

Power of Adiabatic Quantum Computation with Stoquastic Hamiltonians

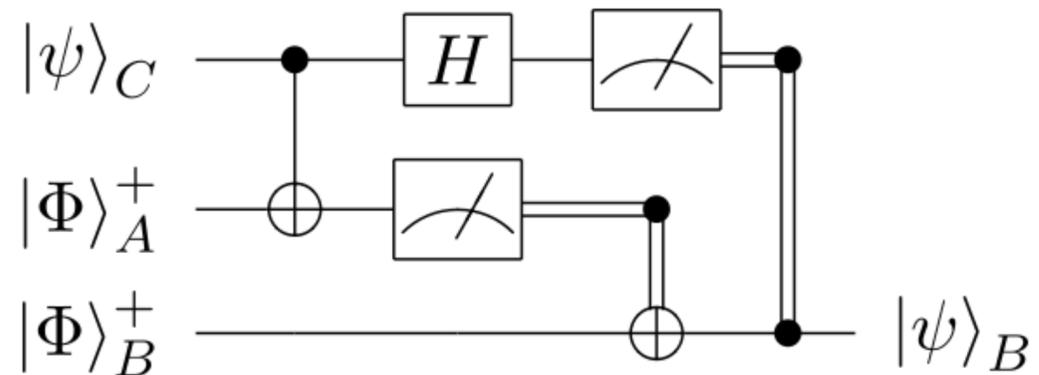
Based on a paper by Gilyén, Hastings and Vazirani, 2021*

Shrinidhi Teganahally Sridhara

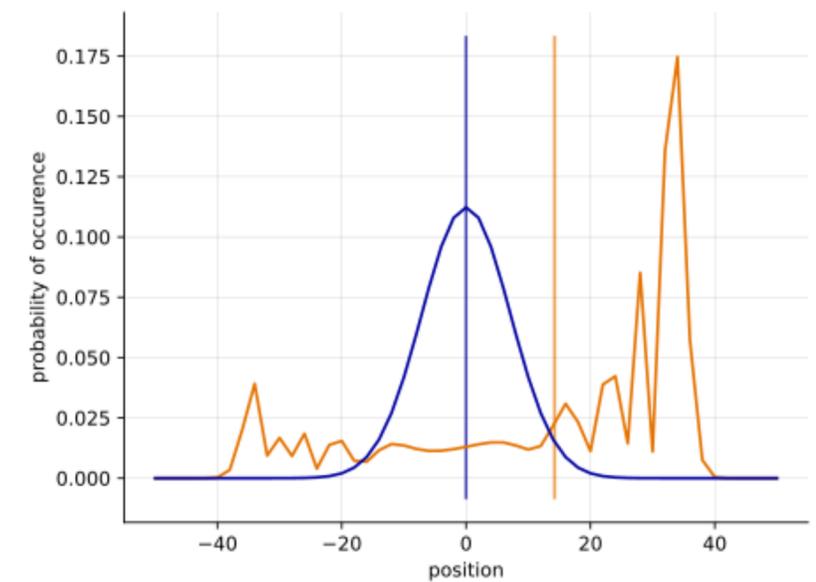
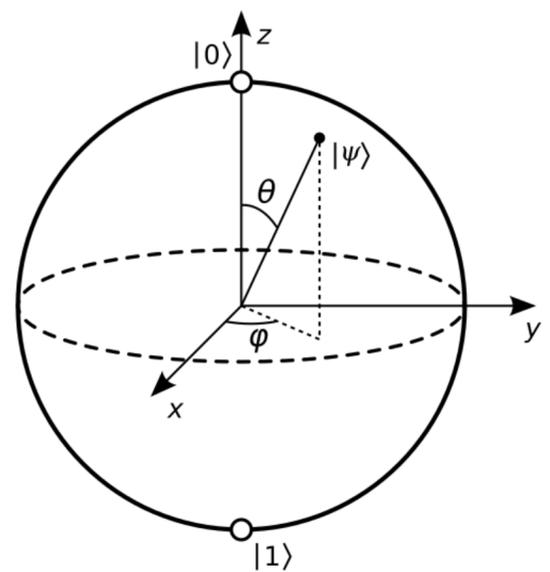
LaBRI

* <https://dl.acm.org/doi/10.1145/3406325.3451060>

Quantum Computation

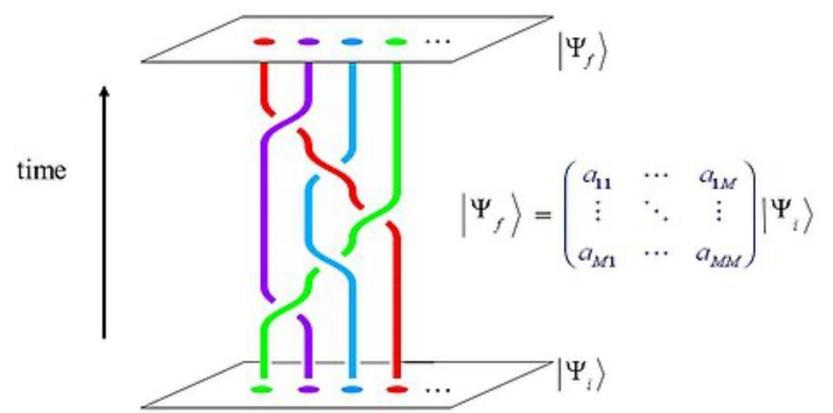


Circuit model

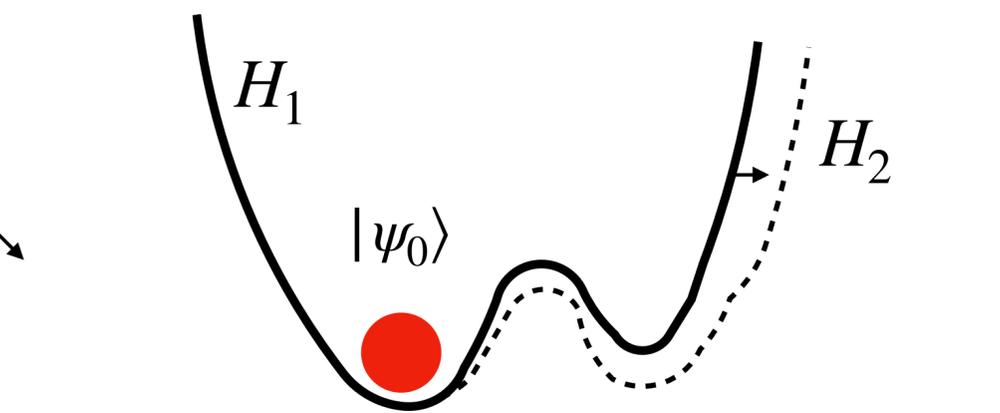


Quantum walks

Topological Quantum Computation



Topological QC



Adiabatic Quantum Computation

Quantum Adiabatic Computation

Start with ground state of H_i
Aim is to evolve to ground state of H_f

Use Hamiltonian Simulation

No guarantee of staying in ground state

Phase estimation needed at the end

Might take exponential time in total

Use Adiabatic Computation

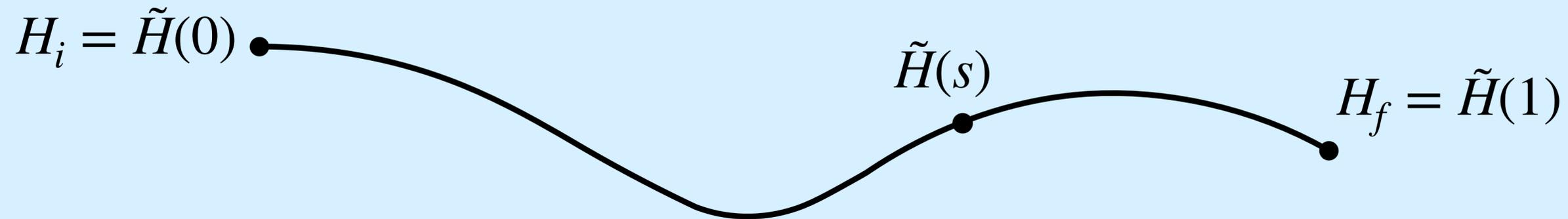
Guaranteed of staying in ground state

But needs to evolve slowly

\exists some **simple** H_f whose ground state can be prepared in *poly* time **adiabatically** **This talk**

Quantum Adiabatic Theorem

Let $\tilde{H}(s) \in \mathbb{C}^{2^n \times 2^n}$ be a smoothly varying hamiltonian for $s \in [0,1]$



$$\tilde{H}(s) = \sum_{j=0}^{N-1} \lambda_j(s) |\lambda_j(s)\rangle\langle\lambda_j(s)|, \text{ where } \lambda_0(s) < \lambda_1(s) \leq \lambda_2(s) \leq \dots \leq \lambda_{N-1}(s)$$

Suppose $|\psi(0)\rangle = |\lambda_0(0)\rangle$ and we evolve according to $\partial_s |\psi(s)\rangle = -i\tilde{H}(s) |\psi(s)\rangle$

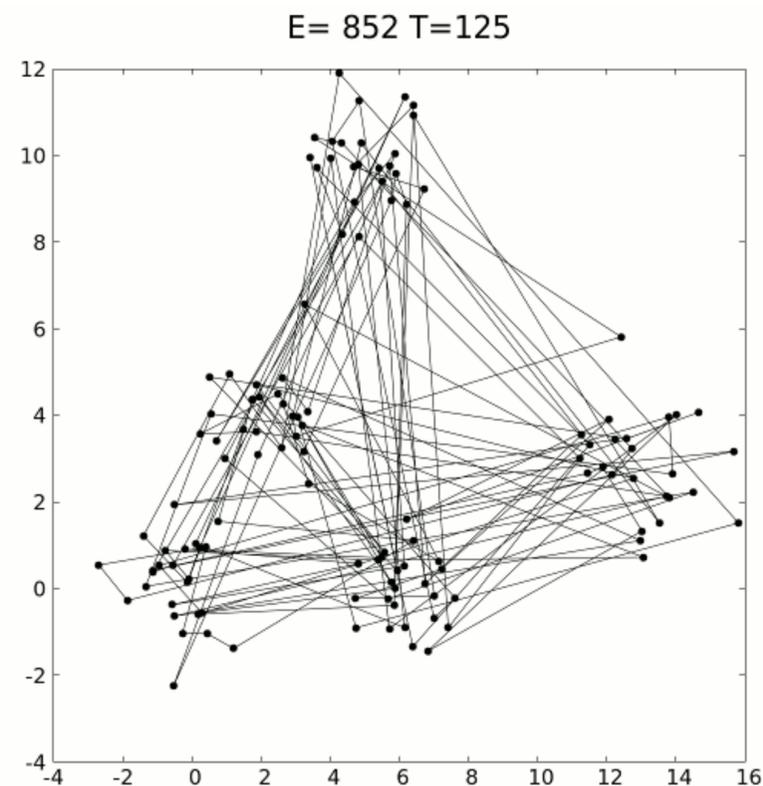
If evolved “slowly” along the path, then $|\langle\lambda_0(1) | \psi(1)\rangle|^2 \approx 1$

Example: Line interpolation
 $\tilde{H}(s) = (1-s)\tilde{H}(0) + s\tilde{H}(1)$

Quantum Adiabatic Theorem

A comparison

Simulated Annealing in Markov Chains



Simulated annealing of TSP solutions

Inverse temperature parameter β

$$\beta_1 > \beta_2 > \dots > \beta_i > \dots > \beta_{n-1} > \beta_n$$

$$\pi_{\beta_1} \rightarrow \pi_{\beta_2} \rightarrow \dots \rightarrow \pi_{\beta_i} \rightarrow \dots \rightarrow \pi_{\beta_{n-1}} \rightarrow \pi_{\beta_n}$$

Sample from π_{β_i} acts as a warm start for MC with stationary distribution $\pi_{\beta_{i+1}}$

Hence can prepare (sample from) the target π_{β_n} quickly

Quantum Adiabatic Theorem

A comparison and an example

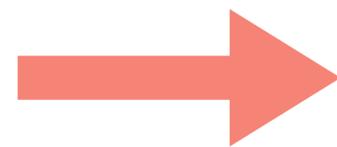
Classical Optimization problems

$$h : \{0,1\}^n \rightarrow \mathbb{R}$$

Ex: MAX-SAT

$$(x_0 \vee x_1) \wedge (x_0 \vee \neg x_1) \wedge (x_3 \vee x_2) \wedge (x_2 \vee \neg x_1)$$

$$h(x) = \# \text{ Unsatisfied clauses}$$



$$H_f = \begin{bmatrix} h(0) & & & & \\ & h(1) & & & \\ & & \ddots & & \\ & & & h(i) & \\ & & & & \ddots \\ & & & & & h(2^n) \end{bmatrix} \quad H_f = \sum_{z \in \{0,1\}^n} h(z) |z\rangle\langle z|$$

$$H_i = - \sum_i X_i$$

$$|\psi(0)\rangle = |+\rangle^{\otimes n} = \frac{1}{2^n} \sum_{z \in \{0,1\}^n} |z\rangle$$

Adiabatic evolution



$$H_f = \sum_{z \in \{0,1\}^n} h(z) |z\rangle\langle z|$$

$$|\psi(1)\rangle \rightarrow \text{MAX-SAT solution}$$

But doesn't mean all classical optimization problems can be solved very fast....

Quantum Adiabatic Theorem

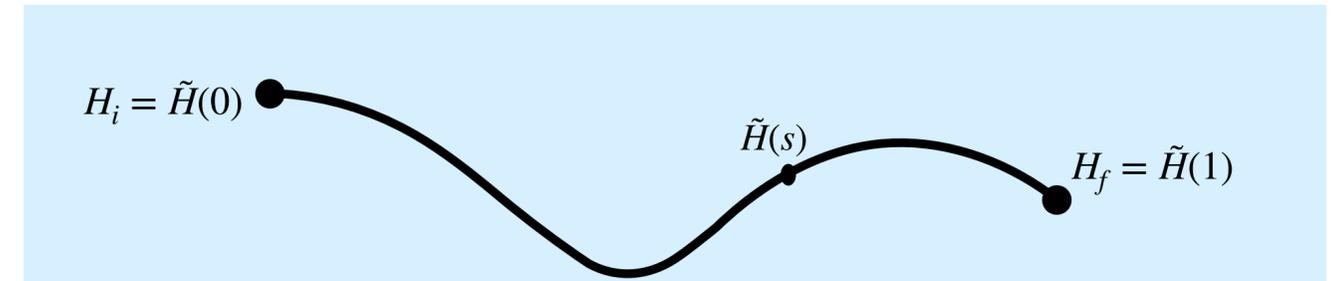
Let $s = t/T$

So, $\tilde{H}(s) = \tilde{H}(t/T)$ where T is the total simulation time

Evolve according to $\partial_t |\psi(t)\rangle = -i\tilde{H}(t/T) |\psi(t)\rangle$

Then as $T \rightarrow \infty$, $|\langle \lambda_0(1) | \psi(T) \rangle|^2 \rightarrow 1$ (Adiabatic theorem)

For large T , $|\psi(T)\rangle \approx |\lambda_0(1)\rangle$



$$\tilde{H}(s) = \sum_{j=0}^{N-1} \lambda_j(s) |\lambda_j(s)\rangle \langle \lambda_j(s)|$$

, where

$$\lambda_0(s) < \lambda_1(s) \leq \lambda_2(s) \leq \dots \leq \lambda_{N-1}(s)$$

But how large must it be?

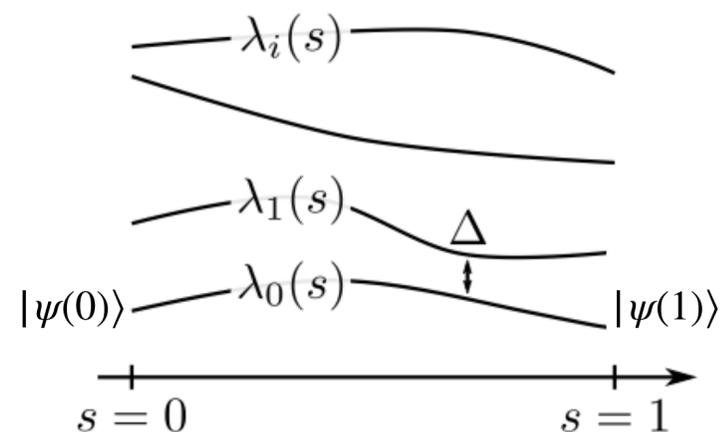
Quantum Adiabatic Theorem

Approximately Adiabatic Evolution

The total simulation time required for adiabaticity depends on the spectrum of the Hamiltonian

$$\text{Gap: } \Delta(s) = \lambda_1(s) - \lambda_0(s) \text{ and } \Delta = \min_{s \in [0,1]} \Delta(s)$$

Theorem [Aharonov et al]. Let the spectral gap of $\tilde{H}(s)$ be at least Δ for all $0 \leq s \leq 1$. For any $\epsilon > 0$, there exists $T = \text{poly} \left(\frac{1}{\Delta}, \frac{1}{\epsilon}, \|H_i\| + \|H_f\| \right)$ such that $|\psi(T)\rangle$ will be ϵ -close to $|\lambda_0(1)\rangle$.



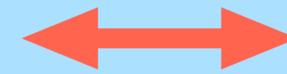
If ground state of H_0 can be **easily prepared** and $\Delta > 1/\text{poly}(n)$, then ground state of H_1 can be prepared in $\text{poly}(n)$ time!

Unfortunately for many NP-Hard problems, Δ is exponentially small, hence doesn't work in general for all optimization problems

Main ideas

What is the computational power of Quantum Adiabatic Computation?

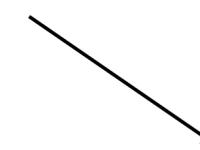
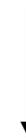
If $H(s) = (1 - s)H_0 + sH_1$ where H_0 and $H_1 \in \mathbb{C}^{2^n \times 2^n}$ are local hamiltonians with $\Delta(s) \geq 1/\text{poly}(n)$



BQP [Aharonov et al., 2004]

What if $H(s)$ is promised to be **stoquastic**?

Stoquastic - All non-diagonal elements of $H(s)$ are non-positive (in some basis)



Related work on stoquastic H

[Bravyi and Terhal, 2010] - for frustration-free stoquastic H , computing the ground state is classically tractable.

Frustration-free: $H = \sum_i H_i$ and ground state of H is ground state of each H_i as well.

Is it true for arbitrary stoquastic H as well?

[Hastings and Freedman, 2013] - There are topological obstructions for the **classical** “Quantum Monte Carlo” method when H is stoquastic.

Are there some other classical methods that could solve this?

Related work on stoquastic H

[Hastings, 2020] - There exists a stoquastic $H \in \mathbb{C}^{2^n \times 2^n}$ whose ground state can be computed efficiently i.e. in $poly(n)$ time by QAC but **any** classical algorithm requires $\Omega(n^{poly(\log(n))})$ time.

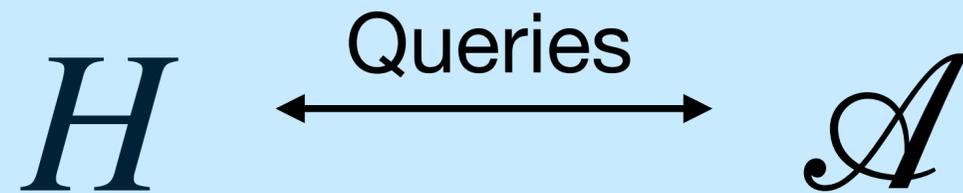
[Gilyén, Hastings and Vazirani, 2021] - There exists a stoquastic $H \in \mathbb{C}^{2^n \times 2^n}$ whose ground state can be computed efficiently i.e. in $poly(n)$ time by QAC but **any** classical algorithm requires $\Omega(2^{n^{\frac{1}{5}-o(1)}})$ time.

This talk

Open problem - Is there a stoquastic $H \in \mathbb{C}^{2^n \times 2^n}$ whose ground state can be computed efficiently i.e. in $poly(n)$ time by QAC but **any** classical algorithm requires $2^{\Omega(poly(n))}$ time?

Query model

All the separations are proved with respect to the query model



Query access to H

A can access $\langle i | H | j \rangle$, the (i, j) th entry in one query

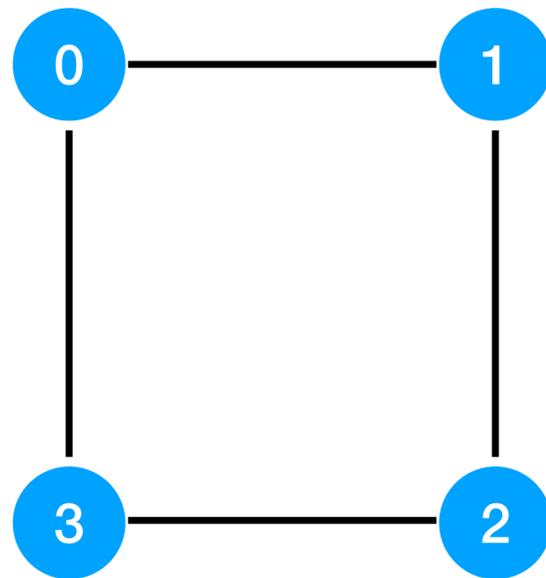
Algorithms are analyzed based on the number of queries made to H

No other information about H is given

For this talk: Queries \approx Time

Example of a stoquastic H

Let $G(V, E)$ be a graph as below,



Let $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ be states representing the vertices

$$H = - \sum_{i,j} A_{i,j} |i\rangle\langle j|$$

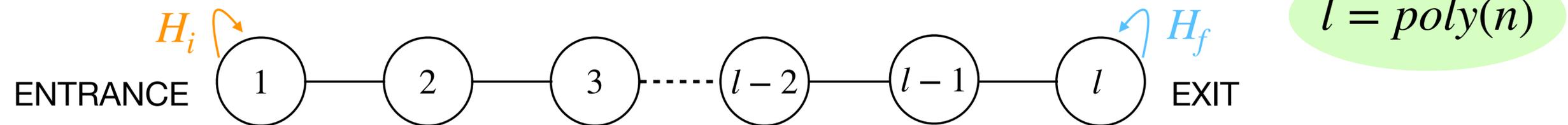
$H = -A$ Is a stoquastic hamiltonian

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

Graph G on 2^n vertices $\rightarrow H$ on n qubits

A simple search problem

Start at one endpoint of a path on l vertices and reach the other endpoint



Let A_l denote the adjacency matrix of the path $\langle k | A_l | k + 1 \rangle = \langle k + 1 | A_l | k \rangle = 1$

Initial hamiltonian

$$H_i = - |1\rangle\langle 1|$$

Ground state: $|1\rangle$

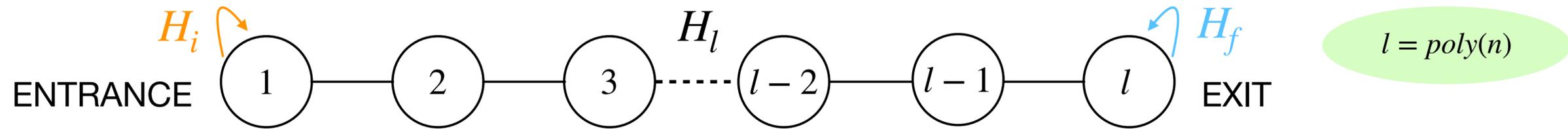
Stoquastic

Final hamiltonian

$$H_f = - |l\rangle\langle l|$$

Ground state: $|l\rangle$

A simple search problem

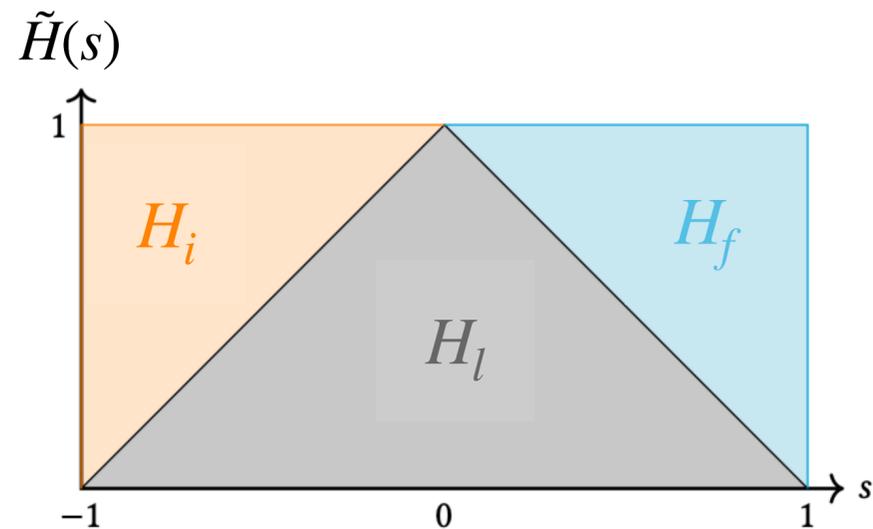


Stoquastic hamiltonian $H_l = -A_l$

$$\tilde{H}(s) = \begin{cases} (1+s)H_l - sH_i, & s \in [-1, 0] \\ (1-s)H_l + sH_f, & s \in [0, 1] \end{cases}$$

Adiabatic evolution $\tilde{H}(s)$ maps

$|ENT\rangle := |1\rangle$ to $|EXT\rangle := |l\rangle$

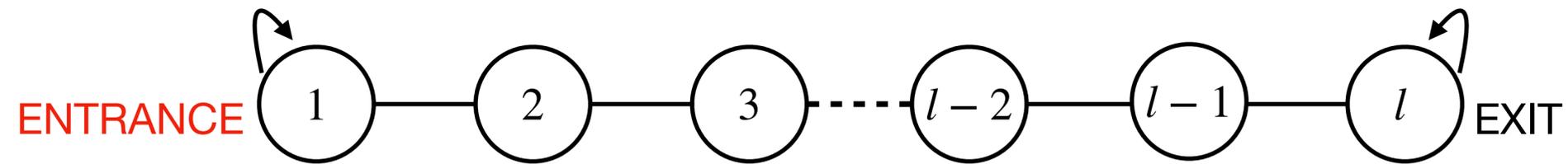


This takes just $O(l^2) = poly(n)$ queries because of the good spectral gap

Spectral gap of $\tilde{H}(s) - \Omega(1/l^2)$

A simple search problem

Start at one endpoint of a path on l vertices and reach the other endpoint

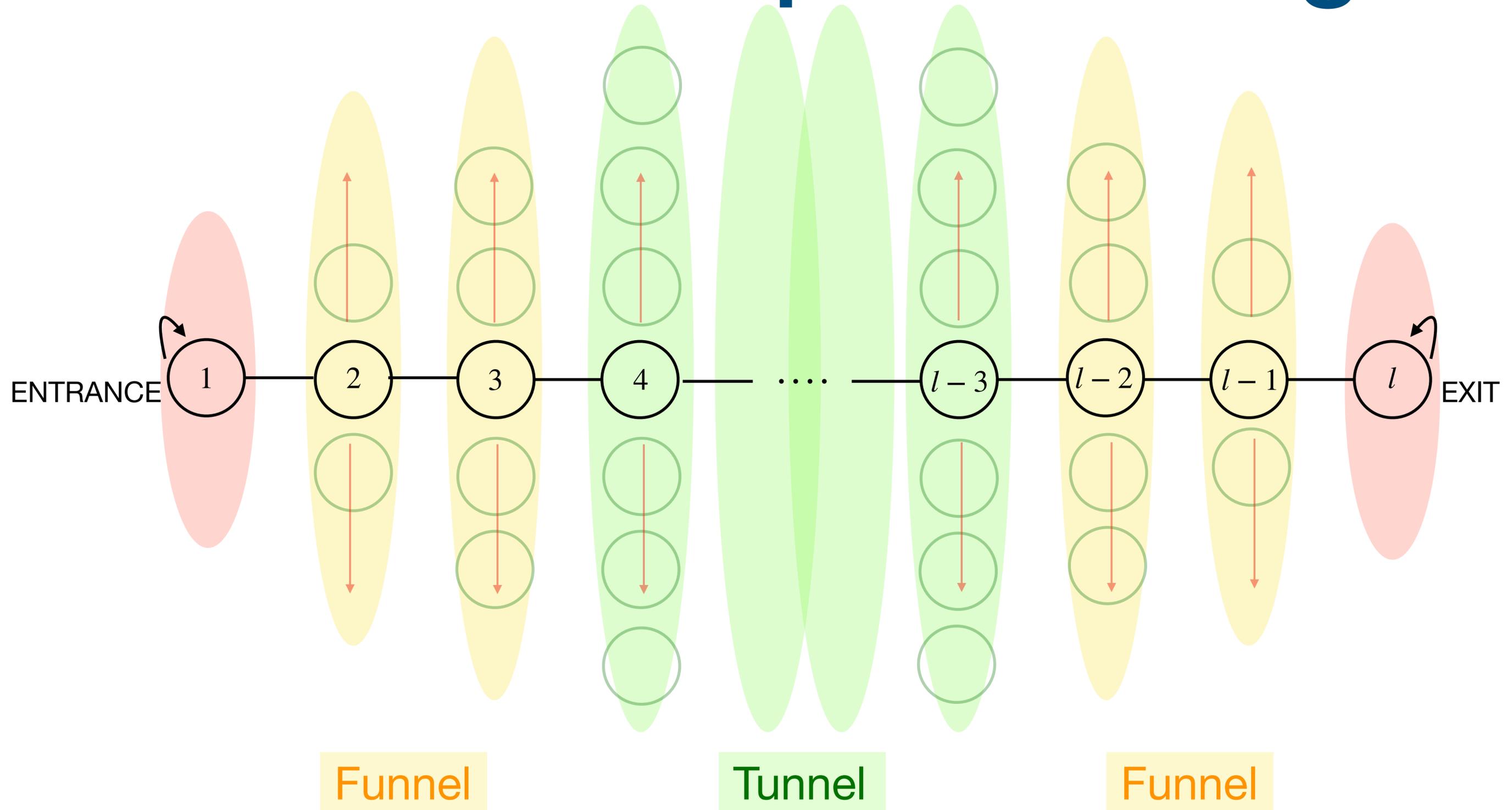


A non-backtracking walk can reach in just l steps!

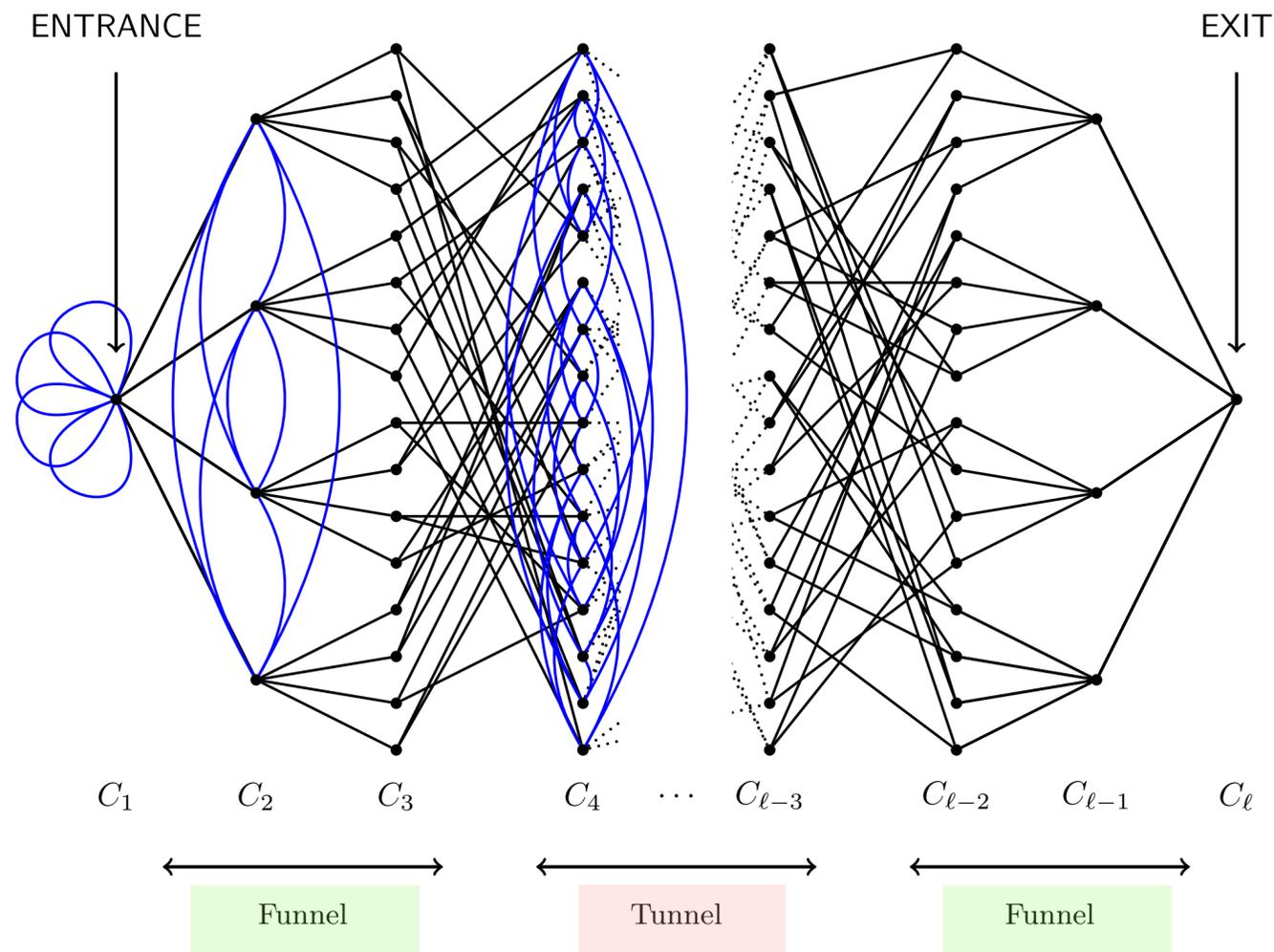
So, this is easy for both classical and QAC



Obfuscation of path of length l



Obfuscation of path of length l



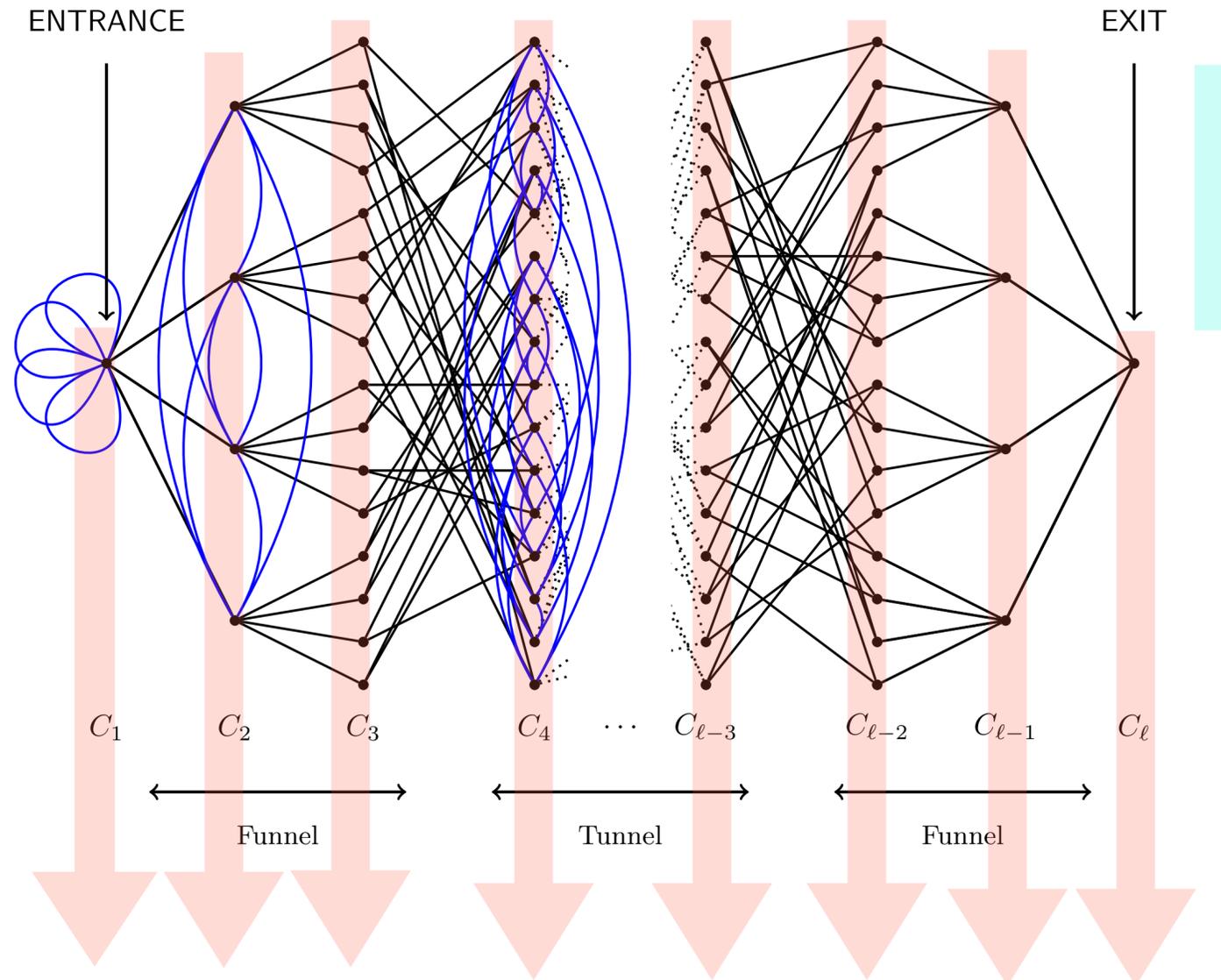
Here $k = 2$, $m = 2$

Vertices of path replaced by clusters of vertices
First and last k many clusters form a m^2 -ary tree (Funnel)
The middle clusters (size m^{2k}) are joined by m random matchings (Tunnel)

Blue edges: Random expanders of degree $2m$ added to each cluster

All vertices inside tunnel have degree $4m$
All funnel (except the last cluster of funnel) vertices have degree $m^2 + 1$
Border vertices have degree $3m + 1$

Obfuscation of path of length l



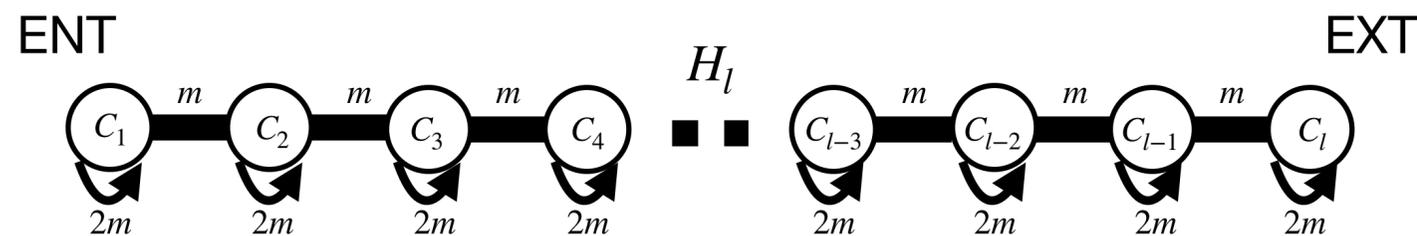
Dynamics can be simplified by projecting on to a line
There is a symmetric subspace here

This is the subspace spanned by the column

$$\text{states } |C_i\rangle = \frac{1}{\sqrt{|C_i|}} \sum_{x \in C_i} |x\rangle \text{ such that}$$

$$\langle C_{i+1} | H_l | C_i \rangle = m \text{ (matchings edges)}$$

$$\langle C_i | H_l | C_i \rangle = 2m \text{ (expanders)}$$



Here $k = 2, m = 2$

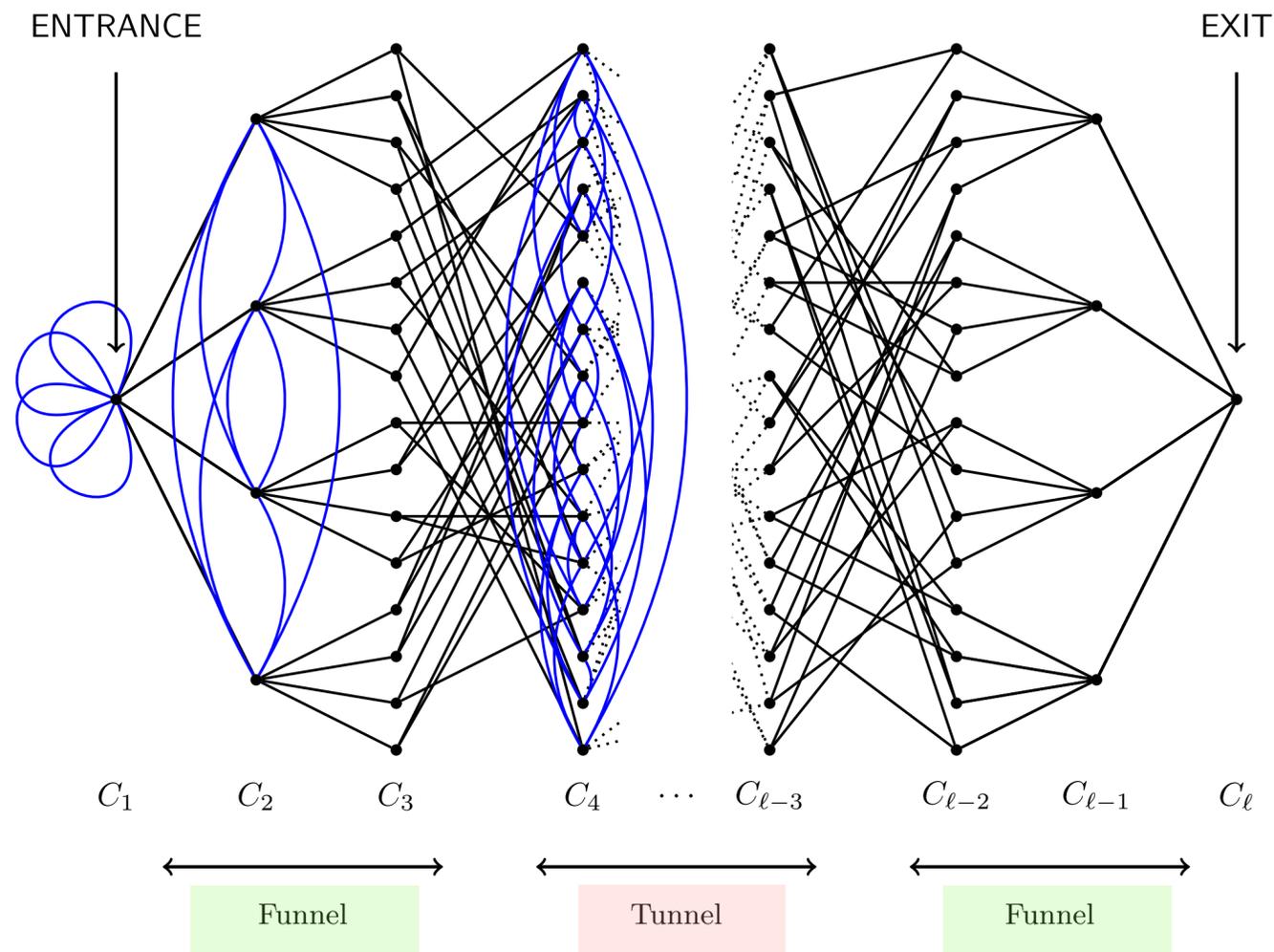
Recall

$$H_i = -2m |ENT\rangle\langle ENT|$$

$$H_f = -2m |EXT\rangle\langle EXT|$$

$$\tilde{H}(s) = \begin{cases} (1+s)H_i - sH_f, & s \in [-1,0] \\ (1-s)H_i + sH_f, & s \in [0,1] \end{cases}$$

Obfuscation of path of length l



Here $k = 2, m = 2$

Why are expanders needed?

They have very large spectral gap $> m$

This ensures that both the first and second largest eigenvectors of $\tilde{H}(s)$ come from the symmetric subspace of clusters

But since the dynamics \approx line graph, this symmetric subspace spanned by C_i s has spectral gap $\Omega(m/l^2)$
(same with weight m edges)

So, if $m = poly(n)$, QAC can find EXT in $poly(n)$ queries! ✓

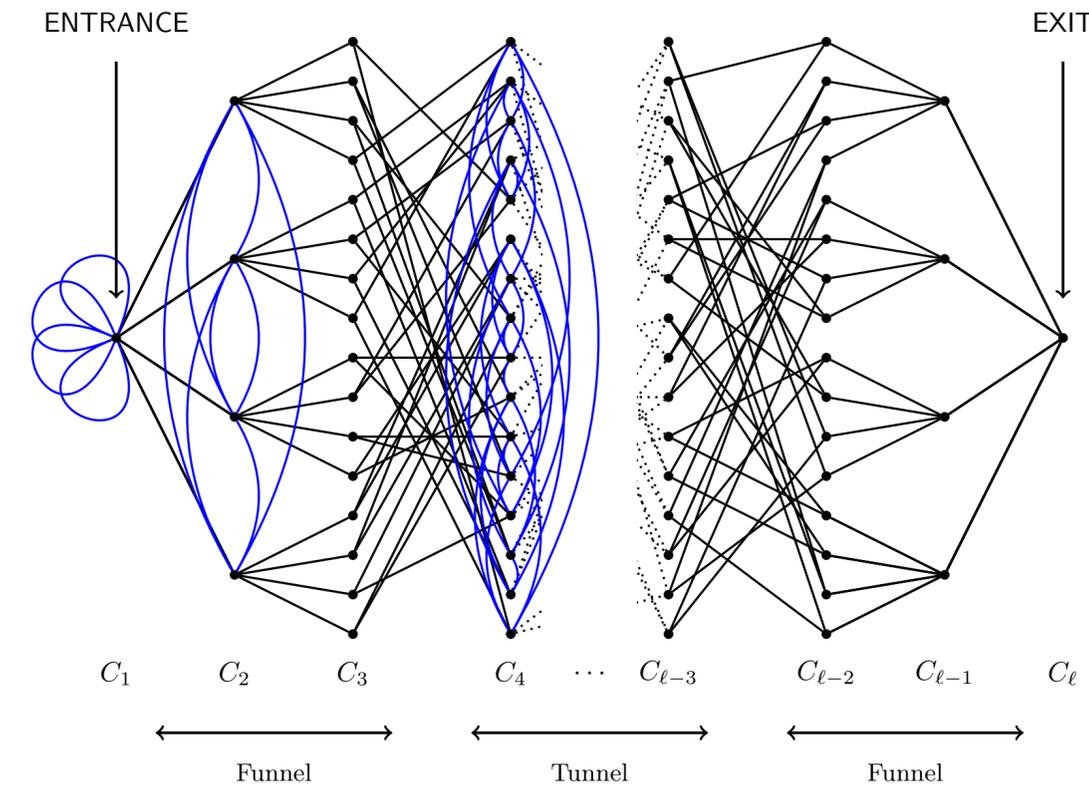
Obfuscation of path of length l

But, this is still classically easy! ❌

Observation: Tunnel vertices and Funnel vertices have different degrees

1. Find a vertex in the starting cluster of tunnel - non-backtracking walk from ENT
2. Reach end of tunnel by performing random walk over the tunnel.
3. Reach EXT by choosing the unique edge towards EXT at every vertex. (Non trivial)

Quick idea: sample many non-backtracking walks from all the neighbors of a vertex, only one has larger distance to the leaves of funnel.



→ Total queries - $poly(n)$

Vertices of path replaced by clusters of vertices

First and last k many clusters form a m^2 -ary tree (Funnel)

The middle clusters (size m^{2k}) are joined by m random matchings (Tunnel)

Obfuscation + Camouflage

Complete trees of fixed depth k are added to all vertices of the original graph - **decorations**

Internal nodes of trees have same degree $O(m)$ as the tunnel vertices

Is this sufficient to make it hard for classical?

No!

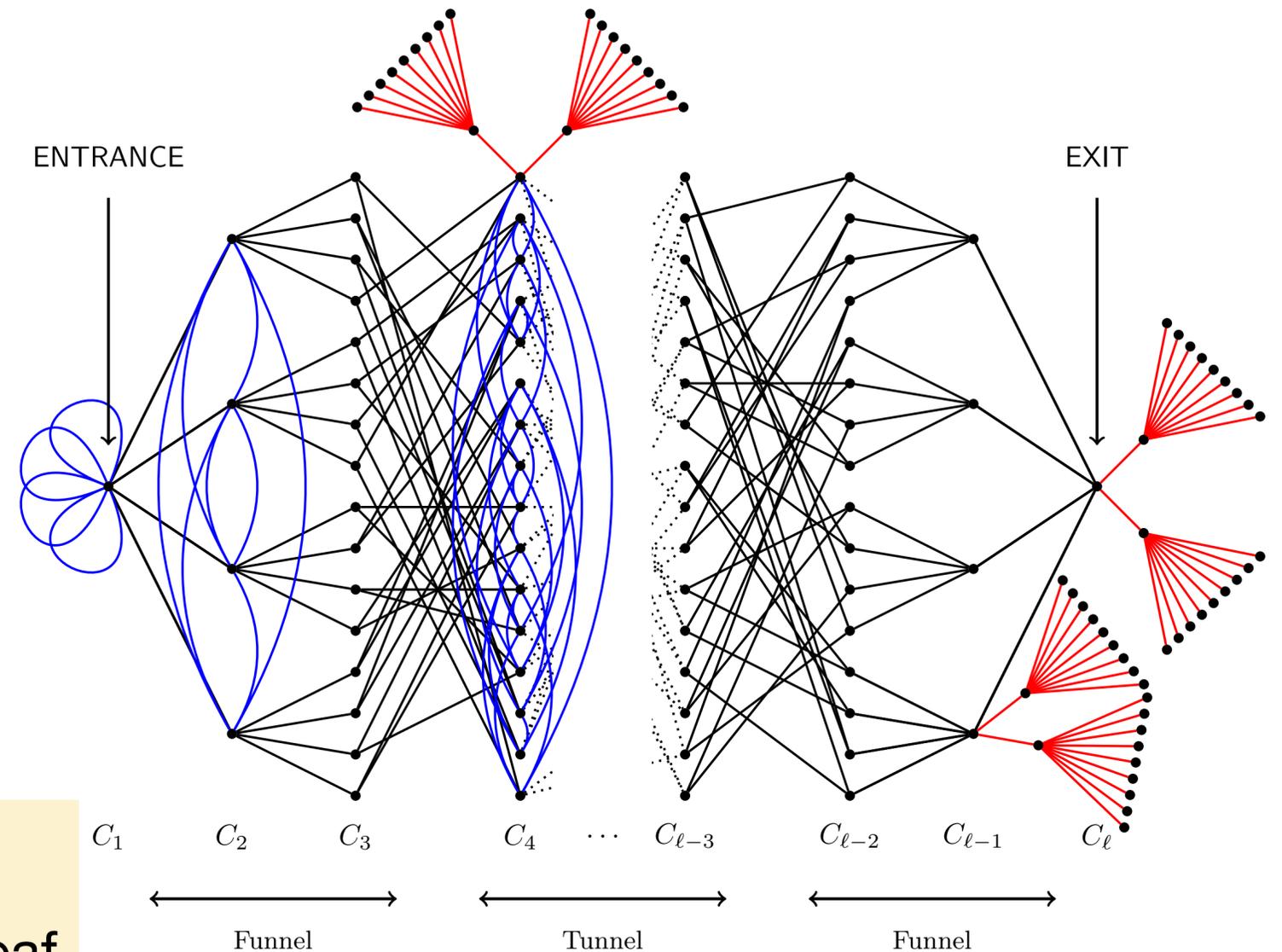


Observation: Degree 1 vertices only in tree

Non-backtracking walk from every vertex till it reaches a leaf

+

Trace back the path to root afterwards helps avoid the decorations



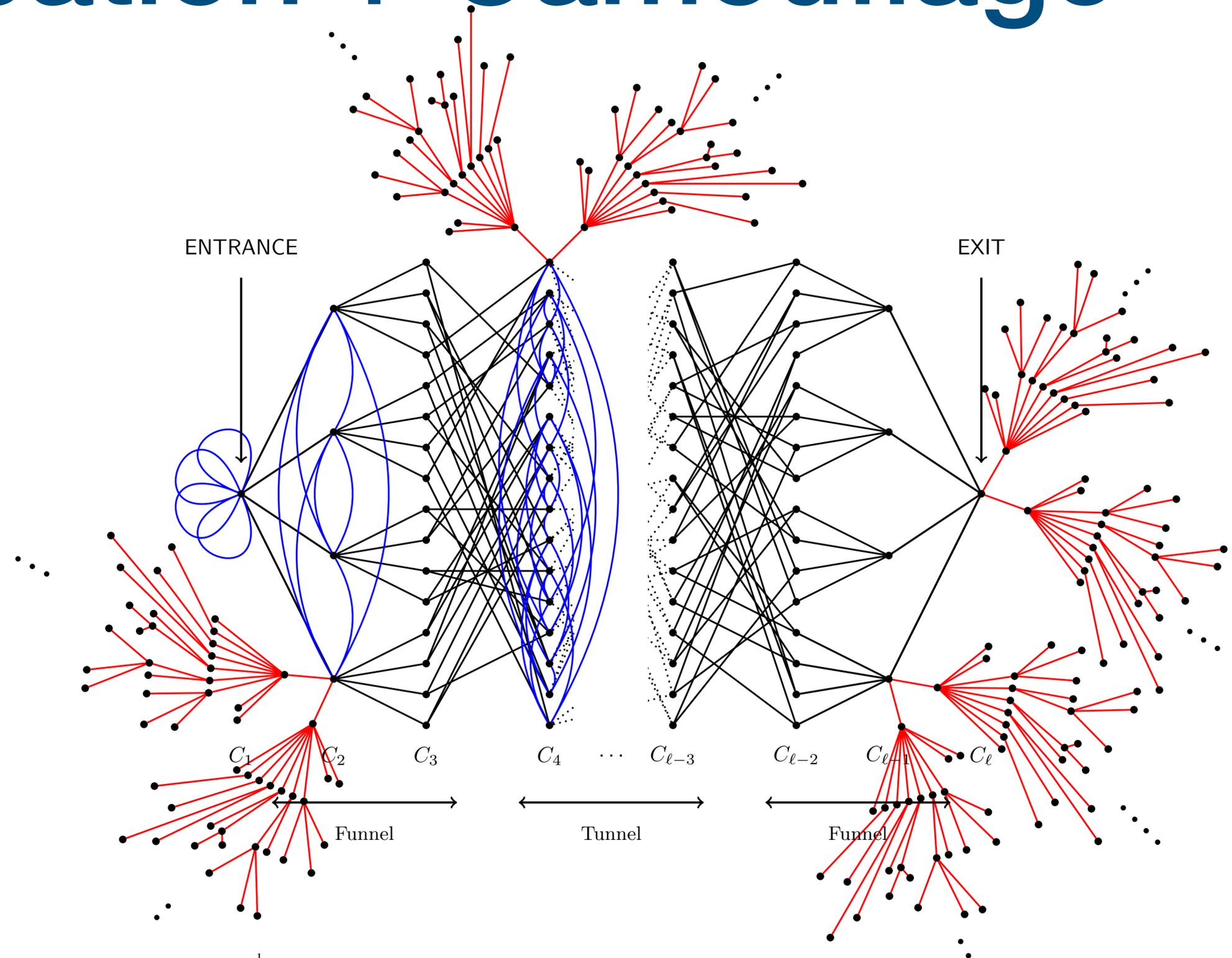
Obfuscation + Camouflage

Another attempt:

Trees have **fractal** shape
It looks similar everywhere

The deeper we move, the
harder it gets to come out
- looks the same always!

If we do only bounded
 $(2^{n^{\frac{1}{5}-o(1)}})$ number of
queries, probability of
detecting/coming out of
decoration once a
classical algorithm enters,
is extremely* small [Lemma 4,
GHV21]



* extremely = $< 2^{-n^{\frac{1}{5}-o(1)}}$

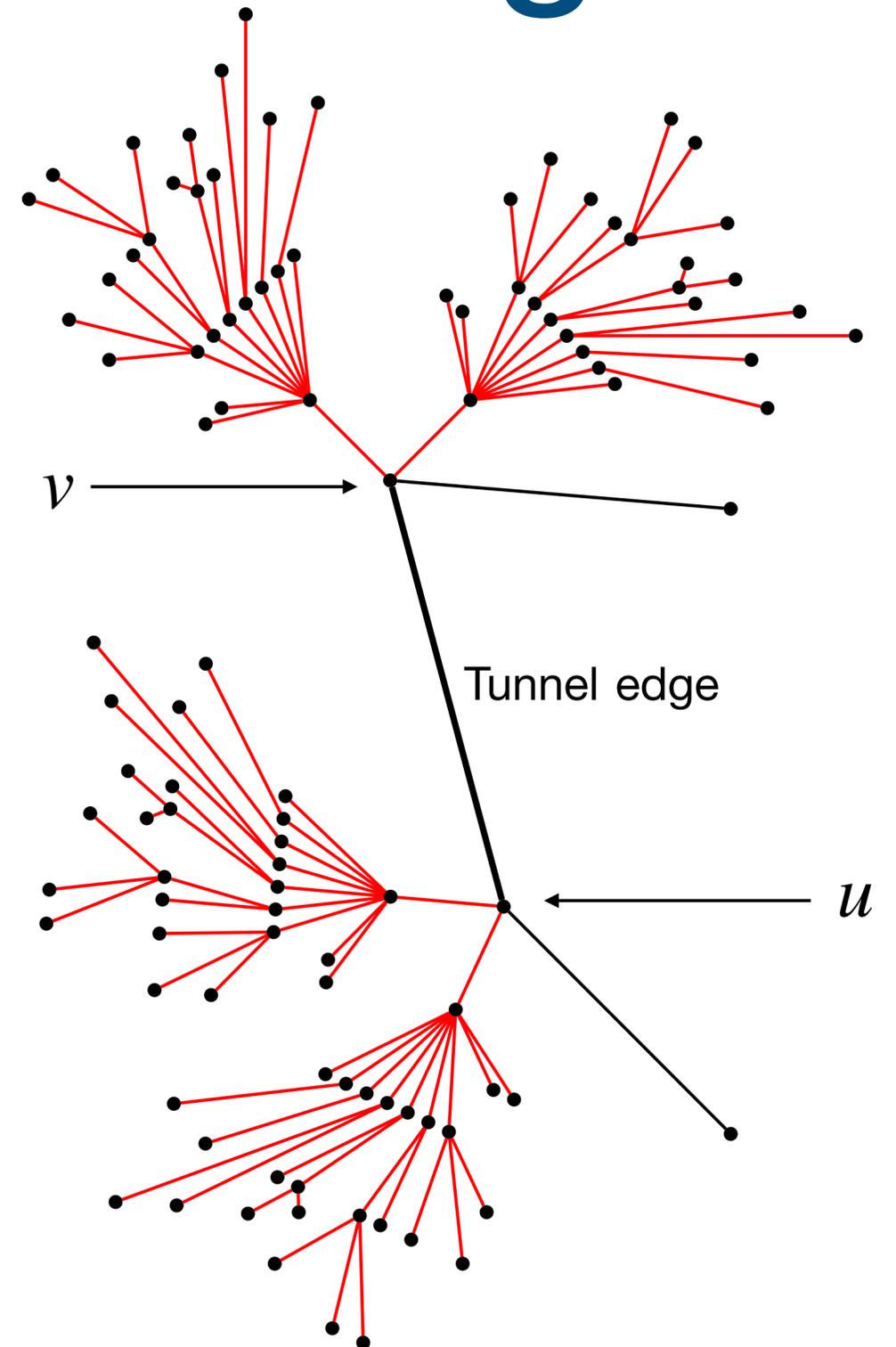
Obfuscation + Camouflage

Suppose the classical algorithm is exploring the graph from v

Due to the decorations, it is hard to detect the edge vu

Might take the red edge instead of vu

Both v and u have same decorations attached!



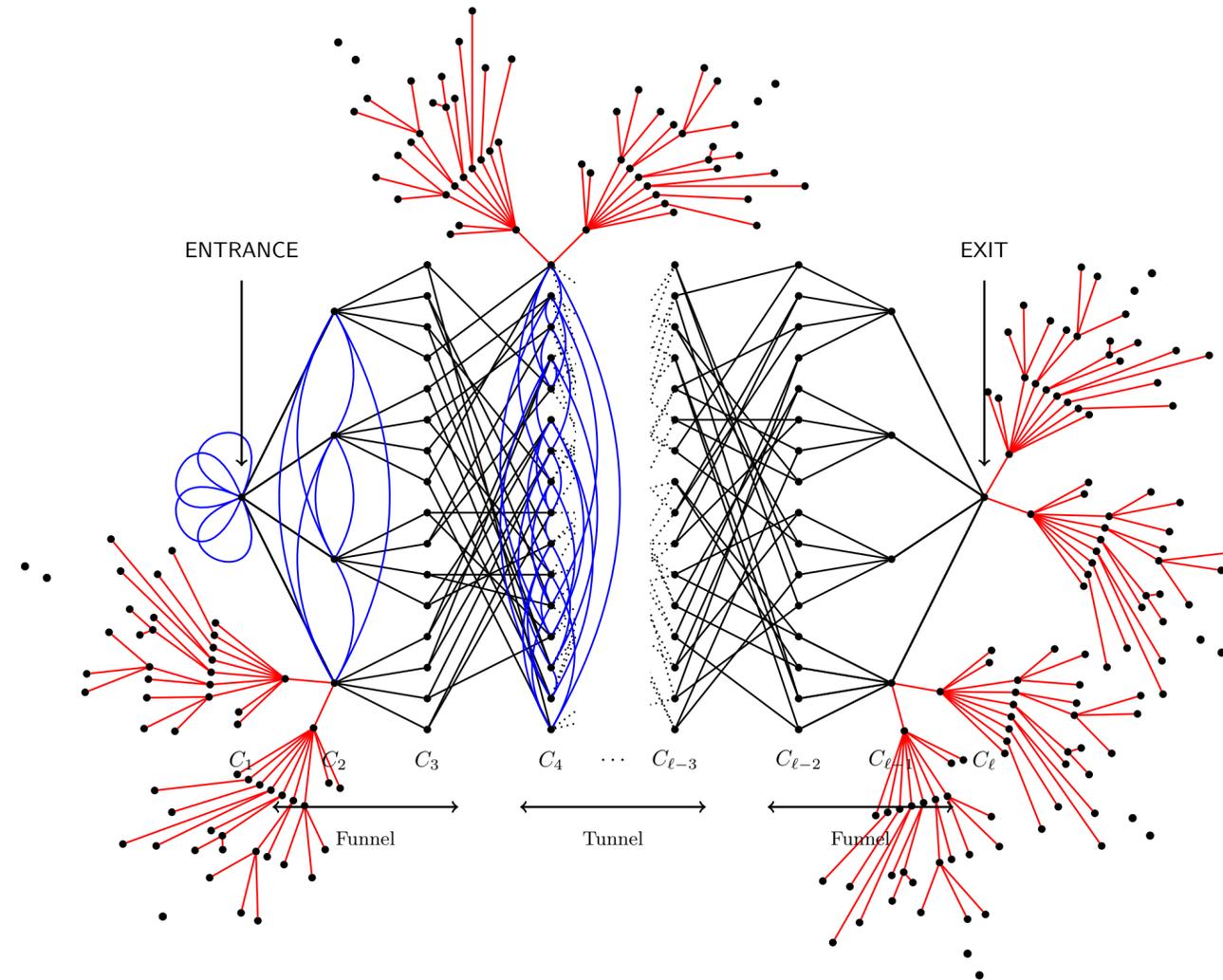
Obfuscation + Camouflage

Classical algorithms are **lured** into the decorations and find it **hard** to detect/come out of it!

On the other hand, spectral properties are preserved as well

$$\max_{\lambda} \text{Tree}_{\leq d} \leq 2\sqrt{d-1} \leq O(\sqrt{m})$$

Hence gap of $\tilde{H}(s)$ remains intact $\Omega(m/l^2)$



QAC still takes $poly(n)$ queries!



Due to *decorations*, **any** classical algorithm needs at least **sub-exponentially** many queries in n to find EXIT
[Lemma 4, GHV21]

Formal construction of decorations

$d_1 > d_2 > d_3 > \dots$ - degree

$l_1 < l_2 < l_3 < \dots$ - depth

Perfect tree $C(d, l)$ - Complete d -ary tree of depth l

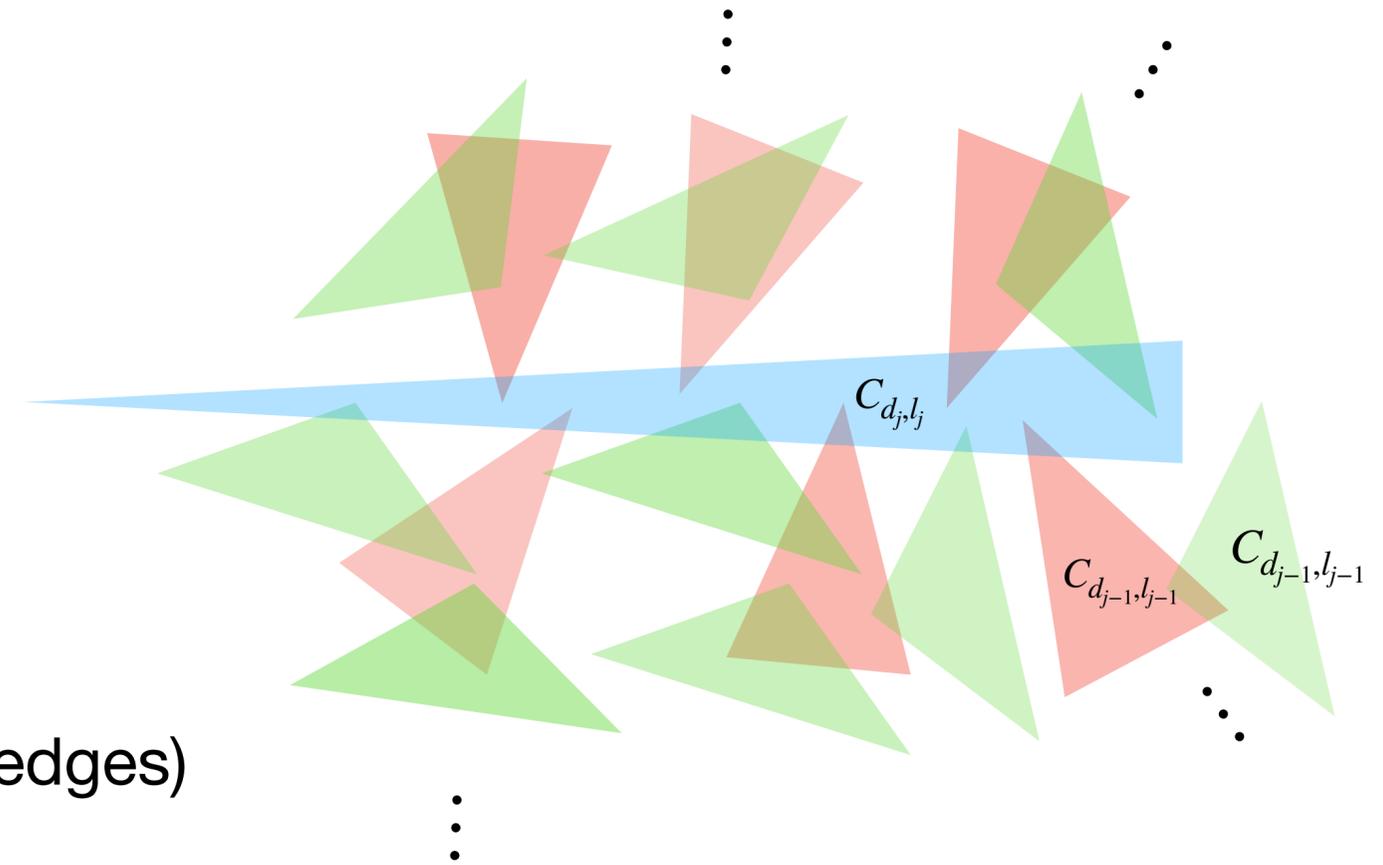
1-level decorated tree T_j : Take C_{d_j, l_j} and attach (via new edges) $d_{j-1} - d_j$ copies of $C_{d_{j-1}, l_{j-1}}$ to each internal node.

Full-level decorated tree \hat{T}_j : Take C_{d_j, l_j} and attach (via new edges)

$d_i - d_{i+1}$ copies of C_{d_i, l_i} to all current internal nodes for

$i = j - 1, j - 2, \dots, 1$.

All internal nodes of T_j have degree $d_{j-1} + 1$, and internal nodes in \hat{T}_j have degree $d_0 + 1$



Formal statement of classical hardness

Given query access to \hat{T}_j and any classical algorithm \mathcal{A} performing $q_{\mathcal{A}} < q_j$ queries

$$\Pr[\mathcal{A} \text{ queries a leaf vertex of } \hat{T}_j] \leq \frac{d_j^{(l_j - l_{j-1})/k}}{d_{j-1}} + q_j \cdot \max_{\mathcal{A}: q_{\mathcal{A}} < q_j/k} \Pr[\mathcal{A} \text{ queries a leaf vertex of } \hat{T}_{j-1}]$$

k denotes the number of distinct trees from which at least one leaf is visited

Conclusion and open questions

There exists a stoquastic H whose ground state can be prepared efficiently by QAC, but is hard for any classical algorithm

Hence QAC on stoquastic H is more powerful than classical computation

Is it possible to show an exponential separation?

Bottlenecks in current construction: Decoration size \gg Original graph ($2^{O(n^\alpha)}$ vs $2^{O(n)}$)

Hence only a sub-exponential separation

Is there a simpler decoration with fewer vertices?

Trees are naturally simple and have good spectral properties

Can stoquastic H with QAC-efficient but classically hard ground state preparation be constructed without using graph adjacency matrices?

From discriminant matrices of Markov Chains perhaps?

Thank you!