Quantum Advantage with Sign-Free Proofs: An Oracle Separation between StoqMA and MA

Yassine Hamoudi* Yvan Le Borgne† Shrinidhi Teganahally Sridhara‡

Université de Bordeaux, CNRS, LaBRI, France

Last update: November 5, 2026

Abstract

Several landmark oracle results, including [Wat00; GNT25; BHNZ26], establish strict separations along the chain $\mathsf{MA} \subseteq \mathsf{QCMA} \subseteq \mathsf{QMA}$, showing that quantum verification and quantum witnesses can each provide additional power over classical proof systems. These separations, however, use substantial coherent quantum computation, often through the quantum Fourier transform. We prove that a minimal quantum resource already suffices to outperform every classical verifier. Namely, we show the separation

$$\mathsf{MA} \subsetneq \mathsf{StoqMA}$$

relative to a classical oracle. The class StoqMA, introduced by Bravyi, Bessen, and Terhal [BBT06], restricts the verifier to classical reversible computation, followed by a single Hadamard-basis measurement. Its power is closely tied to the complexity of *stoquastic* Hamiltonians, a sign-problem-free and seemingly more tractable class of quantum Hamiltonians.

Our separating problem is to determine whether a sparse high-girth graph, given via an adjacency-list oracle, has a large or small Perron–Frobenius eigenvalue. This amounts to a ground-energy estimation task for a related stoquastic Hamiltonian. Our construction builds on a line of work on graph exploration problems with exponential quantum speedups, from welded trees [CCD+03] to recent constructions [GHV21; HBS26].

1 Introduction

Understanding generalisations of the complexity class NP is key to identifying the right quantum analogue of classical proof systems (see [Gha24]). One natural classical generalisation is MA, where the verifier is allowed to use randomness and fail with bounded probability. Among the many quantum generalisations, one is StoqMA, introduced by Bravyi, Bessen and Terhal [BBT06], in which the verifier is a very restricted quantum computer. It may use $|0\rangle$ and $|+\rangle$ ancillas, apply polynomially many classical reversible gates, and finish with a single Hadamard-basis measurement. Estimating the ground-state energy of stoquastic Hamiltonians is StoqMA-complete when they are local [BBT06], or even just sparse [GR26]. Here, stoquastic (or sign-problem-free) Hamiltonians are those whose off-diagonal entries are non-positive in the computational basis, allowing their ground states to be chosen with non-negative amplitudes. Despite its simplicity, this verifier model captures an important regime of Hamiltonian complexity. For example, Cubitt and Montanaro [CM16] give an exhaustive classification of 2-local Hamiltonian problems into four complexity classes: P, NP-complete, StoqMA-complete and QMA-complete. While StoqMA is contained in QMA, it has also been shown to sit in between MA and AM, the two randomized

*ys.hamoudi@gmail.com

†borgne@labri.fr

‡shrinidhi.teganahally-sridhara@u-bordeaux.fr

extensions of NP [BDOT08; BBT06]. The question of whether quantum proofs can be strictly more powerful than classical ones is central to quantum verification. StoqMA provides a natural setting for this question: does its minimal quantum power already suffice to outperform MA?

At least three reasons suggest that MA might be equal to StoqMA. First, its gate set is nearly classical. Second, classical methods such as quantum Monte Carlo algorithms appear to work efficiently on ground-state problems for stoquastic Hamiltonians [BT10; Bra15]. Third, [AGL25] showed that if StoqMA admits error reduction, as most other proof-system classes do, then MA = StoqMA. In this work, we give evidence against this belief by showing that there exists a classical oracle relative to which MA $\subsetneq$ StoqMA, so that a very restricted quantum verifier can solve a problem that no classical verifier with a classical witness can solve efficiently.

Theorem 1.1 (Oracle separation of StoqMA and MA)**.** *There exists a classical oracle $\mathcal{O}$ such that* $\mathsf{StoqMA}^{\mathcal{O}} \neq \mathsf{MA}^{\mathcal{O}}$.

Our separating problem is to distinguish whether a sparse, high-girth graph on 2^m vertices has a large or small top eigenvalue $\lambda_{\max}(A)$ (its Perron–Frobenius eigenvalue) of its adjacency matrix A. The graph is given through its adjacency-list oracle after a uniformly random relabeling of its vertices. Since A is entrywise non-negative, $H = -A$ is a stoquastic Hamiltonian with ground energy $-\lambda_{\max}(A)$. The problem is therefore a ground-energy distinction problem for a stoquastic Hamiltonian, which is exactly what a StoqMA verifier is built to handle.

We construct such an oracle by building on and extending prior work by Gilyén, Hastings, and Vazirani [GHV21], as well as our own prior work [HBS26]. The two graph families have polynomially separated top eigenvalues. The YES instance is a single graph G and for every small seed set U, we construct a NO-instance $F(U)$ that is indistinguishable from G when exploration starts from U. A StoqMA verifier receives the top eigenvector itself as a quantum witness and verifies its energy using a simple Hadamard test, a more elementary procedure than quantum phase estimation. An MA verifier, in contrast, is essentially limited to local graph exploration and cannot detect the global spectral difference using a classical witness and poly(m) queries.

Oracle separations are a central tool for studying the power of quantum verification and quantum proofs. The choice of oracle model is crucial, as illustrated by the separations between QCMA and QMA (where a quantum verifier receives a classical or quantum proof). Aaronson and Kuperberg [AK07] first established this separation relative to *unitary oracles*. Such oracles are particularly powerful because they provide black-box access to an arbitrary unitary transformation U. Fefferman and Kimmel [FK18] later obtained the separation using randomized *in-place permutation oracles*, whose queries permute computational-basis states as $|x\rangle \mapsto |\pi(x)\rangle$. After a long line of refinements, two recent works [BHNZ26; BHV26] finally established the separation in the standard *classical-oracle* model [Deu85], in which a function f is accessed through the reversible XOR map $|x, y\rangle \mapsto |x, y \oplus f(x)\rangle$. The choice of oracle model is particularly important for StoqMA, whose verifier is restricted to classical reversible gates. Allowing arbitrary unitary oracles would violate this restriction. Even phase oracles (obtained from classical oracles by phase kickback) introduce signs. We therefore work with classical XOR oracles, which preserve the sign-free structure of the verifier.

1.1 Relation to prior work

Expander-mixing based separations. Earlier partial progress on the QMA vs. QCMA question came from subspace and subset verification [AK07; FK18], Yamakawa–Zhandry-based oracles [LLPY24; BK24], and expander-mixing problems over graph oracles [Lut11; NN24; BFM23; LMY25]. These results either require special oracle access or rely on substantial coherent quantum computation, often through the quantum Fourier transform. The expander-mixing candidates are closest in spirit to our graph problem, but a fundamental difference poses an obstacle to verification in StoqMA. They use regular graphs, hence the top eigenvalue is

trivial to compute. The difficulty lies instead in testing the second-largest eigenvalue of the adjacency matrix A (equivalently the second-lowest eigenvalue of $-A$). Our problem instead concerns the top eigenvalue itself (whose negative is the ground-state energy of $-A$), precisely the quantity a StoqMA verifier is suited to certify. This reflects a broader distinction for stoquastic Hamiltonians: ground-energy estimation is StoqMA-complete, whereas highest-energy estimation is QMA-complete [JGL10].

Other oracle separations. Several other oracle separations have appeared recently, including QMA_1 vs. QCMA [MPR26] and QMA(2) vs. QMA [BGH+26; GMT26]. Here, the subscript 1 denotes perfect completeness, while QMA(2) allows two unentangled quantum proofs. The result of [MPR26] also separates StoqMA_1 from QCMA, as we sketch in Appendix A, but uses an in-place oracle. The construction in [GMT26] is also close to our setting: its hard instances are signed graphs accessed through a classical adjacency-list oracle, as in the earlier constructions of [NN24; LMY25]. However, its YES and NO instances are both frustration-free. Thus, unlike in our problem, the two cases are distinguished not by the lowest energy over all states, but by the lowest energy attainable by product states.

Adiabatic and guided stoquastic problems. Our work is most directly related to two earlier oracle separations for stoquastic problems: that of Gilyén, Hastings, and Vazirani [GHV21] in the adiabatic setting, and our earlier work [HBS26] in the dequantized guided ground-state setting. The present setting differs in how auxiliary information is supplied. Here it is an untrusted proof, whereas [HBS26] assumes a trusted guide as part of the input, and [GHV21] uses no witness. The graph constructions in all three works inherit ideas from welded trees [CCD+03] but require substantial changes for their respective settings. Although the decision problem studied in [HBS26] belongs to StoqMA, we show in Appendix B that it also belongs to NP, requiring a different construction for an oracle separation. Our YES instances use the same graph construction as [HBS26], combining a high-degree, high-girth base graph with the self-similar decorations of [GHV21]. Here, however, the base graph need not be an expander. Our new NO instances build on a notion of closure that has appeared in earlier work (e.g., [CMM09; BCK15]) and that we study further here.

1.2 Proof overview

We outline the proof of Theorem 1.1, which gives an oracle separation between StoqMA and MA.

Distinguishing problem and graph families. Our separation is based on estimating the top eigenvalue of a sparse graph given oracle access to its adjacency list. We call this the Graph Perron-Eigenvalue Estimation (GPE) problem. We construct two families of graphs on exponentially many vertices ($2^{\Theta(n^3 \log^2 n)}$) whose top eigenvalues are polynomially separated. To make the problem hard for classical algorithms, the vertex names are hidden by a uniformly random relabeling, so an algorithm can learn about the graph only by exploring it *locally* from vertices it already knows.

The YES *family* consists, for each n, of all labelings of a single graph G (Definition 3.7 and Figure 1). To construct G, we start from an n-regular *base graph* of girth $g = \mathrm{poly}(n)$ (Lemma 3.5), similar to that of [HBS26], and attach copies of *self-similar trees* (Definition 3.6) to every vertex of the base graph, similar to those studied in [Has21; GHV21]. Because of its girth, the base graph looks like a tree locally, and the shape of the attached trees mimics this local structure (Proposition 4.1), so that locally the trees are hard to tell apart from the base graph. The resulting graph G has maximum degree less than $2n$, and since it contains the n-regular base graph, whose top eigenvalue is n, we have $\lambda_{\max}(G) \geq n$.

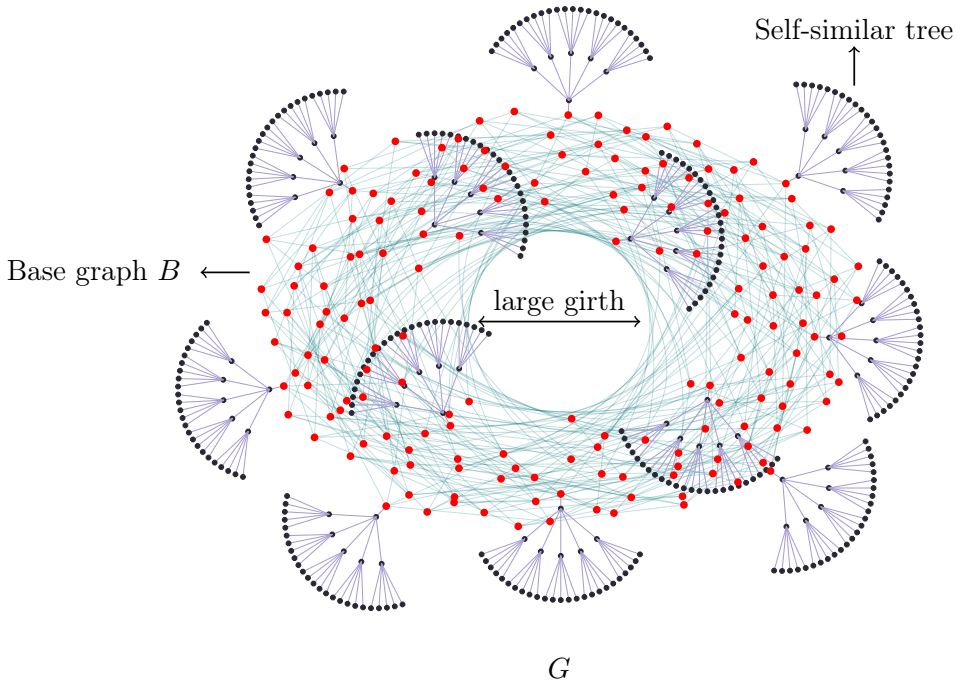


Figure 1: Pictorial representation of the YES graph G. The high-girth, regular base graph B is shown with red vertices and green edges, with the self-similar trees attached to its vertices shown in purple; for simplicity, we show them on only some of the vertices, with only one copy per vertex, and depict each self-similar tree as a simple perfect tree. The actual construction is more complicated: each tree is obtained by attaching shorter-depth trees to a perfect tree (see Definitions 3.6 and 3.7).

The NO *family* is designed to have a small top eigenvalue while looking like G to an algorithm exploring locally, even when given a small set of vertices as starting points (we later derive such a set from the witness via a bit-fixing technique). Concretely, for each *seed set* U of at most $2^{\sqrt{n}}$ vertices, a graph $F(U)$ is obtained by truncating G around U (Definition 3.10 and Figure 2). The truncation must not be recognisable by local exploration starting from U, so we keep everything within a large distance of U and truncate only far away from it. Since the same trees hang off every base vertex, only the base graph distinguishes one place of G from another, so we truncate by distance in the base graph (a tree vertex counts as the base vertex it hangs off).

We build $F(U)$ as follows. We start from U and add all vertices on short paths between its vertices, repeating until nothing new is added; the result, a set of base vertices, is the *closure* of U (Definition 2.7). We then keep everything within a large (polynomial) distance of the closure and delete the rest. By the high girth, for every edge going out of the closure, all vertices reachable from the closure via this edge induce a subgraph that is isomorphic to a particular self-similar tree (Proposition 4.1), and these subgraphs are pairwise disjoint. The truncation is chosen so that the outermost retained base vertices, called the *boundary*, correspond exactly to a particular set of leaves in these trees. These are the only places where $F(U)$ stops while G continues.

These graphs inherit the high girth g of G. Moreover, each $F(U)$ is 2-*degenerate* (Lemma 3.11): it contains no subgraph of minimum degree at least 3 (equivalently, it can be reduced to the empty graph by repeatedly removing a vertex of degree at most 2). Indeed, by the Moore bound (Lemma 2.6), any subgraph of the closure with minimum degree at least 3 would have at least $2^{\Omega(g)}$ vertices. However, the closure is only polynomially larger than its seed set, containing at most $\mathrm{poly}(n) \cdot |U|$ vertices (Lemma 2.8). For our choice of g and $|U| \leq 2^{\sqrt{n}}$, this is too small to contain such a subgraph. The rest of $F(U)$ consists of trees attached to the closure, which

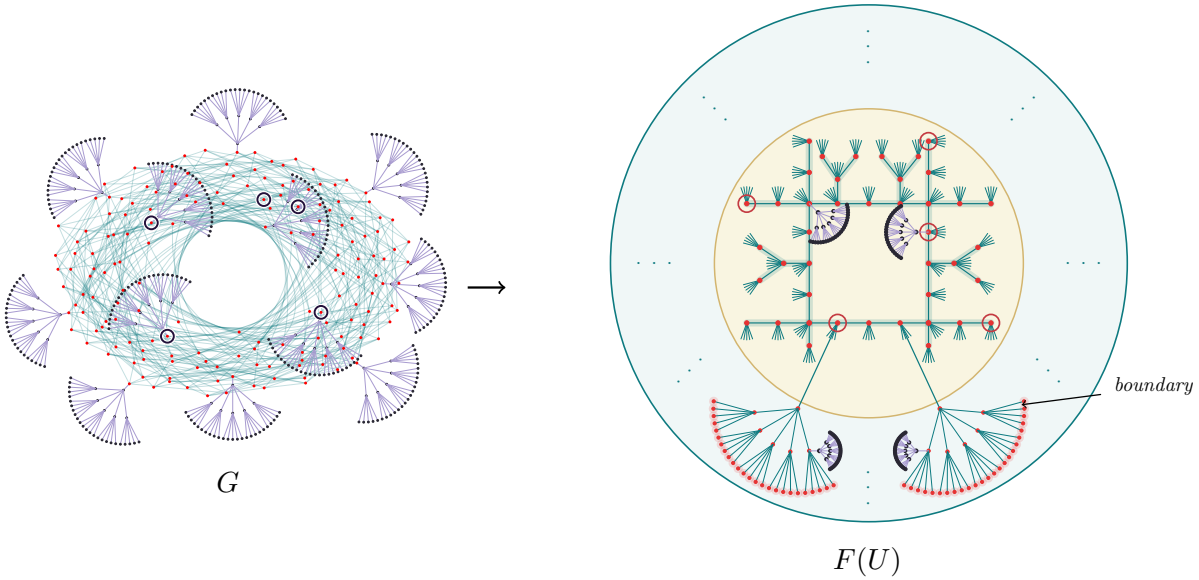


Figure 2: Pictorial representation of the NO graph $F(U)$, obtained as a subgraph of G using U. The vertices of U are circled in both graphs. In $F(U)$, on the right, the yellow region marks the closure of U, and the short paths added between vertices to form the closure are highlighted. For two vertices, the subgraph reachable via an edge leaving the closure is shown together with its attached purple trees. These subgraphs are pairwise disjoint, and each is isomorphic to the same fixed self-similar tree; the *boundary vertices* (highlighted in pink) correspond exactly to a particular set of its leaves. As in the previous figure, self-similar trees are shown attached to only a few vertices, and only one copy is drawn (see Definition 3.10 and Proposition 4.1 for a detailed description).

preserves 2-degeneracy. In contrast, G has degeneracy exactly n since the base graph is n-regular. A central consequence of being sparse and 2-degenerate is that the spectral norm is small, namely $\lambda_{\max}(F(U)) \leq 4\sqrt{n}$ (Proposition 3.12). This gives the spectral separation from the YES instance.

StoqMA protocol. The GPE problem is easily shown to lie in StoqMA because the negative of the adjacency matrix $-A$ is a sparse stoquastic Hamiltonian [GR26]. In Proposition 3.4, we give a direct proof using the Hadamard test. For any graph of maximum degree at most d, our verifier uses $\mathrm{poly}(n)$ adjacency-list queries and accepts any witness $|\psi\rangle$ with probability $1/2 + \langle\psi|A|\psi\rangle/(2d)$. It therefore accepts a top eigenvector of G (YES case) with probability at least $3/4$, while accepting every witness for $F(U)$ (NO case) with probability at most $1/2 + 1/\sqrt{n}$.

Classical hardness when the witness is a set of vertices. As a first step toward showing that GPE is not in MA, we consider the simplest kind of witness: one that just reveals the names of a set U of at most $2^{\sqrt{n}}$ vertices, so that the algorithm knows where U is, while all other vertices stay hidden behind random names. We show that this does not help: no classical algorithm making polynomially many queries can distinguish G from the NO graph $F(U)$ built around U with non-negligible advantage, even though it knows what U is.

An algorithm can tell G from $F(U)$ only by finding a place where they differ. Since all vertices outside U are hidden behind random names, a classical algorithm cannot jump to a vertex it has not reached: except with negligible probability, it only explores the part of the graph connected to U by following edges (this is standard [CCD+03]; see Lemma 4.2). Throughout such an exploration, the two graphs look the same until the algorithm reaches a *boundary* vertex, which is a leaf of $F(U)$ but has further neighbors in G (Proposition 4.1). Hence the algorithm

must find a path from some vertex in U to the boundary.

Why is this hard? Recall that by construction, the part of $F(U)$ reachable after taking an edge leaving the closure of U is a copy of the same self-similar tree, and the boundary vertices correspond to a particular set of leaves of this tree (Proposition 4.1 and Figure 2). So, starting from a vertex of the closure, the algorithm has to walk down the self-similar tree attached to it, all the way to one of these particular leaves, without being able to tell which way leads there. The self-similar trees are designed to hide these leaves from any classical algorithm [GHV21; HBS26]: a polynomial-query algorithm finds such a path with probability at most $2^{-2n \log n}$ (Lemma 4.3).

So far we considered a path starting from one fixed vertex of the closure, that is, a descent down a single self-similar tree, and showed that finding it is hard. To reach the boundary, the algorithm needs such a path from *some* vertex of the closure. Since the trees hanging off distinct closure vertices are disjoint and independently labelled, exploring one reveals nothing about the others, so Lemma 4.3 bounds the success probability in each of them by $2^{-2n \log n}$ regardless of the algorithm's history; a union bound over all of them then applies. The closure is relatively small: it has at most $\mathrm{poly}(n) \cdot |U|$ vertices (Lemma 2.8). Altogether, the probability of reaching the boundary is thus at most $q \cdot \mathrm{poly}(n) \cdot |U| \cdot 2^{-2n \log n}$, where q is the number of queries, which stays negligible even for $|U|$ as large as $2^{\sqrt{n}}$.

From vertices to arbitrary witnesses. An MA witness need not be a set of vertices: it is an arbitrary bit-string of polynomial length, and it may depend on the entire graph, including how its vertices are named. We would like to show that such a string cannot do more than reveal the location of a small set of vertices. If this were true, the proof would conclude as follows. Take a YES instance G and a witness that makes the verifier accept it. By the claim, this witness is as good as revealing the location of some small set U. But an algorithm that only knows where U is cannot tell G from $F(U)$ by the previous paragraph, so the verifier would also accept the NO instance $F(U)$ with the same witness, contradicting soundness.

To make this precise, we use a bit-fixing technique for random permutations [CDG18] (see Lemma 5.1). Informally, it says that polynomially many bits of information about a uniformly random relabeling can be replaced, up to a small error, by revealing the names of a few vertices while keeping all other names uniformly random. (The lemma speaks about queries to the relabeling rather than to the graph, which only costs a polynomial factor in the number of queries.) We apply it with up to $2^{\sqrt{n}}$ revealed vertices: this is enough to make the error $\mathrm{poly}(n)/2^{\sqrt{n}}$ negligible, and it is still within the size of seed sets handled before. The witness then keeps the verifier's acceptance probability on G at least $2/3 - 2^{-\sqrt{n}/2}$, and once the witness is fixed the argument above goes through (Proposition 5.3).

Finally, a standard diagonalisation argument [Zha25; NN24], adapted in Theorem 1.1, lifts this to a relativized separation. We build a single oracle $\mathcal{O}$ whose n-th block is the adjacency-list oracle of a GPE instance on $2^{\Theta(n^3 \log^2 n)}$ vertices from either the YES or the NO family, chosen so that the i-th candidate MA verifier errs on its block. Since the StoqMA protocol above solves GPE for every such oracle, this gives a relativized separation of StoqMA and MA.

2 Preliminaries

2.1 Complexity classes

A promise problem $\mathcal{L} = (\mathcal{L}_{\mathrm{YES}}, \mathcal{L}_{\mathrm{NO}})$ consists of disjoint subsets $\mathcal{L}_{\mathrm{YES}}, \mathcal{L}_{\mathrm{NO}} \subseteq \{0,1\}^*$. We now give the definitions of the two complexity classes considered in this paper.

Definition 2.1 (MA AND $\mathsf{MA}^{\mathcal{O}}$)**.** A promise problem $\mathcal{L} = (\mathcal{L}_{\mathrm{YES}}, \mathcal{L}_{\mathrm{NO}})$ is in the complexity class MA if and only if there exists a polynomial $p(\cdot)$ and a probabilistic polynomial-time verifier $\mathcal{V}(x, w)$ where $\mathcal{V}$ takes as input a bit string $x \in \{0,1\}^*$ and a witness $w \in \{0,1\}^{p(|x|)}$ and outputs either 0 (reject) or 1 (accept) such that:

Completeness: If $x \in \mathcal{L}_{\text{YES}}$, then there exists w such that $\Pr[\mathcal{V}(x,w)=1] \geq 2/3$.

Soundness: If $x \in \mathcal{L}_{\text{NO}}$, then for any w, $\Pr[\mathcal{V}(x,w)=1] \leq 1/3$.

The oracle version $\mathsf{MA}^{\mathcal{O}}$ is defined in the same way, for an oracle $\mathcal{O}:\{0,1\}^* \to \{0,1\}$, except that $\mathcal{L}^{\mathcal{O}}_{\text{YES}}$ and $\mathcal{L}^{\mathcal{O}}_{\text{NO}}$ may depend on $\mathcal{O}$ and the verifier $\mathcal{V}^{\mathcal{O}}(x,w)$ has oracle access to $\mathcal{O}$, which it may query at most polynomially many times.

The completeness and soundness parameters in the definition of MA can be chosen arbitrarily, provided that they are inverse-polynomially separated from each other and from 0 and 1. This is unlike StoqMA, for which gap amplification remains an open problem. We recall the definition introduced in [BBT06].

Definition 2.2 (StoqMA, [BBT06])**.** A *stoquastic verifier* is a tuple $\mathcal{V}=(m,m_w,m_0,m_+,U)$, where m is the number of input bits, m_w is the number of input witness qubits, m_0 is the number of input ancillas in the state $|0\rangle$, m_+ is the number of input ancillas in the state $|+\rangle$, and U is a quantum circuit on $m+m_w+m_0+m_+$ qubits with X, CNOT, and Toffoli gates. The acceptance probability of a stoquastic verifier $\mathcal{V}$ on input string $x \in \{0,1\}^m$ and witness state $|\psi\rangle \in (\mathbb{C}^2)^{\otimes m_w}$ is defined as $\Pr[\mathcal{V}(x,|\psi\rangle)=1] = \langle \psi_{\text{inp}} \,|\, U^\dagger \Pi_{\text{out}} U \,|\, \psi_{\text{inp}}\rangle$, where $|\psi_{\text{inp}}\rangle = |x\rangle\otimes|\psi\rangle\otimes|0\rangle^{\otimes m_0}\otimes|+\rangle^{\otimes m_+}$ is the initial state and $\Pi_{\text{out}} = |+\rangle\langle+|_1 \otimes \mathbb{I}_{\text{else}}$ projects the first qubit onto the state $|+\rangle$.

A promise problem $\mathcal{L}=(\mathcal{L}_{\text{YES}},\mathcal{L}_{\text{NO}})$ belongs to $\mathsf{StoqMA}(a,b)$ if there exists a polynomial-time uniform family of stoquastic verifiers, such that for any input $x \in \mathcal{L}_{\text{YES}} \cup \mathcal{L}_{\text{NO}}$ of m bits, the corresponding stoquastic verifier $\mathcal{V}$ uses at most $p_1(m)$ qubits and $p_2(m)$ gates where both p_1 and p_2 are polynomials and obeys the following:

Completeness: If $x \in \mathcal{L}_{\text{YES}}$, then there is some $|\psi\rangle$ such that $\Pr[\mathcal{V}(x,|\psi\rangle)=1] \geq a(m)$.

Soundness: If $x \in \mathcal{L}_{\text{NO}}$, then for any $|\psi\rangle$, $\Pr[\mathcal{V}(x,|\psi\rangle)=1] \leq b(m)$.

We define StoqMA as the union of $\mathsf{StoqMA}(a,b)$ over all polynomial-time computable threshold functions a,b satisfying $1/2 \leq b(m) < a(m) \leq 1$ and $a(m)-b(m) \geq 1/\operatorname{poly}(m)$.

To define the oracle version $\mathsf{StoqMA}^{\mathcal{O}}(a,b)$, fix an oracle $\mathcal{O}:\{0,1\}^* \to \{0,1\}$. The sets $\mathcal{L}^{\mathcal{O}}_{\text{YES}}$ and $\mathcal{L}^{\mathcal{O}}_{\text{NO}}$ may depend on $\mathcal{O}$ and the verifier $\mathcal{V}^{\mathcal{O}}(x,|\psi\rangle)$ has oracle access to $\mathcal{O}$, which it may query at most polynomially many times via query gates $|y\rangle|c\rangle \mapsto |y\rangle|c\oplus\mathcal{O}(y)\rangle$.

Both definitions extend to an oracle $\mathcal{O}:\{0,1\}^* \to \{0,1\}^m$ with longer outputs, such as the adjacency-list oracle we define later in Definition 3.1: replace each query by m queries $|y,j\rangle|c\rangle \mapsto |y,j\rangle|c\oplus\mathcal{O}(y)_j\rangle$ for $j\in[m]$, which increases the query count by only a polynomial factor. All our results are stated and proved with respect to this multi-bit oracle model.

XOR oracle. We stress that the query gate must be the XOR oracle above and not the phase-flip oracle $|y\rangle|c\rangle \mapsto (-1)^{c\cdot\mathcal{O}(y)}|y\rangle|c\rangle$. For general quantum verifiers the two are equivalent via Hadamard gates on the target qubit, but these are not available to a StoqMA verifier.

2.2 Graph-theoretic notions

We consider simple undirected graphs $G=(V,E)$, where V and E denote the vertex and edge sets, respectively. A graph $G'=(V',E')$ is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$, with every edge in E' having both endpoints in V'. The subgraph of G *induced* by $V' \subseteq V$ is denoted by $G[V'] = (V', E_{|V'})$, where $E_{|V'} = \{\{u,v\} \in E : u,v \in V'\}$. The degree of a vertex $v \in V$, denoted by $\deg(v)$, is the number of vertices adjacent to v, namely $\deg(v) = |\{u \in V : \{u,v\} \in E\}|$.

Definition 2.3 (c-DEGENERATE GRAPH, [LW70])**.** A graph G of n vertices is called c-degenerate if every non-empty subgraph of G contains at least one vertex of degree at most c.

Lemma 2.4 (Spectral norm of c-degenerate graphs, [Hay06])**.** *The spectral norm of the adjacency matrix of a c-degenerate graph G with maximum degree d is at most $2\sqrt{cd}$.*

Proof. By c-degeneracy, we can repeatedly remove a vertex whose degree in the remaining graph is at most c. Orient each edge from the endpoint removed earlier to the endpoint removed later. The resulting directed adjacency matrix B has at most c nonzero entries per row and at most d per column, and satisfies $A_G = B + B^\top$. Thus $\|B\|_\infty \le c$ and $\|B\|_1 \le d$, giving $\|A_G\|_2 \le 2\|B\|_2 \le 2\sqrt{\|B\|_1\|B\|_\infty} \le 2\sqrt{cd}$. □

A *path* of length $\ell \ge 0$ in G is a sequence of distinct vertices $u_0, u_1, \dots, u_\ell$ such that $\{u_{i-1}, u_i\} \in E$ for every $i \in \{1, \dots, \ell\}$. A *cycle* of length $\ell \ge 3$ in G is a sequence of distinct vertices $u_1, \dots, u_\ell$ such that $\{u_i, u_{i+1}\} \in E$ for every $i \in \{1, \dots, \ell - 1\}$ and $\{u_\ell, u_1\} \in E$.

Definition 2.5 (Girth of a graph)**.** The girth of a graph G denoted by $\mathrm{girth}(G)$ is the length of the shortest cycle in G.

The following result gives a lower bound on the number of vertices of a graph in terms of its girth and average degree.

Lemma 2.6 (Average Moore bound, [AHL02])**.** *The number of vertices n in a graph of girth g and average degree at least $d \ge 2$ satisfies: $n \ge (d-1)^{\lfloor \frac{(g-1)}{2} \rfloor}$.*

In constructing the NO-graph family, we use a *closure* operation that enlarges a vertex set until every short path between its vertices lies entirely within the set, and we prove a bound on the size of the result. Related closure operations have been studied in other contexts, such as LP integrality gaps and sum-of-squares lower bounds [CMM09; BCK15].

Definition 2.7 (Closure of a set of vertices)**.** Let $G = (V, E)$ be a finite graph, let $r \ge 2$ be an integer, and let $U \subseteq V$. Starting from U, repeat the following step as long as possible:

> If there is a path of length $\ell \in \{2, \dots, r\}$ whose two endpoints are distinct vertices of the current set U and whose internal vertices all lie outside U, add its internal vertices to U.

Since each step enlarges U, and V is finite, the process terminates. The resulting set does not depend on the order in which the steps are performed; we call it the *closure* of U with parameter r and denote it by $\mathrm{clos}_{G,r}(U)$. We drop the subscript G when the graph is clear from context.

When the girth is large enough, the size of the closure scales linearly in the size of the initial set $|U| = t$ and the distance parameter r. A similar bound was established in [BCK15, Lemma 4.4] under different hypotheses.

Lemma 2.8 (Size of closure)**.** *Let $U \subseteq V$ be a set of vertices such that $|U| = t \ge 1$ and $r \ge 2$ be an integer. Suppose G has girth at least $g \ge 4r\log(2rt) + 4$. Then the procedure of Definition 2.7 makes fewer than $2t$ additions and the final number of vertices is $|\mathrm{clos}_r(U)| < 2rt$.*

Proof. We let U_i denote the set U after i steps of adding paths as in Definition 2.7, so $U_0 = U$ is the original set. Let $v_i = |U_i|$ and e_i be the number of edges of $G[U_i]$. We have $e_{i+1} - e_i \ge v_{i+1} - v_i + 1$, since the added path adds exactly $\ell - 1$ vertices and at least ℓ edges in the induced subgraph (for some ℓ). Applying this inequality i times and using $v_0 = t$, $e_0 \ge 0$ gives $e_i - v_i \ge e_0 - v_0 + i \ge i - t$. Moreover, $v_i \le t + i(r-1)$ since each step adds at most $r - 1$ vertices.

Suppose for contradiction that the process makes at least $2t$ steps. Then $v_{2t} < 2rt$ and $e_{2t} - v_{2t} \ge t$, so $G[U_{2t}]$ has average degree $d = 2e_{2t}/v_{2t} \ge 2 + 2t/v_{2t} > 2 + 1/r$. Since $G[U_{2t}]$ is a subgraph of G, it has girth at least g. By applying the Moore bound (Lemma 2.6), we get $v_{2t} \ge (d-1)^{\lfloor (g-1)/2 \rfloor} \ge (1 + 1/r)^{\lfloor (g-1)/2 \rfloor}$. Using $1 + x \ge \exp(\frac{x}{1+x})$ with $x = 1/r$, together with $\lfloor (g-1)/2 \rfloor \ge 2r\log(2rt)$ and $r + 1 \le 2r$, we get $v_{2t} > 2rt$, contradicting $v_{2t} < 2rt$.

Hence at most $2t - 1$ paths are added, and the final set satisfies $|\mathrm{clos}_r(U)| < 2rt$. □

3 Problem definition and setup

In this section we introduce the problem that we show to lie in StoqMA but not in MA. The task is to estimate the top eigenvalue of a graph given adjacency-list oracle access under a random labeling of its vertices. We first show that a stoquastic verifier solves this problem for every sparse graph, placing it in StoqMA (Proposition 3.4). We then construct two families of sparse graphs whose top eigenvalues are separated by a polynomial gap in m, providing the YES and NO instances (Sections 3.2 and 3.3) used in our separation.

We access graphs through the standard adjacency-list oracle model, defined below.

Definition 3.1 (Oracle of a labelled graph)**.** Let G be a simple graph on 2^m vertices with maximum degree at most d, and let $\pi : V(G) \to [2^m]$ be a permutation of its vertices, which we call a *labeling*.

For $x \in [2^m]$ and $i \in [d]$, let $N(x)_i \in [2^m]$ be the label of the ith neighbor of $\pi^{-1}(x)$ (ordered by increasing label), or x if this vertex has fewer than i neighbors.

- *(Classical oracle)* Given $(x, i) \in [2^m] \times [d]$, the oracle returns $\mathcal{O}_{G,\pi}(x, i) = N(x)_i$.
- *(Quantum oracle)* The oracle implements the unitary $\mathcal{O}_{G,\pi} : |x\rangle|i\rangle|z\rangle \mapsto |x\rangle|i\rangle|z \oplus N(x)_i\rangle$.

For a labelled graph (G, π), let $A_{G,\pi} := P_\pi A_G P_\pi^{-1}$ be its permuted adjacency matrix, where P_π is the permutation matrix with $(P_\pi)_{x,u} = 1$ if $\pi(u) = x$ and 0 otherwise. It has the same spectrum as A_G, and if $|\psi_G\rangle$ is the top eigenvector of A_G then $|\psi_\pi\rangle = P_\pi|\psi_G\rangle$ is the top eigenvector of $A_{G,\pi}$.

When clear from the context, we may drop G from the subscript.

3.1 The Graph Perron-Eigenvalue Estimation problem

Our separating problem asks whether a sparse graph, accessed through its adjacency-list oracle, has a large or small Perron eigenvalue.

Definition 3.2 (Graph Perron-Eigenvalue Estimation)**.** The *Graph Perron-Eigenvalue Estimation* (GPE) problem is defined as follows. We are given oracle access to the adjacency list of a labelled graph G on 2^m vertices (see Definition 3.1). The graph is assumed to be poly(m)-sparse, meaning that G has maximum degree at most $d = \text{poly}(m)$. The top eigenvalue $\lambda_{\max}(G)$ of its adjacency matrix is promised to satisfy exactly one of the following:

$$\text{YES: } \lambda_{\max}(G) \geq a \qquad \text{NO: } \lambda_{\max}(G) \leq b,$$

where $a - b \geq 1/\text{poly}(m)$. The goal is to output 1 in the YES case and 0 in the NO case.

The matrix $-A_{G,\pi}$ is a stoquastic sparse Hamiltonian whose ground-state energy is the negative of the top eigenvalue of $A_{G,\pi}$. Estimating this energy can be done in StoqMA given the adjacency-list oracle, by a recent result of Grilo and Rozos [GR26]. For completeness, we give a direct proof by constructing a stoquastic verifier that estimates the top-eigenvalue of the graph.

Lemma 3.3 (Stoquastic verifier for GPE)**.** *Let G be a simple graph on 2^m vertices of maximum degree at most $d = \text{poly}(m)$. Then there exists a stoquastic verifier (Definition 2.2) with access to the quantum oracle $\mathcal{O}_{G,\pi}$ (Definition 3.1) which, given any m-qubit state $|\psi\rangle$ as input, makes at most $\text{poly}(m)$ queries to $\mathcal{O}_{G,\pi}$ and accepts with probability*

$$\frac{1}{2} + \frac{\langle\psi \,|\, A_{G,\pi} \,|\, \psi\rangle}{2d}.$$

Proof. For adjacent vertices with labels x and y, let $i_{x\to y} \in [d]$ denote the unique index satisfying $N(x)_{i_{x\to y}} = y$. Since G is undirected, the reverse position $i_{y\to x}$ also exists, although the two positions need not coincide. Define, for any $x \in [2^m]$, $i \in [d]$, $b \in \{0, 1\}$,

$$U_{G,\pi}|x, i, b\rangle = \begin{cases} |y, i_{y\to x}, b\rangle, & \text{if } N(x)_i = y \text{ and } x \neq y, \\ |x, i, 1 - b\rangle, & \text{if } N(x)_i = x. \end{cases}$$

This map is an involution on basis states and hence defines a unitary. Its action can be computed using $O(d)$ queries and $\mathrm{poly}(m)$ classical reversible gates and ancilla qubits initialized to $|0\rangle$.

Let $|u_d\rangle = \frac{1}{\sqrt{d}} \sum_{i=1}^{d} |i\rangle = |+\rangle^{\otimes \log d}$ (we assume that d is a power of two). Then $U_{G,\pi}$ satisfies the block-encoding identity,

$$(\mathbb{I} \otimes \langle u_d| \otimes \langle 0|)\, U_{G,\pi}\, (\mathbb{I} \otimes |u_d\rangle \otimes |0\rangle) = \frac{A_{G,\pi}}{d}. \tag{1}$$

Indeed, each edge contributes $1/d$, while unoccupied positions flip the last qubit and therefore contribute zero. Given a witness $|\psi\rangle$, the verifier applies controlled-$U_{G,\pi}$ on $|+\rangle|\psi\rangle|u_d\rangle|0\rangle$, obtaining $\frac{1}{\sqrt{2}}(|0\rangle|\psi\rangle|u_d\rangle|0\rangle + |1\rangle U_{G,\pi}(|\psi\rangle|u_d\rangle|0\rangle))$. It then measures the control qubit in the Hadamard basis and accepts on outcome $+$. By Equation (1), its acceptance probability is $\frac{1}{2} + \frac{1}{2}(\langle\psi|\langle u_d|\langle 0|)U_{G,\pi}(|\psi\rangle|u_d\rangle|0\rangle) = \frac{1}{2} + \frac{1}{2d}\langle\psi|A_{G,\pi}|\psi\rangle$. □

The verifier above converts the eigenvalue promise in GPE into a completeness–soundness gap, yielding the following containment in StoqMA.

Proposition 3.4. *The GPE problem (Definition 3.2) belongs to* StoqMA.

Proof. By Lemma 3.3, there is a stoquastic verifier that uses $\mathrm{poly}(m)$ queries to $\mathcal{O}_{G,\pi}$ and accepts an m-qubit witness $|\psi\rangle$ with probability $\frac{1}{2} + \frac{1}{2d}\langle\psi|A_{G,\pi}|\psi\rangle$. If G is a YES instance, then $\lambda_{\max}(G) \geq a$, so the top eigenstate $|\psi_\pi\rangle$ of $A_{G,\pi}$ is accepted with probability at least $\frac{1}{2} + \frac{a}{2d}$. If G is a NO instance, then $\lambda_{\max}(G) \leq b$, so every witness is accepted with probability at most $\frac{1}{2} + \frac{b}{2d}$. Hence GPE belongs to $\mathsf{StoqMA}\big(\frac{1}{2} + \frac{a}{2d}, \frac{1}{2} + \frac{b}{2d}\big)$. Since $d \leq \mathrm{poly}(m)$ and $a - b \geq 1/\mathrm{poly}(m)$, the completeness–soundness gap is inverse polynomial, proving the claim. □

3.2 The YES-instances: High-degree, high-girth graphs

We describe the YES instance G_n that underlies the separation. For each value of the parameter n there is a single such graph, so the YES instances form the sequence $(G_n)_n$. Throughout, n is a free integer parameter, which will later be chosen as $n = \mathrm{poly}(m)$. Following [HBS26], the graph G_n is obtained from a high-degree, high-girth base graph (which, unlike in [HBS26], need not be an expander) by decorating it with self-similar trees and adding dummy isolated vertices.

We first define the *base graph* B_n using Dahan's construction [Dah14]. This generalizes the Lubotzky–Phillips–Sarnak construction of Ramanujan graphs [LPS88] to arbitrary sufficiently large degrees while preserving large girth.

Lemma 3.5 (High-degree, high-girth base graph B_n [Dah14])**.** *For every sufficiently large integer n, there is an explicitly constructible connected simple n-regular graph B_n on N_B vertices satisfying*

$$2^{n^3 \log^2 n} \leq N_B \leq 2^{2n^3 \log^2 n} \qquad \textit{and} \qquad \mathrm{girth}(B_n) \geq g_n := 100 n^{5/2} \log n.$$

We then *decorate* every vertex of B_n with copies of self-similar trees. This fractal-like decoration has appeared in constructions of stoquastic Hamiltonians [Has21; GHV21; HBS26], and we use the version of [HBS26].

Definition 3.6 (Self-similar trees $\widehat{T}_{n,k}$)**.** Let $n \geq 1$ and $1 \leq k \leq \sqrt{n}$. Define the degree and depth sequences $d_{n,k} = 2n - k\sqrt{n}$ and $\ell_{n,k} = k \cdot 10\, n^{3/2} \log n$. The *perfect tree* $C_{n,k}$ is the rooted tree of depth $\ell_{n,k}$ in which every internal vertex has exactly $d_{n,k} - 1$ children. Its leaves are called the 0-*level decorated leaves*.

The self-similar tree $\widehat{T}_{n,k}$ is obtained from $C_{n,k}$ as follows. For each $j = 1, \ldots, k-1$, attach to every internal vertex of $C_{n,k}$ $d_{n,j} - d_{n,j+1} = \sqrt{n}$ new edges, and join the other endpoint of each new edge to the root of a fresh copy of $\widehat{T}_{n,j}$. The base case is $\widehat{T}_{n,1} = C_{n,1}$.

Finally, we add sufficiently many isolated vertices that they form an overwhelming majority of the graph. This is a standard trick [CCD+03] to force classical algorithms to explore the nontrivial part of the graph by following edges from known vertices (Lemma 4.2). The resulting graph G_n, which serves as the YES instance, is defined below.

Definition 3.7 (Decorated base graph G_n)**.** Let $n \geq 1$ and $K = \sqrt{n}$. The graph G_n is built in three steps.

1. Start from the n-regular base graph B_n on $2^{n^3 \log^2 n} \leq N_B \leq 2^{2n^3 \log^2 n}$ vertices with girth at least $g_n = 100n^{5/2} \log n$ (see Lemma 3.5).
2. For each $k = 1, \ldots, K-1$ and each vertex of B_n, attach $d_{n,k} - d_{n,k+1} = \sqrt{n}$ new edges, joining the other endpoint of each to the root of a fresh copy of the self-similar tree $\widehat{T}_{n,k}$ (see Definition 3.6).
3. Add isolated vertices until the total number of vertices is $2^{15n^3 \log^2 n}$.

The vertex set of G_n is $V = V_B \cup V_T \cup V_I$, where V_B consists of the *base graph vertices* (the vertices of B_n), V_T of the *tree vertices* (those in the copies of $\widehat{T}_{n,k}$), and V_I of the *isolated vertices*. When several graphs are in play, we write $V(G_n) = V$, and similarly for the other sets.

To analyse G_n, we will need to relate the tree vertices to their attachment points in the base graph. We do so using a projection onto the base graph, which also induces a notion of distance between vertices. These notions will also be central to the construction and analysis of our NO instances.

Definition 3.8 (Base projection and base distance)**.** For each $v \in V_B \cup V_T$, let $\mathrm{proj}_B(v) \in V_B$ denote its *base projection*, defined as the unique base vertex closest to v in G_n.

The *base distance* between $u, v \in V_B \cup V_T$ is $\mathrm{dist}_B(u, v) = \mathrm{dist}(\mathrm{proj}_B(u), \mathrm{proj}_B(v))$, where dist denotes graph distance in B. Equivalently, this is the number of base graph edges on a shortest path from u to v in G_n. We set $\mathrm{dist}_B(u, v) = +\infty$ if either vertex lies in V_I. For $U \subseteq V$, we define $\mathrm{dist}_B(U, v) = \min_{u \in U} \mathrm{dist}_B(u, v)$.

Apart from the isolated vertices, the degree of every vertex in G_n is either $d_{n,1}$ or 1 (the degree-one vertices are exactly the leaves of the self-similar trees). The graph G_n also has the same girth as B. The top eigenvalue is at least n, so we can set $a = n$ in the associated GPE problem (Definition 3.2).

Lemma 3.9 (Large Perron eigenvalue)**.** *The top eigenvalue of G_n is at least n.*

Proof. The base graph B is an n-regular induced subgraph of G_n, so its top eigenvalue is exactly n. The top eigenvalue of a graph is at least that of any of its subgraphs, so $\lambda_{\max}(G_n) \geq n$. □

3.3 The NO-instances: Decorated closure graphs

We construct the NO family as a collection of subgraphs of G_n, one for each set U of at most $2^{\sqrt{n}}$ vertices. The construction relies on the closure operation (Definition 2.7) and is defined below.

Definition 3.10 (Decorated closure graph $F_n(U)$)**.** Let $U \subseteq V$ be a subset of vertices of G_n. The graph $F_n(U)$ is built in three steps.

1. Let $\overline{U} = \mathrm{clos}_{2\ell_{n,K}+3}(\mathrm{proj}_B(U)) \subseteq V_B$ be the closure vertices of the base projection of U (Definitions 2.7 and 3.8).
2. Let $W = \{v \in V : \mathrm{dist}_B(\overline{U}, v) \leq \ell_{n,K}\} \cup \{v \in V_B : \mathrm{dist}_B(\overline{U}, v) = \ell_{n,K} + 1\}$. The first set contains all *base and tree* vertices of G_n at base distance at most $\ell_{n,K}$ from $\overline{U}$, while the second adds only *base* vertices at base distance exactly $\ell_{n,K} + 1$. We call the vertices in the second set the *boundary vertices* of U, denoted by ∂U.

3. Define the graph $F_n(U) = (V, E_{|W})$ by extending the vertex set of $G_n[W]$ to V, with all added vertices isolated.

The resulting graph $F_n(U)$ has the same vertex set as G_n and exactly the edges of the induced subgraph $G_n[W]$. It is the member of the NO family defined by U.

The key combinatorial property of the closure graph $F_n(U)$ is that it is 2-degenerate whenever the seed set U is sufficiently small.

Lemma 3.11 (2-DEGENERACY OF CLOSURE)**.** *If $|U| \leq 2^{\sqrt{n}}$ then the graph $F_n(U)$ is 2-degenerate.*

Proof. First, we show that $G_n[\overline{U}]$ is 2-degenerate. Suppose otherwise. By Definition 2.3, it contains a non-empty subgraph G' with minimum degree at least 3. Since G' has girth at least g_n, the Moore bound (Lemma 2.6) gives $|V(G')| \geq (3-1)^{\lfloor (g_n-1)/2 \rfloor}$. On the other hand, Lemma 2.8 gives $|V(G')| \leq |\overline{U}| < 2(2\ell_{n,K}+3)2^{\sqrt{n}} < 2^{\lfloor (g_n-1)/2 \rfloor}$ for sufficiently large n, since $\ell_{n,K} = 10n^2 \log n$ and $g_n = 100n^{5/2} \log n$. This contradiction proves that $G_n[\overline{U}]$ is 2-degenerate.

Next, we show that the subgraph $G_n[W]$ remains 2-degenerate. We claim that $G_n[W]$ is obtained from $G_n[\overline{U}]$ by attaching trees at some vertices. Notice that such an operation preserves 2-degeneracy. To prove the claim, take a breadth-first forest of the base vertices in W, rooted at $\overline{U}$. Each tree has depth at most $R := \ell_{n,K} + 1$. Any base edge belonging to neither this forest nor $G_n[\overline{U}]$ would join vertices in the same tree, creating a cycle of length at most $2R+1$, or vertices in different trees, creating a path of length between 2 and $2R+1$ between distinct vertices of $\overline{U}$ with all internal vertices outside $\overline{U}$. The former contradicts the girth bound $g_n > 2R+1$, while the latter contradicts the closure property, since $2R+1 = 2\ell_{n,K}+3$. Hence every base edge outside $G_n[\overline{U}]$ belongs to the breadth-first forest. Neither these edges nor the decoration edges create new cycles, proving the claim.

Finally, $F_n(U)$ is 2-degenerate as well since it adds no edges over $G_n[W]$. □

It follows from the 2-degenerate property that the graph $F_n(U)$ has a small top eigenvalue.

Proposition 3.12 (SMALL PERRON EIGENVALUE)**.** *If $|U| \leq 2^{\sqrt{n}}$ then the top eigenvalue of $F_n(U)$ is at most $4\sqrt{n}$.*

Proof. This follows directly by applying Lemma 2.4, Lemma 3.11 and the fact that the maximum degree of the decorated closure graph is at most that in G_n, i.e., $d_{n,1} = 2n - \sqrt{n}$. □

It follows that the YES and NO instances constructed above can be distinguished in StoqMA.

Corollary 3.13 (StoqMA PROTOCOL FOR G_n VS. $F_n(U)$)**.** *There is a* StoqMA *protocol distinguishing* YES*: $G_{n,\pi}$ from* NO*: $F_{n,\pi}(U)$, where π is any labeling and $U \subset V$ any subset of at most $2^{\sqrt{n}}$ vertices.*

Proof. By Lemma 3.9 and Proposition 3.12, the top eigenvalues of the two graphs differ by at least $n - 4\sqrt{n}$. Moreover, both graphs have degree at most $2n - \sqrt{n}$. Hence, they represent valid instances of the Graph Perron-Eigenvalue Estimation problem (Definition 3.2) and can be solved by a StoqMA verifier (Proposition 3.4). □

4 Lower bound when advice is a set of vertices

In this section we show that no randomised algorithm can efficiently distinguish the two graph families even when given some subset of vertices as advice. More precisely, given any set of vertices U and any fixed labeling of them as input, we prove that there is no randomised algorithm that can distinguish between G_n and $F_n(U)$ by making at most polynomially many queries to the graph oracle.

We first show that there is a partial isomorphism between G_n and $F_n(U)$ for any set of vertices U. Recall that both have the same vertex set V and only differ in the edges adjacent to some vertices which is made precise in the following proposition.

Proposition 4.1 (Isomorphism up to the boundary). *Let $U \subseteq V$, and let W and the boundary $\partial U \subseteq W$ be as in Definition 3.10. Every vertex of $W \setminus \partial U$ has the same neighborhood in G_n and $F_n(U)$. The neighborhoods differ only at vertices of $\partial U \cup Y$, where Y consists of the vertices outside W that are non-isolated in G_n. More precisely,*

(a) $N_{F_n(U)}(v) = N_{G_n}(v)$ *for every* $v \in W \setminus \partial U$;

(b) $N_{F_n(U)}(v) = N_{G_n}(v) \cap W \subsetneq N_{G_n}(v)$ *for every* $v \in \partial U$;

(c) every vertex outside W is isolated in $F_n(U)$.

Consequently, for every labeling π and every query (x, i) with $\pi^{-1}(x) \in W \setminus \partial U$, $\mathcal{O}_{F_n(U),\pi}(x, i) = \mathcal{O}_{G_n,\pi}(x, i)$.

Moreover, the graph $G_n[W]$ is isomorphic to the graph obtained from $G_n[\overline{U}]$ by attaching to each $u \in \overline{U}$ as many copies of $\widehat{T}_{n,k}$ as it has in G_n (for $k = 1, \ldots, K-1$), together with as many copies of $\widehat{T}_{n,K}$ as the number of base graph neighbors of u outside $\overline{U}$. The isomorphism can be chosen so that it maps the boundary vertices, i.e., the base graph vertices at distance exactly $\ell_{n,K} + 1$ from $\overline{U}$ in G_n, bijectively onto the 0-level leaves of the attached copies of $\widehat{T}_{n,K}$.

Proof. The neighbors of vertices in $W \setminus \partial U$ are contained in W by step 2 of Definition 3.10. The claim about the same neighborhood and (a) follow from step 2 and 3 of Definition 3.10, and so does (c). For (b), step 2 does not put the tree vertices attached to a boundary vertex into W, and its base neighbors at distance $\ell_{n,K} + 2$ from $\overline{U}$ are not in W either, so $N_{F_n(U)}(v) = N_{G_n}(v) \cap W \subsetneq N_{G_n}(v)$ for every $v \in \partial U$.

We now prove the second part of the claim. Consider any vertex $u \in \overline{U}$. Let $v \in V_B \setminus \overline{U}$ be a base graph neighbor of u, and let $T(u, v)$ denote the induced subgraph of $G_n[W]$ consisting of the vertices reachable from v *after removing* the edge (u, v). It suffices to show that the subgraphs $\{T(u, v)\}_{u,v}$ are disjoint and that $T(u, v)$ is isomorphic to $\widehat{T}_{n,K}$ for all u, v.

The breadth-first forest argument in the proof of Lemma 3.11 shows that every base edge of $G_n[W]$ outside $G_n[\overline{U}]$ belongs to a forest rooted at $\overline{U}$. Since the decorations are themselves trees attached at single vertices, each edge (u, v) considered above separates $T(u, v)$ from the rest of $G_n[W]$. In particular, the subgraphs $T(u, v)$ are pairwise disjoint.

Finally, since the base graph is n-regular, the base vertices of $T(u, v)$ form a perfect tree of depth $\ell_{n,K}$ with exactly $n - 1 = d_{n,K} - 1$ children at each internal vertex. All self-similar trees attached to base vertices at distance less than $\ell_{n,K}$ from v are retained in W, whereas the base vertices at distance $\ell_{n,K}$ are exactly the boundary vertices and retain no decorations. This matches the definition of $\widehat{T}_{n,K}$ in Definition 3.6, with the boundary vertices corresponding to its 0-level leaves. See Figure 3 for an illustration. □

We can restrict attention to algorithms that explore the graph locally, meaning that the queried vertices form connected components rooted at the vertices of U, together with some isolated vertices. Indeed, the proportion of non-isolated vertices in G_n is exponentially small, so if a classical algorithm that starts from U queries a vertex that is not connected to the vertices it has already explored, this vertex is isolated in G_n with overwhelming probability. In particular, the algorithm queries a vertex that is isolated only in $F_n(U)$ with negligible probability, unless it reaches this vertex by exploring from U.

Lemma 4.2 (Local exploration, adapted from [HBS26]). *Let $U \subseteq V$ be a subset of vertices of G_n with $|U| \leq |V|/2$ and a fixed labeling, and let $G \in \{G_n, F_n(U)\}$. Let $\mathcal{A}$ be a classical algorithm that is given $\pi(U)$ and makes q queries to the adjacency-list oracle $\mathcal{O}_{G,\pi}$, for a uniformly random π subject to the fixed labels of U. Then, except with probability at most $q \cdot 2^{-2n^3 \log^2 n}$, the set of all vertices queried by $\mathcal{A}$ at any point during its execution induces a collection of connected components rooted at the vertices in U, together with some vertices that are isolated in G_n (and hence also in $F_n(U)$).*

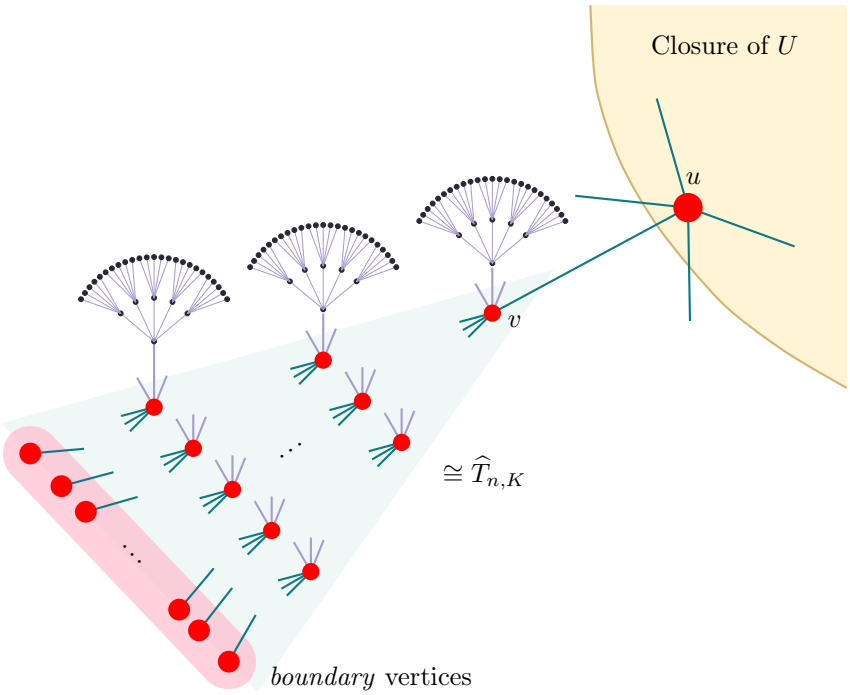


Figure 3: Pictorial representation of the subgraph reachable from a vertex u in the closure of U via an edge uv with v outside the closure. The green triangle depicts the base vertices, with their green branches indicating the n base-graph neighbors, and the purple branches attached to them depict copies of self-similar trees. No such trees are attached to the farthest base vertices, which form the boundary. Together, this subgraph is isomorphic to $\widehat{T}_{n,K}$, as proved in Proposition 4.1.

The proof follows the same argument as in [CCD+03] or [HBS26], so we omit it. Thus, we can now focus on the hardness of reaching a boundary vertex starting from U. By Proposition 4.1, any path from a vertex in U to a boundary vertex must use an edge joining a vertex in the closure to a vertex outside it, and hence contains a root-to-leaf path in a copy of the self-similar tree $\widehat{T}_{n,K}$. We use the following result on the hardness of exploring self-similar trees, first shown in [GHV21], in the variant suited to this construction from [HBS26].

Lemma 4.3 (Hardness of exploring self-similar trees, [HBS26]). *Let $K = \sqrt{n}$, as in Definition 3.7. Suppose a classical algorithm explores the tree $\widehat{T}_{n,K}$ under a uniformly random labeling, starting at its root and querying only vertices adjacent to those it has already visited. If the algorithm makes fewer than $2^{\sqrt{n}}$ queries, the probability that it ever reaches a 0-level leaf of $\widehat{T}_{n,K}$ is at most $2^{-2n\log n}$.*

We are now ready to prove indistinguishability against any classical randomised algorithm making few queries.

Theorem 4.4 (Classical hardness of distinguishing). *Let U be a set of $|U| \le 2^{\sqrt{n}}$ vertices of G_n, and fix any partial labeling $\sigma : U \to [2^m]$. Let π be a uniformly random labeling subject to $\pi|_U = \sigma$. Let $\mathcal{A}$ be any classical randomised algorithm that is given $\pi(U)$ and makes at most $q < 2^{\sqrt{n}}$ queries to the adjacency-list oracle $\mathcal{O}_{G,\pi}$, where $G \in \{G_n, F_n(U)\}$. Then*

$$\left| \Pr[\mathcal{A}^{\mathcal{O}_{G_n,\pi}}(\pi(U)) = 1] - \Pr[\mathcal{A}^{\mathcal{O}_{F_n(U),\pi}}(\pi(U)) = 1] \right| \le 2^{-n\log n}.$$

Proof. By Proposition 4.1, G_n and $F_n(U)$ differ only on the boundary ∂U and on the set Y. Vertices of Y are not isolated in G_n, so by Lemma 4.2, except with probability at most

$q \cdot 2^{-2n^3 \log^2 n}$, $\mathcal{A}$ queries a vertex of Y only after exploring to it from U. By Proposition 4.1(a), this requires first exploring a vertex of ∂U. Hence the advantage of $\mathcal{A}$ is at most

$$q \cdot 2^{-2n^3 \log^2 n} + \Pr[\mathcal{A} \text{ explores a vertex of } \partial U].$$

By Proposition 4.1, the vertices of ∂U correspond exactly to the 0-level leaves of the at most $n|\overline{U}|$ copies of $\widehat{T}_{n,K}$ attached to $\overline{U}$, one for each base-graph edge leaving the closure; these copies are pairwise disjoint and independently labelled. Fix one such copy T. Although $\mathcal{A}$'s queries to T may be interleaved adaptively with its queries elsewhere, we may simulate every query of $\mathcal{A}$ outside T using the rest of the graph together with the given labels $\pi(U)$, which expose no internal vertex of T, and only charge to T the at most q queries that $\mathcal{A}$ makes inside it. Since $q < 2^{\sqrt{n}}$, Lemma 4.3 applied to this restricted exploration bounds the probability that $\mathcal{A}$ reaches a 0-level leaf of T by $2^{-2n \log n}$, independently of what $\mathcal{A}$ learned from the other copies. A union bound over the at most $n|\overline{U}|$ copies, together with $|\overline{U}| \le 2(2\ell_{n,K}+3)|U|$ (Lemma 2.8), then gives

$$\Pr[\mathcal{A} \text{ explores a vertex of } \partial U] \;\le\; n|\overline{U}|\, 2^{-2n \log n} \;\le\; 2(2\ell_{n,K}+3)\, n\, 2^{\sqrt{n}-2n \log n}.$$

Using the definition of $\ell_{n,K}$, the two error terms together are at most $2^{-n \log n}$. □

5 Lower bound when advice is arbitrary

The previous section shows that polynomially many classical queries cannot distinguish G_n from $F_n(U)$, even when the labels of the seed set U are given as input. An MA witness, however, may be any polynomial-length bit string depending on the entire labelled graph. Since completeness only requires the existence of *some* accepted witness for each labeling, we may fix, without loss of generality, one such witness $w_\pi \in \{0,1\}^{p(m)}$ for every labeling π, for some polynomial p. To handle such witnesses, we use a presampling lemma for random permutations from the cryptographic literature [CDG18; CDGS18].

Informally, the following lemma says that an algorithm that receives a $p(m)$-bit string w_π depending on the whole labeling π, and then queries π, can be simulated, up to a small error, by an algorithm that receives the same string but queries a labeling that is *uniformly random* subject only to the labels of some t vertices being fixed. In other words, the $p(m)$-bit string reveals no more about π than the labels of t vertices, chosen depending on the string. More precisely, consider the following experiment, which draws π, π' jointly:

1. draw π uniformly at random among all bijections $\pi : V \to [2^m]$ (labelings of V), and let w_π be its witness;
2. draw a partial labeling $\sigma : U \to [2^m]$, with $|U| \le t$, from a distribution $\mathcal{D}_{w_\pi}$ depending on w_π;
3. draw π' uniformly at random among all bijections $\pi' : V \to [2^m]$ subject to $\pi'|_U = \sigma$.

The lemma states that querying π on input w_π, and querying π' on the same input w_π, give nearly the same output distribution.

Lemma 5.1 (Arbitrary witness to label-fixing, adapted from [CDG18])**.** *Consider $\pi \mapsto w_\pi \in \{0,1\}^{p(m)}$, an arbitrary map, defined on labelings π of a vertex set V with $|V| = 2^m \ge 16$. For any $1 \ge \gamma > 0$ and $t \in \mathbb{N}$, there is a family $\{\mathcal{D}_w\}_{w \in \{0,1\}^{p(m)}}$ of distributions over partial labelings $\sigma : U \to [2^m]$ with $|U| \le t$, such that for π, π' drawn as in the experiment above, and for any classical algorithm $\mathcal{B}$ that takes a $p(m)$-bit input, outputs a bit, and makes at most q queries to the labeling, each either a forward query $v \mapsto \pi(v)$ or a backward query $x \mapsto \pi^{-1}(x)$,*

$$\left| \Pr_{\pi}\left[\mathcal{B}^{\pi}(w_\pi) = 1\right] - \Pr_{\pi,\pi'}\left[\mathcal{B}^{\pi'}(w_\pi) = 1\right] \right| \le \frac{2\,(p(m) + \log 1/\gamma) \cdot q}{t} + \gamma.$$

In our case, the algorithm queries the relabelled adjacency-list oracle $\mathcal{O}_{G,\pi}$ rather than the labeling π directly. The corollary below shows that an algorithm querying π can simulate one querying $\mathcal{O}_{G,\pi}$, at a cost of a factor $2n$ in the number of queries. Hence Lemma 5.1 applies, and the witness can again be treated as essentially only a set of vertices with fixed labels.

Corollary 5.2 (Simulating adjacency-list queries by labeling queries)**.** *Let $G \in \{G_n, F_n(U)\}$ for some set U. For p, γ, t and π, π' as in Lemma 5.1, and every classical algorithm $\mathcal{A}$ that takes a $p(m)$-bit input and makes at most q queries to the adjacency-list oracle $\mathcal{O}_{G,\pi}$ (Definition 3.1),*

$$\left|\Pr_{\pi}\left[\mathcal{A}^{\mathcal{O}_{G,\pi}}(w_\pi) = 1\right] - \Pr_{\pi,\pi'}\left[\mathcal{A}^{\mathcal{O}_{G,\pi'}}(w_\pi) = 1\right]\right| \le \frac{2\,(p(m) + \log 1/\gamma) \cdot 2nq}{t} + \gamma.$$

Proof. There is an algorithm $\mathcal{B}$ that simulates $\mathcal{A}$ using only labeling queries. $\mathcal{B}$ has G hard-wired, so it can look up neighborhoods in G for free. It runs $\mathcal{A}$ and answers each query (x, i) as follows: it computes $u = \pi^{-1}(x)$ with one backward query, then queries $\pi(v)$ for every neighbor v of u in G, sorts these labels, and returns the ith one (or x if $i > \deg_G(u)$). This gives exactly the answer of $\mathcal{O}_{G,\pi}$, so $\mathcal{B}^\pi(w)$ and $\mathcal{A}^{\mathcal{O}_{G,\pi}}(w)$ have the same output distribution for every labeling π and every input w. In particular,

$$\Pr[\mathcal{B}^\pi(w_\pi) = 1] = \Pr[\mathcal{A}^{\mathcal{O}_{G,\pi}}(w_\pi) = 1] \quad \text{and} \quad \Pr[\mathcal{B}^{\pi'}(w_\pi) = 1] = \Pr[\mathcal{A}^{\mathcal{O}_{G,\pi'}}(w_\pi) = 1].$$

Since G_n, hence also $F_n(U)$, has maximum degree less than $2n$, each query of $\mathcal{A}$ costs $\mathcal{B}$ at most $1 + \deg_G(u) \le 2n$ queries, so $\mathcal{B}$ makes at most $2nq$ queries. The family $\{\mathcal{D}_w\}$ from Lemma 5.1 does not depend on the algorithm, so applying that lemma to $\mathcal{B}$, with $2nq$ in place of q, and substituting the two equalities above gives the claimed bound. □

We now combine this corollary with Theorem 4.4 to show that arbitrary polynomial-length witnesses cannot help distinguish the two graph families.

Proposition 5.3. *There is no* MA *protocol that solves the GPE problem using polynomial witness length and polynomially many queries to $\mathcal{O}_{G,\pi}$, where $G \in \{G_n, F_n(U)\}$ and U ranges over all sets of at most $2^{\sqrt{n}}$ vertices. In fact, any such protocol errs on some instance with parameter n for every sufficiently large n.*

Proof. Fix a protocol with a verifier $\mathcal{V}$ that makes at most $q \le \mathrm{poly}(n)$ queries and uses witnesses of length at most $p(m) = \mathrm{poly}(n)$, and suppose that it solves the problem on all instances with parameter n, for a sufficiently large n. For every π it satisfies:

- *Completeness.* There is a witness $w \in \{0,1\}^{p(m)}$ with $\Pr[\mathcal{V}^{\mathcal{O}_{G_n,\pi}}(w) = 1] \ge 2/3$. Fix one such witness w_π for every π.
- *Soundness.* For every U with $|U| \le 2^{\sqrt{n}}$ and every witness w, $\Pr[\mathcal{V}^{\mathcal{O}_{F_n(U),\pi}}(w) = 1] \le 1/3$.

Let $t = 2^{\sqrt{n}}$ and $\gamma = 2^{-n}$. Applying Corollary 5.2 and lemma 5.1 to $\mathcal{V}^{\mathcal{O}_{G_n,\pi}}$, for large enough n, the experiment of Lemma 5.1 gives

$$\left|\Pr_{\pi}\left[\mathcal{V}^{\mathcal{O}_{G_n,\pi}}(w_\pi) = 1\right] - \Pr_{\pi,\pi'}\left[\mathcal{V}^{\mathcal{O}_{G_n,\pi'}}(w_\pi) = 1\right]\right| \le \frac{2(p(m) + n) \cdot 2nq}{2^{\sqrt{n}}} + 2^{-n} \le 2^{-\sqrt{n}/2}.$$

By completeness the first probability is at least $2/3$, so the second is at least $2/3 - 2^{-\sqrt{n}/2}$. The second probability is an average over (π, σ), so there are a π and a σ in the support of $\mathcal{D}_{w_\pi}$ such that, writing $w = w_\pi$ and $U = \operatorname{dom}\sigma$ (so $|U| \le t$), and with π' uniform among labelings with $\pi'|_U = \sigma$ (as in the rest of the proof),

$$\Pr_{\pi'}\left[\mathcal{V}^{\mathcal{O}_{G_n,\pi'}}(w) = 1\right] \ge \frac{2}{3} - 2^{-\sqrt{n}/2}. \tag{2}$$

The witness w is fixed before π' is drawn, so it is a constant. Let $\mathcal{A}_w$ be the algorithm that has w hard-wired, takes σ as input, and runs $\mathcal{V}(w)$. Then $\mathcal{A}_w^{\mathcal{O}_{G,\pi'}}(\sigma)$ and $\mathcal{V}^{\mathcal{O}_{G,\pi'}}(w)$ have the same output distribution for both $G \in \{G_n, F_n(U)\}$. Since σ is exactly the fixed assignment in the hypothesis of Theorem 4.4, that theorem applies to $\mathcal{A}_w$:

$$\begin{aligned}&\left|\Pr_{\pi'}\left[\mathcal{V}^{\mathcal{O}_{G_n,\pi'}}(w)=1\right]-\Pr_{\pi'}\left[\mathcal{V}^{\mathcal{O}_{F_n(U),\pi'}}(w)=1\right]\right| \\ &=\left|\Pr_{\pi'}\left[\mathcal{A}_w^{\mathcal{O}_{G_n,\pi'}}(\sigma)=1\right]-\Pr_{\pi'}\left[\mathcal{A}_w^{\mathcal{O}_{F_n(U),\pi'}}(\sigma)=1\right]\right| \le 2^{-n\log n}.\end{aligned}$$

Together with (2) this gives

$$\Pr_{\pi'}\left[\mathcal{V}^{\mathcal{O}_{F_n(U),\pi'}}(w)=1\right] \ge \tfrac{2}{3}-2^{-\sqrt{n}/2}-2^{-n\log n} > \tfrac{1}{3}.$$

So some π' with $\pi'|_U = \sigma$ makes $\mathcal{V}$ accept the NO instance $F_n(U)$ with witness w with probability more than $1/3$. This violates soundness, a contradiction. □

We have shown an *oracle-input* separation for GPE: the graph oracle is itself the instance, and a protocol given a witness and access to it, with no separate input string, decides whether it is a YES or NO instance. Thus GPE belongs to StoqMA but not to MA in this model. The remaining task is to turn this into a genuine separation of the relativized classes $\mathsf{StoqMA}^{\mathcal{O}}$ and $\mathsf{MA}^{\mathcal{O}}$ for an explicit oracle $\mathcal{O}$. This is a standard diagonalization argument, proved formally in [Zha25] and used in several earlier separations; we include a short proof adapted to our setting for completeness.

Theorem 1.1. *There exists a classical oracle $\mathcal{O}$ such that $\mathsf{MA}^{\mathcal{O}} \subsetneq \mathsf{StoqMA}^{\mathcal{O}}$.*

Proof. We will design an oracle $\mathcal{O}$ and a language $\mathcal{L}^{\mathcal{O}}$ such that $\mathcal{L}^{\mathcal{O}} \in \mathsf{StoqMA}^{\mathcal{O}}$ but $\mathcal{L}^{\mathcal{O}} \notin \mathsf{MA}^{\mathcal{O}}$. The oracle will be a sequence of graph oracles from the universe defined below.

For each n, let $m = m(n) = 15n^3\log^2 n$ and $d = 2n - \sqrt{n}$. The universe is $\mathcal{U} = \{\mathcal{U}_n\}_{n\in\mathbb{Z}_+}$, where $\mathcal{U}_n$ is the set of adjacency-list oracles $\mathcal{O}_{G,\pi}\colon [2^m]\times[d]\to[2^m]$ (Definition 3.1), one for each graph $G \in \{G_n\}\cup\{F_n(U) : U \subseteq V(G_n),\ |U| \le 2^{\sqrt{n}}\}$ and each bijective labeling $\pi\colon V(G)\to[2^m]$. As noted after Definition 2.2, these oracles fit $\mathsf{MA}^{\mathcal{O}}$ and $\mathsf{StoqMA}^{\mathcal{O}}$ via the one-bit-at-a-time simulation, at only polynomial overhead.

We build $\mathcal{O}$ in stages, diagonalising against all candidate MA verifiers. The oracle is a family of blocks $\{\mathcal{O}^n\}_{n\in\mathbb{Z}_+}$ with $\mathcal{O}^n \in \mathcal{U}_n$. Initially every block is $\mathcal{O}_{F_n(\emptyset),\mathrm{id}}$, where id is the identity permutation. At stage i we choose an integer $n_i > n_{i-1}$ and replace block n_i by an instance $\mathcal{O}^{n_i} \in \mathcal{U}_{n_i}$, chosen below, while every block $n \notin \{n_i\}$ stays $\mathcal{O}_{F_n(\emptyset),\mathrm{id}}$. Let $\mathcal{L}^{\mathcal{O}}_{\mathrm{yes}} = \{0^n : \mathcal{O}^n = \mathcal{O}_{G_n,\pi} \text{ for some } \pi\}$ and $\mathcal{L}^{\mathcal{O}}_{\mathrm{no}} = \{0,1\}^\star \setminus \mathcal{L}^{\mathcal{O}}_{\mathrm{yes}}$.

As in [Zha25], we pad all witnesses with 0's so that any potential MA verifier runs in quadratic time. Let $\mathcal{V}_1, \mathcal{V}_2, \ldots$ enumerate the quadratic-time randomised verifiers $\mathcal{V}_i^{\mathcal{O}}(x,w)$ with oracle access as in Definition 2.1. At stage i we want an input 0^{n_i} on which $\mathcal{V}_i$ errs. Our lower bound, however, is for protocols whose only input is the witness and which query a single graph oracle. We therefore derive such a protocol from $\mathcal{V}_i$. Let $\mathcal{O}_{i-1}$ be the oracle with blocks $n_1,\ldots,n_{i-1}$ chosen and all other blocks $\mathcal{O}_{F_n(\emptyset),\mathrm{id}}$. For an instance $\mathcal{O}_{G,\pi} \in \mathcal{U}_n$, let $\widetilde{\mathcal{V}}_i^{\mathcal{O}_{G,\pi}}(w)$ simulate $\mathcal{V}_i(0^n, w)$ with the oracle $\mathcal{O}_{i-1}[n \leftarrow \mathcal{O}_{G,\pi}]$, in which block n is replaced by $\mathcal{O}_{G,\pi}$. It hard-codes the blocks $n_1,\ldots,n_{i-1}$, whose size is independent of n and w, so $\widetilde{\mathcal{V}}_i$ runs in polynomial time, up to a constant depending on i. It is an MA protocol of the kind covered by Proposition 5.3.

We choose n_i and $\mathcal{O}^{n_i} \in \mathcal{U}_{n_i}$ such that both of the following hold:

- $\widetilde{\mathcal{V}}_i^{\mathcal{O}^{n_i}}(w)$ decides $\mathcal{O}^{n_i}$ incorrectly: either $\mathcal{O}^{n_i} = \mathcal{O}_{G_{n_i},\pi}$ for some π and every witness w is accepted with probability less than $2/3$, or $\mathcal{O}^{n_i} = \mathcal{O}_{F_{n_i}(U),\pi}$ and some witness is accepted with probability more than $1/3$.

- $n_i > n_{i-1}$ is large enough that the queries to block n_i are longer than the running time of $\mathcal{V}_j$ on inputs $(0^{n_j}, w)$, over all witnesses w of the padded length, for all $j < i$.

Such n_i and $\mathcal{O}^{n_i}$ exist because the lower bound of Proposition 5.3 holds for every sufficiently large n and applies to $\widetilde{\mathcal{V}}_i$.

We claim that $\mathcal{L}^{\mathcal{O}} = \mathcal{L}^{\mathcal{O}}_{\text{yes}}$ is in $\mathsf{StoqMA}^{\mathcal{O}}$. For strings other than 0^n we reject by a simple check. For 0^n we run the StoqMA protocol of Proposition 3.4 on block $\mathcal{O}^n$, which decides whether 0^n is a YES instance.

We now show that $\mathcal{L}^{\mathcal{O}} \notin \mathsf{MA}^{\mathcal{O}}$. Suppose not, and let $\mathcal{V}$ be an MA verifier for it. After padding it runs in quadratic time, so $\mathcal{V} = \mathcal{V}_i$ for some i in our enumeration. On input $(0^{n_i}, w)$, $\mathcal{V}_i$ cannot query the blocks n_j for $j > i$, by the choice of n_j, and the blocks that are not chosen are the fixed empty-graph blocks. So it behaves exactly as $\mathcal{V}_i(0^{n_i}, w)$ with oracle $\mathcal{O}_{i-1}[n_i \leftarrow \mathcal{O}^{n_i}]$, which is $\widetilde{\mathcal{V}}_i^{\mathcal{O}^{n_i}}(w)$ by definition. But $\widetilde{\mathcal{V}}_i^{\mathcal{O}^{n_i}}$ decides $\mathcal{O}^{n_i}$ incorrectly, so $\mathcal{V}_i$ decides 0^{n_i} incorrectly. This contradicts $\mathcal{V}$ being a valid verifier for $\mathcal{L}^{\mathcal{O}}$. □

Acknowledgements

We used AI (ChatGPT 5.6) during our search for a suitable NO-instance construction. It helped us identify an NP protocol for a previous construction studied in [HBS26] (see Appendix B) and pointed out that another of our earlier candidate graphs was 2-degenerate, suggesting a structural property that could be exploited to construct an alternative NO instance. This observation motivated our subsequent use of the closure and 2-degeneracy, which we developed ourselves for the final family of NO instances. We also used AI-based tools to assist with writing the TikZ code for the figures, as well as for language editing and proofreading.

This work was supported by the Maison du Quantique de Nouvelle-Aquitaine "HybQuant", as part of the HQI initiative and France 2030, under the French National Research Agency (ANR) grant ANR-22-PNCQ-0002. It was also supported by the PEPR integrated project EPiQ under ANR grant ANR-22-PETQ-0007.

References

[AGL25] D. Aharonov, A. B. Grilo, and Y. Liu. "StoqMA vs. MA: The Power of Error Reduction". In: *Quantum* 9 (2025), p. 1853 (cit. on pp. 2, 21).

[AHL02] N. Alon, S. Hoory, and N. Linial. "The Moore Bound for Irregular Graphs". In: *Graphs and Combinatorics* 18.1 (2002), pp. 53–57 (cit. on p. 8).

[AK07] S. Aaronson and G. Kuperberg. "Quantum Versus Classical Proofs and Advice". In: *Theory of Computing* 3.7 (2007), pp. 129–157 (cit. on p. 2).

[BBT06] S. Bravyi, A. J. Bessen, and B. M. Terhal. *Merlin-Arthur Games and Stoquastic Complexity*. arXiv:quant-ph/0611021. 2006 (cit. on pp. 1, 2, 7).

[BCK15] B. Barak, S. O. Chan, and P. K. Kothari. "Sum of Squares Lower Bounds from Pairwise Independence". In: *Proceedings of the 47th Symposium on Theory of Computing (STOC)*. 2015, pp. 97–106 (cit. on pp. 3, 8).

[BDOT08] S. Bravyi, D. P. DiVincenzo, R. Oliveira, and B. M. Terhal. "The Complexity of Stoquastic Local Hamiltonian Problems". In: *Quantum Information & Computation* 8.5 (2008), pp. 361–385 (cit. on p. 2).

[BFM23] R. Bassirian, B. Fefferman, and K. Marwaha. "On the Power of Nonstandard Quantum Oracles". In: *Proceedings of the 18th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC)*. 2023, 11:1–11:25 (cit. on p. 2).

[BGH+26] J. Bostanci, S. Grewal, J. Haferkamp, A. Huang, Y. Hwang, A. Natarajan, and C. Nirkhe. *A quantum oracle separation between QMA(2) and QMA*. arXiv:2609.02865 [quant-ph]. 2026 (cit. on p. 3).

[BHNZ26] J. Bostanci, J. Haferkamp, C. Nirkhe, and M. Zhandry. "Separating QMA from QCMA with a Classical Oracle". In: *Proceedings of the 58th Symposium on Theory of Computing (STOC)*. 2026, pp. 595–606 (cit. on pp. 1, 2).

[BHV26] J. Bostanci, A. Huang, and V. Vaikuntanathan. *Separating Quantum and Classical Advice with Good Codes*. arXiv:2602.09385 [quant-ph]. 2026 (cit. on p. 2).

[BK24] S. Ben-David and S. Kundu. "Oracle Separation of QMA and QCMA with Bounded Adaptivity". In: *Proceedings of the 51st International Colloquium on Automata, Languages, and Programming (ICALP)*. 2024, 21:1–21:18 (cit. on p. 2).

[Bra15] S. Bravyi. "Monte Carlo Simulation of Stoquastic Hamiltonians". In: *Quantum Information & Computation* 15.13-14 (2015), pp. 1122–1140 (cit. on p. 2).

[BT10] S. Bravyi and B. Terhal. "Complexity of Stoquastic Frustration-Free Hamiltonians". In: *SIAM Journal on Computing* 39.4 (2010), pp. 1462–1485 (cit. on p. 2).

[CCD+03] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman. "Exponential Algorithmic Speedup by a Quantum Walk". In: *Proceedings of the 35th Symposium on Theory of Computing (STOC)*. 2003, pp. 59–68 (cit. on pp. 1, 3, 5, 11, 14).

[CDG18] S. Coretti, Y. Dodis, and S. Guo. "Non-Uniform Bounds in the Random-Permutation, Ideal-Cipher, and Generic-Group Models". In: *Proceedings of the 38th International Cryptology Conference (CRYPTO)*. 2018, pp. 693–721 (cit. on pp. 6, 15).

[CDGS18] S. Coretti, Y. Dodis, S. Guo, and J. Steinberger. "Random Oracles and Non-uniformity". In: *Proceedings of the 37th International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*. 2018, pp. 227–258 (cit. on p. 15).

[CM16] T. Cubitt and A. Montanaro. "Complexity Classification of Local Hamiltonian Problems". In: *SIAM Journal on Computing* 45.2 (2016), pp. 268–316 (cit. on p. 1).

[CMM09] M. Charikar, K. Makarychev, and Y. Makarychev. "Integrality Gaps for Sherali-Adams Relaxations". In: *Proceedings of the 41st Symposium on Theory of Computing (STOC)*. 2009, pp. 283–292 (cit. on pp. 3, 8).

[Dah14] X. Dahan. "Regular Graphs of Large Girth and Arbitrary Degree". In: *Combinatorica* 34.4 (2014), pp. 407–426 (cit. on p. 10).

[Deu85] D. E. Deutsch. "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer". In: *Proceedings of the Royal Society of London Series A* 400.1818 (1985), pp. 97–117 (cit. on p. 2).

[FK18] B. Fefferman and S. Kimmel. "Quantum vs. Classical Proofs and Subset Verification". In: *Proceedings of the 43rd Symposium on Mathematical Foundations of Computer Science (MFCS)*. 2018, 22:1–22:23 (cit. on pp. 2, 20, 21).

[Gha24] S. Gharibian. "Guest Column: The 7 Faces of Quantum NP". In: *SIGACT News* 54.4 (2024), pp. 54–91 (cit. on p. 1).

[GHV21] A. Gilyén, M. B. Hastings, and U. Vazirani. "(Sub)Exponential Advantage of Adiabatic Quantum Computation with No Sign Problem". In: *Proceedings of the 53rd Symposium on Theory of Computing (STOC)*. 2021, pp. 1357–1369 (cit. on pp. 1–3, 6, 10, 14).

[GMT26] S. Grewal, S. Mutreja, and K. Tomer. *A classical oracle separation between QMA and QMA(2)*. arXiv:2610.06760 [quant-ph]. 2026 (cit. on p. 3).

[GNT25] F. Le Gall, H. Nishimura, and D. Thakkar. "Group Order is in QCMA". In: *Proceedings of the 66th Symposium on Foundations of Computer Science (FOCS)*. 2025, pp. 1128–1143 (cit. on p. 1).

[GR26] A. B. Grilo and M. Rozos. *The Complexity of Stoquastic Sparse Hamiltonians*. arXiv:2605.02845 [cs.CC]. 2026 (cit. on pp. 1, 5, 9).

[Has21] M. B. Hastings. "The Power of Adiabatic Quantum Computation with No Sign Problem". In: *Quantum* 5 (2021), p. 597 (cit. on pp. 3, 10).

[Hay06] T. Hayes. "A Simple Condition Implying Rapid Mixing of Single-site Dynamics on Spin Systems". In: *Proceedings of the 47th Symposium on Foundations of Computer Science (FOCS)*. 2006, pp. 39–46 (cit. on p. 8).

[HBS26] Y. Hamoudi, Y. Le Borgne, and S. T. Sridhara. *Dequantization Barriers for Guided Stoquastic Hamiltonians*. arXiv:2602.23183 [quant-ph]. 2026 (cit. on pp. 1–3, 6, 10, 13, 14, 18, 21).

[JGL10] S. P. Jordan, D. Gosset, and P. J. Love. "Quantum-Merlin-Arthur–Complete Problems for Stoquastic Hamiltonians and Markov Matrices". In: *Physical Review A* 81 (2010), p. 032331 (cit. on p. 3).

[LLPY24] X. Li, Q. Liu, A. Pelecanos, and T. Yamakawa. "Classical vs Quantum Advice and Proofs Under Classically-Accessible Oracle". In: *Proceedings of the 15th Innovations in Theoretical Computer Science Conference (ITCS)*. 2024, 72:1–72:19 (cit. on p. 2).

[LMY25] J. Liu, S. Mutreja, and H. Yuen. "QMA vs QCMA and Pseudorandomness". In: *Proceedings of the 57th Symposium on Theory of Computing (STOC)*. 2025, pp. 1520–1531 (cit. on pp. 2, 3).

[LPS88] A. Lubotzky, R. Phillips, and P. Sarnak. "Ramanujan Graphs". In: *Combinatorica* 8.3 (1988), pp. 261–277 (cit. on p. 10).

[Lut11] A. Lutomirski. *Component Mixers and a Hardness Result for Counterfeiting Quantum Money*. arXiv:1107.0321 [quant-ph]. 2011 (cit. on p. 2).

[LW70] D. R. Lick and A. T. White. "k-Degenerate Graphs". In: *Canadian Journal of Mathematics* 22.5 (1970), pp. 1082–1096 (cit. on p. 7).

[MPR26] D. Miloschewsky, S. Podder, and D. Rudolph. "En Route to a Standard QMA_1 vs. QCMA Oracle Separation". In: *Proceedings of the 51st Symposium on Mathematical Foundations of Computer Science (MFCS)*. 2026, 32:1–32:19 (cit. on pp. 3, 21).

[NN24] A. Natarajan and C. Nirkhe. "A Distribution Testing Oracle Separation between QMA and QCMA". In: *Quantum* 8 (2024), p. 1377 (cit. on pp. 2, 3, 6).

[Wat00] J. Watrous. "Succinct Quantum Proofs for Properties of Finite Groups". In: *Proceedings of the 41st Symposium on Foundations of Computer Science (FOCS)*. 2000, pp. 537–546 (cit. on p. 1).

[Zha25] M. Zhandry. "Toward Separating QMA from QCMA with a Classical Oracle". In: *Proceedings of the 16th Innovations in Theoretical Computer Science Conference (ITCS)*. 2025, 95:1–95:19 (cit. on pp. 6, 17).

A An in-place oracle separation for StoqMA

In the Preimage Checking problem [FK18], an unknown set $S \subseteq [N^2]$ has size $|S| = N$. Writing S_{even} for its even elements, the promise is YES : $|S_{\text{even}}| = 2N/3$, NO : $|S_{\text{even}}| = N/3$. The in-place oracle applies $\mathcal{O}_\pi|x\rangle = |\pi(x)\rangle$ for a permutation π satisfying $\pi(S) = [N]$.

Proposition A.1. *The Preimage Checking problem has a one-query stoquastic verifier with completeness at least* $5/6$ *and soundness at most* $2/3$.

Proof. The witness is assumed to be a superposition over $[N^2]$, written $|\psi\rangle = \sum_{x\in[N^2]} \beta_x |x\rangle$. The verifier computes the parity into a fresh qubit and applies the oracle to the original register, obtaining $\sum_x \beta_x |\pi(x)\rangle|x \bmod 2\rangle$. It then performs a SWAP test with the state $(\frac{1}{\sqrt{N}} \sum_{y\in[N]} |y\rangle)|0\rangle$.

The acceptance probability is $p_{\text{acc}} = \frac{1}{2} + \frac{1}{2N} |\sum_{x\in S_{\text{even}}} \beta_x|^2 \leq \frac{1}{2} + \frac{|S_{\text{even}}|}{2N}$, by the Cauchy–Schwarz inequality. The uniform witness $|S_{\text{even}}\rangle$ attains equality in the YES case. The promised cardinalities give completeness at least $5/6$ and soundness at most $2/3$. □

Fefferman and Kimmel [FK18] proved a QCMA lower bound for Preimage Checking in a randomized oracle model in which each query uses a fresh uniformly random permutation π satisfying $\pi(S) = [N]$. This per-query randomness was recently removed for a closely related preimage-parity problem [MPR26], yielding $\mathsf{QMA}_1^{\mathcal{O}} \not\subseteq \mathsf{QCMA}^{\mathcal{O}}$ for a fixed in-place permutation oracle $\mathcal{O}$. The verifier above also applies to the promise in [MPR26], with completeness 1 and soundness at most $3/4$. Combined with the QCMA lower bound established in that work, this yields $\mathsf{StoqMA}_1^{\mathcal{O}} \not\subseteq \mathsf{QCMA}^{\mathcal{O}}$.

This result does not contradict the (unrelativized) containment $\mathsf{StoqMA}_1 \subseteq \mathsf{MA}$ shown in [AGL25]. The proof of that containment requires inverting the verifier's circuit, which is generally not possible with an in-place oracle. This separation also says nothing about StoqMA vs. MA relative to classical (XOR) oracles.

B An NP protocol for the earlier graph problem

The graph-testing problem underlying the lower bound in our previous work [HBS26] is to distinguish two graphs $\mathcal{G}_n$ and $\mathcal{F}_n$. The graph $\mathcal{G}_n$ is constructed as G_n in the present paper. The graph $\mathcal{F}_n$ uses a base graph of degree $n - \sqrt{n}$ instead of n, with additional tree decorations ensuring that both graphs have the same degree at every non-isolated, non-leaf vertex. We show that this problem lies in NP, although [HBS26] proves it hard for classical algorithms given samples from a guiding state.

Proposition B.1. *The $\mathcal{G}_n$ (YES) versus $\mathcal{F}_n$ (NO) problem defined in [HBS26] is in* NP.

We use the following general fact: in any graph of minimum degree n, some vertex is incident to at least $n-2$ distinct edges lying on short cycles.

Lemma B.2. *Let E be a graph on N vertices with minimum degree at least $n \geq 2$. Some vertex is incident to at least $n-2$ distinct edges, each lying on a cycle of length at most $2\lceil \log N \rceil + 2$.*

Proof. Set $k = \lceil \log N \rceil + 1$. Construct a greedy spanner J: starting with no edges, process each edge of E, adding it only when its endpoints currently have distance greater than $2k-1$. Then J has girth greater than $2k$. If every vertex of J had degree at least three, a breadth-first tree of depth k would contain at least $1 + 3(2^k - 1) > N$ vertices, since the girth bound prevents collisions. Thus some vertex v has degree at most two in J. At least $n-2$ edges incident to v were omitted, each with an alternative path of length at most $2k-1$, giving the claimed cycles. □

Proof sketch of Proposition B.1. Let N be the number of vertices in the base graph of $\mathcal{G}_n$. By Lemma B.2, this base graph has a vertex incident to $n-2$ distinct edges, each lying on a cycle of length $O(\log N)$. The prover supplies this vertex, these edges, and their cycles; the verifier checks the certificate using adjacency queries. In $\mathcal{F}_n$, each vertex is incident to at most $n - \sqrt{n} < n - 2$ edges lying on cycles, proving soundness. □