(Sub)Exponential Advantage of Adiabatic Quantum Computation with No Sign Problem

András Gilyén
Institute of Quantum Inf. and Matter
California Institute of Technology
Pasadena, California, USA
agilyen@caltech.edu

Matthew B. Hastings
Microsoft Quantum and
Microsoft Research
Redmond, Washington, USA
mahastin@microsoft.com

Umesh Vazirani
Department of Electrical Engineering
and Computer Sciences, UC Berkeley
Berkeley, California, USA
vazirani@eecs.berkeley.edu

ABSTRACT

We demonstrate the possibility of (sub)exponential quantum speed-up via a quantum algorithm that follows an adiabatic path of a gapped Hamiltonian with no sign problem. The Hamiltonian that exhibits this speed-up comes from the adjacency matrix of an undirected graph whose vertices are labeled by n-bit strings, and we can view the adiabatic evolution as an efficient $O(\mathrm{poly}(n))$-time quantum algorithm for finding a specific "EXIT" vertex in the graph given the "ENTRANCE" vertex. On the other hand we show that if the graph is given via an adjacency-list oracle, there is no classical algorithm that finds the "EXIT" with probability greater than $\exp(-n^\delta)$ using at most $\exp(n^\delta)$ queries for $\delta = \frac{1}{5} - o(1)$. Our construction of the graph is somewhat similar to the "welded-trees" construction of Childs et al., but uses additional ideas of Hastings for achieving a spectral gap and a short adiabatic path.

CCS CONCEPTS

• **Theory of computation → Quantum complexity theory**; **Quantum computation theory**.

KEYWORDS

adiabatic quantum computation, sign-problem-free, sparse Hamiltonian, stoquastic, welded-trees, glued-trees, quantum walk

ACM Reference Format:
András Gilyén, Matthew B. Hastings, and Umesh Vazirani. 2021. (Sub)-Exponential Advantage of Adiabatic Quantum Computation with No Sign Problem. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC '21), June 21–25, 2021, Virtual, Italy*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3406325.3451060

1 INTRODUCTION

Adiabatic quantum computing [14] is an interesting model of computation that is formulated directly in terms of Hamiltonians, the quantum analog of constraint satisfaction problems (CSPs). The computation starts in the known ground state of an initial Hamiltonian, and slowly (adiabatically) transforms the acting Hamiltonian into a final Hamiltonian whose ground state encapsulates the answer to the computational problem in question. The final state of the computation is guaranteed, by the quantum adiabatic theorem, to have high overlap with the desired ground state, as long as the running time of the adiabatic evolution is polynomially large in the inverse of the smallest spectral gap of any Hamiltonian along the adiabatic path [3]. This model has been intensely studied, not only because of its inherent interest, but also because it is the zero-temperature limit of quantum annealing.

In general, adiabatic quantum computing is known to be equivalent to standard circuit-based quantum computing [1]. However, a very interesting question is what is the power of adiabatic quantum computing where all Hamiltonians were "stoquastic", i.e., restricted to not having a sign problem. What this means is that in some basis all off-diagonal terms of H are non-positive. Adiabatic quantum computing with no sign problem includes the most natural case where the final Hamiltonian is diagonal, and represents the objective function to be optimized, and the initial Hamiltonian consists of Pauli X operators acting on each qubit, with ground state the uniform superposition on all the n-bit strings. This question was also motivated by understanding the computational limits of the quantum annealers implemented by the company D-Wave, where all the Hamiltonians were stoquastic.

Bravyi and Terhal [8] showed that for frustration-free Hamiltonians without a sign problem, computing the ground state is classically tractable, thereby raising the question of whether this was true for general Hamiltonians without a sign problem. Indeed, a stronger conjecture was that quantum Monte-Carlo, a widely used heuristic in computational condensed matter physics, already provided a technique for an efficient classical simulation. This latter possibility was ruled out by a result of Hastings and Freedman [20], who showed the existence of topological obstructions to the convergence of quantum Monte Carlo on such problems.

The question of classical tractability for general Hamiltonians with no sign problem was open until it was addressed in a recent breakthrough by Hastings [19], who showed a quasipolynomial oracle separation between classical algorithms and adiabatic quantum computation with no sign problem. Subsequently, Gilyén and Vazirani [18] extended and simplified Hastings' result. They showed that there is a (sub)exponential oracle separation of the form 2^{n^δ} between classical algorithms and adiabatic quantum computation with no sign problem. In Section 2 we present the simplified Hamiltonian construction from the latter result, which comes from a sparse graph with a fairly transparent structure. The separation is showed for a problem concerning the underlying graph, which contains two special vertices: ENTRANCE and EXIT. Given the ENTRANCE vertex and oracle access to the adjacency list of the graph, every classical randomized algorithm needs (sub)exponential time

STOC '21, June 21–25, 2021, Virtual, Italy

ACM ISBN 978-1-4503-8053-9/21/06.
https://doi.org/10.1145/3406325.3451060

to find the EXIT vertex on average. However, a simple quantum walk finds the EXIT vertex in polynomial time, and likewise so does a simple adiabatic algorithm which carries out a straight line interpolation between the initial and final Hamiltonian. In Section 3 we also review the key ideas behind Hastings' original construction, and point out some of its aspects that are potentially useful for further extensions of the results.

The simple construction presented in Section 2 highlights the similarities and differences with the well-known "welded-trees" graph (see Fig. 2 and Section 4) which is the basis of the first known example of exponential speedup by quantum walks [10]. The welded-trees graph is not suitable for adiabatic computation, since the ground state has exponentially small support on the roots of the two trees (the ENTRANCE and EXIT vertices).[1] To see this, notice that a quantum walk on the welded trees may be viewed as walking on the symmetric subspace of each level of the trees, and the Hamiltonian effectively reduces to a path of length 2depth + 1. This path has uniform edge weights, except at the middle edge, which has $\sqrt{2}$-times bigger weight. This makes the largest eigenvector of the path graph decay exponentially from the middle towards the two ends. The starting point of our construction is the simple observation that equalizing the edge weights in the level graph on symmetric subspaces has the effect of fixing the exponential decay problem. On the other hand, this necessarily makes the underlying graph non-regular, potentially enabling classical algorithms to detect the structure of the graph [5, 12] and ultimately destroying the lower bound of [10] that heavily builds on the regularity of the graph. In order to restore the classical hardness result we use the approach of Hastings [19], who ensures hardness by "decorating" the graph by means of attaching a cleverly shaped forest to every vertex. Another feature that can be seen very concretely in our simplified construction is the role of the ℓ_2 versus ℓ_1 normalization difference in the behavior of the quantum vs. classical walk.

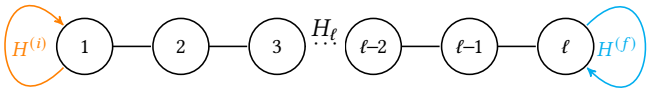
2 MAIN RESULTS

We follow the general framework from [19]. The main idea there was to start with a graph that the adiabatic algorithm can traverse efficiently, and to hide that graph within a larger graph as follows: attach a number of trees to each vertex of the original graph, so that the attached trees form the bulk of the new graph. Now, the intuition is that the behavior of a quantum walk versus classical walk on the attached trees would be governed by their $\ell_2 \to \ell_2$ (i.e., spectral) norm versus $\ell_1 \to \ell_1$ norm respectively, and the latter is quadratically larger. As a result the attached trees only negligibly affect the ℓ_2-weight distribution of the ground state (and so quantum algorithms only suffer from a minor perturbation), while they dramatically shift the ℓ_1-weight distribution of the ground state away from the original graph. Intuitively speaking this enables the trees to lure away classical random walks from the original graph, so that they get lost in the attached "camouflage trees" with very high probability. Furthermore, by choosing the trees to have a confusing enough shape, one can ensure that there is no classical algorithm that can avoid getting drawn into the "camouflage trees". Therefore, classical algorithms fail to quickly explore the original graph, and in our case this ultimately leads to their inability of efficiently finding the EXIT vertex.

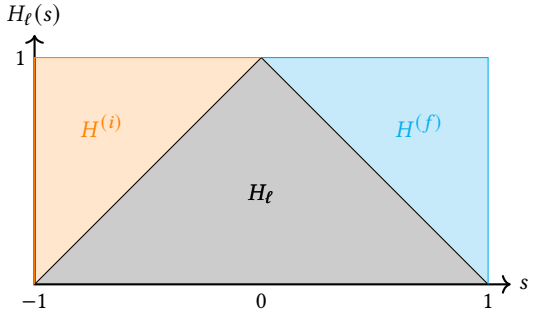
Classical hardness is achieved by constructing hard-to-navigate trees with a fractal-like structure that are built in a recursive manner, via a sequence of so-called "decorations". Hastings' original decorations [19] rapidly blew up the degrees of the vertices in the graph, thereby limited the number of subsequent decoration rounds to logarithmic, permitting only a quasipolynomial separation. We on the other hand use a modified decoration sequence allowing polynomially many rounds of decoration, and ultimately leading to a (sub)exponential separation.

2.1 The Basic Adiabatic Path at the Core of Our Quantum Algorithm

We begin with a simple underlying problem of starting at one endpoint of a path on ℓ vertices, and finding the other endpoint of the path.[2] A simple adiabatic algorithm for this problem is specified as follows: Let A_ℓ denote the adjacency matrix of the path $\langle k|A_\ell|k+1\rangle = \langle k+1|A_\ell|k\rangle = 1$, and let the corresponding Hamiltonian be $H_\ell := -A_\ell$, while $H^{(i)} := -|1\rangle\langle 1|$ and $H^{(f)} := -|\ell\rangle\langle\ell|$.



Consider the simple adiabatic path $H_\ell(s)$ that first interpolates between H_i and H_ℓ, then between H_ℓ and H_f, so that $H_\ell(s) := (1+s)H_\ell - sH^{(i)}$ for $s \in [-1, 0]$ and $H_\ell(s) := (1-s)H_\ell + sH^{(f)}$ for $s \in [0, 1]$.



If one moves slowly enough along this adiabatic path [3, 14], the quantum evolution maps "ENTRANCE":= $|1\rangle$ – the *initial* ground state of $H^{(i)}$ to "EXIT":= $|\ell\rangle$ – the *final* ground state of $H^{(f)}$, since $H(s)$ has a gap of size $\Omega(\frac{1}{\ell^2})$ for all $s \in [-1, 1]$, see Appendix A. Note that if one wishes to use only a simple "straight" adiabatic path, and stops at $s = 0$, a measurement in the computational basis still reveals the state $|\ell\rangle$ with probability at least $\Omega(\ell^{-3})$ since the ground state of H_ℓ has $\Omega(\ell^{-\frac{3}{2}})$ overlap with $|\ell\rangle$, cf. Appendix A.[3]

[1] Nevertheless, if one allows the quantum evolution to also involve higher-energy states, then a corresponding quantum annealing procedure is known to exhibit an exponential advantage [23].

[2] We work with undirected and unweighted graphs, but for simplicity allow parallel edges and self-loops. One can think about parallel edges as simple integer edge weights, since they are represented in the same way in the adjacency matrix. Ultimately we will only use self-loops at the two distinguished vertices "ENTRANCE" and "EXIT".

[3] Alternatively we could increase the success probability to $\Omega(1)$ by accepting any vertex in $\{\ell/2, \ell/2+1, \ldots, \ell\}$, thereby effectively defining multiple exits. This task can be made classically hard as well, similarly to the single EXIT scenario.

2.2 Making the Task of Finding EXIT Classically Hard

In order to prove classical hardness we will hide the EXIT vertex in a larger graph – the new graph will be chosen to allow the quantum adiabatic algorithm to still be efficient, while making the task of any classical algorithm very difficult. The id's of the vertices will be chosen randomly in order to remove any non-structural hints about the whereabouts of the EXIT vertex, and the graph will be specified by oracle access to its adjacency list,[4] together with the ENTRANCE vertex – one of the two vertices with a self-loop.[5] The task is to find the EXIT vertex – the other vertex with a self-loop attached. The graph will have polynomially bounded maximal vertex degree, so the adiabatic evolution can be efficiently performed by a quantum computer using (time-dependent) sparse Hamiltonian simulation techniques [7].

In order to make the task of finding EXIT classically hard we "blow-up" the path graph of length ℓ via two main modifications, that we call *obfuscation* and *decoration*.

Definition 1 (Obfuscation of a path of length ℓ). *We replace every vertex that has distance $d \in [k]$ from terminal vertices $\{ENTRANCE, EXIT\}$ by a cluster C of m^{2d} vertices and call these the* funnel *vertices, and replace the other middle vertices (that have distance $d > k$) by a cluster of m^{2k} vertices, and call those the* tunnel *vertices. Then we add edges between clusters C_j and C_{j+1} corresponding to neighbor vertices j and $j+1$ in P_ℓ, so that we build an m^2-ary tree (with the terminal vertices as roots) on the funnel vertices. Between clusters that correspond to vertices with distance $d \geq k$ we add edges along m random matchings. Additionally, in order to preserve spectral properties we add $2m$ self-loops to the ENTRANCE and the EXIT vertices, and an independently chosen random uniform degree-$(2 \cdot m)$ expander graph on each cluster $C_j : j \in [\ell] \setminus \{1, \ell\}$, as in Appendix C.*[6]

Note that the graph on the tunnel vertices is $4m$-regular. The decoration construction, described next, will hang m trees from each vertex of the obfuscated graph, each of them being a complete $(5m-1)$-ary tree (by a complete tree we mean a tree for which every node has the same number of children except at the bottom layer, which is at a fixed depth) on its first $\mathrm{poly}(m)$ layers, and then having gradually less children at later layers. The construction is motivated by its effect on the tunnel — it will increase the degree of each tunnel vertices to $5m$. Thus, the resulting graph will still be $5m$-regular on the original tunnel vertices, as well as on the surrounding vertices in the first $\mathrm{poly}(m)$ layers of the added trees. This will make it very difficult for any classical algorithm to distinguishing edges between the tunnel vertices from edges that lead away from the tunnel, thereby making the traversal of the tunnel very slow.

If we would everywhere add a complete $(5m-1)$-ary tree of depth d, then the decoration trees would be easy to detect: after traversing an edge perform a non-backtracking walk of length d, if one arrives at a leaf it means that the traversed edge is hanging a decoration tree. Since we cannot add more than $\exp(\mathrm{poly}(m))$ new vertices, the trees must have a bounded depth. Therefore, in order to circumvent such detection algorithms we should construct trees where the distribution of the lengths before a non-backtracking random walk hits the bottom of a tree looks approximately self-similar, i.e., after going one level deeper in the tree the expected distribution should not change by more than a (sub)exponentially small amount. In order to achieve this, the decoration is carried out in $r = m^\delta$ rounds, giving the attached trees a complex fractal-like structure.

Definition 2 (Decoration). *Let $G = (V, E)$ be a graph. A level-j decoration graph G_j is obtained from G by "decorating" every vertex $v \in V$ by attaching $m^{(1-\delta)}$ new trees via an edge to their root. The attached trees are complete $(5m - (j-1)m^{(1-\delta)} - 1)$-ary trees with depth $jm^{(3\delta+o(1))}$. We define $G^{(r)}$ as the r-round decoration of G, which is obtained from G by applying a level-r decoration, then subsequently level-$(r-1)$, level-$(r-2)$, ..., level-1 decorations.*

The above modified definition of decoration is the key to our improved separation result. Hastings [19] added an increasing number of decoration trees in every round so that decoration doubled the maximal degree of the graph in each round. The rapid growth of the vertex degrees prohibited applying more than logarithmically many rounds of decoration, which ultimately limited the separation to being at most quasipolynomial. In contrast, we only add $m^{1-\delta}$ decoration trees in each round, which enables us applying m^δ rounds of decoration, ultimately resulting in a (sub)exponential classical lower bound. In order to keep the desired increase in classical hardness despite using fewer decoration trees, we make them slightly deeper.

The obfuscation construction is motivated by the following consideration: from the viewpoint of the quantum adiabatic algorithm, the obfuscated graph may be viewed as a path on the clusters from the ENTRANCE to the EXIT, with a weight of m on each edge (the expander graph on each cluster helps enforce this structure during the adiabatic evolution). This means that for all practical purposes, the adiabatic quantum algorithm does not notice the obfuscation. On the other hand, for any classical algorithm, the obfuscated graph presents a challenge, because the graph looks locally tree-like, and the underlying path structure is effectively hidden.

Obfuscation is also motivated by the earlier work of Hastings and Freedman [20] that studied obstructions to quantum Monte Carlo algorithms. One way to interpret their findings is that the presence of long cycles in the graph, that are hard to "approximate" by shorter cycles[7] but nevertheless make important spectral contributions, can prevent quantum Monte Carlo algorithms from fast convergence.

[4] Our graph has $N = O(\exp(\mathrm{poly}(m)))$ vertices, each having at most $d = \max(5m, m^2 + 3m + 1)$ neighbors. Let use the notation $[N] := \{1, 2, 3, \ldots, N\}$. The classical adjacency-list oracle $O : [N] \times [d] \mapsto [N] \cup \{\star\}$ can be queried with the id of a vertex and a number k, and as a response tells the k-th neighbor of the vertex with the given id (the neighbors are sorted arbitrarily). If the vertex has less than k neighbors (with multiplicity), then the oracle outputs $\star$ as a response. We assume that the corresponding *reversible* quantum oracle acts as $|i\rangle|k\rangle|0\rangle \to |i\rangle|O(i,k)\rangle|n(i,k)\rangle$, where $n(i,k)$ is the number of $h \in [k-1]$ such that $O(i,h) = O(i,k)$. This is a so-called *in-place* adjacency-list oracle [6, 17], which can save us a $\mathrm{poly}(m)$ factor in the number of queries used by our quantum algorithms.

[5] There are two vertices with a self-loop, and we know the ENTRANCE vertex, so we can simulate both the initial and the final Hamiltonians by using the adjacency-list oracle.

[6] We use random expander graphs as in Definition 7, but condition on their spectral gap being at least m. For large enough m the effect of conditioning is negligible as shown by Corollary 9.

[7] By approximating a cycle with shorter cycles we mean a sequence of shorter cycles in which subsequent cycles differ only in a few edges. Inapproximable cycles could emerge from the topological structure of the graph, for example on a (discretized) torus it is impossible to approximate a cycle wrapping around its "hole" by shorter cycles.

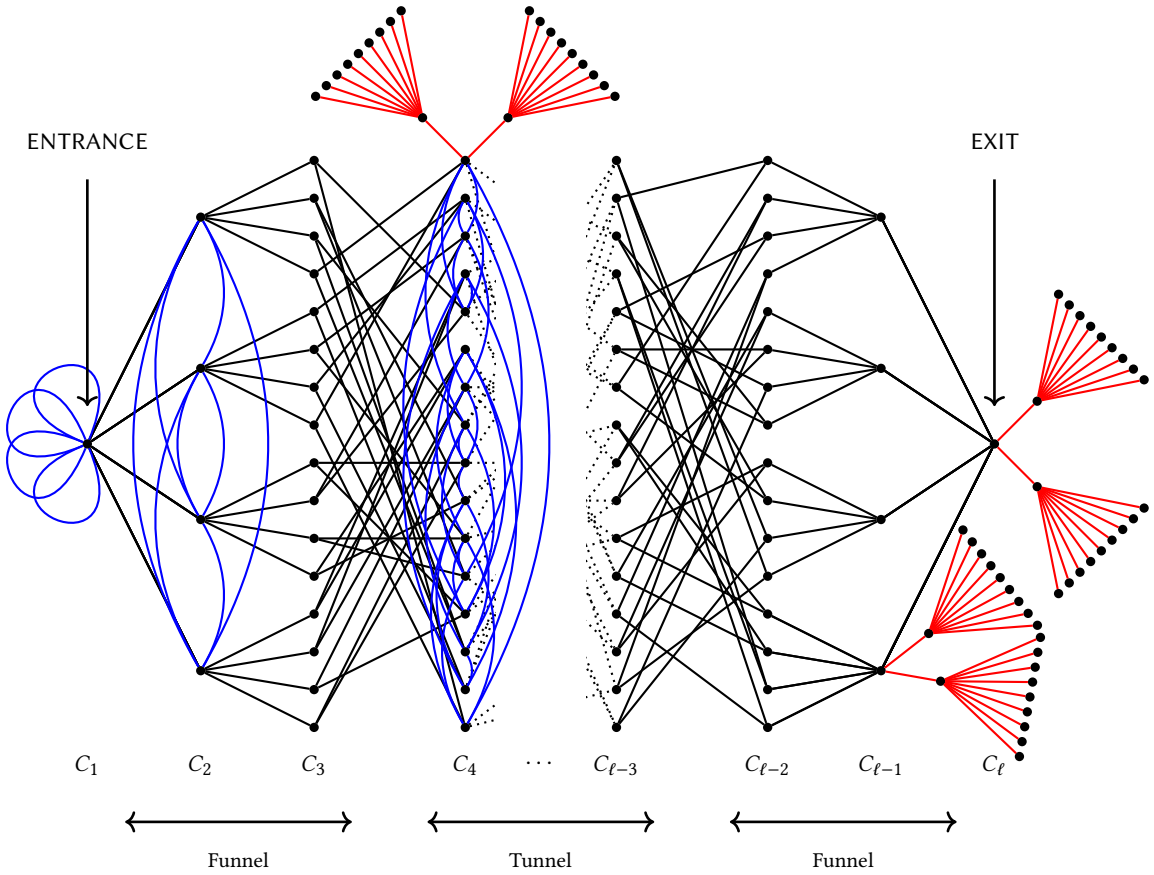


Figure 1: An illustration of a random graph that we construct for the separation. The parameters are $m = 2, k = 2$ and $\delta = 0$ (note that these parameters are non-representative, but it is hard to draw an example with bigger parameters). The edges constructed during obfuscation are in black, the expander graphs on the clusters are in blue, and the decoration trees are in red. For clarity of the picture we only included the expander edges on clusters C_1, C_2, and C_4, but they should be added on all clusters in our construction. Similarly, we only included decoration trees on three vertices (and limited their depth to 1) due to space constraints, but they should be added to every vertex.

The obfuscation step in fact introduces many long cycles that are important from a spectral perspective, but classical algorithms have a hard time recognizing or even finding many of such cycles (see the following two paragraphs) presenting an obstacle to classical algorithms.

However, a random walk with $\Omega(\ell^2)$ steps can still traverse the obfuscated tunnel with high probability, enabling the efficient discovery of the EXIT by a randomized algorithm. The decoration of the obfuscated graph with trees is designed to make the graph even more difficult to navigate for any classical algorithm. Intuitively speaking the trees have a fractal-like structure with $\mathrm{poly}(m)$ levels of self-similarity, and each level of self-similarity will make the graph twice as difficult to navigate – we prove this rigorously following an argument by Hastings [19]. At the same time, the decoration has only an insignificant effect on the adiabatic evolution, since the spectrum of a tree of degree d is bounded by $2\sqrt{d-1}$, and therefore the decoration results in only a slight $O(\sqrt{m})$-magnitude perturbation, which is too small to close the gap in the spectrum throughout the adiabatic path.

We start by stating more quantitatively the intuition that the obfuscated graph on the middle clusters is locally tree-like. After the obfuscation step the tunnel vertices all have degree $4m$. The main observation is that for a random walk, or in fact any classical algorithm that makes only (sub)exponentially many queries, the tunnel section of the graph appears locally like a tree graph of uniform degree $4m$. Indeed, all matchings between the different clusters and all expander graphs within the clusters are chosen randomly, so after q queries the probability that a new edge query will close an inner cycle (i.e., a cycle that only contains edges within the tunnel) has probability $O(q/m^k)$, so by the union bound after q queries the total probability of discovering some inner cycle is at most $O(q^2/m^k)$.

For simplicity let us assume that ℓ is odd, and set $l := (\ell - 1)/2$. Consider any classical algorithm that starts at a vertex, v, in the middle $((l+1)$-st) cluster; it follows that if the classical

algorithm makes at most $m^{\frac{k}{3}}$ queries, then up to $O(m^{-\frac{k}{3}})$ error we can assume that the graph looks like a regular tree up to depth $l-k$. We will argue below that the decoration makes it (sub)exponentially difficult to follow a path of length $l-k$ in the original (undecorated) graph, because local exploration of the graph will be drawn into the decoration trees which are hard to recognize.[8]

It is helpful to understand the case of a single level decoration with depth-d trees. Intuitively, if starting from vertex v, the middle section of the obfuscated graph (the tunnel) were actually a tree (instead of just looking tree-like), and if the classical algorithm was guaranteed to never make it down to a leaf of any decoration tree (with depth d), then we could argue as follows: from the viewpoint of the algorithm it is exploring a regular $(4m+m^{1-\delta})$-ary tree, and we are asking what is the chance that it finds a vertex that is at distance $l-k$ from v. In order to find such a distant vertex the algorithm has to explore at least one path of vertices of length $l-k$. The requirement of not encountering a leaf of a decoration tree forces the algorithm to stay within a (randomly embedded) subtree of degree $4m$ up to depth $l-k-d$ (that is the difference between how deep the exploration has to go and the depth of the decoration trees). Due to the random labeling of the tree this clearly fails with (sub)exponentially high probability at least $1-(1-\Theta(m^{-\delta}))^{l-k-d}$. Now, of course, the tunnel is not actually a tree. But notice that the above intuition can still be made to work as follows: perform a breadth-first search from the start vertex v, and every time a vertex is encountered, make a new copy of it – so that the number of vertices at depth h is exactly $4m^h$. Now we can argue that from the viewpoint of the classical algorithm, the vertices at depth $l-k$ are symmetric under permutation, except when the algorithm discovers a cycle.[9] (But as we argued above the chance of that is (sub)exponentially small.) Noting that attaching an isomorphic collection of graphs to every vertex does not change the above argument, so we get the following statement:

Lemma 3 (Hardness of avoiding the exploration of decoration trees). *Suppose that G is a rooted graph with its root r having degree k, and all vertices up to distance D have degree $k+1$. Let G' be the graph where each vertex v of G gets h distinct complete $(k+h)$-ary trees of depth d attached via and edge to their root. Suppose we have access to a uniformly randomly labeled version of $G'^{(j)}$, and we can perform local exploration starting from the root r. Then the probability that we don't find a cycle, neither discover a leaf of any attached tree in G', but discover a vertex v in G' at distance D form r has probability at most $(k/(k+h))^{D-d}$ irrespective of the number of exploration steps.*

Since upon traversing the tunnel we need to go through at least one middle vertex, the above argument assures that we need to discover at least one leaf of some level-1 decoration tree, in order to traverse the tunnel. Further levels of the decoration ensure that discovering such a leaf is (sub)exponentially unlikely unless the classical algorithm makes at least (sub)exponentially many queries, as shown by the following inductive lemma. Regarding the inductive structure, note that the definition of decoration makes it possible to view G as an induced subgraph of the level-j decorated graph G_j, and in turn as an induced subgraph of the recursively decorated graph $G^{(j)}$, in particular we have $G_j^{(j-1)} = G^{(j)}$. A we already indicated we will choose $r = m^\delta$ rounds of decoration (for simplicity let us assume that both m^δ and $m^{1-\delta}$ are integers), so that $G^{(r)}$ will look roughly uniform with degree $5m$ at every vertex around the original vertices of G.[10]

Lemma 4 (Cf. [19, Lemma 6]). *Suppose we start form the root of a complete $(5m-jm^{(1-\delta)}-1)$-ary tree T of depth $d := (j+1)m^{(3\delta+o(1))}$ (think of T as a tree attached during a $(j+1)$-level decoration), and we are only allowed to explore its decorated version $T^{(j)}$ "locally", i.e., by only querying neighbors of known vertices. If $j \le m^\delta$, then for any algorithm the probability of reaching a leaf vertex of T using 2^j queries is at most $3^{-(m^\delta-j+1)m^\delta}$.*

Proof. We can prove this by induction on j. For $j=1$ the statement is trivial. The induction step is as follows: suppose that the statement is true for $j-1$. What we prove is that it requires at least 2^j queries to find a vertex that has distance at least d from the root of T in the graph $T^{(j)}$ with very high probability. For this we would need to traverse at least $t := m^{3\delta+o(1)}$ edges of T (t is the increment in the depths of the decoration trees of subsequent levels). There are two cases:

Case 1: the explored vertices include leaves of at most one decoration tree in T_j, or

Case 2: the explored vertices include leaves of at least two decoration trees in T_j.

First we bound the probability of Case 1 happening. By assumption, there is a path in $T^{(j)}$ of length $d := (j+1)m^{(3\delta+o(1))}$ going from root to leaf of T: all vertices on the path are explored. Furthermore, there is at most one level-j decoration tree which has an explored leaf. We bound the probability of Case 1 by case separation:

- If the there is no leaf of $T_j \setminus T$ that is explored or the first explored leaf of $T_j \setminus T$ is at a distance at least $d-t/2$ from the root, then we can apply Lemma 3 to show that this event has probability at most $(1-\Theta(m^{-\delta}))^{\frac{t}{2}}$.
- On the other hand, if there is a single decoration tree with an explored vertex whose root is at some depth at most $t/2$, then there is a path of length at least $d-t/2-1$ starting from a vertex of $v \in T$ such that all vertices along the path are explored, but no other leaf of $T_j \setminus T$ is found. We can once again apply Lemma 3 to bound the probability of this happening by $(1-\Theta(m^{-\delta}))^{\frac{t}{2}-1}$.

By the union bound we get that the probability of Case 1 is at most $2(1-\Theta(m^{-\delta}))^{\frac{t}{2}-1} \le \exp(-m^{-2\delta+o(1)}) \le 3^{-(m^{2\delta})}$ irrespective of how many queries are made.

[8] We can force any classical algorithm to only do local exploration by hiding the graph among exponentially many isolated vertices, so that querying an unseen vertex label will lie outside the graph with exponentially high probability. However, this is probably not needed, as the structure itself shall conceal interesting vertices naturally, as we will see.

[9] The same intuition is expressed slightly more rigorously in [10, Section IV], where the authors say that a q-query classical algorithm on a regular graph can be modelled by a random embedding of a tree of size q.

[10] We could in principle modify the definition of decoration by adding more trees to non-maximal degree vertices, so that the resulting graph will be uniform everywhere, except at the leafs of the decoration trees. This would probably make the graph even harder to navigate for a classical algorithm. We will nevertheless stick with the above definition because it has some aspects that are more convenient for our analysis.

Now we bound the probability of Case 2 happening. If we make at most q queries, then we find at most q root vertices of level-$(j-1)$ trees. For each such tree traversing to the bottom of the decoration tree takes at least $2^{(j-1)}$ queries by induction, with probability at least $1-3^{-(m^\delta-j+2))m^\delta}$. So by the union bound in order to traverse to the bottom of 2 such trees one needs at least 2^j queries with probability at least $1-2^j 3^{-(m^\delta-j+2)m^\delta}$.

By applying the union bound on the distinct events Case 1 and Case 2 we can conclude that by using $2^j \le 2^{m^\delta}$ queries we reach the bottom of the tree T with probability at most $q3^{-(m^\delta-j+2))m^\delta} + 3^{-(m^{2\delta})} < 3^{-(m^\delta-j+1)m^\delta}$. □

Consider the following three events:

Event 1: The algorithm finds a leaf of a top-level decoration tree.
Event 2: The algorithm finds a cycle within the tunnel.
Event 3: Neither of the above two events holds but the algorithm traverses the tunnel.

If the algorithm finds the EXIT vertex by local exploration of the graph starting from the ENTRANCE, then it must traverse the tunnel in particular. Therefore the event of discovering the EXIT is covered by the union of the above three events. Now we bound the probability of each of the above three events, assuming that G is the graph that we get by obfuscating a path of length $\ell := m^{4\delta+o(1)}$, with $k := 3m^\delta$ funnel depth, and the (classical) algorithm makes at most $q = 2^{m^\delta}$ queries.

Since the algorithm explores at most q root vertices of a top-level decoration tree, the probability of finding a leaf of any such decoration tree with q queries is at most $q3^{-m^\delta}$, due to Lemma 4 and the union bound. Therefore, the probability of Event 1 is bounded by $\exp(-\Omega(m^\delta))$.

We already discussed that for any algorithm that makes at most q queries the total probability of finding a cycle within the tunnel is at most $O\left(q^2/m^k\right)$. Therefore, the probability of Event 2 is bounded by $O\left(\exp(-m^\delta)\right)$.

Finally, if the algorithm traverses the tunnel it must reach a vertex v in the tunnel from which it discovers a path of length at least $l-k$. As Lemma 3 shows, following a path of length $l-k$ from a middle vertex without discovering a cycle or a leaf of a top-level decoration is (sub)exponentially unlikely: its probability is bounded by $(1-\Theta(m^{-\delta}))^{l-k-d}$, where $d = m^{(4\delta+o(1))}$ – the depth of the top level decoration trees. Therefore, with the right choice of the $o(1)$ term in the definition of ℓ, we can bound the probability of Event 3 by $\exp(-\Omega(m^{3\delta}))$.

We can conclude using the union bound, that any classical algorithm that uses 2^{m^δ} queries can reach the EXIT vertex in the decorated graph with probability at most $\exp(-\Omega(m^\delta))$. In the next subsection we will see that for preserving the gap we shall choose $\ell = \Theta(m^{\frac{1}{4}})$, so we will ultimately choose $\delta = \frac{1}{16} - o(1)$. Note that the graph $G^{(0)}$ has at most $\ell m^{2k} = O\left(\exp(m^{\delta+o(1)})\right)$ vertices and similarly $|G^{(m^\delta)}| \le |G^{(0)}|\left((5m)^{m^{4\delta+o(1)}}\right)^{m^\delta} = \exp(m^{5\delta+o(1)})$. Setting $n := m^{5\delta+o(1)}$, we can see that the vertex labels have n bits, and any classical algorithm needs at least $\exp(n^{\frac{1}{5}-o(1)})$ queries to find the EXIT with probability greater than $\exp(-n^{\frac{1}{5}-o(1)})$, providing the (sub)exponential classical lower bound we claimed.

2.3 Preserving the Adiabatic Path and Its Main Spectral Properties

The actual adiabatic path that we use will be analogous to the simple path that we used at the beginning: $H(s) := (1+s)H - sH_i'$ for $s \in [-1,0]$ and $H(s) := (1-s)H + sH_f'$ for $s \in [0,1]$, where $H := -A$, $H_i' = -m \cdot |\text{ENTRANCE}\rangle\langle\text{ENTRANCE}|$ and $H_f' = -m \cdot |\text{EXIT}\rangle\langle\text{EXIT}|$.

For the sake of analysis we divide the adjacency matrix $A = A_P + A_E + A_D$ to three parts, the edges corresponding to the original path graph (A_P), the edges that belong to the expander graphs on the clusters (A_E), and the edges coming from the decoration (A_D).

The main idea is that for understanding the adjacency matrix after obfuscation $\tilde{A} := A_P + A_E$ we can focus on the "symmetric" subspace, which is spanned by uniform superpositions $|C_j\rangle := \frac{1}{\sqrt{|C_j|}}\sum_{v\in C_j}|v\rangle$ over the clusters $C_j : j \in [\ell]$. In this subspace the adjacency matrix looks like that of the original path graph A_ℓ just with uniform edge weights m. Expander graphs of uniform degree $2m$ are added on each cluster C_j corresponding to individual vertices $j \in [\ell]$ of the original path graph in order to make this "symmetric" subspace have the lowest energy. This enables perfectly preserving the main properties of the adiabatic path, including a spectral gap of size $\Omega(m/\ell^2)$.

Indeed, let us first focus on the adiabatic path corresponding to $\tilde{A}(s) := -H(s) + (1-|s|)A_D$. A_E contains no edges between clusters, but the edges form an expander with uniform degree $2m$ within each cluster C_j, so that we can block-diagonalize A_E according to the clusters. Clearly, then the largest eigenvalue of A_E is $2m$, and has multiplicity ℓ, while the uniform superpositions $|C_j\rangle : j \in [\ell]$ form an orthonormal basis of the subspace U corresponding to the largest eigenvalue, i.e.,[11]

$$\langle C_i|A_E|C_j\rangle = 2m \cdot \delta_{ij}. \tag{1}$$

With the right choice of expander graphs, the spectral gap becomes large, so that for large enough m the second largest eigenvalue of A_E is at most m, as follows from Theorem 8. Now observe that the ℓ-dimensional subspace U is also invariant under the linear map A_P, and that the matrix of A_P on this subspace is isomorphic[12] to the adjacency matrix A_ℓ of the path of length ℓ with uniform edge-weights m, i.e.,[13]

$$\langle C_i|A_P|C_j\rangle = m \cdot \langle i|A_\ell|j\rangle. \tag{2}$$

Since U is also invariant under $\tilde{A}(s)$, we can see that our analysis of the spectrum of the simple adiabatic path $H'(s)$ directly applies here as well, and so we get that the spectral gap within U has size $\Omega(m/\ell^2)$.

Now we show that the second largest eigenvalue of $\tilde{A}(s)$ comes from the spectrum on U. Let $\bar{U}$ denote the orthogonal complement

[11] Here δ_{ij} stands for the Kronecker-delta.
[12] Note that due to the Perron-Frobenius theorem this representation also shows that $\|A_P\| = m\|A_\ell\| \le 2m$.
[13] This can be seen as follows: if S and T are disjoint subsets of the vertices of a graph G with M edges between S and T, then for $|S\rangle := \frac{1}{\sqrt{|S|}}\sum_{v\in S}|v\rangle$ and $|T\rangle := \frac{1}{\sqrt{|T|}}\sum_{w\in T}|w\rangle$ we have that $\langle S|A_G|T\rangle = \frac{M}{\sqrt{|S|\cdot|T|}}$.

of U, and $\Pi_{\bar{U}}$ denote the orthogonal projection to it. The largest absolute eigenvalue of $\tilde{A}$ on the invariant subspace $\bar{U}$ can be written as $\left\|\Pi_{\bar{U}}\tilde{A}\Pi_{\bar{U}}\right\|$, which can be bounded by $\left\|\Pi_{\bar{U}}A_P\Pi_{\bar{U}}\right\| + \left\|\Pi_{\bar{U}}A_E\Pi_{\bar{U}}\right\| \leq \|A_P\| + m$, therefore the largest eigenvalue of $\tilde{A}(s)$ on $\bar{U}$ is at most $(1-|s|)(\|A_P\|+m) \leq (1-|s|)3m$. At the same time, our analysis in Appendix A shows that the second largest eigenvalue of $A_\ell(\alpha) \geq 1$ for $\ell \geq 5$ (cf. Fig. 2) and thereby the second largest eigenvalue of $\tilde{A}(s)$ on U is at least $(1-|s|)3m$. Thus, we can conclude that the overall spectral gap of $\tilde{A}(s)$ has size $\Omega(m/\ell^2)$.

The second main idea is that the graph of edges added during decoration form a forest of maximum degree $5m$ and thus the corresponding adjacency matrix A_D has spectral norm bounded[14] by $2\sqrt{5m}$. Therefore, as long as $\ell \ll m^{\frac{1}{4}}$, perturbation by the matrix $(|s|-1)A_D$ cannot close[15] the gap of $\tilde{A}(s)$ which has size $\Omega(m/\ell^2)$. Therefore, choosing $\ell = \Theta(m^{\frac{1}{4}})$ appropriately ensures that the adiabatic path $H(s)$ has a spectral gap of size at least $\Omega(m/\ell^2)$ around its ground state energy.

The above two main observations show that the adiabatic path still maps the ENTRANCE vertex to the EXIT vertex. Since the main Hamiltonian comes from the adjacency matrix of an undirected and unweighted graph it has no sign problem.

3 EARLIER CONSTRUCTION

The construction in this paper showing a subexponential separation is based on that of [18]. A precursor to that construction is in [19], which showed only a weaker quasipolynomial separation, namely $\exp(\Theta(\log(n)^2))$. Both [18, 19] use the idea of recursively decorating graphs, but choose different graphs to decorate; indeed, the final adiabatic path in [18] turns out to be simpler than that in [19]. In this section, we review two ideas of [19], in the hopes that these ideas will turn out to be useful in a future further strengthening of the result. This review will necessarily be very brief and sketched.

3.1 Modified Query Model

The first idea is the so-called "modified query model". In this model, the oracle gives less information than in the "original query model" considered before. In the modified query model, if the algorithm follows some nonbacktracking path of queries that forms a cycle (for example, querying a vertex i to get some neighbor j, querying j to get some neighbor k, querying k to get i which is a neighbor of k), then it is impossible to determine that one has returned to the start of the cycle (in this case, i) instead of simply going to some other vertex which happens to have the same neighborhood as i.

[14] This is straightforward to show, for example using the argument of Hastings [19]: a forest of degree at most $d+1$ can be embedded into a uniform tree that has d children at every level for some finite depth t. Due to the Perron-Frobenius theorem the largest eigenvalue of the forest can be bounded by the largest eigenvalue of the tree, which can again be bounded by $2\sqrt{d}$. The largest eigenvalue of the uniform tree is easy to bound by reducing it to the path graph of length t with edge weights $\sqrt{d}$; again it suffices to considering the "symmetric" subspace similarly to our previous argument.

[15] This is again straightforward to show using an argument by Hastings [19]: if the Hermitian matrix B has a spectral gap 3γ around its largest eingenvalue, and the Hermitian matrix C has norm at most γ, then $B+C$ has a spectral gap at least γ. Indeed, let λ be the largest eigenvalue and $|\psi\rangle$ the corresponding eigenvector of B, then the largest eigenvalue of $B+C$ can be lower bounded by $\langle\psi|B+C|\psi\rangle \geq \lambda-\gamma$. At the same time by the Courant-Fischer-Weyl min-max principle we have that the second largest eigenvalue of $B+C$ is at most $\max_\phi \langle\phi|(I-|\psi\rangle\langle\psi|)(B+C)(I-|\psi\rangle\langle\psi|)|\phi\rangle \leq \max_\phi \langle\phi|(I-|\psi\rangle\langle\psi|)B(I-|\psi\rangle\langle\psi|)|\phi\rangle + \max_\phi \langle\phi|(I-|\psi\rangle\langle\psi|)C(I-|\psi\rangle\langle\psi|)|\phi\rangle \leq \lambda - 3\gamma + \gamma = \lambda - 2\gamma$.

The construction for the modified query model is closely related to the obfuscation construction of [18].

In detail, we have an infinite set of *labels*. Each label will correspond to some vertex, but the correspondence is many-to-one; we describe this correspondence by some function $F(\cdot)$. The algorithm will initially be given some label l that corresponds to a vertex that is the ground state of H_0, the Hamiltonian at the start of the adiabatic path. A query of the oracle consists of giving it any label m that is either l or is a label that the algorithm has received in response to some previous query, as well as giving it any $s \in [0,1]$. The oracle will return some set S of labels such that $F(S)$ is the set of vertices with edges to $F(m)$ (as well as returning the values of the corresponding matrix elements if the off-diagonal matrix elements are non-uniform), and also return the appropriate diagonal matrix element $\langle F(m)|H_s|F(m)\rangle$. Distinct labels in S will have different images under $F(\cdot)$ so that $|S|$ is equal to the number of neighbors.

The labels in S will be chosen as follows: if label m was received in response to some previous query on a label n, so that $\langle F(n)|H_s|F(m)\rangle$ is nonzero and hence $F(n) \in F(S)$, then label n will be in S, i.e., we will "continue to label $F(n)$ by label n". However, for *all* other vertices j with a nonvanishing matrix element to $F(m)$, there will be a new label (distinct from all previous labels) to label the given vertex j, i.e., a new label o such that $F(o) = j$.

Thus, a sequence of queries by the algorithm, can be described by a tree, each vertex of which is some label, with neighboring vertices in the tree corresponding to vertices which are neighbors.

We claim that it suffices to prove a separation in the modified query model in order to obtain (almost the same) separation in the original query model. To do this, we "blow up" each vertex, replacing it with a cluster of exponentially many (in n) vertices; for example, we may replace each with 2^n vertices. We then replace each edge by 2^n edges, these edges randomly matching the vertices in the corresponding clusters.

We also add edges to the graph within each cluster of vertices on some expander graph within the cluster. Define an isometry from the Hilbert space of the original graph to that of the blown up graph which maps each vertex of the original graph to the uniform superposition of vertices in the corresponding cluster. We choose these added edges so that the ground state and first excited states are given by, up to small error, by applying this isometry to the ground and first excited states on the original graph.

We can regard the 2^n vertices in each cluster of the "blown up" graph as corresponding to different labels for vertices in the original graph. Then, in using the original query model for this "blown up" graph, it becomes exponentially unlikely to receive the same label twice unless one follows a backtracking path, and so indeed it suffices to consider the modified query model.

3.2 Estimating Ground State Energy and Distinguishing Graphs

In [19] an $O(\text{poly}(1/\epsilon))$-time adiabatic (sign-problem-free) protocol is given for estimating the ground state energy of a sign-problem-free Hamiltonian (with polynomially bounded norm) up to ϵ-precision, given a basis state, that we call the "start vertex", which has $\Omega(\epsilon)$ overlap with the ground state. To be more precise about what is meant by "estimating": given a polynomially

bounded sign-problem-free Hamiltonian G and a start vertex, and given the promise that the ground state energy of G is not in the interval $[E-\epsilon, E]$ for some E and the further promise[16] that G has a spectral gap $\Omega(1/\mathrm{poly}(\epsilon))$, then there is an adiabatic path[17] with gap $\Omega(1/\mathrm{poly}(\epsilon))$ so that measurement of the final state in the computational basis solves the decision problem of whether the ground state energy is $> E$ or $< E-\epsilon$.

On the other hand, a pair of graphs are constructed in [19], with $\mathrm{poly}(1/n)$-difference in their ground state energies, such that distinguishing them necessarily takes much longer classically. This then almost gives the needed separation between the power of adiabatic quantum computation with no sign problem and classical algorithms. However, there are two issues. First, the adiabatic path constructed to solve the decision problem does not satisfy the requirement that the ground state at the end of the path be a single basis state, but rather may be a superposition. Second, a classical algorithm which simply "guesses" which graph it is given answers the decision problem correctly with probability $1/2$. To resolve both these problems, some further technical steps are done as sketched in Section 3.2.2. These technical steps show that to prove a bound on the number of classical queries to determine (with at least a non-negligible probability of being correct) the final basis state at the end of an adiabatic path, it suffices to prove a lower bound on the number of queries in a problem of distinguishing graphs.

In this problem of distinguishing graphs, there are two sparse graphs, C and D, and the ground state energy of C is smaller than that of D by at least an inverse polynomial amount. Each graph has a privileged vertex, called the "start vertex". The problem is to correctly determine, given a graph randomly chosen to be C or D with equal probability, which graph it was with probability at least $2/3$. There is a promise that the ground state has amplitude at least inverse polynomial amplitude on the start vertex and that the Hamiltonian on each graph has at least an inverse polynomial spectral gap. Queries of the graph are in the modified query model, with the first vertex being the start vertex; further, every query of a label corresponding to the start vertex returns the additional information that it is the start vertex.

Note that if we were not given a start vertex, but simply asked to estimate the ground state energy of some graph, this would clearly require an exhaustive search of the graph (for example, all vertices might be degree 0, except for one vertex in C with a self-loop, in which case distinguishing them would require a search to find this self-loop). However, the start vertex gives additional information, similar to an adiabatic path.

3.2.1 Solving the Decision Problem. The path to solve the decision problem for the ground state energy is as follows. Consider a system with computational basis states labelled by vertices of a graph G as well as by an additional basis state $|0\rangle$. We will label the basis state corresponding to the start vertex of G by $|s\rangle$ (hopefully no confusion will arise with the use of s as a parameter in the path). Consider the two parameter family of Hamiltonians

$$H(t,U) = -U|0\rangle\langle 0| + t\Big(|0\rangle\langle s| + |s\rangle\langle 0|\Big) + H(G), \tag{3}$$

where $H(G)$ is the Hamiltonian corresponding to the graph G. We take $t < 0$ so that the Hamiltonian has no sign problem.

Now consider a path of Hamiltonians starting at very negative U and with $t = 0$ (so that initially the ground state is $|0\rangle$ regardless of G), then making t slightly negative and increasing U to $E - \epsilon/2$, followed by returning t to zero. At the end of this path, if the ground state energy is $> E$, the ground state of the Hamiltonian will still be $|0\rangle$ but if the ground state energy is $< E - \epsilon$, the ground state of the Hamiltonian will be the ground state of $H(G)$. Note that it is necessary to have a nonzero t in the middle of this path; if t were kept equal to zero then there would be a level crossing at which the gap would vanish; instead, the nonzero t and nonzero overlap of the start vertex with the ground state ensure the gap stays open. Indeed, it is possible to choose t, U along the path so that the gap of this path is inverse polynomial.

3.2.2 Ending the Path at a Single Basis State. Unfortunately, our path of Hamiltonians does not satisfy the condition that the ground state of the Hamiltonian at the end of the path be a single basis state rather than a superposition.

We might try to solve this by concatenating the path of Hamiltonians above with an additional path that decreases the Hamiltonian on the graph to zero while adding an additional negative term on the start vertex so that if the graph is C the final basis state is on the start vertex, while if the graph is D the final basis state is $|0\rangle$. This still however does not give us a good lower bound on the classical number of queries: for this path H_s, a classical algorithm can determine the ground state at the end of the path (which we will assume to occur at $s = 1$) by querying the oracle three times, first querying $\langle 0|H_1|0\rangle$, then querying the oracle to find the neighbors of $|0\rangle$ in the middle of the path (so that it can determine the start vertex s), and finally querying $\langle s|H_1|s\rangle$.

So, we use an additional trick, the additional "technical steps" mentioned above. We consider n different copies of the problem defined by Hamiltonian Eq. (3) "in parallel", i.e., taking the sum of Hamiltonians on the tensor product Hilbert space. For each of these n copies, we choose G to be either C or D independently, so that there are 2^n possible instances; let G_i be the graph on copy i. At the end of the path, the ground state is a tensor product of $|0\rangle$ on some copies and the ground state of $H(C)$ on some other copies. We then use this property of the ground state as a kind of key to find an entry in a database: we add an additional diagonal term which is large and negative on tensor product basis states which are products of basis state $|0\rangle_i$ for all copies for which $G_i = D$ and of basis states corresponding to vertices of G_i for all copies on which $G_i = C$. Finally, we use a modified version of the trick above of concatenating with an additional path that decreases the Hamiltonian on each graph to zero while adding an additional negative term on the start vertex; now we add the term on the start vertex for both graph C and D so that queries of this term do not reveal the graph. These additional diagonal terms are both added after $t = 0$, so the Hamiltonian does not couple $|0\rangle$ to $|s\rangle$. The large negative term added on the given tensor product states ensures that the ground state stays in the space spanned by those tensor product states, with a large gap to all other states. Then, the unique ground state of H_1 is given by the product of $|0\rangle$ on all copies for which $G_i = D$ and of the start vertex for all copies on which $G_i = C$.

[16] This further promise is in fact only required if the ground state energy of G is $< E - \epsilon$.

[17] If the promise is violated, then we do not lower bound the gap of the path.

If the algorithm cannot distinguish graphs C, D with sufficiently large probability, it has no way to find this particular choice given the 2^n possible choices of graphs.

4 RELATION TO THE WELDED-TREES CONSTRUCTION AND QUANTUM WALKS

Our graph in Section 2 is reminiscent to the welded-trees construct [10]. The main difference from the welded-trees construction is that our graph has highly non-uniform degree distribution. This is necessary for getting a polynomially large overlap between the ENTRANCE and EXIT vertices, and the largest eigenvalue of the adjacency matrix. (Indeed, a uniform degree-d graph has largest eigenvalue d, and the corresponding eigenvector is a uniform superposition over all vertices.) Therefore the analysis of [10] that heavily relied on the uniformity of the welded-trees graph, does not apply here, and a different construction was necessary. Our modification is very natural: simply weld the trees with less edges, in order to get the sought polynomial overlap with the ENTRANCE and EXIT vertices. But there is a difficulty here arising from the fact that the non-uniform degrees can give away the structure of the trees, allowing fast traversal [5, 12]. A first attempt is to add a longer middle section, or tunnel, between the trees — but unfortunately a random walk can still traverse such a tunnel. Nevertheless, the tunnel is uniform and looks locally tree-like, and we can utilize these properties to hide the edges of the original graph, by adding "camouflaged" decoration trees, motivated by [19]. This combines the advantages of the welded-trees construction [10] with that of Hastings [19] to obtain the subexponential separation.

The similarity to the welded-trees construction hints at the possibility to use a natural quantum walk algorithm in addition to the adiabatic path that we described. Since we have adjacency-list access to the graph, we can implement an efficient *block-encoding* [17] of its adjacency matrix divided by the maximal degree $(m^2 + m + 1)$ (i.e., a quantum circuit which corresponds to a unitary matrix U whose top-left corner equals $A/(m^2 + m + 1)$), using just 2 queries. Such a block-encoding can be used for running a Szegedy-type discrete quantum walk [2, 9, 24]. Let $|\psi\rangle$ denote the top eigenvector of A; we can approximately "project" this block-encoding to a block-encoding U' of $|\psi\rangle\langle\psi|$ via quantum singular value transformation [17, 22] according to a low-degree $\widetilde{O}\left(m^{\frac{3}{2}}\right)$ polynomial approximation of the threshold function filtering out all non-maximal eigenvalues [16, 21]; this U' can be implemented using $\widetilde{O}\left(m^{\frac{3}{2}}\right)$ queries. Applying this block-encoding to $|\text{ENTRANCE}\rangle$ results in a polynomially large overlap with the $|\text{EXIT}\rangle$ vertex, since $\langle \text{EXIT}||\psi\rangle\langle\psi||\text{ENTRANCE}\rangle = |\langle\psi|\text{ENTRANCE}\rangle|^2 = \Omega(\ell^{-3}) = \Omega(m^{-\frac{3}{4}})$. Using amplitude amplification this gives an $\widetilde{O}\left(m^{\frac{9}{4}}\right)$-time quantum query algorithm.

Alternatively, the above approximate threshold polynomial can be decomposed into a linear combination of Chebyshev polynomials, where the coefficients have ℓ^1-weight at most $\widetilde{O}\left(m^{\frac{3}{4}}\right)$. Since the Szegedy quantum walk effectively applies a Chebyshev polynomial to the block-encoded adjacency matrix [2, 9, 16], it enables a linear combination of unitaries (LCU) [11, 13, 17] based implementation of an $\widetilde{\Omega}(m^{\frac{3}{4}})$-subnormalized block-encoding of $|\psi\rangle\langle\psi|$. This algorithm can be simplified [4] by randomly picking a time $t \in [\widetilde{O}(m^{3/2})]$, and applying t-steps of the Szegedy walk to the input state $|\text{ENTRANCE}\rangle$. It is not difficult to show [4] that measuring the final state of this plain quantum-walk-based algorithm finds the EXIT with probability $\Omega(1/\text{poly}(m))$.

5 DISCUSSION

In this paper we have demonstrated the possibility of a (sub)exponential quantum speedup via an adiabatic evolution where the instantaneous Hamiltonians have spectral norm at most $\text{poly}(n)$ and the spectral gap is at least $1/\text{poly}(n)$, furthermore the Hamiltonian has no sign problem, i.e., all of its matrix elements are non-positive. In order to prove such a big separation we worked in an oracle model, where a $\text{poly}(n)$-sparse graph is given via its adjacency list, whose adjacency matrix described the Hamiltonian. The adiabatic path has very nice additional properties: the initial and the final Hamiltonians are diagonal, and the path consists of two "straight lines"; in fact only following the first "line segment" already provides the (sub)exponential quantum speedup, but then the final Hamiltonian is non-diagonal. Our result heavily builds on ideas recently introduced by Hastings [19], but simplifies and improves those in several ways: Hastings' original result only showed a quasipolynomial quantum speed-up, and used significantly more complicated Hamiltonian paths. Additionally, our problem features a natural quantum walk algorithm, providing a new example of a (sub)exponential speedup via a simple quantum walk.

In order to get a (sub)exponential quantum speedup, we used a graph where the top eigenvector's ℓ^2-weight is concentrated on the essential structural parts of the graph, whereas the ℓ^1-weight is concentrated on some "camouflaging decorations". This ℓ^1-weight shift already happened after one level of decoration which intuitively ruled out efficient simplistic Monte Carlo algorithms. In order to rule out any efficient classical algorithm we added a polynomial number of decoration layers, effectively hiding the essential structure of the graph from any classical algorithm.

We conjecture that it should be possible to improve the exponent of our (sub)exponential lower bound to $\exp(n^{1-o(1)})$ for some simple adiabatic path with no sign problem. In fact, we think that a fine-tuned version of the construction discussed in this paper might already exhibit such a separation. We already hinted at the possibility of some improved decoration structure in Footnote 10, and there are other potential improvements that shall improve the exponent. For example the use of the union bound over all encountered roots inside the proof of Lemma 4 is probably unnecessary, and ultimately one might not need to increase the depth of the decoration trees between various levels as rapidly as we do in Definition 2.

The big open question that remains is whether one can get a superpolynomial speedup via an adiabatic path that has no sign problem, and comes from a local Hamiltonian. Such a speedup could have important practical implications, since D-Wave's quantum annealers have such restrictions. However, proving such a result requires proving a superpolynomial circuit lower-bound for a non-oracular problem, which is beyond the reach of currently known techinques in theoretical computer science.

ACKNOWLEDGMENTS

Part of this work was done while visiting the Simons Institute for the Theory of Computing; we gratefully acknowledge its hospitality.

A.G. acknowledges funding provided by Samsung Electronics Co., Ltd., for the project "The Computational Power of Sampling on Quantum Computers", and additional support by the Institute for Quantum Information and Matter, an NSF Physics Frontiers Center (NSF Grant PHY-1733907). U.V. was supported by the Vannevar Bush faculty fellowship N00014-17-1-3025, and US DOE National Quantum Initiative Science Research Centers, Quantum Systems Accelerator (QSA).

A THE SPECTRAL GAP OF THE ADIABATIC EVOLUTION ON THE PATH GRAPH

In this appendix we prove that the energy gap around the ground state of $H_\ell(s)$ is at least $\Omega(1/\ell^2)$ for all $s \in [-1,1]$. Since $\|H_\ell(s)\| = \Theta(1)$, this is equivalent to proving that $H_\ell(s)/\|H_\ell(s)\|$ has a spectral gap of size $\Omega(1/\ell^2)$ around its ground state energy. We prove the latter, but using a more convenient parametrization.

Consider the Hamiltonian $A_\ell(\alpha) := \alpha|0\rangle\langle 0| + A_\ell$, for $\alpha \in \mathbb{R}_+$. We will show that the smallest eigenvalue gap of $A_\ell(\alpha)$ is at least $\Omega(1/\ell^2)$. This will imply that the spectral gap of $A_\ell(\alpha)$ around the largest eigenvalue is $\Omega((1+\alpha)/\ell^2)$, which in turn implies that the spectral gap around the ground state energy of $-A_\ell(\alpha)/\|A_\ell(\alpha)\|$ is at least $\Omega(1/\ell^2)$ proving that $H'(s)$ has an $\Omega(1/\ell^2)$-large gap around its ground state energy.

Now we turn to analyzing $A_\ell(\alpha)$. We claim that the eigenvectors and eigenvalues of $A_\ell(\alpha)$ are all associated to solutions of the quasimomenta equation

$$f_\ell(p) := \frac{\sin((\ell+1)p)}{\sin(\ell p)} = \alpha. \tag{4}$$

Indeed the vector $|\psi_p\rangle := \sum_{j=1}^{\ell} \sin(jp)|j\rangle$ is always an eigenvector of $A_\ell(\alpha)$ with eigenvalue $2\cos(p)$, whenever p is a solution of Eq. (4). Since $f_\ell(p)$ and $\cos(p)$ are both symmetric and 2π periodic it suffices to concentrate on solutions within the interval $[0,\pi]$.

We show that for $\alpha \in [0, \frac{\ell+1}{\ell})$ there are ℓ distinct $p \in [0,\pi]$ solutions to *Eq.* (4), and there are $\ell-1$ such solutions otherwise. For $\alpha = \frac{\ell+1}{\ell}$ the additional eigenvalue is 2 which can be obtained as the limit of the largest eigenvalue as α goes to $\frac{\ell+1}{\ell}$ from below and corresponds to the "pseudo-solution" $p = 0+$ corresponding to the eigenvector $|\psi_{0+}\rangle := \sum_{j=1}^{\ell} j|j\rangle$. If $\alpha > \frac{\ell+1}{\ell}$, then there is a complex solution to Eq. (4), which corresponds to the unique real solution x of $\frac{\sinh((\ell+1)x)}{\sinh(\ell x)} = \alpha$ giving eigenvalue $2\cosh(x)$ and eigenvector $|\phi_x\rangle := \sum_{j=1}^{\ell} \sinh(jx)|j\rangle$.

We proceed by showing that on every interval $(\frac{j-1}{\ell}\pi, \frac{j}{\ell}\pi)$ for $j \in [\ell]$ the function $f_\ell(p)$ is strictly monotone decreasing, and the range of $f_\ell(p)$ equals $\mathbb{R}$ on these intervals apart from the first and last intervals. For this observe that for all $j \in [\ell-1]$

$$\lim_{\varepsilon \downarrow 0} f_\ell\left(\frac{j}{\ell}\pi \pm \varepsilon\right) = \lim_{\varepsilon \downarrow 0}\left(\frac{\sin(j\pi + \frac{j}{\ell}\pi \pm (\ell+1)\varepsilon)}{\sin(j\pi \pm \ell\varepsilon)}\right)$$
$$= \lim_{\varepsilon \downarrow 0}\left(\frac{\sin(\frac{j}{\ell}\pi \pm (\ell+1)\varepsilon)}{\sin(\pm\ell\varepsilon)}\right) = \pm\infty.$$

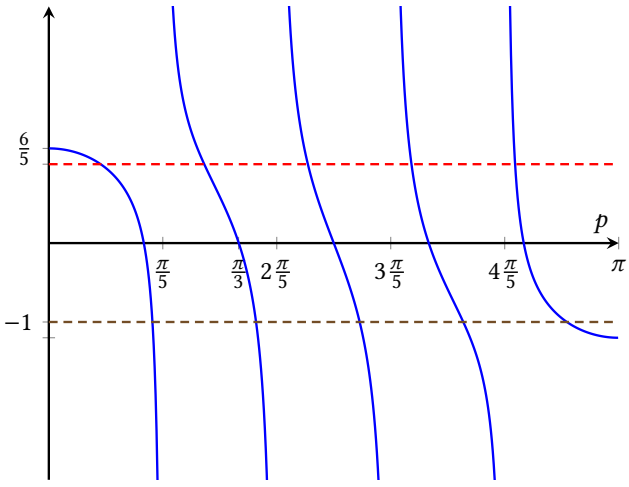


Figure 2: Plot of $f_\ell(p) = \frac{\sin((\ell+1)p)}{\sin(\ell p)}$ for $\ell = 5$; dashed lines show the solutions of Eq. (4) for $\alpha = \pm 1$.

Further observe that the derivative

$$f'_\ell(p) = f_\ell(p)((\ell+1)\cot((\ell+1)p) - \ell\cot(\ell p))$$

is non-positive for all $p \in [0,\pi] \setminus \mathbb{N}\frac{\pi}{\ell}$. To see this, observe that $f_\ell(p)$ changes sign where either $\sin(\ell p)$ or $\sin((\ell+1)p)$ changes sign, that is at values $p \in S_\ell := \{\frac{j}{\ell}\pi \colon j \in [\ell-1]\} \cup \{\frac{j}{\ell+1}\pi \colon j \in [\ell]\}$. Now we show that $g_\ell(p) := ((\ell+1)\cot((\ell+1)p) - \ell\cot(\ell p))$ also changes sign at exactly the same set of points. Since $\ell\cot(\ell p)$ jumps from $-\infty$ to ∞ at $p \in \{\frac{j}{\ell}\pi \colon j \in [\ell-1]\}$ and $(\ell+1)\cot((\ell+1)p)$ jumps from $-\infty$ to ∞ at $p \in \{\frac{j}{\ell+1}\pi \colon j \in [\ell]\}$ we get that $g_\ell(p)$ changes sign at $p \in S_\ell$. One can also see that $g_\ell(p) \neq 0$ for any $p \in (0,\pi)$ implying that $g_\ell(p)$ changes sign only at points of S_ℓ. Indeed

$$g_\ell(p) = 0$$
$$\Updownarrow$$
$$\cot((\ell+1)p) = \ell(\cot(\ell p) - \cot((\ell+1)p))$$
$$\Downarrow$$
$$\cos((\ell+1)p) = \ell\overbrace{\frac{\sin((\ell+1)p)\cos(\ell p) - \cos((\ell+1)p)\sin(\ell p))}{\sin(\ell p)}}^{\sin(p)},$$

where the last equality clearly does not hold, since $\cos((\ell+1)p) \in [-1,1]$, whereas in contrast $|l\sin(p)|/|\sin(\ell p)| > 1$ for all $p \in (0,\pi)$. This implies that $f'_\ell(p) = f_\ell(p)g_\ell(p)$ is never positive.

We can conclude that $f_\ell(p)$ is strictly monotone decreasing on the intervals $(\frac{j-1}{\ell}\pi, \frac{j}{\ell}\pi)$ for all $j \in [\ell]$, moreover the range of $f_\ell(p)$ equals $\mathbb{R}$ on these intervals for all $j \in \{2,3,\ldots,\ell-1\}$, while the range of $f_\ell(p)$ on $(0, \frac{\pi}{\ell})$ equals $(-\infty, \frac{\ell+1}{\ell})$ and its range on $(\frac{\ell-1}{\ell}\pi, \pi)$ equals $(\frac{\ell+1}{\ell}, \infty)$. This proves our claim about the number of solutions of Eq. (4) within the interval $(0,\pi)$.

Next, we prove our lower bound on the spectral gap. Firs we show that for any $\alpha \in \mathbb{R}$ the ℓ or $\ell-1$ different real solutions of Eq. (4) have an $\Omega(1/\ell)$ gap in between. This implies that the corresponding eigenvalues also have gaps of size at least $\Omega(1/\ell^2)$ due to the following little lemma:

Lemma 5. *Suppose that $x < y \in [0,\pi]$, then $|\cos(x)-\cos(y)| \geq 1-\cos(y-x) \geq (y-x)^2/\pi^2$.*

Proof.

$$\begin{aligned}\cos(x)-\cos(y) &= \int_x^y \sin(z)dz \\ &\geq \int_0^{y-x} \sin(z)dz \\ &= \cos(0)-\cos(y-x) \\ &= 1-\cos(y-x) \geq \frac{(y-x)^2}{\pi^2}. \qquad \square\end{aligned}$$

For lower bounding the gaps between the solutions to Eq. (4), observe that for $\alpha = 1$ the solutions are $\{\frac{2j-1}{2\ell+1}\pi\colon j \in [\ell]\}$, and similarly for $\alpha=-1$ the solutions are $\{\frac{2j}{2\ell+1}\pi\colon j\in[\ell]\}$. Since $f_\ell(p)$ is strictly monotone decreasing within each interval $[\frac{(j-1)}{\ell}\pi, \frac{j}{\ell}\pi]$ for all $j\in[\ell]$ we get that for any $\alpha\in[-1,1]$ the j-th solution of Eq. (4) lies in the interval $[\frac{2j-1}{2\ell+1}\pi, \frac{2j}{2\ell+1}\pi]$. Thus for every $\alpha\in[-1,1]$ any two subsequent solutions have a gap at least $\frac{2j+1}{2\ell+1}\pi - \frac{2j}{2\ell+1}\pi = \frac{\pi}{2\ell+1}$. Similarly for $|\alpha|>1$ the solutions lie outside the intervals $[\frac{2j-1}{2\ell+1}\pi, \frac{2j}{2\ell+1}\pi]\colon j\in[\ell]$ and are therefore also at least $\frac{\pi}{2\ell+1}$ apart. We can conclude that the different real solutions of Eq. (4) have gaps of size at least $\frac{\pi}{2\ell+1}$ in between.

We also need to treat the case when $\alpha \geq \frac{\ell+1}{\ell}$, so that there are only $(\ell-1)$ real solutions to Eq. (4). then the largest eigenvalue is $2\cosh(x)$ for some $x\in\mathbb{R}$. Moreover, $\langle 1|A_\ell(\alpha)|1\rangle = \alpha$, so we get that the largest eigenvalue is at least $\max\{2,\alpha\}$. On the other hand the second largest eigenvalue is $2\cos(p)$ for some $p\in(\frac{1}{\ell}\pi, \frac{2}{\ell}\pi)$, and since $2\cos(p) < 2-p^2/\pi^2$ for all $p\in(0,\pi)$, we get that the second largest eigenvalue is at most $2-1/\ell^2$. Thus the eigenvalue gap is at least $\max\{1/\ell^2, \alpha-2+1/\ell^2\} = \Omega((\alpha+1)/\ell^2)$.

Thus we have shown that for any $\alpha\geq 0$ the matrix $A_\ell(\alpha)$ has a spectral gap at least $\Omega((\alpha+1)/\ell^2)$. Since $\|A_\ell(\alpha)\| \leq \alpha + \|H_\ell\| = O(\alpha+1)$ we get that the spectral gap of $A_\ell(\alpha)/\|A_\ell(\alpha)\|$ is at least $\Omega(1/\ell^2)$ finishing our proof of the fact that $H_\ell(s)$ has an $\Omega(1/\ell^2)$-large gap around its ground state energy for every $s\in[-1,0]$, and due to symmetry this result extends to $s\in[-1,1]$.

Finally, let us understand the solutions and eigenvalues for the unperturbed case when $\alpha=0$. Clearly the ℓ different solutions of Eq. (4) are $\{\frac{j}{\ell+1}\pi\colon j\in[\ell]\}$. In this case the largest eigenvalue of $A_\ell(0) = A_\ell$ is $2\cos(1/(\ell+1))$, and the corresponding eigenstate is proportional to $\sum_{j=1}^{\ell}\sin(\frac{j}{\ell+1}\pi)|j\rangle$. In particular the normalized eigenstate has an overlap of at least $\Omega(\ell^{-\frac{3}{2}})$ with any vertex $|j\rangle$.

B THE EFFECT OF DECORATION ON THE TOP EIGENVECTOR: ℓ^1 VS. ℓ^2 WEIGHT

In this appendix we study the effect of decoration on the top eigenvector of the adjacency matrix. We first show that for any graph G with top-eigenvector ψ, the top eigenvector ψ' of the decorated graph G' is proportional to ψ on the original vertices of G. (This result also applies to the Hamiltonians that come from intermediate $s\neq 0$ Hamiltonians.) Moreover, we show that in the top eigenvector $\psi^{(r)}$ of $G^{(r)}$ – the graph that we construct in this paper – the overwhelming majority of the ℓ^2-weight is supported on the original vertices of G. On the other hand the ℓ^1-weight proportion of the original vertices of G in $\psi^{(r)}$ is (sub)exponentially small.

Suppose we have a graph $G=(V,E)$, and we attach k-copies of a connected graph T to every vertex of G via an edge to a distinguished vertex t of T resulting in the new graph G'. Let λ_G be the top eigenvalue of G with corresponding eigenvector ψ. For $\gamma\in\mathbb{R}_+$ let $T(\gamma)$ be the graph where we add a self-loop to the vertex t with weight γ and let $\lambda_T(\gamma)$ its largest eigenvalue with $\phi(\gamma)$ the corresponding eigenvector normalized such that the amplitude $\phi_t(\gamma)$ at t equals 1. Then we claim that the top eigenvalue $\lambda_{G'}$ equals $\phi(\gamma)$, where γ is the unique solution[18] to the equation

$$\lambda_G + \frac{k}{\gamma} = \lambda_T(\gamma). \tag{5}$$

It is easy to verify that the corresponding eigenvector is $\psi' = \psi + \bigoplus_{v\in V}\frac{\psi_v}{\gamma}\left(\bigoplus_{j\in[k]}\phi(\gamma)\right)$, where $\bigoplus_{j\in[k]}\phi(\gamma)$ stands for the direct sum of the eigenvectors $\phi(\gamma)$ corresponding to the k copies of T attached to the vertex v.

Now that we have an analytic description of how the eigenvectors and eigenvalues change under decoration it is time to understand the quantity $\lambda_T(\gamma)$, when T is a complete tree with $d=\Theta(m)$ children at every level up to depth ℓ' for some $\ell'\in\{m^{3\delta}, m^{3\delta}+1,\dots,m^{4\delta}\}$, and w is the root of T like in our decorations. In our scenario we have $2m\leq\lambda_G$, $k=m^{1-\delta}$, and for large enough m

$$m\leq\gamma\leq\lambda_G+1. \tag{6}$$

To see the latter, consider Eq. (5) and observe that $\lambda_T(m)\leq m+2\sqrt{d} = m+o(m)$ which is smaller than λ_G for large enough m, while $\lambda_T(\lambda_G+1)\geq\lambda_G+1$ which is larger than $\lambda_G+k/\gamma\leq\lambda_G+O\left(m^{-\delta}\right)$ for large enough m.

It is again useful to consider the adjacency matrix of $T(\gamma)$ in the "symmetric" subspace, which is spanned by uniform superpositions $|L_j\rangle = \frac{1}{\sqrt{|L_j|}}\sum_{v\in L_j}|v\rangle$ over the level sets $L_j=\{$vertices of T at distance j form the root$\}$. The symmetric subspace is again invariant under the linear map $A_T(\gamma)$, and due the Perron-Frobenius theorem it contains the largest eigenvector, moreover its matrix looks exactly like the adjacency matrix of a path graph of length-ℓ' with a self-loop of weight γ at the root, and uniform $\sqrt{d}$ edge weights, cf. Footnote 13. When $\gamma\geq 2\sqrt{d}$ by our analysis of such graphs in Appendix A we know that the largest eigenvalue will be $\sqrt{d}\cosh(x)$, where x is the unique solution of the equation

$$\sinh((\ell'+2)x) = \frac{\gamma}{\sqrt{d}}\sinh((\ell'+1)x), \tag{7}$$

and the corresponding eigenvector is $\sum_{j=0}^{\ell'}\sinh((\ell'+1-j)x)|L_j\rangle$ up to normailzation. Thus, the eigenvector $\phi(\gamma)$ can be written as

$$\phi(\gamma) = \sum_{j=0}^{\ell'}\frac{\sinh((\ell'+1-j)x)}{\sinh((\ell'+1)x)}|L_j\rangle. \tag{8}$$

[18] Since the right-hand side of Eq. (5) is strictly monotone increasing for $\gamma\in\mathbb{R}_+$ and the left-hand side is strictly monotone decreasing there is at most one solution. Since both sides are continuous, and in the $\gamma\to 0$ limit the left hand side is $+\infty$ and in the $\gamma\to+\infty$ limit the left-hand side is $+\infty$ there always must be a solution to Eq. (5).

Let us now bound the ℓ^2-weight of $\phi(\gamma)$ as follows

$$\|\phi(\gamma)\|_2 \leq \sum_{j=0}^{\ell'} \left|\frac{\sinh((\ell'+1-j)x)}{\sinh((\ell'+1)x)}\right| \left\||L_j\rangle\right\|_2 = \sum_{j=0}^{\ell'} \prod_{i=0}^{j-0} \frac{\sinh((\ell'-i)x)}{\sinh((\ell'+1-i)x)} \leq \sum_{j=0}^{\ell'} \left(\frac{\sqrt{d}}{\gamma}\right)^j, \quad (9)$$

where the last inequality follows from Eq. (7) and the fact[19] that $\frac{\sinh((h-1)x)}{\sinh(hx)}$ is monotone increasing in h for every $x \geq 0$. Since $\frac{\sqrt{d}}{\gamma} = O\left(m^{-\frac{1}{2}}\right)$, Eq. (9) implies that $\|\phi(\gamma)\|_2 = \Theta(1)$. Then the ℓ^2-weight of $\left\|\frac{1}{\gamma}\bigoplus_{j\in[k]}\phi(\gamma)\right\|_2 = \frac{\sqrt{k}}{\gamma}\|\phi(\gamma)\|_2 = O\left(m^{-\frac{1+\delta}{2}}\right)$. Therefore, after one level of decoration the ℓ^2-weight ratio of the original eigenvector ψ within the new eigenvector ψ' is $1 - O\left(m^{-(1+\delta)}\right)$, and after m^δ levels of decoration the ratio still remains as large as $1 - O(1/m)$.

Finally, let us bound the ℓ^1-weight of $\phi(\gamma)$ by observing that the solution of Eq. (7) satisfies $x \leq \ln(\gamma/\sqrt{d})$.[20]

$$\begin{aligned}
\|\phi(\gamma)\|_1 &= \sum_{j=0}^{\ell'} \frac{\sinh((\ell'+1-j)x)}{\sinh((\ell'+1)x)} \left\||L_j\rangle\right\|_1 \\
&\geq \sum_{j=0}^{\ell'} \frac{\sinh((\ell'+1-j)x)}{2\exp((\ell'+1)x)} \left\||L_j\rangle\right\|_1 \\
&= \sum_{j=0}^{\ell'} \frac{\exp((\ell'+1-j)x)}{\exp((\ell'+1)x)} (1-\exp(-2(\ell'+1-j)x)) \left\||L_j\rangle\right\|_1 \\
&\geq (1-\exp(-2x)) \sum_{j=0}^{\ell'} \frac{\exp((\ell'+1-j)x)}{\exp((\ell'+1)x)} \left\||L_j\rangle\right\|_1 \\
&= (1-\Theta(d/\gamma^2)) \sum_{j=0}^{\ell'} \exp(-jx) d^{\frac{j}{2}} \\
&\geq (1-\Theta(d/\gamma^2)) \sum_{j=0}^{\ell'} \left(\frac{d}{\gamma}\right)^j. \quad (10)
\end{aligned}$$

In our case the obfuscated graph satisfies $\lambda_G < 4m$, and for the first decoration we have $d = 4m + m^{1-\delta}$ so that $\frac{d}{\gamma} = 1 + \Omega(m^{-\delta})$, therefore Eq. (10) implies $\|\phi(\gamma)\|_1 = \exp(\Omega(m^{2\delta}))$. Thus, already after one level of decoration the ℓ^1-weight ratio of the original eigenvector ψ within the new eigenvector ψ' is $\exp(-\Omega(m^{2\delta}))$. Since later decorations also satisfy $\frac{d}{\gamma} = 1 + \Omega(m^{-\delta})$, after applying all m^δ levels of decoration the ratio becomes as small as $\exp(-\Omega(m^{3\delta}))$.

Finally, note that essentially the same argument as above shows that the ℓ^2-weight of the top eigenvector is concentrated on the original vertices of the obfuscated graph throughout the entire adiabatic path. However, the above argument breaks down for showing that only a tiny a fraction of the ℓ^1-weight is located on the original graph. Indeed, for $|s| = 1$ the top eigenstate is supported on the original graph, and due to continuity it implies that for $|s| \approx 1$ most of the ℓ^1-weight is located on the original graph.

[19] This can be seen by $-\cosh(2x) \leq -1 \Rightarrow e^{2hx}+e^{-2hx}-e^{2x}-e^{-2x} \leq e^{2hx}+e^{-2hx}-2 \Rightarrow 4\sinh((h+1)x)\sinh((h-1)x) \leq 4\sinh^2(hx) \Rightarrow \frac{\sinh((h-1)x)}{\sinh(hx)} \leq \frac{\sinh(hx)}{\sinh((h+1)x)}$.

[20] Our analysis of Eq. (4) revealed that Eq. (7) has at most 1 real solution. Then the function $\sinh((\ell'+2)x)/\sinh((\ell'+1)x)$ must be strictly monotone increasing on $\mathbb{R}_+$ due to its continuous behavior. Since for every $c > 1$ we have $\sinh((\ell'+2)\ln(c))/\sinh((\ell'+1)\ln(c)) = c(1-c^{-2(\ell'+2)})/(1-c^{-2(\ell'+1)}) > c$ we get $x \leq \ln(\gamma/\sqrt{d})$.

C SAMPLING RANDOM REGULAR EXPANDER GRAPHS

Definition 6 (Sampling a random cycle). *For any $n \in \mathbb{N}$ we can sample a uniformly random cycle on the vertex set $[n]$ as follows: sample an arbitrary permutation of n elements, and define the corresponding edge set as $\{(i,j) \in [n]\times[n] : \pi(i) - \pi(j) \equiv \pm 1 \bmod n\}$.*

Note that there is an alternative way to sample uniformly random cycles which is equivalent to the above: sample a cyclic permutation π of $[n]$ independently and uniformly at random, and define the set of edges so that $(i,j) \in E$ iff $\pi^{\pm 1}(i) = j$. However, in our analysis Definition 6 is more convenient.

Definition 7 (Random d-regular graphs). *For an even d an $n \in \mathbb{N}$ we denote by $\mathcal{H}_{n,d}$ a distribution of undirected d-regular graphs $G = (V, E)$ on n vertices. Identifying the vertices with the set $V := [n]$ the distribution is defined by independently sampling $d/2$ random cycles as in Definition 6, and taking the union of their edges (with multiplicity).*

Theorem 8 ([15, Theorem 1.2]). *Fix a real $\varepsilon > 0$ and an even positive integer d. Then there is a constant, c, such that for a random graph, G, in $\mathcal{H}_{n,d}$ we have that with probability at least $1 - c/n^\tau$ the second largest eigenvalue $\lambda_2(A_G) \leq 2\sqrt{d-1} + \varepsilon$, where $\tau = \tau_{\text{fund}} = \lceil\sqrt{d-1}\rceil - 1$.*

Corollary 9. *There is a universal constant C, such that for every $m, n \in \mathbb{N}$ satisfying $n \geq m^2$ we have that a random G sampled from $\mathcal{H}_{n,2m}$ satisfies $\lambda_2(A_G) \leq m$ with probability at least $1 - \frac{C}{m^5}$.*

Proof. Let $k := \lfloor m/8 \rfloor$ and $r := m - 8k$ for some $k \in \mathbb{N}$. Then we can obtain a sample G by taking k independent samples H_i from $\mathcal{H}_{n,16}$, and a sample from H_0 from $\mathcal{H}_{n,2r}$ and taking the union of their edges (with multiplicity). Setting $\varepsilon = 0.1$ and $d = 16$ in Theorem 8, and observing that $2\sqrt{15} + 0.1 < 7.9$, we get that there is some universal constant $c \in \mathbb{R}_+$ such that for each $i \in [k]$

$$\Pr(\lambda_2(H_i) > 7.9) \leq cn^{-3} \leq cm^{-6}. \quad (11)$$

By the union bound we get that with probability at least $1 - cm^{-5}$ Eq. (11) holds for all $i \in [k]$ simultaneously, and therefore we get that $\Pr(\lambda_2(G) > 7.9k + 2r = m + r - 0.1k) \leq cm^{-5}$. If $m \geq 560$, then $r \leq 7 \leq 0.1k$, so for such large m we also get $\Pr(\lambda_2(G) > 2m) \leq cm^{-5}$. Choosing $C := \max(560^5, c)$ establishes the statement of the theorem. □

REFERENCES

[1] Dorit Aharonov, Wim van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, and Oded Regev. 2007. Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation. *SIAM Journal on Computing* 37, 1 (2007), 166–194. https://doi.org/10.1137/S0097539705447323 arXiv: quant-ph/0405098

[2] Andris Ambainis, András Gilyén, Stacey Jeffery, and Martins Kokainis. 2020. Quadratic Speedup for Finding Marked Vertices by Quantum Walks. In *Proceedings of the 52nd ACM Symposium on the Theory of Computing (STOC)*. 412–424. https://doi.org/10.1145/3357713.3384252 arXiv: 1903.07493

[3] Andris Ambainis and Oded Regev. 2004. An Elementary Proof of the Quantum Adiabatic Theorem. (2004). arXiv: quant-ph/0411152

[4] Simon Apers, András Gilyén, and Stacey Jeffery. 2021. A Unified Framework of Quantum Walk Search. In *Proceedings of the 38th Symposium on Theoretical Aspects of Computer Science (STACS)*. 6:1–6:13. https://doi.org/10.4230/LIPIcs.STACS.2021.6 arXiv: 1912.04233

[5] Shalev Ben-David, Andrew M. Childs, András Gilyén, William Kretschmer, Supartha Podder, and Daochen Wang. 2020. Symmetries, graph properties, and quantum speedups. In *Proceedings of the 61st IEEE Symposium on Foundations of Computer Science (FOCS)*. 649–660. https://doi.org/10.1109/FOCS46700.2020.00066 arXiv: 2006.12760

[6] Dominic W. Berry, Andrew M. Childs, and Robin Kothari. 2015. Hamiltonian Simulation with Nearly Optimal Dependence on all Parameters. In *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science (FOCS)*. 792–809. https://doi.org/10.1109/FOCS.2015.54 arXiv: 1501.01715

[7] Dominic W. Berry, Andrew M. Childs, Yuan Su, Xin Wang, and Nathan Wiebe. 2020. Time-dependent Hamiltonian simulation with L^1-norm scaling. *Quantum* 4 (2020), 254. https://doi.org/10.22331/q-2020-04-20-254 arXiv: 1906.07115

[8] Sergey Bravyi and Barbara Terhal. 2010. Complexity of Stoquastic Frustration-Free Hamiltonians. *SIAM Journal on Computing* 39, 4 (2010), 1462–1485. https://doi.org/10.1137/08072689X arXiv: 0806.1746

[9] Andrew M. Childs. 2010. On the relationship between continuous- and discrete-time quantum walk. *Communications in Mathematical Physics* 294, 2 (2010), 581–603. https://doi.org/10.1007/s00220-009-0930-1 arXiv: 0810.0312

[10] Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. 2003. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the 35th ACM Symposium on the Theory of Computing (STOC)*. 59–68. https://doi.org/10.1145/780542.780552 arXiv: quant-ph/0209131

[11] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. 2017. Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision. *SIAM Journal on Computing* 46, 6 (2017), 1920–1950. https://doi.org/10.1137/16M1087072 arXiv: 1511.02306

[12] Andrew M. Childs and Daochen Wang. 2020. Can graph properties have exponential quantum speedup? (2020). arXiv: 2001.10520

[13] Andrew M. Childs and Nathan Wiebe. 2012. Hamiltonian simulation using linear combinations of unitary operations. *Quantum Information and Computation* 12, 11&12 (2012), 901–924. https://doi.org/10.26421/QIC12.11-12 arXiv: 1202.5822

[14] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. 2000. Quantum computation by adiabatic evolution. (2000). arXiv: quant-ph/0001106

[15] Joel Friedman. 2003. A Proof of Alon's Second Eigenvalue Conjecture. In *Proceedings of the 35th ACM Symposium on the Theory of Computing (STOC)*. 720–724. https://doi.org/10.1145/780542.780646 arXiv: cs/0405020

[16] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. 2018. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics [Full version]. arXiv: 1806.01838

[17] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. 2019. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st ACM Symposium on the Theory of Computing (STOC)*. 193–204. https://doi.org/10.1145/3313276.3316366 arXiv: 1806.01838

[18] András Gilyén and Umesh Vazirani. 2020. (Sub)Exponential advantage of adiabatic quantum computation with no sign problem. (2020). arXiv: 2011.09495

[19] Matthew B. Hastings. 2020. The Power of Adiabatic Quantum Computation with No Sign Problem. (2020). arXiv: 2005.03791

[20] Matthew B. Hastings and Michael H. Freedman. 2013. Obstructions to Classically Simulating the Quantum Adiabatic Algorithm. *Quantum Information and Computation* 13, 11&12 (2013), 1038–1076. https://doi.org/10.26421/QIC13.11-12 arXiv: 1302.5733

[21] Lin Lin and Yu Tong. 2020. Optimal polynomial based quantum eigenstate filtering with application to solving quantum linear systems. *Quantum* 4 (2020), 361. https://doi.org/10.22331/q-2020-11-11-361 arXiv: 1910.14596

[22] Guang Hao Low and Isaac L. Chuang. 2017. Hamiltonian Simulation by Uniform Spectral Amplification. (2017). arXiv: 1707.05391

[23] Rolando D. Somma, Daniel Nagaj, and Mária Kieferová. 2012. Quantum Speedup by Quantum Annealing. *Physical Review Letters* 109, 5 (2012), 050501. https://doi.org/10.1103/PhysRevLett.109.050501 arXiv: 1202.6257

[24] Mário Szegedy. 2004. Quantum speed-up of Markov chain based algorithms. In *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science (FOCS)*. 32–41. https://doi.org/10.1109/FOCS.2004.53 arXiv: quant-ph/0401053