(i) For all $x \in \mathbb{R}$ we have $|B_\tau(x)| \le \lceil \frac{2\tau}{\delta} \rceil$;

(ii) and for all $x \in \mathbb{R}$ such that $|x| \le n\delta/4$ it holds that $|B_\tau(x)| \ge \frac{\tau}{\delta}$.

Proof. $\{M_i + x\}_{i\in[n]}$ is an arithmetic progression with difference δ. Thus, since $\delta \le 2\tau$, every open interval of size 2τ contains at most $\lceil \frac{2\tau}{\delta} \rceil$ points. This proves that $|B_\tau(x)| \le \lceil \frac{2\tau}{\delta} \rceil$ for every $x \in \mathbb{R}$.

For the second statement observe that, by the assumption that $\tau \le n\delta/8$ and $|x| \le n\delta/4$, the interval $[-x-\tau, -x+\tau]$ is fully contained in the range of the progression $(M_i)_{i\in[n]}$. Therefore, the number of points of the progression lying in the interval $[-x-\tau, -x+\tau]$ is at least $\lfloor \frac{2\tau}{\delta} \rfloor \ge \frac{\tau}{\delta}$. □

Now we can prove the main technical component that provides the bound of $|H_v(x,y)| = O(s_v(x,y)\text{poly}(m/\delta)\log(n))$, thus showing that the magnitude of $|H_v(x,y)|$ is sublinear in n. This will be used in showing that the noise term $|\Delta_q(x,y)|$ is small enough so that it falls strictly into the range of the regularizer.

Lemma 10. *For any solution (x,y) to* GDA-FIXEDPOINT *and any $v \in V$ we have:*

$$|H_v(x,y)| \le \frac{16}{\delta} m^2 e^{13} s_v(x,y)\log(2n\delta) + 2mn\sqrt{\varepsilon}.$$

Proof. Firs notice that our choice of parameters δ, n satisfy the assumptions of Lemma 9 with $\tau = 1$.

If $i \in B_1(\Delta_v(x,y))$, then $\|x_i^v - y_i^v\|_1 \le m$, and by Lemma 9 item *(i)*, $|B_1(\Delta_v(x,y))| \le 1 + 2/\delta \le 4/\delta$. Thus, when $i \in B_1(\Delta_v(x,y))$ we get:

$$\sum_{i\in B_1(\Delta_v(x,y))} \|x_i^v - y_i^v\|_1 \le \frac{4m}{\delta} \tag{14}$$

On other hand, if $i \notin B_1(\Delta_v(x,y))$, then $|M_i + \Delta_v(x,y)| \ge 1$, and we have:

$$\begin{aligned}\|x_i^v - y_i^v\|_1 &= \sum_{j\in[m]} |x_{i,j}^v - y_{i,j}^v| \\ &\le m^2 e^{13} \frac{s_v(x,y)}{|M_i + \Delta_v(x,y)|} + m\sqrt{\varepsilon}. && \text{(by Lemma 7)}\end{aligned}$$

Moreover, by Lemma 8 with $\tau = 1$, we can sum over these component to obtain:

$$\begin{aligned}\sum_{i\notin B_1(\Delta_v(x,y))} \|x_i^v - y_i^v\|_1 &\le m^2 e^{13} s_v(x,y) \sum_{i\notin B_1(\Delta_v(x,y))} \frac{1}{|M_i + \Delta_v(x,y)|} + mn\sqrt{\varepsilon} \\ &\le \frac{4}{\delta} m^2 e^{13} s_v(x,y)\log(2n\delta) + mn\sqrt{\varepsilon},\end{aligned} \tag{15}$$

where in the last inequality we used that $n\delta \geq 1$ and $1 = \tau \geq \delta$.Finally, recalling that $H_v(x,y) = \sum_{i\in[n]} \langle F(\frac{x_i^v+y_i^v}{2}) - \frac{x_i^v+y_i^v}{2}, y_i^v - x_i^v\rangle$ and combining Equation (14) and Equation (15) we get:

$$\begin{aligned}
|H_v(x,y)| &\leq \sum_{i\in[n]} \left\| F\left(\frac{x_i^v+y_i^v}{2}\right) - \frac{x_i^v+y_i^v}{2} \right\|_\infty \|x_i^v - y_i^v\|_1 \\
&\leq 2\sum_{i\in[n]} \|x_i^v - y_i^v\|_1 \\
&= 2\sum_{i\in B_1(\Delta_v(x,y))} \|x_i^v - y_i^v\|_1 + 2\sum_{i\notin B_1(\Delta_v(x,y))} \|x_i^v - y_i^v\|_1 \\
&\leq 8\frac{m}{\delta} + 2\sum_{i\notin B_1(\Delta_v(x,y))} \|x_i^v - y_i^v\|_1 && \text{(by Equation (14))} \\
&\leq 8\frac{m}{\delta} + \frac{8}{\delta}m^2 e^{13} s_v(x,y)\log(2n\delta) + 2mn\sqrt{\varepsilon} && \text{(by Equation (15))} \\
&\leq \frac{16}{\delta}m^2 e^{13} s_v(x,y)\log(2n\delta) + 2mn\sqrt{\varepsilon},
\end{aligned}$$

concluding the proof. □

E.1 Small Signal

Lemma 11. *Consider a solution (x,y) of* GDA-FIXEDPOINT *and any node q such that $s_q(x,y) = 0$, then $b(q) = 0$.*

Proof. From Lemma 9, we get that $|B_\delta(\Delta_q(x,y))| \leq 2$ and, trivially, also that for all $i \in B_\delta(\Delta_q(x,y))$ we have $\|x_i^q - y_i^q\|^2 \leq m$. On the other hand, if $i \notin B_\delta(\Delta_q(x,y))$ then we can invoke Lemma 7, with $s_q(x,y) = 0$, which implies that

$$|x_{i,j}^q - y_{i,j}^q| \leq \sqrt{\frac{\varepsilon}{|M_i + \Delta_q(x,y)|}} \leq \sqrt{\frac{\varepsilon}{\delta}},$$

since for the i's we are considering $|M_i + \Delta_q(x,y)| \geq \delta$. This implies that

$$\|x_i^q - y_i^q\|^2 = \sum_{j\in[m]} (x_{i,j}^q - y_{i,j}^q)^2 \leq m\varepsilon/\delta.$$

Summing the two cases we get

$$\|x^q - y^q\|^2 \leq 2m + mn\varepsilon/\delta \leq 3m,$$

since our choice of parameters guarantees that $\varepsilon \leq \delta/n$. The proof is concluded by the definition of the decoded solution b and the definition of $\mathcal{E}_q$ □

E.2 Large Signal

Lemma 12. *Let (x,y) be a solution to* GDA-FIXEDPOINT. *Then:*

$$|B_1(\Delta_q(x,y)| \geq 1/\delta.$$

Proof. This follows easily from Lemma 10. Indeed from the decomposition of Lemma 5:

$$\begin{aligned}
|\Delta_q(x,y)| &\leq \sum_{w\in \text{Out}(q)} |H_w(x,y)| \cdot |\phi'_{3m,3m+1}(\|x^q - y^q\|^2)| \cdot |\partial_{\mathcal{E}_q(x,y)} s_w(x,y)| \\
&\leq me^{14}\left(\frac{16}{\delta}m^2 e^{13}\log(2n\delta) + 2mn\sqrt{\varepsilon}\right) \leq \frac{n\delta}{4}.
\end{aligned}$$

where we used $|\text{Out}(q)| \leq m$, $\|\phi'_{3m,3m+1}\|_\infty \leq e^2$ and $\|s'_v\|_\infty \leq e^{12}$. The proof is then concluded by straightforward application of the second item of Lemma 9. □

Lemma 13. *Consider any $q \in V$ such that $s_q(x,y) = 1$, then, if there is no $i \in [n]$ such that $\frac{x_i^q + y_i^q}{2}$ is a ρ-approximate solution to* SMOOTHBROUWER*, then $b(q) = 1$.*

Proof. In this proof, for all $i \in [n], j \in [m]$ and $q \in V$, it is convenient to define $\xi_i^q := \frac{x_i^q + y_i^q}{2} \in [0,1]^m$ and $\xi_{i,j}^q := \frac{x_{i,j}^q + y_{i,j}^q}{2} \in [0,1]$. By Lemma 12, there are at least $1/\delta$ many $i \in B_1(\Delta_q(x,y))$. We now show that for such i's we have $\|x_i^q - y_i^q\|_1 \geq \frac{\rho^2}{10e^{13}}$. By contradiction, assume that $\|x_i^q - y_i^q\|_1 < \frac{\rho^2}{10e^{13}}$. Lemma 6 let us write for all $j \in [m]$:

$$\partial_{x_{i,j}^q} H_q(x,y) = -(F_j(\xi_i^q) - \xi_{i,j}^q) + R_{i,j,q}(x,y), \tag{16}$$

and,

$$\partial_{y_{i,j}^q} H_q(x,y) = F_j(\xi_i^q) - \xi_{i,j}^q + R_{i,j,q}(x,y), \tag{17}$$

where $|R_{i,j,q}(x,y)| \leq \frac{e^{12}}{2}\|x_i^q - y_i^q\|_1 \leq \frac{\rho^2}{10}$, where the first inequality is by Lemma 6, while the second inequality holds by contradiction' assumption. Then consider, any $j \in [m]$, and the optimality conditions of the first player together with Equation (16):

$$\left(F_j(\xi_i^q) - \xi_{i,j}^q - R_{i,j,q} - 2(M_i + \Delta_q(x,y))(x_{i,j}^q - y_{i,j}^q)\right)(z - x_{i,j}^q) \leq \varepsilon,$$

for all $z \in [0,1]$ which implies that

$$\begin{aligned}
(F_j(\xi_i^q) - \xi_{i,j}^q)(z - x_{i,j}^q) &\leq \varepsilon + (R_{i,j,q} + 2(M_i + \Delta_q(x,y))(x_{i,j}^q - y_{i,j}^q))(z - x_{i,j}^q) \\
&\leq \varepsilon + \frac{\rho^2}{10} + 2|M_i + \Delta_q(x,y)| \cdot |x_{i,j}^q - y_{i,j}^q| \qquad (|z - x_{i,j}^q| \leq 1) \\
&\leq \varepsilon + \frac{\rho^2}{10} + 2\frac{\rho^2}{10e^{13}} \leq \rho^2. \qquad (i \in B_1(\Delta_q(x,y)))
\end{aligned}$$

which can also be written as

$$(F_j(\xi_i^q) - \xi_{i,j}^q)(F_j(\xi_i^q) - x_{i,j}^q) \leq \rho^2, \tag{18}$$

after specializing it for $z = F_j(\xi_i^q) \in [0,1]$.

Similarly, considering the optimality condition of the second player together with Equation (17), we get for all $z \in [0,1]$:

$$\left(F_j(\xi_i^q) - \xi_{i,j}^q + R_{i,j,q} - 2(M_i + \Delta_q(x,y))(x_{i,j}^q - y_{i,j}^q)\right)(z - y_{i,j}^q) \leq \varepsilon$$

which implies that

$$\begin{aligned}
\left(F_j(\xi_i^q) - \xi_{i,j}^q\right)(z - y_{i,j}^q) &\leq \varepsilon + (R_{i,j,q} - 2(M_i + \Delta_q(x,y))(x_{i,j}^q - y_{i,j}^q))(y_{i,j}^q - z) \\
&\leq \varepsilon + \frac{\rho^2}{10} + 2|x_{i,j}^q - y_{i,j}^q| \leq \varepsilon + \frac{\rho^2}{10} + 2\frac{\rho^2}{10e^{13}} \leq \rho^2,
\end{aligned}$$

which is

$$(F_j(\xi_i^q) - \xi_{i,j}^q)(F_j(\xi_i^q) - y_{i,j}^q) \leq \rho^2, \tag{19}$$

after specializing it for $z = F_j(\xi_i^q)$.

By averaging Equation (18) with Equation (19) we get:

$$(F_j(\xi_i^q) - \xi_{i,j}^q)^2 \leq \rho^2 \implies |F_j(\xi_i^q) - \xi_{i,j}^q| \leq \rho \quad \forall j \in [m].$$

which implies that $\xi_i^q = \frac{x_i^q + y_i^q}{2}$ is a solution to SMOOTHBROUWER, which is a contradiction. Thus there are at least $1/\delta$ many i such that $\|x_i^q - y_i^q\|_1 \geq \frac{\rho^2}{10e^{13}}$, and

$$\begin{aligned}
\|x^q - y^q\|^2 &\geq \sum_{i \in B_1(\Delta_q(x,y))} \|x_i^q - y_i^q\|^2 \\
&\geq \frac{1}{m} \sum_{i \in B_1(\Delta_q(x,y))} \|x_i^q - y_i^q\|_1^2 \\
&\geq \frac{\rho^4}{100e^{26}\delta m} = 4m \geq 3m + 1,
\end{aligned}$$

concluding the proof, since by definition $b(q) = \mathcal{E}_q(\|x^q - y^q\|^2) = 1$. □

E.3 Proof of Lemma 3

Lemma 3 (Dichotomy)**.** *Let (x, y) be a solution to* GDA-FIXEDPOINT*. Then either there exists $i \in [n], v \in V$ such that $(x_i^v + y_i^v)/2$ is a solution to* SMOOTHBROUWER*, or $b : V \to \{0, 1, \perp\}$, as defined in Equation (5), is a solution to* ORACLEPURECIRCUIT*.*

Proof. By Lemma 13 we have that either there is a $(i, q) \in [n] \times V$ such that $(x_i^q + y_i^q)/2$ is a solution to SMOOTHBROUWER, or that for all $q \in V$ such that $s_q(x, y) = 1$ then $b(q) = 1$. On the other hand, Lemma 11 always guarantees that $s_q(x, y) = 0$ implies that $b(q) = 0$. Thus, we can assume that $s_v(x, y) = 1 \implies b(v) = 1$ and $s_v(x, y) = 0 \implies b(v) = 0$ for all $v \in V$. Now, we show that in this case, the assignment b satisfies all the gates:

- $(u, v, w) \in \mathcal{G}_{\text{NOR}}$. Assume that $\mathcal{E}_u(x, y) = \mathcal{E}_v(x, y) = b(u) = b(v) = 0$, then $s_w(x, y) = g(\mathcal{E}_u(x, y) + \mathcal{E}_v(x, y)) = g(0) = 1$. By assumption, this implies that also $b(w) = 1$ and thus the gate is satisfied. On the other hand, if either $b(u) = 1$ or $b(v) = 1$, then $\mathcal{E}_u(x, y) + \mathcal{E}_v(x, y) \geq 1$ and thus $s_w(x, y) = 0$. By assumption, this implies that $b(w) = 0$, and thus NOR gates are satisfied.
- $(u, v, w) \in \mathcal{G}_{\text{PURIFY}}$. If $b(u) = 0$ then $\mathcal{E}_u(x, y) = 0$ and thus $s_v(x, y) = s_w(x, y) = \ell(\mathcal{E}_u(x, y) \pm 1/4) = 0$, which by assumption implies that $b(v) = b(w) = 0$. Similarly, if $b(u) = 1$ then $\mathcal{E}_u(x, y) = 1$ and $s_v(x, y) = s_w(x, y) = \ell(\mathcal{E}_u(x, y) \pm 1/4) = 1$, which by assumption implies that $b(v) = b(w) = 1$. On the other hand, if $b(u) = \perp$, then $\mathcal{E}_u(x, y) \in (0, 1)$ but either $\mathcal{E}_u(x, y) - 1/4 \leq 5/12$ or $\mathcal{E}_u(x, y) + 1/4 \geq 7/12$, which implies that either $s_v(x, y) = 0$ or $s_w(x, y) = 1$, thus satisfying the gate's constraints.
- $(u_1, \ldots, u_N, v) \in \mathcal{G}_{\text{ORACLE}}$. Assume that $(b(u_1), \ldots, b(u_N)) \in \{0, 1\}^N$. Then $\mathcal{E}_{u_i}(x, y) = b(u_i)$ for all $i \in [N]$, and $s_v(x, y) = \mathcal{H}((b(u_i))_{i \in [N]}, \mathcal{L})$. By Lemma 2, we get that $s_v(x, y) = \mathcal{L}(b(u_1), \ldots, b(u_N)) \in \{0, 1\}$, which trivially shows that the gate is satisfied. By assumption we have that $b(v) = \mathcal{L}(b(u_1), \ldots, b(u_N))$ as desired.

This proves that b is a valid assignment to ORACLEPURECIRCUIT. □

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The paper's abstract correctly reflects the main results.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: In the final section of the paper, we discussed potential improvements to our results.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: All the proofs of our theoretical results are contained in either the main body of the supplementary material.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [N/A]

   Justification: This work does not include an experimental evaluation.

   Guidelines:

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

5. **Open access to data and code**

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

Answer: [N/A]

Justification: This work does not include an experimental evaluation.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [N/A]

   Justification: This work does not include an experimental evaluation.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [N/A]

   Justification: This work does not include an experimental evaluation.

   Guidelines:

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

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

8. **Experiments compute resources**

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

   Answer: [N/A]

   Justification: This work does not include an experimental evaluation.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This research complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: Our work shows a computational lower bound and has neither direct negative nor positive societal consequences.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: There are no such risks associated with this research.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [N/A]

    Justification: This work does not rely on existing assets.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: This work is theoretical and does not introduce new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

    Justification: This work does not involve research with human subjects.

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [N/A]

    Justification: This work does not involve research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: This work has not relied on LLM in any non-trivial way.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.