Dequantization Barriers for Guided Stoquastic Hamiltonians

Yassine Hamoudi* Yvan Le Borgne† Shrinidhi Teganahally Sridhara‡

Université de Bordeaux, CNRS, LaBRI, France

Last update: February 26, 2026

Abstract

We construct a probability distribution, induced by the Perron–Frobenius eigenvector of an exponentially large graph, which cannot be efficiently sampled by any classical algorithm, even when provided with the *best-possible* warm-start distribution. In the quantum setting, this problem can be viewed as preparing the ground state of a stoquastic Hamiltonian given a guiding state as input, and is known to be efficiently solvable on a quantum computer. Our result suggests that no efficient classical algorithm can solve a broad class of stoquastic ground-state problems.

Our graph is constructed from a class of *high-degree, high-girth spectral expanders* to which self-similar trees are attached. This builds on and extends prior work of Gilyén, Hastings, and Vazirani [Has21; GHV21], which ruled out dequantization for a specific stoquastic adiabatic path algorithm. We strengthen their result by ruling out any classical algorithm for guided ground-state preparation.

1 Introduction

Understanding and analyzing the ground state $|\psi_H\rangle$ of a many-body system represented by a Hamiltonian H acting on m qubits is of fundamental interest in quantum computation with critical applications in quantum chemistry (molecular modeling), materials science (high-temperature superconductivity), and quantum machine learning, where ground states are used to represent optimal model parameters. One of the main obstructions to efficient preparation of ground state or estimation of the ground-state energy is the exponential size of the matrix H. In fact, in the absence of other information apart from the Hamiltonian, the problem of estimating the ground-state energy is computationally hard even for quantum computers; the problem is famously QMA-complete for local Hamiltonians [KSV02]. In practice, we often assume access to *guiding states* $|\psi_{\text{in}}\rangle$, which are states close to the actual ground state (measured by the overlap $\langle\psi_{\text{in}}\,|\,\psi_H\rangle$), as a computational resource for the ground-state preparation/energy estimation problems. We refer to this setting as the *Guided Ground-State Preparation* (GGSP) problem and to the related energy estimation problem as *Guided Ground-State Energy Estimation* (GGSE).

Guiding states can be seen as analogous to *warm starts* in Markov Chain Monte Carlo algorithms [AF02], ansätze in variational quantum algorithms [TCC+22], or the intermediate states along a path of adiabatic quantum computation [AL18]. Guiding states can also be classified based on their closeness to the actual ground state and the existence of efficient algorithms to prepare them or query their amplitudes. The well-studied types include guiding states with constant or $1/\operatorname{poly}(m)$ *global* overlap, *strong* guiding states with constant or $1/\operatorname{poly}(m)$ *pointwise* overlap [Bra15], *succinct* guiding states with efficient classical circuits for amplitude evaluation

*ys.hamoudi@gmail.com

†borgne@labri.fr

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

[Liu21; GG22; Jia25; Gal25], etc. In the presence of these guiding states, the GGSP/GGSE problems have been proved to be efficiently solvable by quantum algorithms and, in some cases, even by classical algorithms under additional assumptions on H and $|\psi_{\text{in}}\rangle$.

Dequantization. Recently, there have been several results on *dequantized* algorithms [GG22; Gal25; CFG+23] for the GGSE problem (in the case of local Hamiltonians). These *classical* algorithms usually assume sampling access to the guiding state, along with amplitude access to the guiding state via an evaluation oracle. For instance, Le Gall [Gal25] showed that when $|\langle\psi_{\text{in}}\,|\,\psi_H\rangle| = \Omega(1)$ and the required precision is $1/\log(m)$, there exists a classical algorithm for energy estimation that runs in $\text{poly}(m)$ time. Another line of classical algorithms for the GGSP problems is the Quantum Monte Carlo (QMC) method, a widely used heuristic in computational condensed matter physics. Bravyi and Terhal [BT10] showed that when H is stoquastic and frustration-free, then the ground-state preparation is classically tractable. Later, in [BCGL23], it was shown that there exists a rapidly mixing Markov Chain that can sample from the ground state of a *gapped*, local H when given access to ratios of amplitudes of the ground state via an oracle, along with a starting state satisfying a mild technical condition. However, implementing the amplitude-ratio oracle is unknown for general Hamiltonians, including stoquastic Hamiltonians that are not frustration-free.

General barriers. Dequantization results have led to the question of *to what extent classical algorithms can suffice to solve the GGSP/GGSE problems*. This question has received most of its attention when the guiding state is *untrusted*, i.e., claimed by the prover in prover-verifier protocols. While the general energy estimation problem in this setting is famously QMA-complete [KSV02; GHLS14], restricting to Hamiltonians that admit specific guiding states makes it QCMA-complete [WFC24] or MA-complete [Jia25]. Turning to the *trusted* case, where a guiding state is given as input (as in the present work), GGSE is BQP-complete when the required precision is $1/\text{poly}(m)$ [CFG+23; GG22]. For weaker precision, the problem becomes solvable in BPP [GG22; Gal25].

Stoquastic barriers. A further restriction on Hamiltonians is to assume them to be *stoquastic*, namely, that all their off-diagonal entries are real and non-positive in some given basis. Such Hamiltonians are particularly interesting because of their classical flavor. Indeed, the ground state of a stoquastic Hamiltonian can be chosen to have non-negative real entries, which allows it to be interpreted as a probability distribution. Most Hamiltonians arising in quantum architectures like D-Wave come from Transverse Field Ising Models (TIM), which are stoquastic in the standard basis. Other examples of stoquastic Hamiltonians include the Heisenberg model on bipartite graphs and the bosonic Hubbard model. Since a classical objective function can be seen as a diagonal Hamiltonian and hence stoquastic, the ground-state properties of stoquastic Hamiltonians are of considerable importance in optimization. The general energy estimation problem becomes StoqMA-complete when H is stoquastic, with $\text{MA} \subseteq \text{StoqMA} \subseteq \text{SBP} \cap \text{QMA}$ [BDOT08; BBT06]. In the untrusted setting, the problem is MA-complete under different notions of guiding states [Bra15; Liu21]. In the trusted setting, the problem is only known to be BPP-hard [Wai25].

Connection to Adiabatic Quantum Computation. Adiabatic quantum computation is a leading framework for solving ground-state preparation problems on a quantum computer when a target Hamiltonian has no available guiding state initially [AT07]. The paradigm relies on designing a sequence (or *path*) of slowly varying Hamiltonians whose spectral gap remains open, ensuring – by the adiabatic theorem – that the system stays close to its instantaneous ground state throughout the evolution. From a computational perspective, this process can be viewed as a *bootstrapped* guided ground-state preparation algorithm, where access to intermediate ground

states enables the preparation of the final target state. The possibility of dequantizing adiabatic algorithms has been raised in several prior works. For instance, [BDOT08] asked whether *"every efficient adiabatic path using stoquastic Hamiltonians can be simulated by a polynomial-time probabilistic machine"*. Positive results in this direction include a BPP algorithm for simulating any path of stoquastic frustration-free Hamiltonians [BT10], and a PostBPP algorithm for simulating paths of general stoquastic Hamiltonians [BDOT08]. On the negative side, recent breakthrough results rule out the possibility of dequantizing a *specific* stoquastic path in BPP, relative to an oracle [Has21; GHV21]. Our results align with these latter works but have broader implications: we rule out any dequantization of an adiabatic path – stoquastic or not – that must pass through the stoquastic Hamiltonian constructed in our paper.

Connection to other exponential speedups. Our contribution can also be viewed as part of a line of work showcasing exponential quantum speedups based on oracular graph constructions with particular spectral properties. A foundational result in this direction is due to Childs et al. [CCD+03] for the so-called *welded-tree problem*. These ideas have since been extended to demonstrate exponential quantum speedups in a variety of settings, including quantum-walk-based search algorithms [CCD+03; LLL24; BLH25; JZ23; LZ25], quantum property testing [BCG+24], quantum annealing [SNK12], adiabatic computing [Has21; GHV21], quantum distributed computing [GLMR25], and quantum optimization [LWWZ25].

1.1 Contributions

Our main result is a no-go theorem for the efficient dequantization of GGSP for stoquastic Hamiltonians, even when the algorithm has access to an arbitrarily good guiding state. Below, we give an informal definition of the problem.

Definition 1.1 (Informal definition of GGSP)**.** We are given an m-qubit stoquastic Hamiltonian H with a unique *ground state* $|\psi_H\rangle$ and a spectral gap of at least $1/\operatorname{poly}(m)$. In addition, we are given $\operatorname{poly}(m)$ copies of a *guiding state* $|\psi_{\text{in}}\rangle$ satisfying $|\langle\psi_{\text{in}}\,|\,\psi_H\rangle| \geq 1/\operatorname{poly}(m)$. The goal is to output a *refined state* $|\psi_{\text{out}}\rangle$ such that $|\langle\psi_{\text{out}}\,|\,\psi_H\rangle| \geq 3/4$.

For classical algorithms, the input is represented by $\operatorname{poly}(m)$ i.i.d. samples from the distribution $(\langle x\,|\,\psi_{\text{in}}\rangle^2)_x$, and the output must be a sample from the distribution $(\langle y\,|\,\psi_{\text{out}}\rangle^2)_y$ *that is also* 1/5-independent from the input samples.

Note that we allow guiding states that need not be easy to prepare, including guiding states that are exactly equal to the ground state. This generality enables us to understand the power of a wide range of guiding-state preparation procedures, such as more general adiabatic paths than those previously considered (e.g., in [Has21; GHV21]) and arbitrary ansätze used as guiding states.

We focus on GGSP rather than ground-state energy estimation because the ability to sample from or prepare the ground state is useful for a variety of tasks beyond energy estimation, such as estimating expectation values of general observables on the ground state, or simulating its real-time dynamics. It also helps decouple the complexity of these problems from the hardness of preparing good guiding states. For instance, the dequantized energy estimation algorithm of [Gal25] runs in $\operatorname{poly}\big(\frac{1}{|\langle\psi_{\text{in}}\,|\,\psi_H\rangle|^{\Delta}}, m\big)$ (where Δ is the desired energy precision), which could be improved given the ability to boost the overlap $|\langle\psi_{\text{in}}\,|\,\psi_H\rangle|$ via an independent GGSP algorithm. Our result largely rules out this possibility by establishing classical hardness of GGSP for a stoquastic Hamiltonian, even in the presence of an arbitrary guiding state. Informally, we establish this classical hardness by proving the following result.

Theorem 3.2 and Proposition 3.1 (Informal). *There exists an m-qubit stoquastic Hamiltonian H for which the GGSP problem can be solved efficiently (in* $\operatorname{poly}(m)$ *time) by a quantum algorithm, whereas no classical algorithm can solve it using at most 2^{m^c} queries to the oracle of H, for some constant $c < 1$. This result holds* regardless *of the guiding state provided as input.*

On the independence requirement. An important requirement of Definition 1.1 – in the classical setting – is that the output sample must be (almost) independent of the input samples. Without this condition, an algorithm could sometimes reuse the same input as output (for example, when the guiding state already coincides with the ground state). This would pose a significant practical limitation. For instance, standard concentration bounds, such as Chernoff or Chebyshev inequalities, rely on (nearly) independent samples to hold. Producing multiple correlated samples would therefore prevent estimation of statistics of the ground-state distribution via such bounds. A similar consideration arises in the MCMC literature. For instance, when estimating partition functions via simulated annealing, Markov chains are initialized from a warm-start distribution (playing the role of the guiding state) and run long enough to produce fresh, independent samples from the stationary distribution (see, e.g., [JSV04, Section 6] and [ŠVV09, Section 7]). One may wonder why we do not impose a similar independence requirement in the quantum setting. Due to the no-cloning theorem, it is generally impossible to produce the ground state without entirely consuming a copy of the guiding state. Nevertheless, quantum statistical estimators can behave differently from classical ones. For example, quantum speedups for MCMC methods can rely on non-destructive quantum primitives [HW20; CH23], which make it possible to use a single copy of the ground state $|\psi_H\rangle$ as a fixed point, together with multiple applications of the reflection $\mathbb{I} - 2|\psi_H\rangle\langle\psi_H|$.

1.2 Proof overview

We provide a high-level overview of the main ideas behind the proof of Theorem 3.2. Our result holds relative to an oracle representing the adjacency matrix of a graph G constructed in this paper.

Underlying Graph. We construct a graph G (Definition 2.7 and Figure 1) that is obtained by taking a graph E which is a high-degree, high-girth spectral expander and then attaching many copies of self-similar trees $\widehat{T}$ to its nodes (these are similar to the trees studied in [Has21; GHV21]). G is constructed in a way that it is degree-regular everywhere except at the leaves. The expander results in G having a large spectral gap and no short cycles. The stoquastic Hamiltonian H acting on m qubits is the negative of the adjacency matrix of G over 2^m vertices, i.e., $H = -A_G$. The only assumption on the guiding state $|\psi_{\text{in}}\rangle$ is that it has at least inverse-polynomial overlap with $|\psi_H\rangle$ since, without this assumption, even a quantum computer may not be able to efficiently solve the GGSP problem. We show that *no such guiding state* can simultaneously enable an efficient classical algorithm.

Hardness of classical exploration. The graph G is designed so that it is difficult for classical algorithms to distinguish locally between the expander part and the tree part (Lemmas 2.11 and 3.7). Indeed, since the expander has high girth and the final graph is degree-regular except at the leaves, it is difficult to distinguish the expander vertices from the tree vertices in a small neighborhood. Moreover, the self-similar trees have a sufficiently confusing structure that makes it difficult to infer one's position in the graph from their leaves (Lemma 3.8 and Appendix B). For instance, a simple random walk on G is insufficient for sampling from the ground state of H, since its stationary distribution is the degree distribution rather than the ground-state distribution. While more sophisticated classical algorithms might sample from the correct distribution, our main result is to show that none can do so efficiently.

Localization property. The classical hardness is established by examining a specific *localization* property of the ground state $|\psi_H\rangle$, which must also be approximately satisfied by the guiding state $|\psi_{\text{in}}\rangle$ and the output state $|\psi_{\text{out}}\rangle$. This property states that, upon sampling multiple vertices from the ground-state distribution, the distance between the *nearest vertices* on the expander

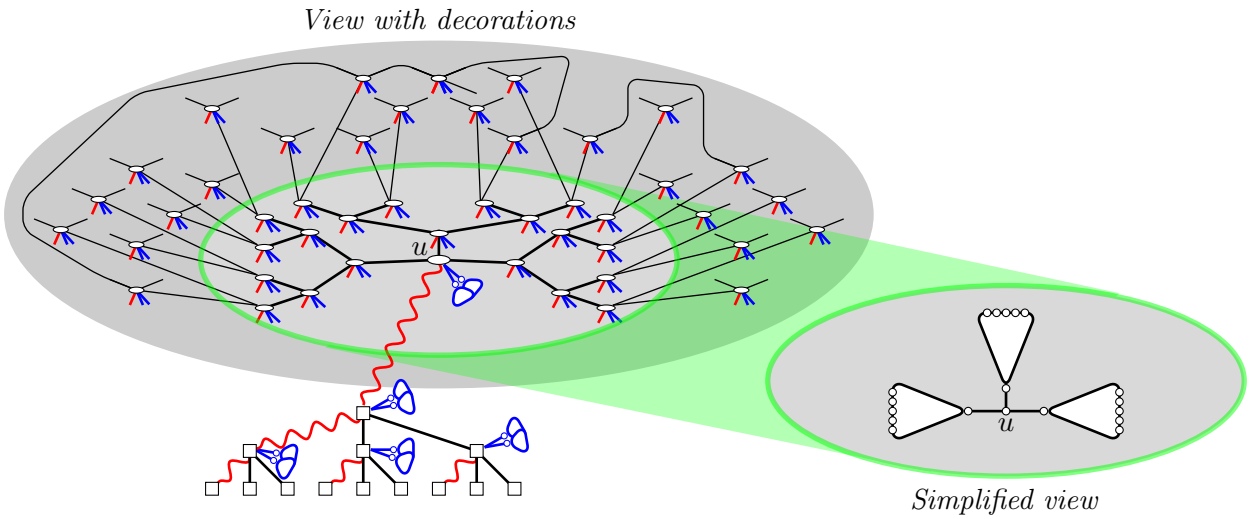

Figure 1: Pictorial representation of the graph G, with the expander E shown inside the gray ellipse (circular vertices) and details of the self-similar trees attached to a vertex u (square vertices). The expander is locally tree-like within a ball of radius equal to half the girth around u (shown by the green boundary). The graph G locally resembles a self-similar tree – see Lemma 2.11. The simplified view depicts the local tree structure around u, drawing the incident edges within the expander, but omitting the decorations arising from the blue and red edges.

graph E is large with very high probability (Lemma 3.3). This follows essentially from the fact that the marginal ground-state distribution over the expander vertices is uniform (Lemma 2.9). The localization property must also hold for the distributions arising from $|\psi_{\text{in}}\rangle$ and $|\psi_{\text{out}}\rangle$ with constant probability (Proposition 3.5).

Hardness of classical sampling. Finally, we demonstrate that any classical algorithm making fewer than 2^{m^c} queries to H for some $c < 1$ (i.e., sub-exponentially many queries) cannot reproduce the above localization property in its output distribution with more than exponentially small probability (Proposition 3.9). By standard arguments, a classical algorithm can be assumed to explore the graph G by growing connected components around the input sampled vertices (Lemma 3.6). However, since the graph is difficult to explore, none of these connected components can reach an expander vertex sufficiently far from its starting point (Lemma 3.7). It follows that such an algorithm cannot satisfy the localization property with high probability, and therefore cannot solve the GGSP problem on H.

2 Definition of the setup

2.1 The Guided Ground-State Preparation (GGSP) problem

The Guided Ground-State Preparation problem (GGSP) is formally defined below. In the quantum setting, the goal is to prepare the ground state of a gapped Hamiltonian, given access to copies of a guiding state. In the classical setting, quantum states are replaced by classical samples drawn from the corresponding standard basis measurement distributions. We use $\text{poly}(m)$ as shorthand for m^c for some absolute constant c. The constants $3/4$ and $1/5$ in the output condition are chosen arbitrarily.

Definition 2.1. The *Guided Ground-State Preparation* (GGSP) problem is defined as follows. We are given an m-qubit stoquastic Hamiltonian H with bounded norm $\|H\| \leq \mathrm{poly}(m)$, which has a unique *ground state* $|\psi_H\rangle$ and a spectral gap at least $1/\mathrm{poly}(m)$. In addition, we are given the description of $t = \mathrm{poly}(m)$ copies of a *guiding state* $|\psi_{\mathrm{in}}\rangle$ that satisfies $|\langle\psi_{\mathrm{in}} \,|\, \psi_H\rangle| \geq 1/\mathrm{poly}(m)$. We should output the description of a *refined state* $|\psi_{\mathrm{out}}\rangle$ such that $|\langle\psi_{\mathrm{out}} \,|\, \psi_H\rangle| \geq 3/4$. The descriptions of the guiding and refined states are given in one of the following two models:

- *(Quantum input-output)* The algorithm is given the quantum state $|\psi_{\mathrm{in}}\rangle^{\otimes t}$ and must output the quantum state $|\psi_{\mathrm{out}}\rangle$.
- *(Classical input-output)* Let p_{in} and p_{out} denote the probability distributions on $\{0,1\}^m$ obtained by measuring $|\psi_{\mathrm{in}}\rangle$ and $|\psi_{\mathrm{out}}\rangle$, respectively, in the standard basis. The algorithm is given i.i.d. samples $x_1, \ldots, x_t \sim p_{\mathrm{in}}$, and must output one sample $y \sim p_{\mathrm{out}}$ such that the joint distribution of $(x_1, \ldots, x_t, y)$ is 1/5-close in total variation distance to the product distribution $p_{\mathrm{in}}^{\otimes t} \times p_{\mathrm{out}}$.

In the classical setting, the conditions on the input-output states could equivalently be defined in terms of the *fidelities* $F(p_{\mathrm{in}}, p_H) = |\langle\psi_{\mathrm{in}} \,|\, \psi_H\rangle|^2 \geq 1/\mathrm{poly}(m)$ and $F(p_{\mathrm{out}}, p_H) = |\langle\psi_{\mathrm{out}} \,|\, \psi_H\rangle|^2 \geq (3/4)^2$, where $p_H(x) = |\langle x \,|\, \psi_H\rangle|^2$ is the ground-state distribution.

The GGSP problem can be solved quantumly by filtering out the ground-state component of the guiding state. Because this state already has non-negligible overlap with the ground state, quantum algorithms can rapidly amplify this overlap, something classical algorithms cannot do efficiently in general. Below, we describe an elementary approach to achieve this goal (modern tools, such as QSVT, provide alternative methods with similar functionality).

Proposition 2.2 (Quantum algorithm for the GGSP problem)**.** *There exists a quantum algorithm that solves the GGSP problem by outputting the exact ground state $|\psi_{\mathrm{out}}\rangle = |\psi_H\rangle$ of H with cost $\mathrm{poly}(m) \cdot T_H$, where T_H is the cost of simulating the unit-time evolution e^{-iH}.*

Proof. The algorithm runs quantum phase estimation on H using the guiding state $|\psi_{\mathrm{in}}\rangle$ as input. Because the spectral gap is at least inverse polynomial, it suffices to use inverse-polynomial precision to distinguish the smallest-eigenvalue component $|\psi_H\rangle$ from the others. Measuring the energy register then collapses the state to that component with probability equal to the squared overlap $|\langle\psi_{\mathrm{in}} \,|\, \psi_H\rangle|^2 \geq 1/\mathrm{poly}(m)$. The success probability can be amplified to arbitrarily close to 1 either by repeating the procedure on fresh copies of $|\psi_{\mathrm{in}}\rangle$ until the state collapses to $|\psi_H\rangle$, or by using amplitude amplification on a single copy of $|\psi_{\mathrm{in}}\rangle$. □

The simulation cost T_H of H can vary depending on the properties and input specification of H. Examples where T_H is generally small include local Hamiltonians, sparse Hamiltonians with oracle access to their nonzero entries (as is the case in the present work), or more generally when H can be implemented efficiently via a block-encoding.

2.2 Graph construction

We describe the main graph G_n that will lead to hard instances of the GGSP problem. This graph combines a high-degree, high-girth spectral expander with a particular type of self-similar trees. We use a free parameter n in our constructions, which will later be chosen as $n = \mathrm{poly}(m)$ when defining the Hamiltonian derived from the graph.

We first state the properties required of the spectral expanders. A key difficulty is that the vertex degree must grow (logarithmically) with the number of vertices, which rules out constructions with constant degree. Instead, we use the probabilistic method, which shows that random regular graphs satisfy the desired requirements.

Proposition 2.3 (High-degree, high-girth expanders E_n)**.** *There exists a family of regular graphs $(E_n)_n$ on $N_E = 2^{21n^2 \log^2 n}$ vertices such that, for n large enough, E_n has degree $d_n = n$, spectral gap at least $d_n/2$ and girth at least $g_n = 40n^2 \log n + 8$.*

Proof. It follows from [Sar23, Theorem 1.1] and [MWW04, Corollary 1] that a graph chosen uniformly at random from all d_n-regular graphs on N_E vertices has, asymptotically almost surely, a spectral gap of at least $d_n/2$ and girth at least g_n. □

We believe that such expanders can be constructed explicitly, for certain values of n, using number-theoretic Ramanujan graphs [LPS88; Mor94]. Moreover, we expect that the graph constructed in [GHV21] can be derandomized by relying on similar expanders, together with high-girth bipartite regular graphs in place of the random bipartite matchings used in that work.

Next, we describe a family of trees with a fractal-like structure, which we call *self-similar trees*. The construction is based on the notion of *graph decoration*: given two graphs G and G', one attaches a distinct copy of G' to each vertex of the base graph G by adding a new edge connecting that vertex to a distinguished vertex of the corresponding copy of G'. Such a process has been studied in the context of stoquastic Hamiltonian constructions in [Has21; GHV21], and our construction of the trees bears strong similarities to these works. One difference is that we do not decorate the leaves of the base graph. This has the advantage of making the final graph slightly smaller, and ensures that it is degree-regular everywhere except at its root and its leaves.

Definition 2.4 (Self-similar trees $\widehat{T}_{n,k}$)**.** For $n \geq 1$ and $1 \leq k \leq \sqrt{n}$, consider the degree sequence $d_{n,k} = 2n - k\sqrt{n}$ and the depth sequence $\ell_{n,k} = k \cdot 10n^{\frac{3}{2}} \log n$. Then, the following trees are defined,

- The *perfect tree* $C_{n,k}$ is the tree in which every internal vertex has exactly $d_{n,k} - 1$ children and all leaves are at depth $\ell_{n,k}$. The 0*-level decorated tree* $T_{n,k,0}$ is defined to be $C_{n,k}$.
- The r*-level decorated tree* $T_{n,k,r}$ is defined recursively, for $r = 1, \ldots, k-1$, by adding $d_{n,k-r} - d_{n,k-r+1}$ edges to each internal node of $T_{n,k,r-1}$ and attaching the other endpoint of each edge to the root of a new copy of $C_{n,k-r}$.
- The *full-level decorated tree* $\widehat{T}_{n,k}$ is defined to be $T_{n,k,k-1}$.

The r*-level leaves* is the set of all new leaves added in the construction of $T_{n,k,r}$ by decorating $T_{n,k,r-1}$.

The above definition is equivalent to the following construction, which will prove convenient for some of the later analysis.

Fact 2.5 (Bottom-up construction of $\widehat{T}_{n,k}$)**.** *The full-level decorated tree $\widehat{T}_{n,k}$ can equivalently be obtained by, for each $r = 1, \ldots, k-1$, adding $d_{n,r} - d_{n,r+1}$ edges to each internal node of $C_{n,k}$ and attaching the other endpoint of each edge to the root of a new copy of $\widehat{T}_{n,r}$.*

These trees have the following properties.

Fact 2.6. *All vertices in $T_{n,k,r}$ (except the root and the leaves) have degree $d_{n,k-r}$. The number of vertices in $\widehat{T}_{n,k}$ is at most $2^{12n^3 \log^2 n}$.*

We now describe the main graph, which is obtained by decorating the expander defined above with self-similar trees. We also add dummy isolated vertices for technical reasons, following prior work on classical graph exploration (e.g., [CCD+03]). These isolated vertices are used to force classical algorithms to explore a connected component, since otherwise a queried vertex would be isolated with very high probability.

Definition 2.7 (Main graph G_n)**.** For $n \geq 1$ and $K = \sqrt{n}$, define the graph G_n as follows:

Fix any high-degree, high-girth spectral expanders E_n satisfying Proposition 2.3.

For $k = 1, \dots, K-1$, add $d_{n,k} - d_{n,k+1}$ edges to each node of E_n and attach the other endpoint of each edge to the root of a new copy of $\widehat{T}_{n,k}$.

Pad the construction with $2^{15n^3 \log^2 n} - N_E \cdot (1 + \sum_{k=1}^{K-1} |\widehat{T}_{n,k}|)$ isolated vertices.

We represent the vertices of G_n by the abstract sets $V = V_E \cup V_T \cup V_I$, where V_E are the vertices coming from E_n (called the *expander vertices*), V_T are the vertices coming from the trees $\widehat{T}_{n,k}$ (called the *tree vertices*), and V_I are the other *isolated vertices*.

For each tree vertex $v \in V_E \cup V_T$, we let $\mathrm{ex}(v) \in V_E$ denote the closest expander vertex to v in the graph (this is v itself when $v \in V_E$, and the expander vertex to which the tree containing v is attached when $v \in V_T$).

The adjacency matrix of G_n is denoted by A_n, and its top eigenvector by $|\psi_n\rangle$.

Fact 2.8. *All vertices in G_n (except the leaves and the isolated vertices) have degree $d_{n,1}$. The graph G_n has the same girth as E_n. The number of vertices is $|V| = 2^{15n^3 \log^2 n}$.*

There are two crucial properties of G_n that will be used to demonstrate a quantum speedup later. First, G_n inherits the spectral properties of the expander E_n, in the sense that it has a large spectral gap and that its top eigenvector has a large overlap with the ground state of E_n.

Lemma 2.9 (Spectral properties of the adjacency matrix A_n)**.** *The difference between the two largest eigenvalues of A_n is at least $n/2 - 4\sqrt{2n}$. The top eigenvector $|\psi_n\rangle$ of A_n can be written as,*

$$|\psi_n\rangle = |\psi_E\rangle + \sum_{v \in V_E} |\psi_{v,T}\rangle$$

where $|\psi_E\rangle$ is a uniform (unnormalized) superposition over the expander vertices $\{|v\rangle\}_{v \in V_E}$, and each $|\psi_{v,T}\rangle$ is a superposition over the vertices in the trees attached to v. Moreover, all states in $\{|\psi_{v,T}\rangle\}_{v \in V_E}$ are identical (up to a basis permutation), and $\||\psi_E\rangle\|^2 \geq 1 - O(1/n)$.

Proof. The proof is deferred to Appendix A. It relies on generic spectral properties of decorated graphs, similar to those used in [Has21; GHV21]. □

The second property is that, locally, G_n resembles the tree $\widehat{T}_{n,K}$ along any edge within the expander graph. This will make it very difficult for classical algorithms to distinguish between these two graphs, although their spectral properties are radically different. We illustrate a simplified version of that in Figure 1 and in the next lemma. A stronger version is given in Figure 2 and Lemma 3.7, where we consider multiple connected components rooted at arbitrary vertices. The property is based on the following distance measure defined on the graph G_n.

Definition 2.10 (Distance dist_E)**.** Given two vertices $u, v \in V$, the distance $\mathrm{dist}_E(u, v)$ is defined as the number of *expander* vertices along the shortest path connecting u and v in G_n. Given a subset $U \subseteq V$ of vertices, we define $\mathrm{dist}_E(U, v) = \min_{u \in U} \mathrm{dist}_E(u, v)$.

Note that dist_E is not a metric, since the distance between two distinct vertices $u \neq v$ can be zero (if they belong to trees attached to the same expander vertex). Nevertheless, it satisfies the triangle inequality. We use dist_E to study locality in the graph G_n as follows.

Lemma 2.11 (Local structure of G_n)**.** *Let $u \in V_E$ be an expander vertex and consider the subgraph $B(u) \subseteq G_n$ induced by vertices within distance $g_n/4 - 1$ of u (according to the distance measure dist_E). Then $B(u)$ admits an embedding into the graph G_n modified as follows: for each edge from u to an expander neighbor, replace it with an edge attached to u and to the root of a new copy of $\widehat{T}_{n,K}$. Moreover, this embedding maps the expander vertices at distance $g_n/4 - 1$ from u to the 0-level leaves of the trees $\widehat{T}_{n,K}$.*

Proof. The subgraph $B(u)$ is necessarily a tree, since the expander graph E_n has no cycles of length less than g_n by the girth property. The expander vertices in $B(u)$ form a full tree of depth $g_n/4 - 1$ and degree $d_n + 1$, to which are attached the decorations specified in Definition 2.7. Observe that, for any expander neighbor v of u, the subtree rooted at v matches exactly the definition of the self-similar tree $\widehat{T}_{n,K}$ from Definition 2.4 and Fact 2.5, using the identities $g_n/4 - 2 = \ell_{n,K}$ and $d_n = d_{n,K}$. □

2.3 Hamiltonian construction

We describe the stoquastic Hamiltonian used to prove a sub-exponential speedup for the GGSP problem. It is given by minus the adjacency matrix A_n of the graph G_n defined in the previous section, with the vertices encoded via a random permutation. Without such a random labeling, the GGSP problem would present no difficulty, since the ground-state distribution would depend only on n and not on the oracle. The randomness of the labeling is used crucially in the proofs of Lemmas 3.6 and 3.8. The Hamiltonian is specified to the algorithm via an adjacency-list oracle.

Definition 2.12 (Hamiltonian H_π)**.** For m sufficiently large, let $n = \mathrm{poly}(m)$ be the integer such that G_n has 2^m vertices. Let $\pi : V \to [2^m]$ be a bijection (a labeling) from the vertex set V of G_n to the range $[2^m]$. Define the m-qubit Hamiltonian H_π as,

$$H_\pi = -P_\pi A_n P_\pi^{-1}$$

where P_π is the permutation matrix corresponding to π (i.e., $(P_\pi)_{x,u} = 1$ if $\pi(u) = x$, and $(P_\pi)_{x,u} = 0$ otherwise). The ground state of H_π is denoted $|\psi_\pi\rangle$.

The Hamiltonian H_π can be queried via the following adjacency-list oracles:

- *(Classical oracle)* Given $x \in [2^m]$, the oracle $\mathcal{O}_\pi$ returns the list $\mathcal{O}_\pi(x) = \Gamma_x \subset [2^m]$ of labels that correspond to the neighbors of the vertex $\pi^{-1}(x)$ in G_n (equivalently, the non-zero entries in the x-th row of H_π).
- *(Quantum oracle)* The oracle implements the unitary operation $\mathcal{O}_\pi|x\rangle|0\rangle \mapsto |x\rangle|\Gamma_x\rangle$.

The spectral properties of H_π follow directly from that of the graph G_n (Lemma 2.9). In particular, its ground state $|\psi_\pi\rangle$ coincides with the top eigenvector $|\psi_n\rangle$ of A_n.

Fact 2.13 (Spectral properties of H_π)**.** *The Hamiltonian H_π is stoquastic, it has a non-degenerate ground space with spectral gap at least $n/2$, and its ground state is $|\psi_\pi\rangle = P_\pi|\psi_n\rangle$.*

Finally, given a guiding state $|\psi_{\mathrm{in}}\rangle$ for the unlabelled graph $-A_n$, we associate the corresponding family of guiding states $\{|\psi_{\mathrm{in},\pi}\rangle\}_\pi$ for the Hamiltonians H_π.

Definition 2.14 (Guiding state $|\psi_{\mathrm{in},\pi}\rangle$)**.** Given a guiding state $|\psi_{\mathrm{in}}\rangle$ for the Hamiltonian $-A_n$, we let $|\psi_{\mathrm{in},\pi}\rangle = P_\pi|\psi_{\mathrm{in}}\rangle$ denote the corresponding guiding state for the Hamiltonian H_π.

3 Main result

In this section, we prove our main result (Theorem 3.2): any classical algorithm that makes only a sub-exponential number of queries to the adjacency-list oracle of the Hamiltonian H_π, even when provided with copies of a guiding state $|\psi_{\mathrm{in},\pi}\rangle$, cannot sample from the ground-state distribution with sufficiently high precision. Before presenting the classical lower bound, we first show that quantum algorithms can solve the problem efficiently.

Proposition 3.1 (Quantum algorithm)**.** *There exists a quantum algorithm that solves the GGSP problem on every Hamiltonian H_π (Definition 2.12), using a single copy of a guiding state $|\psi_{\mathrm{in},\pi}\rangle$, and performing* $\mathrm{poly}(n)$ *quantum queries to the adjacency-list oracle of H_π.*

Proof. The graph G_n (and hence the Hamiltonian H_π) is $d_{n,1}$-sparse, where $d_{n,1} = O(n)$ (Fact 2.8). The algorithm follows immediately from Proposition 2.2, by noting that simulating a $O(n)$-sparse poly(n)-qubit Hamiltonian H requires $T_H = \text{poly}(n)$ quantum gates and queries to its adjacency-list oracle (see, for instance, [AT07, Lemma 1]). □

The main theorem we prove is that classical algorithms performing only sub-exponentially many queries cannot succeed in solving the GGSP problem on H_π with high probability.

Theorem 3.2 (Classical hardness)**.** *Let $a, b \geq 1$ be any constant and $(|\psi_{\text{in},\pi}\rangle)_\pi$ a family of guiding states for $\{H_\pi\}_\pi$ (Definition 2.14) with $|\langle\psi_{\text{in},\pi}\,|\,\psi_\pi\rangle| \geq 1/n^a$ for all π. Suppose that a classical algorithm is given access to H_π for a random labeling π, as well as $t = n^b$ independent samples $x_1, \ldots, x_t \sim p_{\text{in},\pi}$ from the distribution $p_{\text{in},\pi}(x) = \langle x\,|\,\psi_{\text{in},\pi}\rangle^2$.*

Then, for n large enough, any algorithm must make at least $2^{\sqrt{n}}$ queries to the adjacency-list oracle of H_π in order to output a sample $y \sim p_{\text{out},\pi}$ from a distribution $p_{\text{out},\pi}(y) = \langle y\,|\,\psi_{\text{out},\pi}\rangle^2$ arising from a refined state $|\psi_{\text{out},\pi}\rangle$ such that $|\langle\psi_{\text{out},\pi}\,|\,\psi_\pi\rangle| \geq 3/4$, and such that the joint distribution J_π of $(x_1, \ldots, x_t, y)$ is within total variation distance $1/5$ of $p_{\text{in},\pi}^{\otimes t} \times p_{\text{out},\pi}$.

The theorem is proved by analyzing a certain localization property of the ground state $|\psi_\pi\rangle$ of H_π, which must also be satisfied (at least to some extent) by the output state $|\psi_{\text{out},\pi}\rangle$. This property corresponds to the event that the output vertex lies at distance at least $g_n/2$ from the input vertices, measured according to dist_E (Definition 2.10). First, we show in Section 3.1 that any input-output distribution that correctly solves the GGSP problem satisfies this property with probability at least $1/10$.

Proposition 3.5 (Input-Output state localization)**.** *Suppose that a classical algorithm solves the GGSP problem on $\{H_\pi\}_\pi$ under the conditions of Theorem 3.2. Let J_π denote the joint distribution of the input-output samples $(x_1, x_2, \ldots, x_t, y)$ of the algorithm under the vertex labeling π. Then, the probability that the output vertex y lies at distance at least $g_n/2$ from the input vertices satisfies,*

$$\Pr_{(x_1,\ldots,x_t,y)\sim J_\pi}\left[\text{dist}_E(\pi^{-1}(\{x_1, \ldots, x_t\}), \pi^{-1}(y)) \geq g_n/2\right] \geq 1/10. \tag{1}$$

Conversely, we prove in Section 3.2 that for any classical algorithm making at most $2^{\sqrt{n}}$ queries, the probability of the above event is exponentially small. We stress that this result holds for a random π, in contrast to the proposition above which is valid for all π.

Proposition 3.9 (Hardness of exploring the graph G_n)**.** *Suppose that a classical algorithm is given access to H_π for a random labeling π, as well as input samples $x_1, \ldots, x_t$ from a distribution as specified in Theorem 3.2. Suppose that it outputs an additional sample y, and let J_π denote the joint distribution of the input-output samples $(x_1, x_2, \ldots, x_t, y)$. If the algorithm makes fewer than $2^{\sqrt{n}}$ queries, then the probability the output vertex y lies at distance at least $g_n/2$ from the input vertices satisfies,*

$$\Pr_{\pi,(x_1,\ldots,x_t,y)\sim J_\pi}\left[\text{dist}_E(\pi^{-1}(\{x_1, \ldots, x_t\}), \pi^{-1}(y)) \geq g_n/2\right] \leq 2^{-n\log n}. \tag{2}$$

We can now conclude the proof of the main theorem.

Proof of Theorem 3.2. Fix $|\psi_{\text{in}}\rangle$ to be any guiding state for the Hamiltonian $-A_n$. Consider a classical algorithm that solves the GGSP problem on every Hamiltonian H_π, using the guiding state $|\psi_{\text{in},\pi}\rangle = P_\pi|\psi_{\text{in}}\rangle$ as specified in Definition 2.14 and Theorem 3.2. Suppose, for the sake of contradiction, that it makes fewer than $2^{\sqrt{n}}$ queries. Then, by Proposition 3.9, there exists at least one labeling π for which its input-output distribution J_π violates Equation (1). It implies by Proposition 3.5 that the algorithm cannot be correct on H_π. □

3.1 Proof of ground-state localization properties

In this section, we lower bound the probability that a successful algorithm solving the GGSP problem samples a vertex at a large distance from its input. First, we prove the statement under the exact ground-state distribution.

Lemma 3.3 (Ground-state localization)**.** *Let $U \subseteq V$ be any subset of vertices of G_n. For any integer g, the probability that a vertex sampled from the ground-state distribution $p_n(v) = \langle v \,|\, \psi_n \rangle^2$ of $-A_n$ lies at distance at least g from U is at least,*

$$\Pr_{v \sim p_n}[\mathrm{dist}_E(U, v) \geq g] \geq 1 - \frac{|U| \cdot n^g}{N_E}$$

where N_E is the size of the expander E_n in Proposition 2.3.

Proof. First, we prove the lemma when $U = \{u\}$ is of size 1. Let p_E be the conditional distribution of p_n on the set of expander vertices V_E. By Lemma 2.9, when v is sampled according to p_n, the corresponding expander vertex $\mathrm{ex}(v)$ follows the distribution p_E. Hence,

$$\Pr_{v \sim p_n}[\mathrm{dist}_E(u, v) < g] = \Pr_{v \sim p_E}[\mathrm{dist}_E(u, v) < g]. \tag{3}$$

By Lemma 2.9, p_E is the uniform distribution over V_E. Moreover, by construction, the expander graph over V_E has degree $d_n = n$. Hence, the number of expander vertices v that are at distance at most g from $\mathrm{ex}(u)$ is at most $1 + n + n^2 + \cdots + n^{g-1} \leq n^g$. Consequently, the probability of sampling v at distance $< g$ from $\mathrm{ex}(u)$, under the distribution p_E, can be upper bounded by the ratio between the number of expander vertices at distance at most g from $\mathrm{ex}(u)$ and the total number N_E of expander vertices:

$$\Pr_{v \sim p_E}[\mathrm{dist}_E(u, v) < g] \leq \frac{n^g}{N_E}. \tag{4}$$

Equations (3) and (4) imply that $\Pr_{v \sim p_n}[\mathrm{dist}_E(u, v) \geq g] \geq 1 - \frac{n^g}{N_E}$.

When U is of arbitrary size, the lemma follows by applying the above result together with a union bound over the $|U|$ vertices $u \in U$. □

The previous lemma analyzes the ideal case in which the output vertex is sampled according to the *ideal* ground-state distribution p_n. However, the GGSP problem allows outputting vertices from a perturbed distributions p_{out}, which may adversarially affect the occurrence of the above event. The next result shows that the event still occurs in this setting, at least with constant probability, when $|U| = t$ is of polynomial size and $g = g_n/2$ is half the girth of the expander E_n.

Corollary 3.4. *Consider a joint distribution J over $t + 1$ vertices $(u_1, u_2, \ldots, u_t, v)$ of G_n. Suppose that J is within total variation distance $1/5$ of a product distribution $p_{\mathrm{in}}^{\otimes t} \times p_{\mathrm{out}}$, where $F(p_{\mathrm{out}}, p_n) \geq (3/4)^2$. Then, for n large enough, the probability that the vertex v lies at distance at least $g_n/2$ from the other vertices satisfies,*

$$\Pr_{(u_1, \ldots, u_t, v) \sim J}[\mathrm{dist}_E(\{u_1, \ldots, u_t\}, v) \geq g_n/2] \geq 1/10.$$

Proof. The result is shown by upper-bounding the total variation distance between the distributions $p_{\mathrm{in}}^{\otimes t} \times p_n$ and J, and then applying the result from Lemma 3.3.

The total variation distance is bounded as follows,

$$\begin{aligned}
\mathrm{TV}(p_{\mathrm{in}}^{\otimes t} \times p_n, J) &\leq \mathrm{TV}(p_{\mathrm{in}}^{\otimes t} \times p_n, p_{\mathrm{in}}^{\otimes t} \times p_{\mathrm{out}}) + \mathrm{TV}(p_{\mathrm{in}}^{\otimes t} \times p_{\mathrm{out}}, J) \quad \text{(by the triangle inequality)} \\
&\leq \sqrt{1 - F(p_{\mathrm{out}}, p_n)} + \mathrm{TV}(p_{\mathrm{in}}^{\otimes t} \times p_{\mathrm{out}}, J) \\
&\qquad \text{(by relating the TV distance and the fidelity)} \\
&\leq \sqrt{1 - \frac{9}{16}} + \frac{1}{5} \leq 0.87. \qquad \text{(by the assumptions of the proposition)}
\end{aligned}$$

Hence, the probability of sampling a vertex at distance $g_n/2$ under J is at least,

$$\begin{aligned}
&\Pr_{(u_1,\ldots,u_t,v)\sim J}[\mathrm{dist}_E(\{u_1,\ldots,u_t\},v) \geq g_n/2] \\
&\quad \geq \Pr_{(u_1,\ldots,u_t,v)\sim p_{\mathrm{in}}^{\otimes t}\times p_n}[\mathrm{dist}_E(\{u_1,\ldots,u_t\},v) \geq g_n/2] - \mathrm{TV}(p_{\mathrm{in}}^{\otimes t}\times p_n, J) \\
&\qquad\qquad \text{(by definition of the TV distance)} \\
&\quad \geq \min_{U\subseteq V:|U|=t} \Pr_{v\sim p_n}[\mathrm{dist}_E(U,v) \geq g_n/2] - \mathrm{TV}(p_{\mathrm{in}}^{\otimes t}\times p_n, J) \\
&\qquad\qquad \text{(by the independence of the distributions)} \\
&\quad \geq 1 - \frac{t\cdot n^{g_n/2}}{N_E} - 0.87 \qquad \text{(by Lemma 3.3)} \\
&\quad \geq 1/10. \qquad \text{(for } n \text{ large enough)}
\end{aligned}$$

□

As a direct corollary, the same result holds for the input-output samples of any algorithm solving the GGSP problem, since the distributions J and J_π are the same object up to a relabeling of the vertices.

Proposition 3.5 (Input-Output state localization)**.** *Suppose that a classical algorithm solves the GGSP problem on $\{H_\pi\}_\pi$ under the conditions of Theorem 3.2. Let J_π denote the joint distribution of the input-output samples $(x_1, x_2, \ldots, x_t, y)$ of the algorithm under the vertex labeling π. Then the probability that the output vertex y lies at distance at least $g_n/2$ from the input vertices satisfies,*

$$\Pr_{(x_1,\ldots,x_t,y)\sim J_\pi}\left[\mathrm{dist}_E(\pi^{-1}(\{x_1,\ldots,x_t\}),\pi^{-1}(y)) \geq g_n/2\right] \geq 1/10.$$

3.2 Proof of hardness for classical algorithms

This section proves a converse result to the previous one. Namely, it demonstrates that *no* classical algorithm can output a vertex far from the input vertices unless it makes at least sub-exponentially many queries to H_π.

First, we show that any classical algorithm operating on the graph G_n is forced to explore the graph by querying connected components around its input vertices. This is similar to the argument in [CCD+03].

Lemma 3.6 (Local exploration)**.** *Let $U \subseteq V$ be any subset of vertices of G_n with $|U| \leq |V|/2$. Suppose that a classical algorithm is given access to the adjacency-list oracle $\mathcal{O}_\pi$, for a random π, together with $\pi(U)$, and that it makes q queries. Then, except with probability at most $q\cdot 2^{-2n^3\log^2 n}$, the set of all vertices queried by the algorithm at any point during its execution induces a collection of connected components rooted at the vertices in U, together with some isolated vertices in V_I.*

Proof. This is because the set of isolated vertices V_I in Definition 2.7 constitutes an overwhelming majority of the vertices of the graph G_n. Hence, if an algorithm attempts to query a vertex that is not connected to any of its previously queried vertices, then, due to the randomness of the labeling π, it has overwhelming probability of querying an isolated vertex.

In more detail, by Definition 2.7, the ratio between the number of non-isolated vertices and the total number of (non-input) vertices in G_n is at most $(|V_T| + |V_E|)/(|V| - |U|) \leq 2(|V_T| + |V_E|)/|V| = 2N_E(1 + \sum_{k=1}^{K-1}|\widehat{T}_{n,k}|)/2^{15n^3\log^2 n} \leq 2^{-2n^3\log^2 n}$. Hence, by a union bound over the q queries made by an algorithm, the probability that it queries a non-isolated vertex that is not connected to any of its previously explored vertices is at most $q\cdot 2^{-2n^3\log^2 n}$. □

We now analyze how local graph exploration around an input set of vertices U in G_n behaves. The case $|U| = 1$ was addressed in Figure 1 and Lemma 2.11, where we showed that, within a ball of radius $g_n/4$, the graph G_n is identical to the self-similar tree $\widehat{T}_{n,K}$. The next lemma treats the case $|U| > 1$. The analysis is more involved, as different connected components may intersect and cycles may appear in the exploration graph. We show that such events must be confined to a small superset $\overline{U} \supseteq U$ of the input vertices. The remainder of the local exploration departing from these vertices can then be described by the tree $\widehat{T}_{n,K}$. This is illustrated in Figure 2.

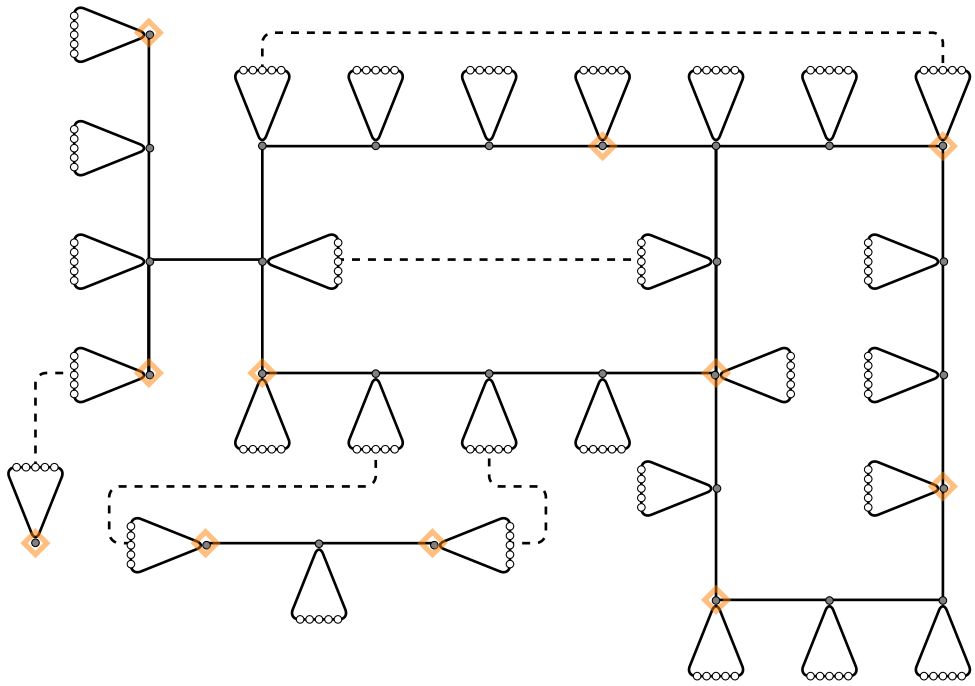

Figure 2: Pictorial representation of the local structure of the graph G_n with girth $g_n = 12$ (only expander vertices are shown). The orange diamonds represent the vertices in a set U. The solid edges connect pairs of vertices in U at distance $< g_n/2$ from one another, so that the set $\overline{U}$ corresponds to the greyed vertices. The triangles depict the local tree structure within distance $< g_n/4$ around $\overline{U}$. The white circles represent the vertices identified with the 0-level leaves of $\widehat{T}_{n,K}$. The dashed lines indicate other paths in the expander graph that may create cycles but lie too far from $\overline{U}$ to be detected by a local exploration.

Lemma 3.7 (Local structure of G_n around multiple roots)**.** *Let $U \subseteq V$ be any subset of vertices of G_n. For any two vertices $u_1, u_2 \in U$, let $\mathrm{path}(u_1, u_2)$ denote the subset of expander vertices along a shortest path between u_1 and u_2. Define the subset of expander vertices,*

$$\overline{U} = \{\mathrm{ex}(u) : u \in U\} \cup \{\mathrm{path}(u_1, u_2) : u_1, u_2 \in U, \mathrm{dist}_E(u_1, u_2) < g_n/2\} \subseteq V_E. \tag{5}$$

Consider the subgraph $B(\overline{U}) \subseteq G_n$ induced by vertices within distance $g_n/4 - 1$ of $\overline{U}$ (according to the distance measure dist_E). Then $B(\overline{U})$ admits an embedding into the graph G_n modified as follows: for each edge from $u \in \overline{U}$ to an expander neighbor outside $\overline{U}$, replace it with an edge attached to u and to the root of a new copy of $\widehat{T}_{n,K}$. Moreover, this embedding maps the expander vertices at distance $g_n/4 - 1$ from $\overline{U}$ to the 0-level leaves of the trees $\widehat{T}_{n,K}$.

Proof. We will use the fact that, between any two expander vertices, there is at most one path of length less than $g_n/2$. This follows from the expander having no cycle of length less than g_n.

Consider any vertex $u \in \overline{U}$. Let $v \in V_E \setminus \overline{U}$ be an expander neighbor of u, and let $T(u, v)$ denote the subgraph of $B(\overline{U})$ consisting of the vertices reachable from v *after removing* the

edge (u, v). The lemma can be reformulated as showing 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.

We first claim that $T(u, v)$ consists exactly of the vertices at distance at most $g_n/4-2$ from v. Suppose, for contradiction, that $T(u, v)$ contains a vertex $w \in V_E$ at distance $g_n/4-1$. Then $\mathrm{dist}_E(u, w) = g_n/4$. By the definition of $\overline{U}$, there must exist another vertex $u' \in \overline{U}$ such that $\mathrm{dist}_E(u', w) \leq g_n/4-1$. By the triangle inequality, it follows that $\mathrm{dist}_E(u, u') \leq g_n/2-1$. Hence, all vertices on $\mathrm{path}(u, u')$ must belong to $\overline{U}$, including the vertex v, contradicting the assumption that $v \in V_E \setminus \overline{U}$.

Next, we show that for any two pairs $(u_1, v_1) \neq (u_2, v_2)$, the subgraphs $T(u_1, v_1)$ and $T(u_2, v_2)$ are disjoint. If $\mathrm{dist}_E(u_1, u_2) \geq g_n/2$ then it follows from the previous paragraph that $T(u_1, v_1)$ and $T(u_2, v_2)$ cannot intersect. If $\mathrm{dist}_E(u_1, u_2) < g_n/2$ and they do intersect, this would create a cycle of length at most $(g_n/4-1)+(g_n/4-1)+(g_n/2-1) = g_n-3$, contradicting the girth condition.

Finally, $T(u, v)$ induces the tree $\widehat{T}_{n,K}$ by the same arguments as Lemma 2.11: it is necessarily a tree by the girth property, and the expander vertices within it form a full tree of depth $g_n/4-2$ and degree d_n+1, to which are attached the decorations specified in Definition 2.7. This matches the definition of $\widehat{T}_{n,K}$ from Definition 2.4 and Fact 2.5, using the identities $g_n/4-2 = \ell_{n,K}$ and $d_n = d_{n,K}$. □

The next result studies the hardness of exploring the tree $\widehat{T}_{n,K}$, starting from its root and attempting to reach a 0-level leaf.

Lemma 3.8 (Hardness of exploring self-similar trees)**.** *Let $K = \sqrt{n}$, as in Definition 2.7. Suppose a classical algorithm explores the tree $\widehat{T}_{n,K}$, 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 $2^{-2n\log n}$.*

Proof. The proof essentially follows [GHV21]. We provide a detailed version in Appendix B, adapted to our choice of parameters and to our definition of self-similar trees. The lemma is reformulated there as Proposition B.6. □

By combining the three lemmas above, we show that exploring the main graph G_n to reach expander vertices far from a given set U of input vertices is hard.

Proposition 3.9 (Hardness of exploring the graph G_n)**.** *Suppose that a classical algorithm is given access to H_π for a random labeling π, as well as input samples $x_1, \ldots, x_t$ from a distribution as specified in Theorem 3.2. Suppose that it outputs an additional sample y, and let J_π denote the joint distribution of the input-output samples $(x_1, x_2, \ldots, x_t, y)$. If the algorithm makes fewer than $2^{\sqrt{n}}$ queries, then the probability that the output vertex y lies at distance at least $g_n/2$ from the input vertices satisfies,*

$$\Pr_{\pi,(x_1,\ldots,x_t,y)\sim J_\pi}\left[\mathrm{dist}_E(\pi^{-1}(\{x_1,\ldots,x_t\}), \pi^{-1}(y)) \geq g_n/2\right] \leq 2^{-n\log n}.$$

Proof. Let $U = \pi^{-1}(\{x_1, \ldots, x_t\})$ denote the (random) set of input vertices sampled under J_π. By Lemma 3.6, except with probability $2^{\sqrt{n}} \cdot 2^{-2n^3\log^2 n} < 2^{-2n\log n}$, we may assume that the consecutive queries of the algorithm form a set of connected components in G_n rooted at the vertices in U.

Let $v = \pi^{-1}(y)$ be the vertex in G_n output by the algorithm. Observe that $\mathrm{dist}_E(U, v) \geq g_n/2$ implies $\mathrm{dist}_E(\overline{U}, v) \geq g_n/4-1$ (where $\overline{U}$ is defined in Equation (5)), hence we can focus on the latter event to happen.

By Lemma 3.7, if the algorithm were to explore a vertex at distance $g_n/4-1$ from $\overline{U}$, then there must exist an exploration tree attached to one of the vertices in $\overline{U}$ that can be embedded into $\widehat{T}_{n,K}$ and reaches one of its 0-level leaves. However, by Lemma 3.8, reaching a 0-level leaf

from the root of such a tree can occur with probability at most $2^{-2n\log n}$ under $2^{\sqrt{n}}$ queries. Since the number of possible roots to consider for the embedding is at most $|\overline{U}|\cdot d_n \leq t^2\cdot g_n\cdot d_n$, a union bound then implies that the overall success probability is at most $t^2\cdot g_n\cdot d_n\cdot 2^{-2n\log n}\leq 2^{-\frac{3}{2}n\log n}$, for n large enough. □

Acknowledgements

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

[AF02] D. Aldous and J. A. Fill. *Reversible Markov Chains and Random Walks on Graphs*. Unfinished monograph, recompiled 2014, available at http://www.stat.berkeley.edu/~aldous/RWG/book.html. 2002 (cit. on p. 1).

[AL18] T. Albash and D. A. Lidar. "Adiabatic Quantum Computation". In: *Reviews of Modern Physics* 90 (2018), p. 015002 (cit. on p. 1).

[AT07] D. Aharonov and A. Ta-Shma. "Adiabatic Quantum State Generation". In: *SIAM Journal on Computing* 37.1 (2007), pp. 47–82 (cit. on pp. 2, 10).

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

[BCG+24] S. Ben-David, A. M. Childs, A. Gilyén, W. Kretschmer, S. Podder, and D. Wang. "Symmetries, Graph Properties, and Quantum Speedups". In: *SIAM Journal on Computing* 53.6 (2024), FOCS20-368–FOCS20-415 (cit. on p. 3).

[BCGL23] S. Bravyi, G. Carleo, D. Gosset, and Y. Liu. "A Rapidly Mixing Markov Chain from Any Gapped Quantum Many-Body System". In: *Quantum* 7 (2023), p. 1173 (cit. on p. 2).

[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 pp. 2, 3).

[BLH25] S. Balasubramanian, T. Li, and A. W. Harrow. "Exponential Speedups for Quantum Walks in Random Hierarchical Graphs". In: *Communications in Mathematical Physics* 406.9 (2025), p. 209 (cit. on p. 3).

[BP94] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. Vol. 9. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1994 (cit. on p. 18).

[Bra15] S. Bravyi. "Monte Carlo Simulation of Stoquastic Hamiltonians". In: *Quantum Information & Computation* 15.13–14 (2015), pp. 1122–1140 (cit. on pp. 1, 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 pp. 2, 3).

[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. 3, 7, 12).

[CFG+23] C. Cade, M. Folkertsma, S. Gharibian, R. Hayakawa, F. Le Gall, T. Morimae, and J. Weggemans. "Improved Hardness Results for the Guided Local Hamiltonian Problem". In: *Proceedings of the 50th International Colloquium on Automata, Languages, and Programming (ICALP)*. 2023, 32:1–32:19 (cit. on p. 2).

[CH23] A. Cornelissen and Y. Hamoudi. "A Sublinear-Time Quantum Algorithm for Approximating Partition Functions". In: *Proceedings of the 34th Symposium on Discrete Algorithms (SODA)*. 2023, pp. 1245–1264 (cit. on p. 4).

[Gal25] F. Le Gall. "Classical Algorithms for Constant Approximation of the Ground State Energy of Local Hamiltonians". In: *Proceedings of the 33rd European Symposium on Algorithms (ESA 2025)*. 2025, 73:1–73:19 (cit. on pp. 2, 3).

[GG22] S. Gharibian and F. Le Gall. "Dequantizing the Quantum singular value transformation: hardness and applications to Quantum chemistry and the Quantum PCP conjecture". In: *Proceedings of the 54th Symposium on Theory of Computing (STOC)*. 2022, pp. 19–32 (cit. on p. 2).

[GHLS14] S. Gharibian, Y. Huang, Z. Landau, and S. W. Shin. "Quantum Hamiltonian Complexity". In: *Foundations and Trends in Theoretical Computer Science* 10.3 (2014), pp. 159–282 (cit. on p. 2).

[GHV21] A. Gilyén, M. B. Hastings, and U. Vazirani. "(Sub)Exponential Advantage of Adiabatic Quantum Computation with no Sign Prioblem". In: *Proceedings of the 53rd Symposium on Theory of Computing (STOC)*. 2021, pp. 1357–1369 (cit. on pp. 1, 3, 4, 7, 8, 14, 17, 22, 23).

[GLMR25] F. Le Gall, M. Luce, J. Marchand, and M. Roget. *Exponential Quantum Advantage for Message Complexity in Distributed Algorithms*. arXiv:2510.01657 [quant-ph]. 2025 (cit. on p. 3).

[Has21] M. B. Hastings. "The Power of Adiabatic Quantum Computation with No Sign Problem". In: *Quantum* 5 (2021), p. 597 (cit. on pp. 1, 3, 4, 7, 8, 17).

[HJ12] R. A. Horn and C. R. Johnson. *Matrix Analysis*. 2nd ed. Cambridge University Press, 2012 (cit. on p. 21).

[HW20] A. W. Harrow and A. Y. Wei. "Adaptive Quantum Simulated Annealing for Bayesian Inference and Estimating Partition Functions". In: *Proceedings of the 31st Symposium on Discrete Algorithms (SODA)*. 2020, pp. 193–212 (cit. on p. 4).

[Jia25] J. Jiang. "Local Hamiltonian Problem with Succinct Ground State is MA-Complete". In: *PRX Quantum* 6 (2025), p. 020312 (cit. on p. 2).

[JSV04] M. Jerrum, A. Sinclair, and E. Vigoda. "A Polynomial-Time Approximation Algorithm for the Permanent of a Matrix with Nonnegative Entries". In: *Journal of the ACM* 51.4 (2004), pp. 671–697 (cit. on p. 4).

[JZ23] S. Jeffery and S. Zur. "Multidimensional Quantum Walks". In: *Proceedings of the 55th Symposium on Theory of Computing (STOC)*. 2023, pp. 1125–1130 (cit. on p. 3).

[KSV02] A. Y. Kitaev, A. H. Shen, and M. N. Vyalyi. *Classical and Quantum Computation*. Vol. 47. American Mathematical Society, 2002 (cit. on pp. 1, 2).

[Liu21] Y. Liu. "StoqMA Meets Distribution Testing". In: *Proceedings of the 16th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC)*. 2021, 4:1–4:22 (cit. on p. 2).

[LLL24] G. Li, L. Li, and J. Luo. "Recovering the Original Simplicity: Succinct and Exact Quantum Algorithm for the Welded Tree Problem". In: *Algorithmica* 86.12 (2024), pp. 3719–3758 (cit. on p. 3).

[LPS88] A. Lubotzky, R. Phillips, and P. Sarnak. "Ramanujan Graphs". In: *Combinatorica* 8.3 (1988), pp. 261–277 (cit. on p. 7).

[LWWZ25] J. Leng, K. Wu, X. Wu, and Y. Zheng. *(Sub)Exponential Quantum Speedup for Optimization*. arXiv:2504.14841 [quant-ph]. 2025 (cit. on p. 3).

[LZ25] J. Li and S. Zur. "Multidimensional Electrical Networks and their Application to Exponential Speedups for Graph Problems". In: *Quantum* 9 (2025), p. 1733 (cit. on p. 3).

[Mor94] M. Morgenstern. "Existence and Explicit Constructions of $q+1$ Regular Ramanujan Graphs for Every Prime Power q". In: *Journal of Combinatorial Theory, Series B* 62.1 (1994), pp. 44–62 (cit. on p. 7).

[MWW04] B. D. McKay, N. C. Wormald, and B. Wysocka. "Short Cycles in Random Regular Graphs". In: *The Electronic Journal of Combinatorics* 11.1 (2004) (cit. on p. 7).

[Sar23] A. Sarid. "The Spectral Gap of Random Regular Graphs". In: *Random Structures & Algorithms* 63.2 (2023), pp. 557–587 (cit. on p. 7).

[SNK12] R. D. Somma, D. Nagaj, and M. Kieferová. "Quantum Speedup by Quantum Annealing". In: *Physical Review Letters* 109 (2012), p. 050501 (cit. on p. 3).

[ŠVV09] D. Štefankovič, S. S. Vempala, and E. Vigoda. "Adaptive Simulated Annealing: A Near-Optimal Connection between Sampling and Counting". In: *Journal of the ACM* 56.3 (2009), pp. 1–36 (cit. on p. 4).

[TCC+22] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, and J. Tennyson. "The Variational Quantum Eigensolver: A Review of Methods and Best Practices". In: *Physics Reports* 986 (2022), pp. 1–128 (cit. on p. 1).

[Wai25] G. Waite. *The Guided Local Hamiltonian Problem for Stoquastic Hamiltonians*. arXiv:2509.25829 [quant-ph]. 2025 (cit. on p. 2).

[WFC24] J. Weggemans, M. Folkertsma, and C. Cade. "Guidable Local Hamiltonian Problems with Implications to Heuristic Ansatz State Preparation and the Quantum PCP Conjecture". In: *Proceedings of the 19th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC)*. 2024, 10:1–10:24 (cit. on p. 2).

A Spectral properties of the main graph

In this section, we analyze the spectral properties of the main graph. The study of the top eigenvector of its adjacency matrix is closely related to the analysis in Appendix B of [GHV21]. However, their argument proceeds via an iterative decoration process in which identical complete trees are attached to all vertices at each stage. Our construction differs in that we do not decorate the leaves created at intermediate levels of the iteration as in Definitions 2.4 and 2.7. Consequently, the spectral analysis in [GHV21] does not apply directly to our setting. We therefore present a self-contained proof that analyzes the top-eigenvector of the main graph directly, rather than via an iterative argument.

Below is a lemma proved in [Has21] about the spectral gap of the sum of two Hermitian matrices of same dimension in terms of the spectral gap of one and the spectral norm of another.

Lemma A.1 (Spectral gap of sum of matrices, (Lemma 4 in [Has21]; Footnote 15 in [GHV21])). *Let A_1 and A_2 be any two Hermitian matrices of the same dimension. If A_1 has a spectral gap of δ and A_2 has largest eigenvalue at most γ, then $A_1 + A_2$ has a spectral gap of at least $\delta - 2\gamma$.*

Applying the above result to the main graph, we obtain the following, which proves the first part of Lemma 2.9:

Lemma A.2 (Spectral gap of G_n)**.** *The difference between the two largest eigenvalues of A_n is at least $n/2 - 4\sqrt{2n}$.*

Proof. The adjacency matrix A_n of the main graph G_n can be written as $A_n = A_E + A_T$, with A_E denoting the expander part and A_T denoting the tree edges added to it. By the assumptions of Proposition 2.3, the spectral gap of A_E is at least $n/2$. The graph formed by the added tree edges is a forest of maximum degree $2n$, and thus the corresponding adjacency matrix A_T has spectral norm bounded by $2\sqrt{2n}$.

Applying Lemma A.1 with $A_1 = A_E$, $A_2 = A_T$, $\delta = n/2$ and $\gamma = 2\sqrt{2n}$ proves the lemma. □

From now on, we will study the top-eigenvector of the main graph G_n in order to prove the second part of Lemma 2.9. Since the graph is connected, the top-eigenvector is also the Perron–Frobenius eigenvector of G_n. We start by defining trees with self-loops attached to the root.

Definition A.3 (Self-loop tree eigenpair)**.** Let T be a finite tree with distinguished root vertex t, and let $\alpha > 0$. Let $T(\alpha)$ denote the graph obtained from T by adding a self-loop of weight α at t. Let $A_{T(\alpha)}$ denote its adjacency matrix. We write $\lambda_{T(\alpha)}$ for the largest eigenvalue of $A_{T(\alpha)}$, and $\psi_{T(\alpha)}$ for the corresponding Perron–Frobenius eigenvector, normalized so that $\psi_{T(\alpha)}(t) = 1$.

We use the notation ψ (rather than $|\psi\rangle$) when the vector is not necessarily of unit norm. We give a convenient characterization of the Perron–Frobenius eigenvector $\psi_{T(\alpha)}$ of a tree with a self-loop at the root, parametrized by the resulting top eigenvalue.

Lemma A.4 (Top eigenvector of self-looped tree)**.** *Let T be a finite tree with root t and largest eigenvalue λ_T. For any $\lambda > \lambda_T$, define $\alpha > 0$ by $\alpha^{-1} := \left[(\lambda I - A_T)^{-1}\right]_{t,t}$. Then the top eigenvector of the tree with a self-loop of weight α at t, normalized so that the entry at t equals 1, is*

$$\psi_{T(\alpha)} = \alpha(\lambda I - A_T)^{-1} e_t,$$

where e_t denotes the standard basis vector at t. Moreover $\lambda_{T(\alpha)} = \lambda$.

Proof. Write the adjacency matrix as $A_{T(\alpha)} = A_T + \alpha\, e_t e_t^T$, where e_t is the standard basis vector at the root t. We check that $(\lambda I - A_T)^{-1} e_t$ is an eigenvector of $A_{T(\alpha)}$ with eigenvalue λ:

$$\begin{aligned}
A_{T(\alpha)}(\lambda I - A_T)^{-1} e_t &= A_T(\lambda I - A_T)^{-1} e_t + \alpha\, e_t e_t^T (\lambda I - A_T)^{-1} e_t \\
&= A_T(\lambda I - A_T)^{-1} e_t + \alpha\, e_t \cdot \frac{1}{\alpha} && \text{(by definition of } \alpha\text{)} \\
&= (A_T - \lambda I + \lambda I)(\lambda I - A_T)^{-1} e_t + e_t \\
&= \lambda \cdot (\lambda I - A_T)^{-1} e_t.
\end{aligned}$$

Since A_T is non-negative and $\lambda > \lambda_T$, the matrix $\lambda I - A_T$ is a non-singular M-matrix (see [BP94], Chapter 6), whose inverse is entry-wise positive. Hence $(\lambda I - A_T)^{-1} e_t$ is a positive vector. By the Perron–Frobenius theorem, it is proportional to the top eigenvector of $A_{T(\alpha)}$, so $\lambda = \lambda_{T(\alpha)}$. Normalizing the t-entry to 1 gives $\psi_{T(\alpha)} = \alpha \cdot (\lambda I - A_T)^{-1} e_t$, as claimed. □

Let G be obtained by attaching multiple copies of various trees to each vertex of a base graph E via edges to a distinguished vertex in each tree; we describe the Perron–Frobenius eigenvector of G in terms of that of E and the trees, which will be useful for analyzing G_n in Definition 2.7 with G corresponding to G_n and E to E_n.

Lemma A.5 (Top-eigenvector under multiple decorations)**.** *Let E be a connected graph with adjacency matrix A_E. Fix a Perron–Frobenius vector ψ_E with arbitrary normalization, and let λ_E denote the corresponding eigenvalue. For each $k \in [K-1]$, let T_k be a finite tree with distinguished root t_k, and let $\psi_{T_k(\cdot)}, \lambda_{T_k(\cdot)}$ be defined as in Lemma A.4. Let G be obtained by attaching β copies of each T_k to every vertex $v \in V_E$ via edges connecting v to the root t_k of each copy. Denote the Perron–Frobenius eigenvalue of A_G by λ_G, and for every $k \in [K-1]$, define $\alpha_k^{-1} := \left[(\lambda_G I - A_{T_k})^{-1}\right]_{t_k,t_k}$. Then we have the following:*

1. $\lambda_G = \lambda_E + \beta \sum_{k=1}^{K-1} \alpha_k^{-1}$.
2. $\lambda_G = \lambda_{T_k(\alpha_k)}$ *for all* $k \in [K-1]$.
3. *The restriction of any Perron–Frobenius vector of G to E is proportional to ψ_E. Let ψ_G be the Perron–Frobenius vector of G whose restriction to E exactly equals ψ_E. Then, ψ_G decomposes as*

$$\psi_G = \psi_E \;\oplus\; \bigoplus_{v \in V_E} \psi_E(v) \left(\bigoplus_{k=1}^{K-1} \bigoplus_{j=1}^{\beta} \alpha_k^{-1} \cdot \psi_{T_k(\alpha_k)} \right).$$

Proof. For each $k \in [K-1]$ and each vertex $v \in V_E$, we index the β copies of T_k attached to v by $j \in [\beta]$ and denote the corresponding copy by $T_{k,j}$. All copies are isomorphic to T_k; we therefore denote their adjacency matrices uniformly by A_{T_k}.

The adjacency matrix of G admits the block decomposition

$$A_G = \begin{pmatrix} A_E & B \\ B^\top & \bigoplus_{v \in V_E} \bigoplus_{k=1}^{K-1} \bigoplus_{j=1}^{\beta} A_{T_k} \end{pmatrix},$$

where B encodes the edges between each vertex $v \in V_E$ and the root t_k of each copy $T_{k,j}$ attached to v. More explicitly, B has entries

$$B_{v,(v,k,j,t_k)} = 1 \quad \text{for each } v \in V_E,\ k \in [K-1],\ j \in [\beta], \tag{6}$$

and all other entries are zero.

Let $\psi_G = (x, y)$ be the Perron–Frobenius vector of A_G, with $x \in \mathbb{R}^{|V_E|}$ and $y \in \mathbb{R}^{|V_E| \cdot K-1 \cdot \beta}$. By symmetry of the construction, the vector y decomposes as $y = \bigoplus_{v \in V_E} \bigoplus_{k=1}^{K-1} \bigoplus_{j=1}^{\beta} y_{v,k,j}$.

The eigenvalue equations are $A_E x + By = \lambda_G x$ and $B^\top x + \bigoplus_{v,k,j} A_{T_k} y = \lambda_G y$. Solving the second one block-wise over each copy $T_{k,j}$ yields, for every (v,k,j), $y_{v,k,j} = (\lambda_G I - A_{T_k})^{-1} e_{t_k} x_v$. Indeed,

$$\begin{aligned}
& B^\top x + \bigoplus_{v,k,j} A_{T_k} y = \lambda_G y \\
\implies & B^\top x + \left(\bigoplus_{v,k,j} A_{T_k} \right) \cdot \left(\bigoplus_{v,k,j} y_{v,k,j} \right) = \lambda_G \left(\bigoplus_{v,k,j} y_{v,k,j} \right) \\
\implies & \bigoplus_{v,k,j} e_{t_k} x_v + \left(\bigoplus_{v,k,j} A_{T_k} \right) \cdot \left(\bigoplus_{v,k,j} y_{v,k,j} \right) = \lambda_G \left(\bigoplus_{v,k,j} y_{v,k,j} \right) \\
& \qquad\qquad \text{(from Equation (6), } (B^T x)_{(v,k,j)} = e_{t_k} x_v) \\
\implies & \bigoplus_{v,k,j} (\lambda_G I - A_{T_k})^{-1} e_{t_k} x_v = \bigoplus_{v,k,j} y_{v,k,j}.
\end{aligned}$$

In the last step, the existence of $(\lambda_G I - A_{T_k})^{-1}$ is due to the fact that $\lambda_G > \lambda_{T_k} = \|A_{T_k}\|$ as T_k is a subgraph of G. Solving the final expression blockwise gives the desired expression for $y_{v,k,j}$.

Substituting this into the first equation, $A_E x + By = \lambda_G x$ entrywise gives, for each v,

$$(A_E x)_v + \sum_{k=1}^{K-1} \sum_{j=1}^{\beta} \left[(\lambda_G I - A_{T_k})^{-1}\right]_{t_k,t_k} x_v = \lambda_G x_v, \qquad (\text{since } (By)_v = \textstyle\sum_{k=1}^{K-1} \sum_{j=1}^{\beta} y_{v,k,j}(t_k))$$

or equivalently,

$$\Big(A_E + \beta \sum_{k=1}^{K-1} \alpha_k^{-1} I\Big) x = \lambda_G x.$$

Since x is positive and is an eigenvector of A_E, the Perron–Frobenius theorem implies that it is proportional to ψ_E, the top eigenvector of E. Thus we can conclude that

$$\lambda_G = \lambda_E + \beta \sum_{k=1}^{K-1} \alpha_k^{-1}.$$

Finally, since T_k is a subgraph of G, the eigenvalue interlacing theorem says that $\lambda_G > \lambda_{T_k}$, Lemma A.4 applied to each T_k implies that $\lambda_{T_k(\alpha_k)} = \lambda_G$ and $\psi_{T_k(\alpha_k)} = \alpha_k \cdot (\lambda_G I - A_{T_k})^{-1} e_{T_k}$. Hence $y_{v,k,j} = {\alpha_k}^{-1}\, \psi_E(v)\, \psi_{T_k(\alpha_k)}$, and combining the expressions for x and y yields the stated decomposition of ψ_G. □

Lemma A.6 (α_k BOUNDS)**.** *Let α_k be defined as per the assumptions of Lemma A.5. If for all $k \in [K-1]$ the degree of T_k is bounded by Δ, then we have for all $k \in [K-1]$,*

$$\lambda_E - 2\sqrt{\Delta} \le \alpha_k \le \lambda_E + \beta K - 1 \cdot (\lambda_E - 2\sqrt{\Delta})^{-1}.$$

Proof. To establish the lower bound, we use the property $\lambda_{T_k} + \alpha_k \ge \lambda_G$. Indeed from Lemma A.5 we have that $\lambda_G = \lambda_{T_k(\alpha_k)}$ and

$$\lambda_{T_k(\alpha_k)} = \left\|A_{T_k(\alpha_k)}\right\| = \left\|A_{T_k} + \alpha_k \cdot e_{t_k} e_{t_k}^T\right\| \le \|A_{T_k}\| + \left\|\alpha_k \cdot e_{t_k} e_{t_k}^T\right\| = \lambda_k + \alpha_k.$$

The spectral norm of a finite tree of degree at most Δ is bounded by $\lambda_{T_k} \le 2\sqrt{\Delta}$. Thus, $\alpha_k \ge \lambda_E - 2\sqrt{\Delta}$.

For the upper bound, we use the fact that the top eigenvalue of a graph with a self-loop is at least the weight of that loop: $\alpha_k \le \lambda_{T_k(\alpha_k)}$. By Lemma A.5, the eigenvalue of G is given by:

$$\alpha_k \le \lambda_{T_k(\alpha_k)} = \lambda_G = \lambda_E + \beta \cdot \sum_{k=1}^{K-1} \alpha_k^{-1} \le \lambda_E + \beta K - 1 \cdot (\lambda_E - 2\sqrt{\Delta})^{-1}.$$

□

Lemma A.7 (L_2-NORM CONCENTRATION ON G_n)**.** *Let G_n be the graph from Definition 2.7, obtained by attaching $\sqrt{n}$ copies of $\widehat{T}_{n,k}$ (degree at most $2n$, see Definition 2.4) to each vertex of the n-regular expander E_n, for $k \in [K-1]$ with $K = \sqrt{n}$.*

Let ψ_n be a Perron–Frobenius vector of A_n (unnormalized). By Lemma A.5, the restriction of ψ_n to E_n is proportional to a Perron–Frobenius vector of A_E; we fix the normalization of ψ_n so that this restriction equals a chosen representative ψ_E of the Perron vectors of E_n. For each $v \in V_E$, let $\psi_{v,T}$ denote the restriction of ψ_n to the tree vertices attached to v. Then we have that,

$$\psi_n = \psi_E + \sum_{v \in V_E} \psi_{v,T},$$

and for sufficiently large n,

$$\|\psi_E\|^2 \ge (1 - O(1/n)) \|\psi_n\|^2.$$

Proof. Without loss of generality, we assume that all isolated vertices of G_n have been removed. We slightly abuse notation by continuing to write ψ_n for the (unnormalized) Perron–Frobenius vector of the resulting graph. This causes no ambiguity, since the Perron–Frobenius vector of the original graph is identically zero on isolated vertices.

Applying Lemma A.5 with $E = E_n$, $G = G_n$, $\beta = \sqrt{n}$, $K = \sqrt{n}$, and $T_k = \widehat{T}_{n,k}$, we get that,

$$\psi_n = \psi_E \oplus \bigoplus_{v \in V_E} \psi_E(v) \left(\bigoplus_{k=1}^{K-1} \bigoplus_{j=1}^{\sqrt{n}} {\alpha_k}^{-1} \cdot \psi_{\widehat{T}_{n,k}(\alpha_k)} \right)$$

with corresponding terms as defined in Definition A.3 and assumptions of Lemma A.5. Combining the components of trees attached to a common vertex v together, we get that,

$$\psi_n = \psi_E + \sum_{v \in V_E} \psi_{v,T}, \text{ where } \psi_{v,T} = \psi_E(v) \cdot \left(\bigoplus_{k=1}^{K-1} \bigoplus_{j=1}^{\sqrt{n}} {\alpha_k}^{-1} \cdot \psi_{\widehat{T}_{n,k}(\alpha_k)} \right)$$

Using all the above, we can write the L_2-norm of ψ_n as

$$\|\psi_n\|_2^2 = \left(1 + \sum_{k=1}^{K-1} \alpha_k^{-2} \sqrt{n} \cdot \left\| \psi_{\widehat{T}_{n,k}(\alpha_k)} \right\|_2^2 \right) \cdot \|\psi_E\|_2^2.$$

Since E is an n-regular expander graph, the Perron-eigenvalue of A_E is $\lambda_E = n$. Also, by Fact 2.8, $\widehat{T}_{n,k}$ is a tree graph with max-degree bounded by $\Delta = 2n$ and hence $\lambda_{\widehat{T}_{n,k}} \leq 2\sqrt{2n}$. By applying Lemma A.6, we have the following,

$$n - 2\sqrt{2n} \leq \alpha_k \leq n + \frac{\sqrt{n} \cdot (\sqrt{n} - 1)}{n - 2\sqrt{2n}} \leq n + 4. \tag{7}$$

Let λ_n denote the Perron-eigenvalue of A_n. From Lemma A.4, we have that $\psi_{\widehat{T}_{n,k}(\alpha_k)} = \alpha_k (\lambda_n \cdot I - A_{\widehat{T}_{n,k}})^{-1} \cdot e_{t_k}$. Multiplying by $(\lambda_n \cdot I - A_{\widehat{T}_{n,k}})$ on both sides and taking inner product with $\psi_{\widehat{T}_{n,k}(\alpha_k)}$:

$$\lambda_n \cdot \left\| \psi_{\widehat{T}_{n,k}(\alpha_k)} \right\|_2^2 - \langle \psi_{\widehat{T}_{n,k}(\alpha_k)}, A_{\widehat{T}_{n,k}} \psi_{\widehat{T}_{n,k}(\alpha_k)} \rangle = \langle \psi_{\widehat{T}_{n,k}(\alpha_k)}, \alpha_k \cdot e_{t_k} \rangle = \alpha_k. \tag{8}$$

Since $\lambda_{\widehat{T}_{n,k}}$ is the largest eigenvalue of $\widehat{T}_{n,k}$, by Courant-Fischer (min-max) theorem for symmetric matrices (see [HJ12], Chapter 4), we obtain for all ψ, $\langle \psi, A_{\widehat{T}_{n,k}} \psi \rangle \leq \lambda_{\widehat{T}_{n,k}} \|\psi\|_2^2$. By plugging this in Equation (8), we get the inequality $(\lambda_n - \lambda_{\widehat{T}_{n,k}}) \left\| \psi_{\widehat{T}_{n,k}(\alpha_k)} \right\|_2^2 \leq \alpha_k$. By using Equation (7) and the fact that $\lambda_n \leq \lambda_E = n$ (since E is a subgraph of G_n), we finally obtain,

$$\left\| \psi_{\widehat{T}_{n,k}(\alpha_k)} \right\|_2^2 \leq \frac{\alpha_k}{(\lambda_n - \lambda_{\widehat{T}_{n,k}})} \leq \frac{n+4}{(n - 2\sqrt{2n})} \leq 1 + O\left(\frac{1}{\sqrt{n}} \right) = O(1).$$

Putting it all together,

$$\begin{aligned} \|\psi_n\|_2^2 &= \left(1 + \sum_{k=1}^{K-1} \alpha_k^{-2} \sqrt{n} \cdot \left\| \psi_{\widehat{T}_{n,k}(\alpha_k)} \right\|_2^2 \right) \cdot \|\psi_E\|_2^2 \\ &\leq \left(1 + \frac{(\sqrt{n} - 1)\sqrt{n}}{(n - 2\sqrt{2n})^2} \cdot O(1) \right) \cdot \|\psi_E\|_2^2 \\ &\leq (1 + O(1/n)) \cdot \|\psi_E\|_2^2. \end{aligned}$$

□

B Hardness of exploring self-similar trees

This section is devoted to proving that any classical algorithm requires a sub-exponential number of queries to explore a 0-level leaf in the full-level decorated tree $\widehat{T}_{n,k}$ starting from its root (see Definition 2.4 for description of these objects). The main result is Lemma 3.8 (restated below as Proposition B.6). The proof proceeds by induction on the index k.

We begin by introducing several probabilistic events and random variables that play a central role in the argument. Observe that $\widehat{\textsc{Exit}}_{n,K}$ is exactly the event whose probability we aim to bound in order to prove Lemma 3.8.

Definition B.1 (Events and random variables)**.** Given a randomized algorithm that explores the 1-level decorated tree $T_{n,k,1}$ starting at its root, we define:

- $\textsc{Exit}_{n,k}$: the event that the algorithm queries a 0-level leaf,
- $W_{n,k}$: the random variable denoting the number of queried 1-level leaves that belong to *distinct* decorations.

The same objects are denoted $\widehat{\textsc{Exit}}_{n,k}$ and $\widehat{W}_{n,k}$ when the algorithm is instead exploring the full-level decorated tree $\widehat{T}_{n,k}$.

For a randomized algorithm $\mathcal{A}$, let $q_{\mathcal{A}}$ denote the maximum number of queries made by that algorithm (over all possible random executions). Our goal is to identify the largest value of $q_{n,k}$ for which the event $\widehat{\textsc{Exit}}_{n,k}$ is unlikely to occur when the algorithm makes at most $q_{n,k}$ queries. This amounts to upper-bounding the quantity $\max_{\mathcal{A}:q_{\mathcal{A}}<q_{n,k}} \Pr[\widehat{\textsc{Exit}}_{n,k}]$. The inductive proof will relate this to $\max_{\mathcal{A}:q_{\mathcal{A}}<q_{n,k-1}} \Pr[\widehat{\textsc{Exit}}_{n,k-1}]$, where the number of queries is decreased to $q_{n,k-1} < q_{n,k}$ and the input tree is $\widehat{T}_{n,k-1}$.

First, it is easy to see that full-level decorated trees are harder to explore than 1-level decorated trees. In particular, we will make use of the following lemma.

Lemma B.2 (Monotonicity of hardness under decorations)**.** *For any number q of queries and integer w, we have*

$$\max_{\mathcal{A}:q_{\mathcal{A}}<q} \Pr[\widehat{\textit{Exit}}_{n,k} \text{ and } \widehat{W}_{n,k} < w] \leq \max_{\mathcal{A}:q_{\mathcal{A}}<q} \Pr[\textit{Exit}_{n,k} \text{ and } W_{n,k} < w].$$

Proof. Any algorithm with query access to $T_{n,k}$ can simulate query access to the full-level decorated graph $\widehat{T}_{n,k}$ by simply simulating the subsequent levels of decoration itself. Hence, the success probability cannot be larger on $\widehat{T}_{n,k}$ than on $T_{n,k}$. □

The next central lemma upper bounds the probability of the event $\textsc{Exit}_{n,k}$ occurring when only a few 1-level leaves of the 1-level decorated tree $T_{n,k}$ are explored. Note that the input graph need not be fully-decorated, nor does the number $q_{\mathcal{A}}$ of queries need to be bounded to apply this lemma.

Lemma B.3 (Hardness of avoiding the decorations)**.** *For any integer w,*

$$\max_{\mathcal{A}:q_{\mathcal{A}}<\infty} \Pr[\textit{Exit}_{n,k} \text{ and } W_{n,k} < w] \leq \left(\frac{d_{n,k}}{d_{n,k-1}}\right)^{(\ell_{n,k}-\ell_{n,k-1})/w}.$$

Proof. The cases $w = 1$ and $w = 2$ correspond essentially to [GHV21, Lemma 3] and [GHV21, Lemma 4, Case 1]. We do not prove the case $w > 2$ here since it is not needed for our application.

We sketch the proof for $w = 1$. We view an algorithm $\mathcal{A}$ as a local exploration that starts at the root of $T_{n,k,1}$ and may only query edges adjacent to previously queried vertices. The events $\textsc{Exit}_{n,k}$ and $W_{n,k} < 1$ require the first queried leaf in $T_{n,k,1}$ to be a 0-level leaf. Thus,

the algorithm must grow an exploration path of length $\ell_{n,k}$ that reaches a 0-level leaf before any 1-level leaf is queried. A path that grows to length $\ell_{n,k}$ avoids reaching a 1-level leaf if and only if it does not enter any decoration before depth $\ell_{n,k} - \ell_{n,k-1}$, since the decorations have depth $\ell_{n,k-1}$. For each node in the base graph, the proportion of its children that do not belong to a decoration is $d_{n,k}/d_{n,k-1}$. Hence, the probability of avoiding any decoration for $\ell_{n,k} - \ell_{n,k-1}$ consecutive steps is $(d_{n,k}/d_{n,k-1})^{\ell_{n,k}-\ell_{n,k-1}}$. □

The next lemma exploits the self-similar structure of $\widehat{T}_{n,k}$ to relate the probability of finding multiple 1-level leaves in $\widehat{T}_{n,k}$ to the probability of the event $\widehat{\textsc{Exit}}_{n,k-1}$. Most importantly, the number of queries allowed to explore $\widehat{T}_{n,k-1}$ is reduced by a factor of $1/w$, which will make the recursion effective later on.

Lemma B.4 (Self similarity)**.** *For any integer w,*

$$\max_{\mathcal{A}:q_{\mathcal{A}}<q} \Pr[\widehat{W}_{n,k} \geq w] \leq q \cdot \max_{\mathcal{A}:q_{\mathcal{A}}<q/w} \Pr[\widehat{EXIT}_{n,k-1}].$$

Proof. This generalizes [GHV21, Lemma 4, Case 2]. Essentially, finding leaves of w distinct outer trees in $\widehat{T}_{n,k}$ amounts to the event $\widehat{\textsc{Exit}}_{n,k-1}$ occurring w times in w distinct subtrees $\widehat{T}_{n,k-1}$ of $\widehat{T}_{n,k}$. By a pigeonhole argument, at least one of these events must happen after making only q/w queries in the corresponding subtree $\widehat{T}_{n,k-1}$. Since at most q subtrees are explored in total (this is a very crude bound), by a union bound argument, the overall probability must be at most $q \cdot \max_{\mathcal{A}:q_{\mathcal{A}}<q/w} \Pr[\widehat{\textsc{Exit}}_{n,k-1}]$. □

Combining the above three results, we obtain the inductive step of the proof:

Corollary B.5 (Inductive argument)**.** *For any integer w,*

$$\max_{\mathcal{A}:q_{\mathcal{A}}<q_{n,k}} \Pr[\widehat{EXIT}_{n,k}] \leq \left(\frac{d_{n,k}}{d_{n,k-1}}\right)^{(\ell_{n,k}-\ell_{n,k-1})/w} + q_{n,k} \cdot \max_{\mathcal{A}:q_{\mathcal{A}}<q_{n,k}/w} \Pr[\widehat{EXIT}_{n,k-1}]. \tag{9}$$

Proof. It is simply a matter of applying Lemmas B.2 to B.4, and basic laws of probabilities,

$$\begin{aligned}
\max_{\mathcal{A}:q_{\mathcal{A}}<q_{n,k}} \Pr[\widehat{\textsc{Exit}}_{n,k}] &\leq \max_{\mathcal{A}:q_{\mathcal{A}}<q_{n,k}} \Pr[\widehat{\textsc{Exit}}_{n,k} \text{ and } \widehat{W}_{n,k} < w] + \max_{\mathcal{A}:q_{\mathcal{A}}<q_{n,k}} \Pr[\widehat{\textsc{Exit}}_{n,k} \text{ and } \widehat{W}_{n,k} \geq w] \\
&\leq \max_{\mathcal{A}:q_{\mathcal{A}}<\infty} \Pr[\textsc{Exit}_{n,k} \text{ and } W_{n,k} < w] + \max_{\mathcal{A}:q_{\mathcal{A}}<q_{n,k}} \Pr[\widehat{W}_{n,k} \geq w] \\
&\leq \left(\frac{d_{n,k}}{d_{n,k-1}}\right)^{(\ell_{n,k}-\ell_{n,k-1})/w} + q_{n,k} \cdot \max_{\mathcal{A}:q_{\mathcal{A}}<q_{n,k}/w} \Pr[\widehat{\textsc{Exit}}_{n,k-1}].
\end{aligned}$$

□

Finally, we solve the above recurrence relation using $w = 2$ and the set of parameters given in Lemma 3.8.

Proposition B.6 (Finding exit is hard - Lemma 3.8 restated)**.** *For $k \in \{1, \ldots, \sqrt{n}\}$, let $d_{n,k} = 2n - k\sqrt{n}$, $\ell_{n,k} = k \cdot 10n^{\frac{3}{2}} \log n$, $q_{n,k} = 2^k$ and $w = 2$. Then,*

$$\max_{\mathcal{A}:q_{\mathcal{A}}<q_{n,k}} \Pr[\widehat{EXIT}_{n,k}] \leq 2^{-2n\log(n)+(\sqrt{n}+1)(k-\sqrt{n})}.$$

In particular, for $K = \sqrt{n}$, the probability of $\widehat{EXIT}_{n,K}$ is at most $2^{-2n\log n}$ for any classical algorithm making at most $2^{\sqrt{n}}$ queries. Moreover, the tree $\widehat{T}_{n,K}$ contains less than $2^{12n^3\log^2 n}$ vertices.

Proof. The base case $k = 1$ is trivial. For the induction, we have,

$$\begin{aligned}\left(\frac{d_{n,k}}{d_{n,k-1}}\right)^{(\ell_{n,k}-\ell_{n,k-1})/2} &= \left(1-\frac{1}{2\sqrt{n}-k}\right)^{(10n^{\frac{3}{2}}\log n)/2}\\ &\le \left(1-\frac{1}{2\sqrt{n}}\right)^{5n^{\frac{3}{2}}\log n}\\ &\le 2^{-\frac{5}{2}n\log n}.\end{aligned}$$

Hence, by Corollary B.5,

$$\begin{aligned}\max_{\mathcal{A}:q_{\mathcal{A}}<q_{n,k}} \Pr[\widehat{\textsc{Exit}}_{n,k}] &\le 2^{-\frac{5}{2}n\log n} + 2^k \cdot 2^{-2n\log n+(\sqrt{n}+1)(k-1-\sqrt{n})}\\ &\le 2^{-\frac{5}{2}n\log n} + 2^{\sqrt{n}} \cdot 2^{-2n\log n+(\sqrt{n}+1)(k-1-\sqrt{n})}\\ &\le 2^{-2n\log n+(\sqrt{n}+1)(k-\sqrt{n})}.\end{aligned}$$

The number of vertices in $\widehat{T}_{n,k}$ is $d_{n,k}^{\ell_{n,k}} \cdot \prod_{i=k-1}^{1}(d_{n,i}-d_{n,i+1})d_{n,i}^{\ell_{n,i}} \le 2^{12n^3\log^2 n}$. □