UNIVERSITY OF WATERLOO

On the complexity of quantum partition functions

David Gosset

Based on:

Sergey Bravyi, Anirban Chowdhury, DG, Pawel Wocjan. *Nature Physics*, 18(11):1367-1370, 2022

Sergey Bravyi, Anirban Chowdhury, DG, Vojtech Havlicek, Guanyu Zhu. arXiv:2302.11454, 2023

Sergey Bravyi, DG. *Phys. Rev. Lett*. 119, 100503, 2017

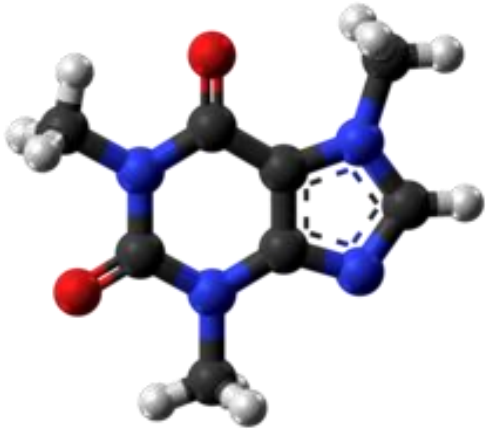
IQC Institute for Quantum Computing

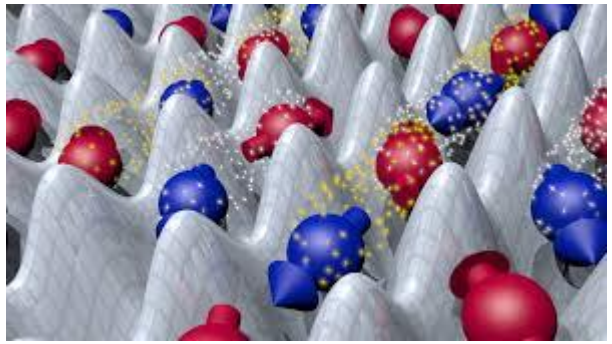
PI PERIMETER INSTITUTE

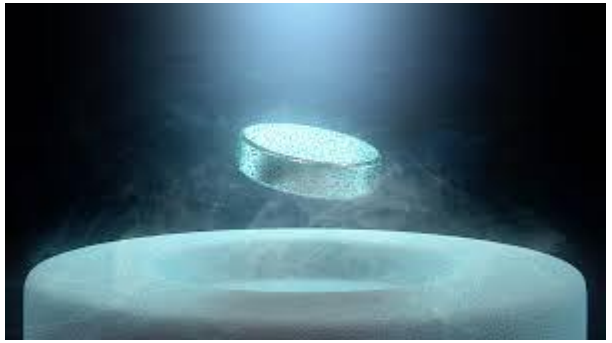
IBM

Complexity of many-body systems

Many-body systems in nature have **local interactions.**





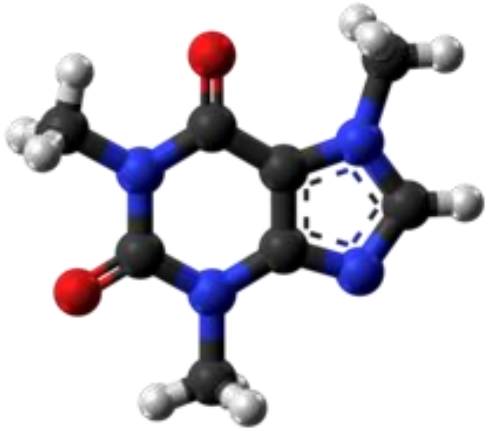


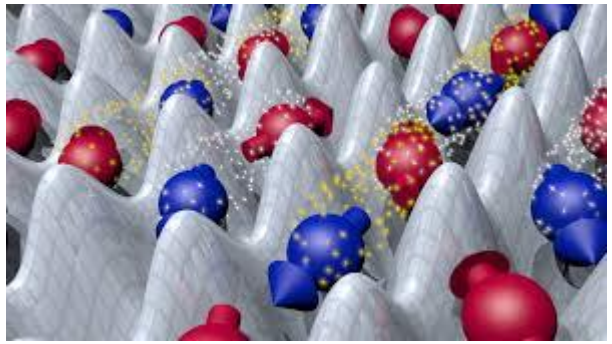
Complexity of many-body systems

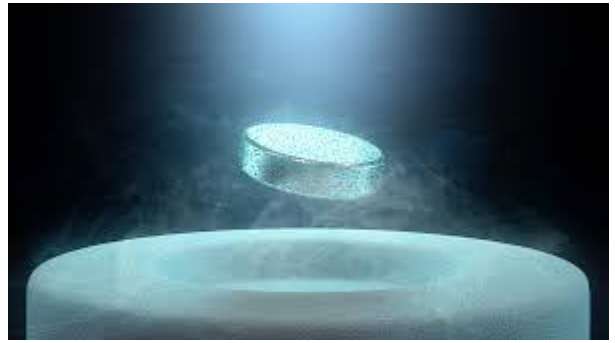
Many-body systems in nature have **local interactions.**

$$H = \sum_c h_c$$

Interactions between particles (qubits, bosons, fermions, ….)





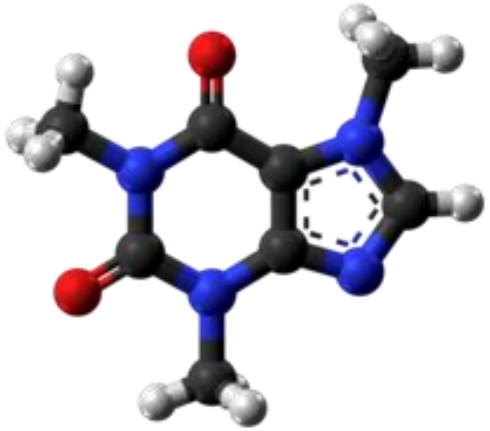


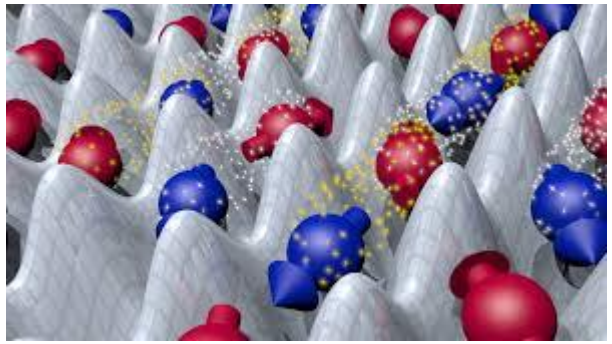
Complexity of many-body systems

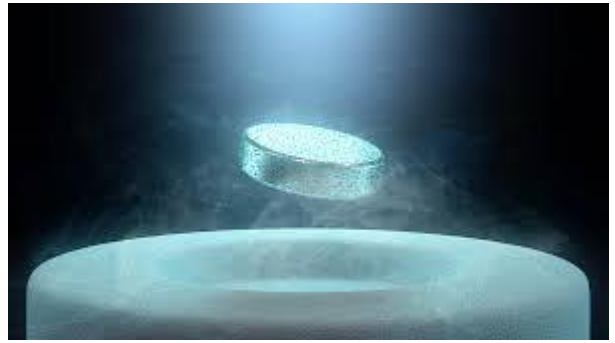
Many-body systems in nature have **local interactions.**

$$H = \sum_c h_c$$

Interactions between particles (qubits, bosons, fermions, ….)







How hard is it to compute their physical properties?

Ground energy problems are intractable optimization problems

Computing the **ground energy** of **classical** locally interacting systems is an NP-hard (intractable) optimization problem

e.g. 3-SAT

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

$C(z) =$ Number of clauses violated by z

Ground energy problems are intractable optimization problems

Computing the **ground energy** of **classical** locally interacting systems is an NP-hard (intractable) optimization problem

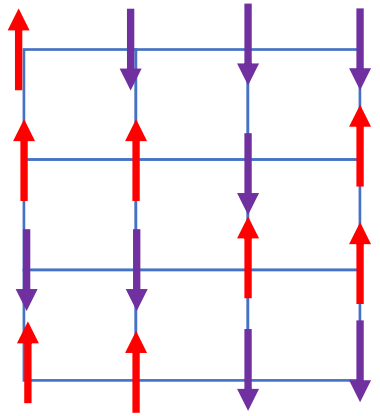
e.g. 3-SAT

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

$C(z) =$ Number of clauses violated by z

e.g. 2D Ising with magnetic field [Barahona 1982]

$$H = \sum_{(i,j) \in E(G)} J_{ij} Z_i Z_j + h_i Z_i$$



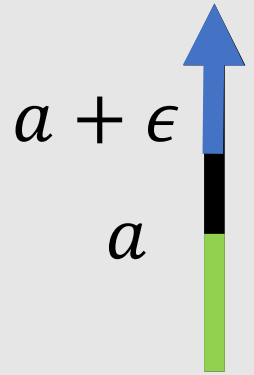
Quantum Cook-Levin theorem (informal):
Computing the ground energy of an n particle quantum system with local interactions to precision $1/\mathrm{poly}(n)$ is an **intractable quantum optimization problem.**

The local Hamiltonian problem

Input: An n-qubit local Hamiltonian H, a number $a \in \mathbb{R}$, a precision $\epsilon = 1/n^{\alpha}$

Output: (YES) The ground energy of H is $\leq a$
(NO) The ground energy of H is $\geq a + \epsilon$

(promised one of these conditions is satisfied)

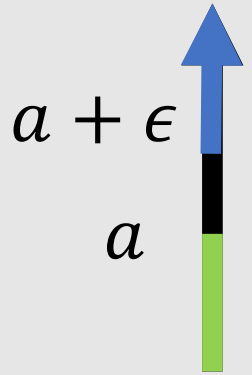


The local Hamiltonian problem

Input: An n-qubit local Hamiltonian H, a number $a \in \mathbb{R}$, a precision $\epsilon = 1/n^{\alpha}$

Output: (YES) The ground energy of H is $\leq a$
(NO) The ground energy of H is $\geq a + \epsilon$

(promised one of these conditions is satisfied)



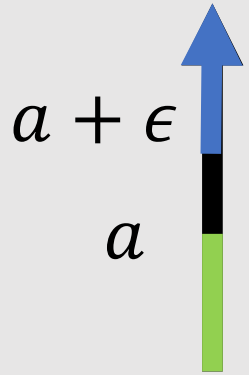
Theorem (Kitaev 1999)
The local Hamiltonian problem is **QMA-complete.**

The local Hamiltonian problem

Input: An n-qubit local Hamiltonian H, a number $a \in \mathbb{R}$, a precision $\epsilon = 1/n^{\alpha}$

Output: (YES) The ground energy of H is $\leq a$
(NO) The ground energy of H is $\geq a + \epsilon$

(promised one of these conditions is satisfied)



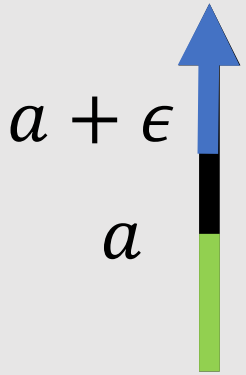
Theorem (Kitaev 1999)
The local Hamiltonian problem is **QMA-complete.** ← Quantum analogue of NP

The local Hamiltonian problem

Input: An n-qubit local Hamiltonian H, a number $a \in \mathbb{R}$, a precision $\epsilon = 1/n^{\alpha}$

Output: (YES) The ground energy of H is $\leq a$
(NO) The ground energy of H is $\geq a + \epsilon$

(promised one of these conditions is satisfied)

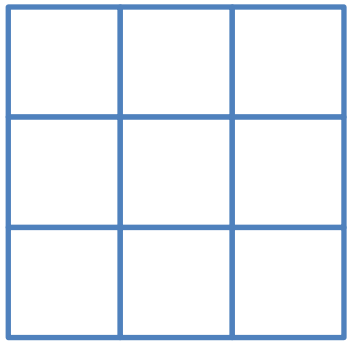


Theorem (Kitaev 1999)
The local Hamiltonian problem is **QMA-complete.** ← Quantum analogue of NP

The local Hamiltonian problem remains QMA-complete even for systems with 2-qubit interactions [Kempe Kitaev Regev 2004]

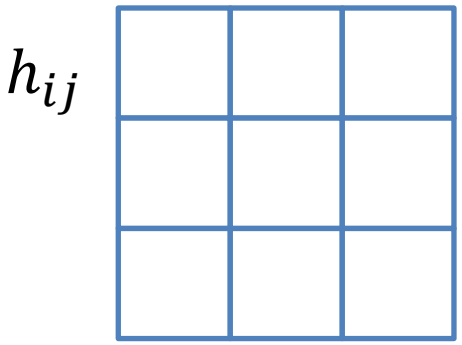
More examples of systems with QMA-complete ground energy problems:

h_{ij}

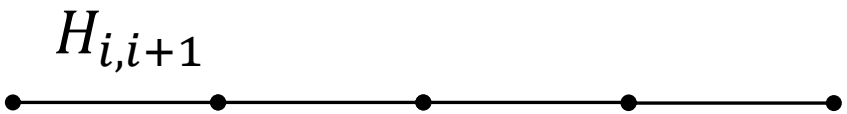


2-local Hamiltonian on a 2D grid
[Oliveira Terhal 2008]

More examples of systems with QMA-complete ground energy problems:

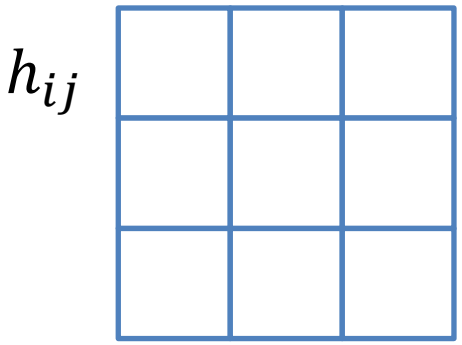


2-local Hamiltonian on a 2D grid
[Oliveira Terhal 2008]

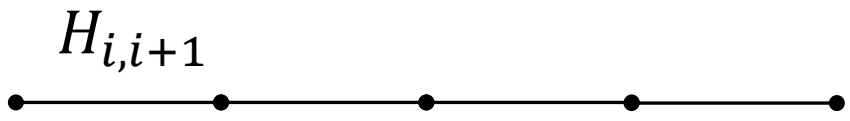


2-local Hamiltonian on a line with qudits
[Aharonov et. al 2009] [Gottesman Irani 2009]

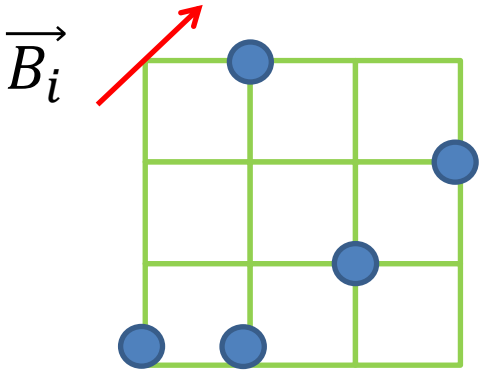
More examples of systems with QMA-complete ground energy problems:



2-local Hamiltonian on a 2D grid
[Oliveira Terhal 2008]

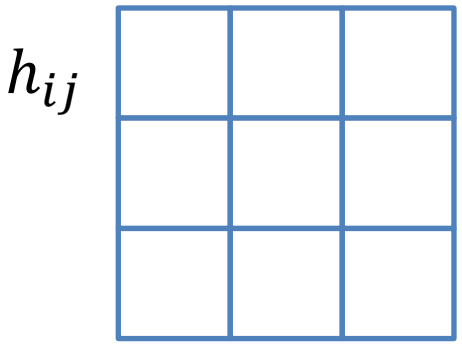


2-local Hamiltonian on a line with qudits
[Aharonov et. al 2009] [Gottesman Irani 2009]

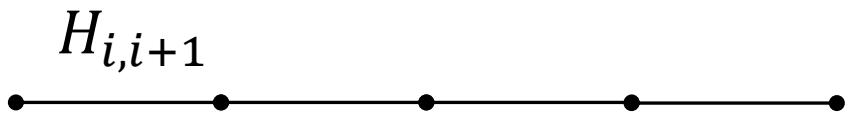


Hubbard model on a 2D grid with site-dependent magnetic field
[Schuch Verstraete 2009].

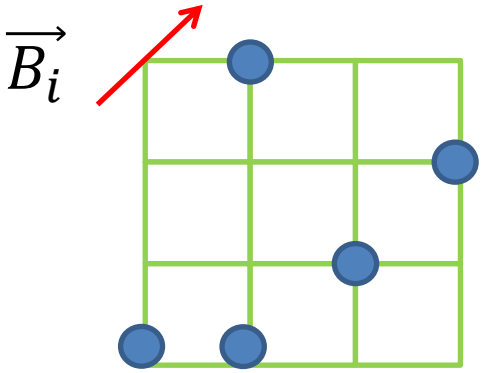
More examples of systems with QMA-complete ground energy problems:



2-local Hamiltonian on a 2D grid
[Oliveira Terhal 2008]



2-local Hamiltonian on a line with qudits
[Aharonov et. al 2009] [Gottesman Irani 2009]



Hubbard model on a 2D grid with site-dependent magnetic field
[Schuch Verstraete 2009].

There is also an important class of ground energy problems associated with **stoquastic** Hamiltonians that lies in between NP and QMA… [Bravyi Divincenzo Oliveira Terhal 2006]

Three flavours of intractable optimization problems

$$H = \sum_{1 \le i < j \le n} h(i,j)$$

		LH problem	
Ising model	$h(i,j) = \alpha_{ij} Z_i Z_j$	NP-complete	**(Classical)**
XY model	$h(i,j) = \alpha_{ij}(X_i X_j + Y_i Y_j)$	QMA-complete [Cubitt Montanaro 2013]	**(Quantum)**

Three flavours of intractable optimization problems

$$H = \sum_{1 \le i < j \le n} h(i,j)$$

		LH problem	
Ising model	$h(i,j) = \alpha_{ij} Z_i Z_j$	NP-complete	(Classical)
Transverse-field Ising model	$h(i,j) = \alpha_{ij} X_i X_j - \gamma_i Z_i - \gamma_j Z_j$	StoqMA-complete [Bravyi, Hastings 2014]	(Stoquastic)
XY model	$h(i,j) = \alpha_{ij}(X_i X_j + Y_i Y_j)$	QMA-complete [Cubitt Montanaro 2013]	(Quantum)

Three flavours of intractable optimization problems

$$H = \sum_{1 \le i < j \le n} h(i,j)$$

		LH problem	
Ising model	$h(i,j) = \alpha_{ij} Z_i Z_j$	NP-complete	(Classical)
Transverse-field Ising model	$h(i,j) = \alpha_{ij} X_i X_j - \gamma_i Z_i - \gamma_j Z_j$	StoqMA-complete [Bravyi, Hastings 2014]	(Stoquastic)
XY model	$h(i,j) = \alpha_{ij}(X_i X_j + Y_i Y_j)$	QMA-complete [Cubitt Montanaro 2013]	(Quantum)

These examples illustrate all nontrivial possibilities within the framework of [Cubitt Montanaro 2013])

Three flavours of intractable optimization problems

$$H = \sum_{1 \le i < j \le n} h(i,j)$$

		LH problem	
Ising model	$h(i,j) = \alpha_{ij} Z_i Z_j$	NP-complete	(Classical)
Transverse-field Ising model	$h(i,j) = \alpha_{ij} X_i X_j - \gamma_i Z_i - \gamma_j Z_j$	StoqMA-complete [Bravyi, Hastings 2014]	(Stoquastic)
XY model	$h(i,j) = \alpha_{ij}(X_i X_j + Y_i Y_j)$	QMA-complete [Cubitt Montanaro 2013]	(Quantum)

These examples illustrate all nontrivial possibilities within the framework of [Cubitt Montanaro 2013])

There are also some very special examples where local Hamiltonian problems are easy.

Beyond ground energy

How hard is it to compute physical properties in **thermal equilibrium**?

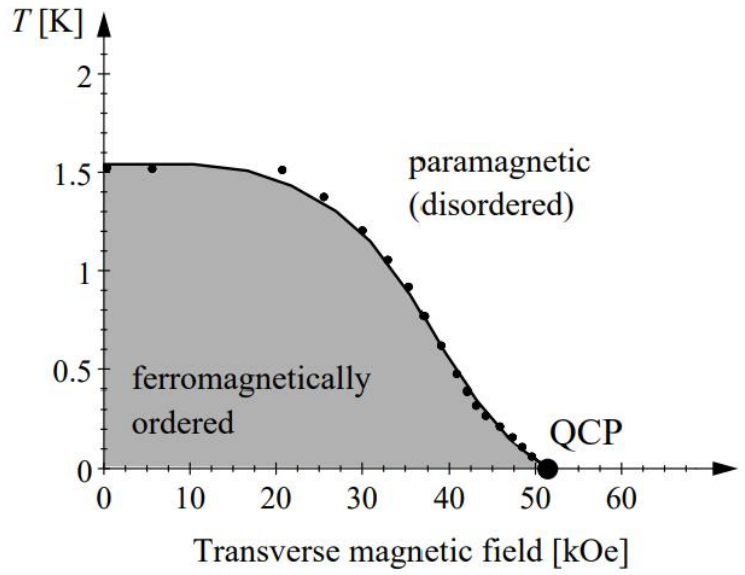


Figure 2. Magnetic phase diagram of $LiHoF_4$ (after [5]). The ordered phase breaks a Z_2 symmetry of the Hamiltonian and is bounded by a line of finite-temperature phase transitions. This line terminates in the quantum critical point (QCP), where ferromagnetic order is destroyed solely by quantum fluctuations.

Vojta M. Quantum phase transitions. Reports on Progress in Physics. 2003 Nov 3;66(12):2069.

Bitko, D., T. F. Rosenbaum, and G. Aeppli. *PRL* 77.5 (1996): 940.

Beyond ground energy

How hard is it to compute physical properties in **thermal equilibrium**?

$$Z = \mathrm{Tr}[e^{-\beta H}]$$

Partition function

$$F = -\beta^{-1}\log(Z)$$

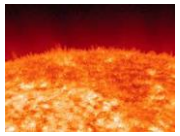
Free energy

$$\langle O \rangle = \frac{\mathrm{Tr}[Oe^{-\beta H}]}{Z}$$

Thermal expected values

Beyond ground energy

How hard is it to compute physical properties in **thermal equilibrium**?



$\beta = 0$

$$Z = \mathrm{Tr}[e^{-\beta H}]$$

Partition function

$$F = -\beta^{-1}\log(Z)$$

Free energy

$$\langle O \rangle = \frac{\mathrm{Tr}[Oe^{-\beta H}]}{Z}$$

Thermal expected values

Complexity of partition functions

$$Z = \mathrm{Tr}[e^{-\beta H}]$$

Partition function

Exact computation as hard as #P

e.g., can count the number of solutions to a 3-SAT formula

Complexity of partition functions

$$Z = \mathrm{Tr}[e^{-\beta H}]$$

Partition function

Exact computation as hard as #P

e.g., can count the number of solutions to a 3-SAT formula

We are interested in relative-error approximation…

Why relative-error approximation?

QPF problem: Given H, β and $\delta > 0$, compute an estimate $\widetilde{Z}$ satisfying

$$(1-\delta)\mathrm{Tr}[e^{-\beta H}] \leq \widetilde{Z} \leq (1+\delta)\mathrm{Tr}[e^{-\beta H}]$$

Defines a robust class of computational problems that are equivalent under polynomial-time reductions.

Why relative-error approximation?

QPF problem: Given H, β and $\delta > 0$, compute an estimate $\widetilde{Z}$ satisfying

$$(1-\delta)\mathrm{Tr}[e^{-\beta H}] \leq \widetilde{Z} \leq (1+\delta)\mathrm{Tr}[e^{-\beta H}]$$

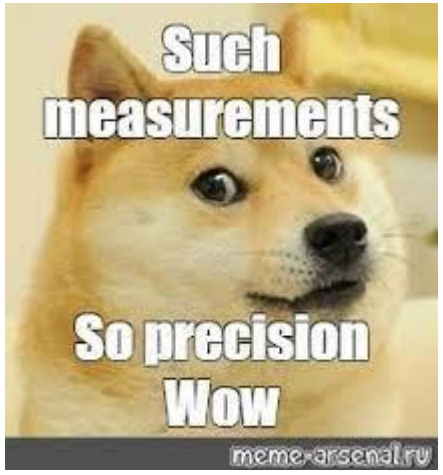
Defines a **robust** class of computational problems that are equivalent under polynomial-time reductions.

Why relative-error approximation?

QPF problem: Given H, β and $\delta > 0$, compute an estimate $\tilde{Z}$ satisfying

$$(1-\delta)\mathrm{Tr}[e^{-\beta H}] \leq \tilde{Z} \leq (1+\delta)\mathrm{Tr}[e^{-\beta H}]$$

Defines a **robust** class of computational problems that are equivalent under polynomial-time reductions.



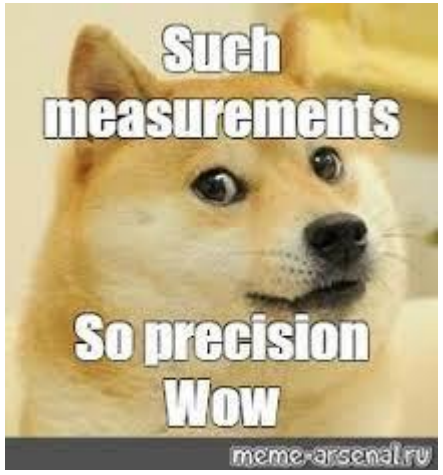
WlOG may take precision parameter to be a constant. In particular $\delta = 0.25$ is equivalent to the general case with $\delta = \frac{1}{poly(n)}$.

Why relative-error approximation?

QPF problem: Given H, β and $\delta > 0$, compute an estimate $\tilde{Z}$ satisfying

$$(1-\delta)\mathrm{Tr}[e^{-\beta H}] \leq \tilde{Z} \leq (1+\delta)\mathrm{Tr}[e^{-\beta H}]$$

Defines a **robust** class of computational problems that are equivalent under polynomial-time reductions.



WlOG may take precision parameter to be a constant. In particular $\delta = 0.25$ is equivalent to the general case with $\delta = \frac{1}{poly(n)}$.

Why? Partition function of L noninteracting copies of H is $\left(\mathrm{Tr}[e^{-\beta H}]\right)^L$
Compute this to constant error and then take the Lth root.

Why relative-error approximation?

QPF problem: Given H, β and $\delta > 0$, compute an estimate $\widetilde{Z}$ satisfying

$$(1-\delta)\mathrm{Tr}[e^{-\beta H}] \leq \widetilde{Z} \leq (1+\delta)\mathrm{Tr}[e^{-\beta H}]$$

Defines a **robust** class of computational problems that are equivalent under polynomial-time reductions.

Why relative-error approximation?

QPF problem: Given H, β and $\delta > 0$, compute an estimate $\widetilde{Z}$ satisfying

$$(1-\delta)\mathrm{Tr}[e^{-\beta H}] \leq \widetilde{Z} \leq (1+\delta)\mathrm{Tr}[e^{-\beta H}]$$

Defines a **robust** class of computational problems that are equivalent under polynomial-time reductions.

General case is equivalent to special case with 2-local Hamiltonians

T. S. Cubitt, A. Montanaro, and S. Piddock, "Universal quantum Hamiltonians" PNAS 115, no. 38, pp. 9497–9502, 2018.

Why relative-error approximation?

QPF problem: Given H, β and $\delta > 0$, compute an estimate $\widetilde{Z}$ satisfying

$$(1-\delta)\mathrm{Tr}[e^{-\beta H}] \leq \widetilde{Z} \leq (1+\delta)\mathrm{Tr}[e^{-\beta H}]$$

Defines a robust class of computational problems that are equivalent under polynomial-time reductions.

Why relative-error approximation?

QPF problem: Given H, β and $\delta > 0$, compute an estimate $\widetilde{Z}$ satisfying

$$(1-\delta)\mathrm{Tr}[e^{-\beta H}] \leq \widetilde{Z} \leq (1+\delta)\mathrm{Tr}[e^{-\beta H}]$$

Defines a robust class of computational problems that are equivalent under polynomial-time reductions.

Estimating thermal mean values (additive error)

Estimating density of states of a local Hamiltonian (relative error)

Quantum approximate counting

…

Why relative-error approximation?

QPF problem: Given H, β and $\delta > 0$, compute an estimate $\widetilde{Z}$ satisfying

$$(1-\delta)\mathrm{Tr}[e^{-\beta H}] \leq \widetilde{Z} \leq (1+\delta)\mathrm{Tr}[e^{-\beta H}]$$

QPF is polynomial time equivalent to:

Estimating thermal mean values (additive error)

Given $H, \beta, \epsilon > 0$, and a Pauli operator P, compute an estimate μ satisfying

$$\left| \mu - \frac{\mathrm{Tr}[Pe^{-\beta H}]}{Z} \right| \leq \epsilon$$

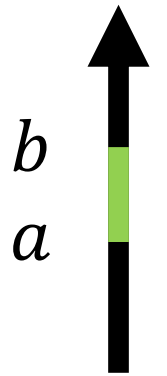
Why relative-error approximation?

QPF problem: Given H, β and $\delta > 0$, compute an estimate $\widetilde{Z}$ satisfying

$$(1-\delta)\mathrm{Tr}[e^{-\beta H}] \leq \widetilde{Z} \leq (1+\delta)\mathrm{Tr}[e^{-\beta H}]$$

QPF is polynomial time equivalent to:

Let $m_{[a,b]}$ be the number of eigenvalues of H in an interval $[a, b]$



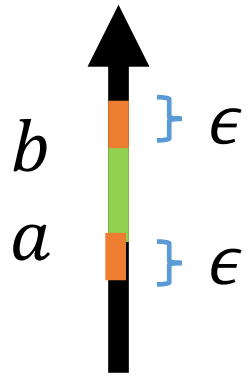
Why relative-error approximation?

QPF problem: Given H, β and $\delta > 0$, compute an estimate $\tilde{Z}$ satisfying

$$(1-\delta)\mathrm{Tr}[e^{-\beta H}] \leq \tilde{Z} \leq (1+\delta)\mathrm{Tr}[e^{-\beta H}]$$

QPF is polynomial time equivalent to:

Let $m_{[a,b]}$ be the number of eigenvalues of H in an interval $[a, b]$



Why relative-error approximation?

QPF problem: Given H, β and $\delta > 0$, compute an estimate $\widetilde{Z}$ satisfying

$$(1-\delta)\mathrm{Tr}[e^{-\beta H}] \leq \widetilde{Z} \leq (1+\delta)\mathrm{Tr}[e^{-\beta H}]$$

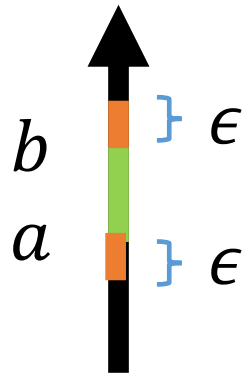
QPF is polynomial time equivalent to:

Let $m_{[a,b]}$ be the number of eigenvalues of H in an interval $[a, b]$

Estimating density of states (relative error)

Given H and two thresholds $a < b$ and precision parameters ϵ, δ output an estimate *m* satisfying

$$(1-\delta)m_{[a,b]} \leq m \leq (1+\delta)m_{[a-\epsilon,b+\epsilon]}$$



QPF problem: Given H, β and $\delta > 0$, compute an estimate $\widetilde{Z}$ satisfying

$$(1-\delta)\mathrm{Tr}[e^{-\beta H}] \leq \widetilde{Z} \leq (1+\delta)\mathrm{Tr}[e^{-\beta H}]$$

Its complexity is largely a mystery…

QPF problem: Given H, β and $\delta > 0$, compute an estimate $\widetilde{Z}$ satisfying

$$(1-\delta)\mathrm{Tr}[e^{-\beta H}] \leq \widetilde{Z} \leq (1+\delta)\mathrm{Tr}[e^{-\beta H}]$$

Its complexity is largely a mystery…

QMA-hard

At least as hard as the local Hamiltonian problem (compute free energy with $\beta = O(n\epsilon^{-1})$ is enough)

QPF problem: Given H, β and $\delta > 0$, compute an estimate $\widetilde{Z}$ satisfying

$$(1-\delta)\mathrm{Tr}[e^{-\beta H}] \leq \widetilde{Z} \leq (1+\delta)\mathrm{Tr}[e^{-\beta H}]$$

Its complexity is largely a mystery…

QMA-hard

At least as hard as the local Hamiltonian problem (compute free energy with $\beta = O(n\epsilon^{-1})$ is enough)

Not harder than $\#P$

This captures the complexity of exactly computing the partition function.

QPF problem: Given H, β and $\delta > 0$, compute an estimate $\widetilde{Z}$ satisfying

$$(1-\delta)\mathrm{Tr}[e^{-\beta H}] \leq \widetilde{Z} \leq (1+\delta)\mathrm{Tr}[e^{-\beta H}]$$

Its complexity is largely a mystery…

QMA-hard

At least as hard as the local Hamiltonian problem (compute free energy with $\beta = O(n\epsilon^{-1})$ is enough)

Not harder than $\#P$

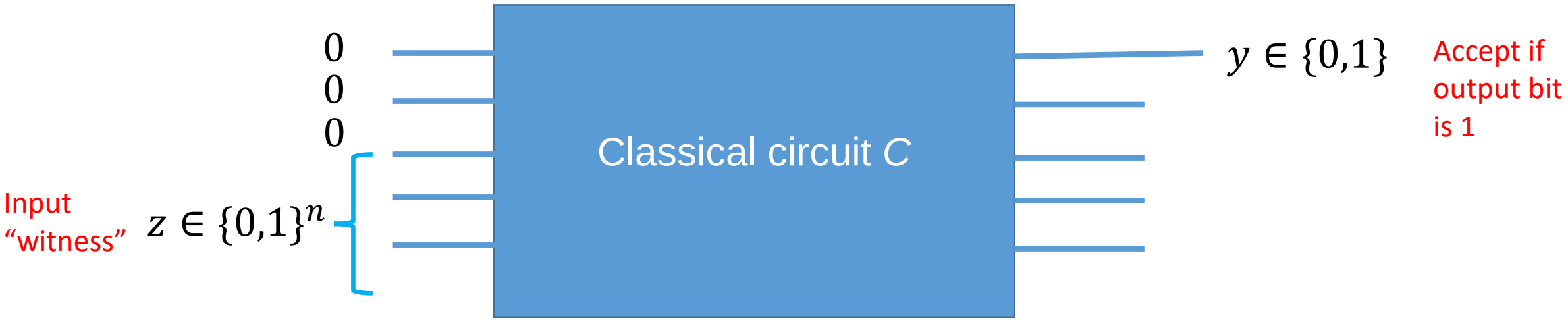
This captures the complexity of exactly computing the partition function.

…which is directly related to our lack of understanding of quantum approximate counting more generally…

Counting : classical, quantum, exact, approximate

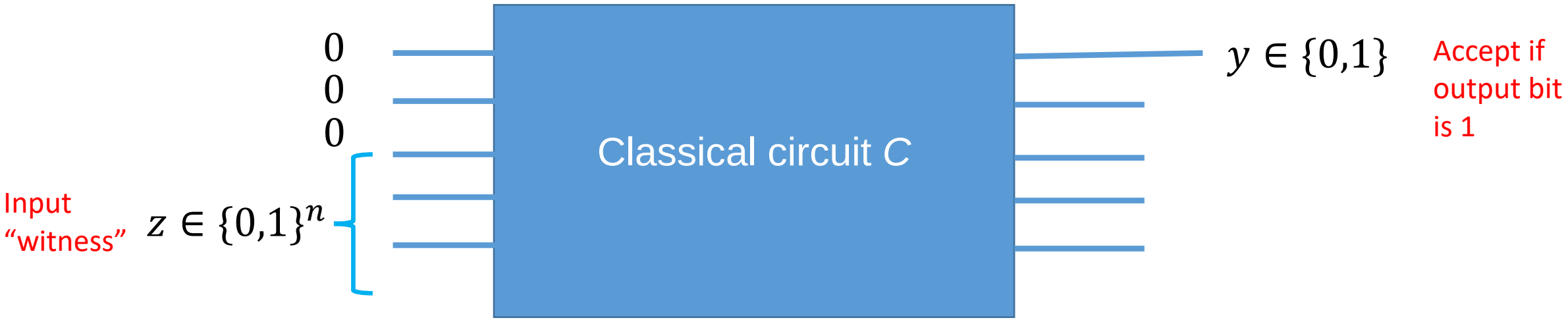
Classical counting problems

Consider a classical (NP) verifier circuit



Classical counting problems

Consider a classical (NP) verifier circuit

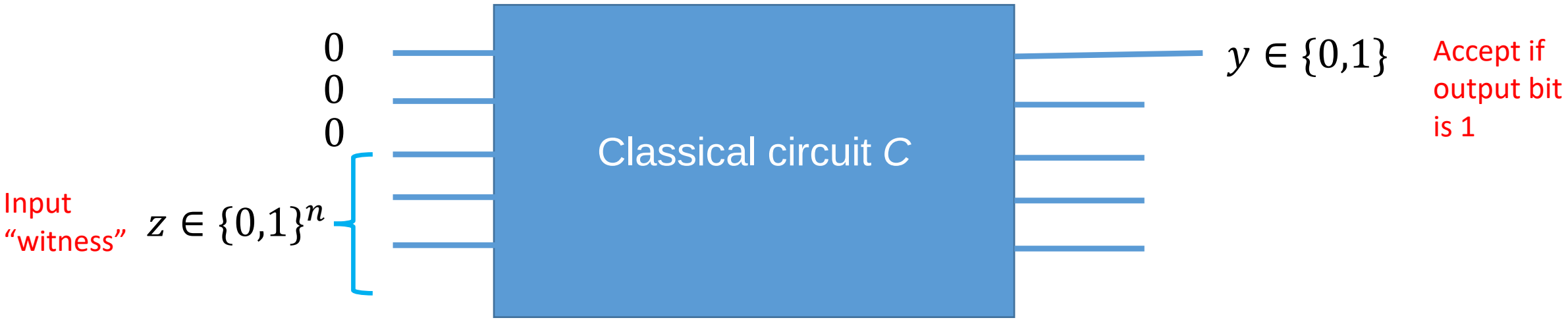


#P is the class of problems (functions) of the form:

Given a verifier circuit, count the number of accepting witnesses.

Classical counting problems

Consider a classical (NP) verifier circuit

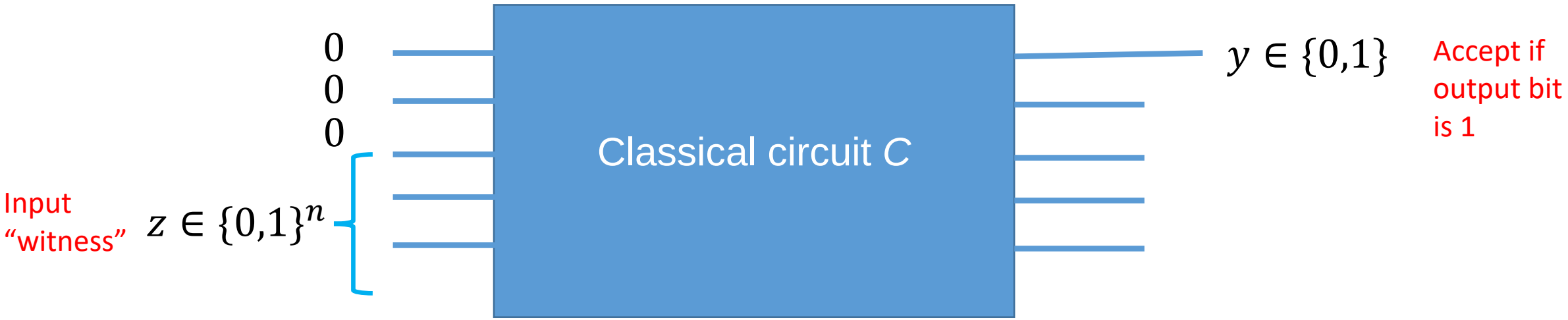


#P is the class of problems (functions) of the form:
Given a verifier circuit, count the number of accepting witnesses.

Can also think of $\#P$ as describing those mathematical quantities which admit a **combinatorial interpretation:** counting a set where membership is efficiently checkable**.**

Classical counting problems

Consider a classical (NP) verifier circuit



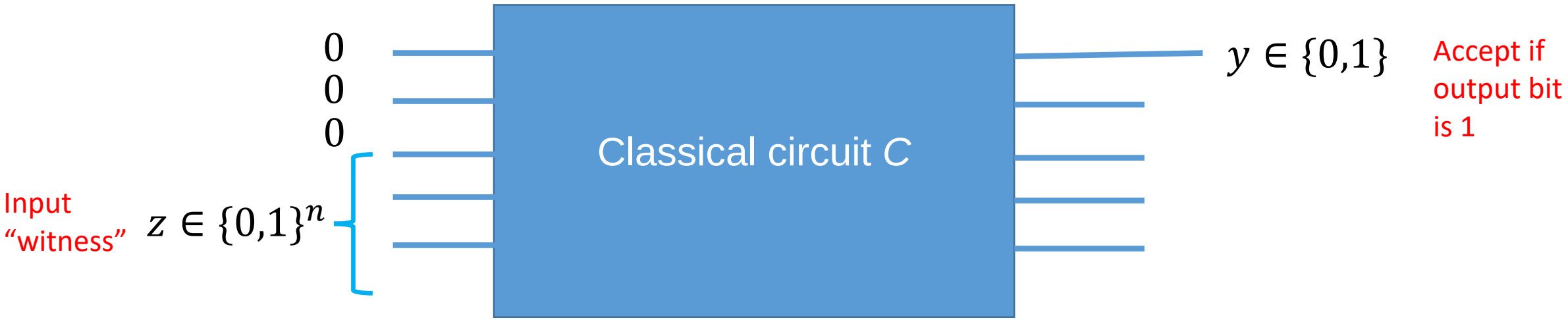
#P is the class of problems (functions) of the form:
Given a verifier circuit, count the number of accepting witnesses.

Examples:

Permanent of a {0,1} matrix (counts perfect matchings)

Classical counting problems

Consider a classical (NP) verifier circuit



#P is the class of problems (functions) of the form:
Given a verifier circuit, count the number of accepting witnesses.

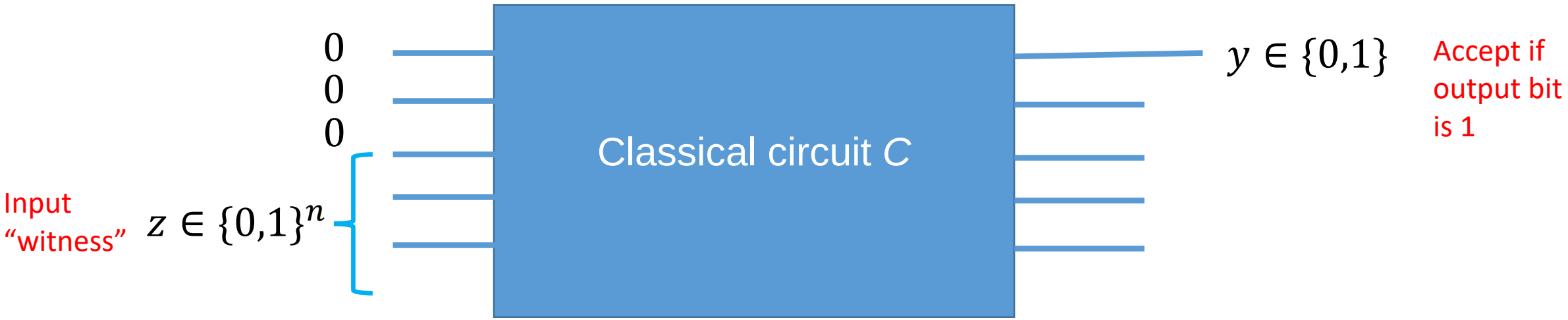
Examples:

Permanent of a {0,1} matrix (counts perfect matchings)

Output probability of a randomized classical circuit (counts computational paths)

Classical counting problems

Consider a classical (NP) verifier circuit

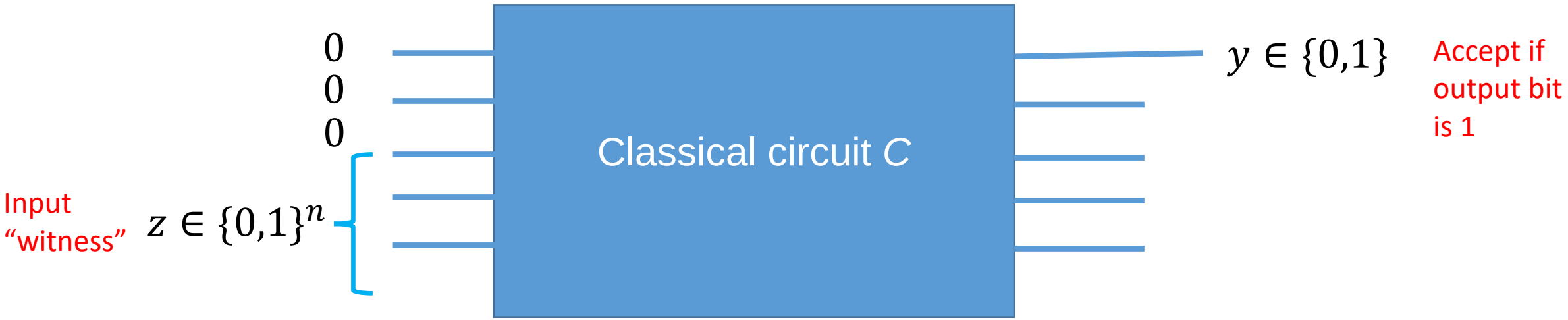


#P is the class of problems (functions) of the form:
Given a verifier circuit, count the number of accepting witnesses.

A related class gapP captures certain "signed" counting problems…

Classical counting problems

Consider a classical (NP) verifier circuit

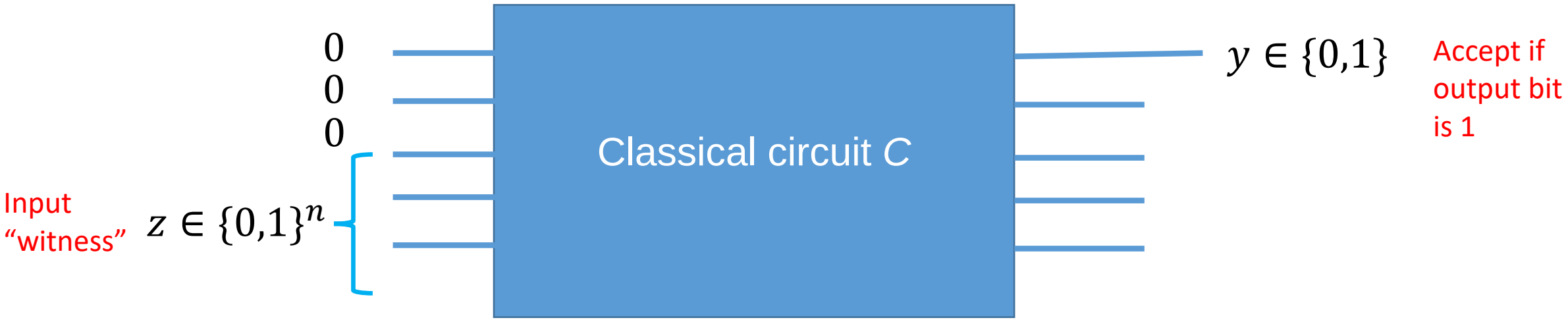


#P is the class of problems (functions) of the form:
Given a verifier circuit, count the number of accepting witnesses.

gapP is the class of functions that can be expressed as the difference of two functions in $\#P$.

Classical counting problems

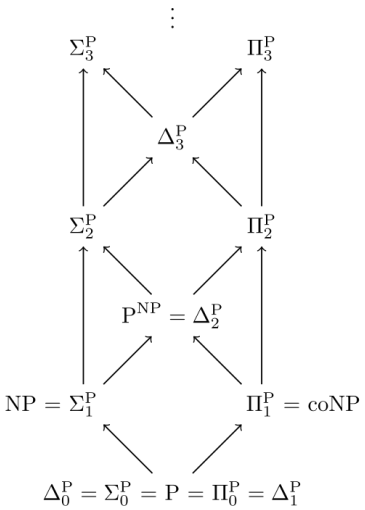
Consider a classical (NP) verifier circuit



#P is the class of problems (functions) of the form:
Given a verifier circuit, count the number of accepting witnesses.
(Example: output probability of randomized classical circuit)

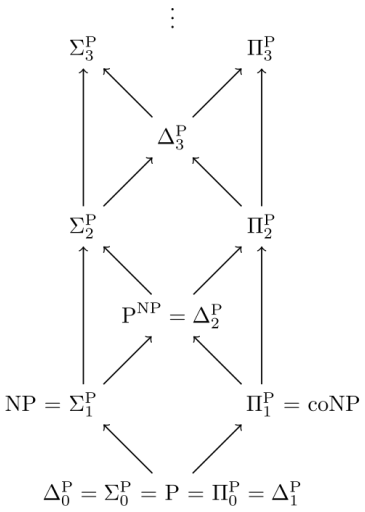
gapP is the class of functions that can be expressed as the difference of two functions in $\#P$. (Example: output probability of quantum circuit)

#P versus gapP



Computing a #P function is just as hard as computing a gapP function.
Both tasks are as powerful as the polynomial hierarchy [Toda's theorem].
So why do we care about the difference between #P and gapP?

#P versus gapP

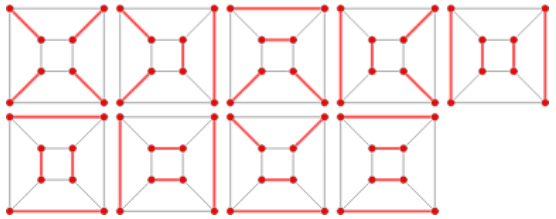


Computing a #P function is just as hard as computing a gapP function. Both tasks are as powerful as the polynomial hierarchy [Toda's theorem].

So why do we care about the difference between #P and gapP?

Math perspective: #P functions have combinatorial interpretations, while gapP functions may not.

$Per(A)$ ⟺

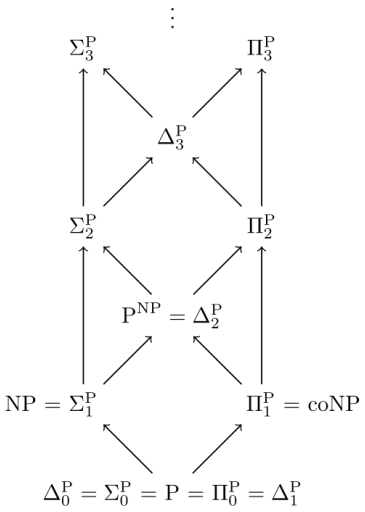


WHAT IS A COMBINATORIAL INTERPRETATION?

IGOR PAK*

ABSTRACT. In this survey we discuss the notion of *combinatorial interpretation* in the context of Algebraic Combinatorics and related areas. We approach the subject from the Computational Complexity perspective. We review many examples, state a workable definition, discuss many open problems, and present recent results on the subject.

#P versus gapP



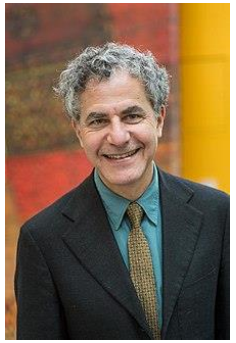
Computing a #P function is just as hard as computing a gapP function. Both tasks are as powerful as the polynomial hierarchy [Toda's theorem].

So why do we care about the difference between #P and gapP?

Computer Science perspective:

There is a vast difference in the complexity of relative-error approximation…

Larry Stockmeyer



Michael Sipser

#P versus gapP

Functions in gapP can be just as hard to approximate as they are to compute exactly.

e.g, output probabilities of quantum circuits are #P-hard to approximate

#P versus gapP

Functions in gapP can be just as hard to approximate as they are to compute exactly.
e.g, output probabilities of quantum circuits are #P-hard to approximate

Functions in #P are vastly easier* to approximate!

***unless PH collapses**

#P versus gapP

Functions in gapP can be just as hard to approximate as they are to compute exactly.
e.g, output probabilities of quantum circuits are #P-hard to approximate

Functions in #P are vastly easier* to approximate!

***unless PH collapses**

Stockmeyer's approximate counting theorem [Stockmeyer 1983]
All functions in #P can be approximated by an efficient randomized algorithm with access to an NP oracle.

#P versus gapP

Functions in gapP can be just as hard to approximate as they are to compute exactly.
e.g, output probabilities of quantum circuits are #P-hard to approximate

Functions in #P are vastly easier* to approximate!

***unless PH collapses**

Stockmeyer's approximate counting theorem [Stockmeyer 1983]
All functions in #P can be approximated by an efficient randomized algorithm with access to an NP oracle.

Proof idea:

$S \subseteq \{0,1\}^n$

Membership can be verified efficiently

Goal: approximate $|S|$

#P versus gapP

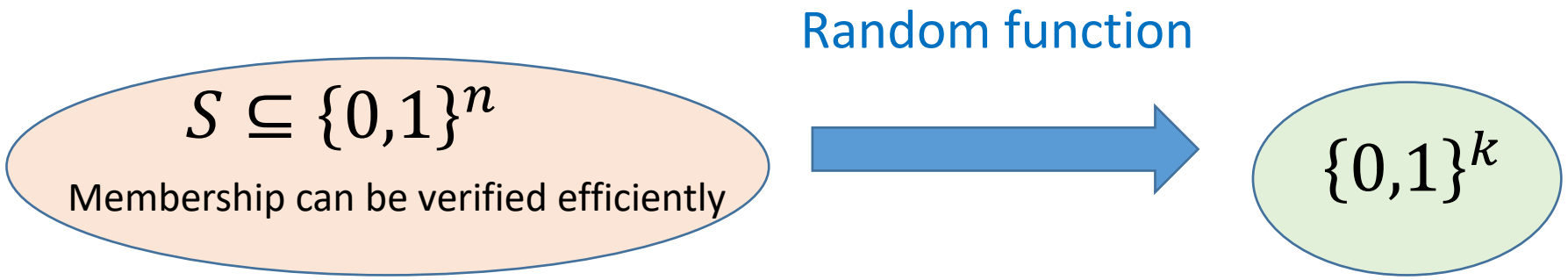
Functions in gapP can be just as hard to approximate as they are to compute exactly.
e.g, output probabilities of quantum circuits are #P-hard to approximate

Functions in #P are vastly easier* to approximate!

***unless PH collapses**

Stockmeyer's approximate counting theorem [Stockmeyer 1983]
All functions in #P can be approximated by an efficient randomized algorithm with access to an NP oracle.

Proof idea:



Goal: approximate $|S|$

#P versus gapP

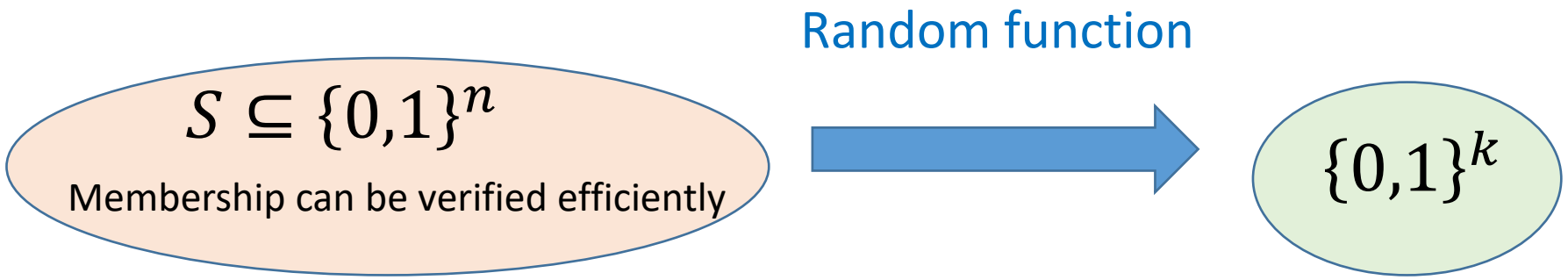
Functions in gapP can be just as hard to approximate as they are to compute exactly.
e.g, output probabilities of quantum circuits are #P-hard to approximate

Functions in #P are vastly easier* to approximate!

***unless PH collapses**

Stockmeyer's approximate counting theorem [Stockmeyer 1983]
All functions in #P can be approximated by an efficient randomized algorithm with access to an NP oracle.

Proof idea:



If $2^k < |S|$ then there is a collision.

Goal: approximate $|S|$

#P versus gapP

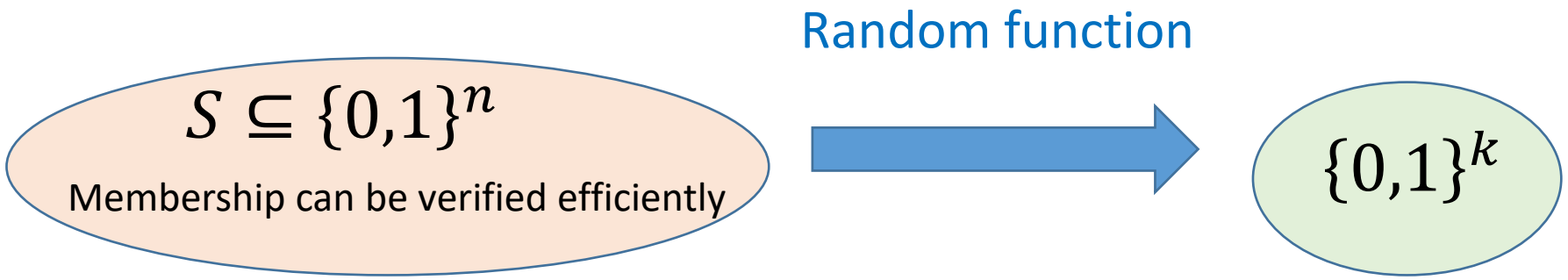
Functions in gapP can be just as hard to approximate as they are to compute exactly.
e.g, output probabilities of quantum circuits are #P-hard to approximate

Functions in #P are vastly easier* to approximate!

***unless PH collapses**

Stockmeyer's approximate counting theorem [Stockmeyer 1983]
All functions in #P can be approximated by an efficient randomized algorithm with access to an NP oracle.

Proof idea:



If $2^k < |S|$ then there is a collision.

Conversely if $2^k \gg |S|$ then there is no collision w.h.p.

.

Goal: approximate $|S|$

#P versus gapP

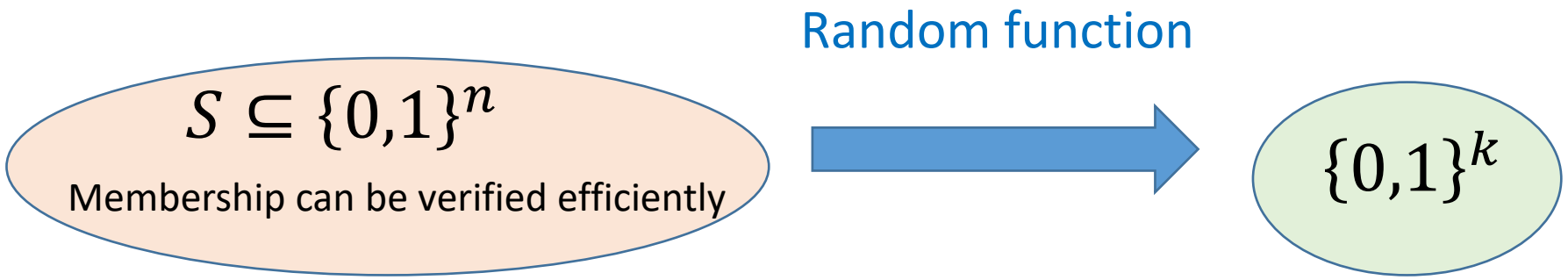
Functions in gapP can be just as hard to approximate as they are to compute exactly.
e.g, output probabilities of quantum circuits are #P-hard to approximate

Functions in #P are vastly easier* to approximate!

***unless PH collapses**

Stockmeyer's approximate counting theorem [Stockmeyer 1983]
All functions in #P can be approximated by an efficient randomized algorithm with access to an NP oracle.

Proof idea:



If $2^k < |S|$ then there is a collision.

Conversely if $2^k \gg |S|$ then there is no collision w.h.p.

You can find a collision by asking the NP oracle.

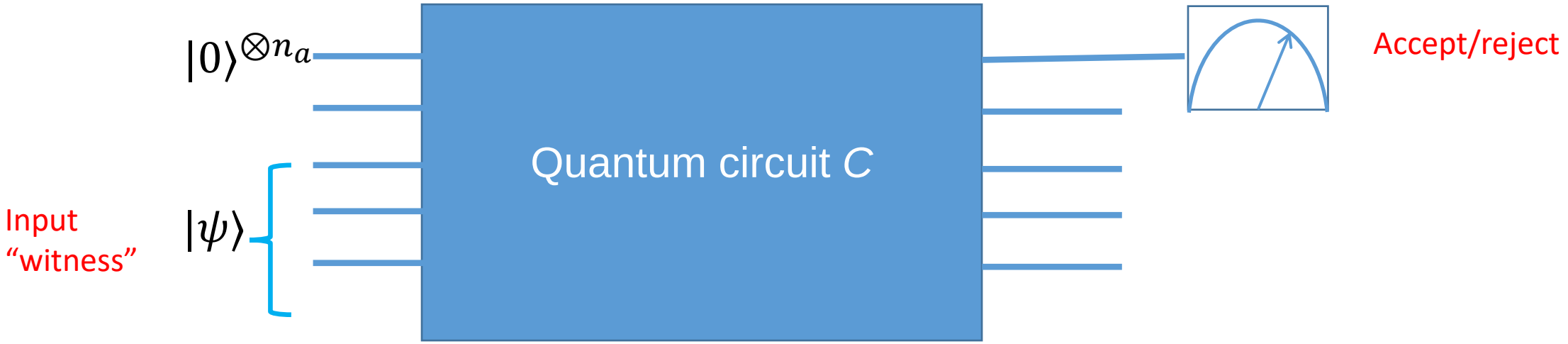
Goal: approximate $|S|$

	Function contained in…	Approximation task upper bound…
Classical counting	$\#P$	$FBPP^{NP}$
Signed counting	gapP	$\#P$-hard

Quantum counting problems

[Brown Flammia Schuch 2010] [Shi Zhang 2009]

Consider a quantum (QMA) verifier circuit

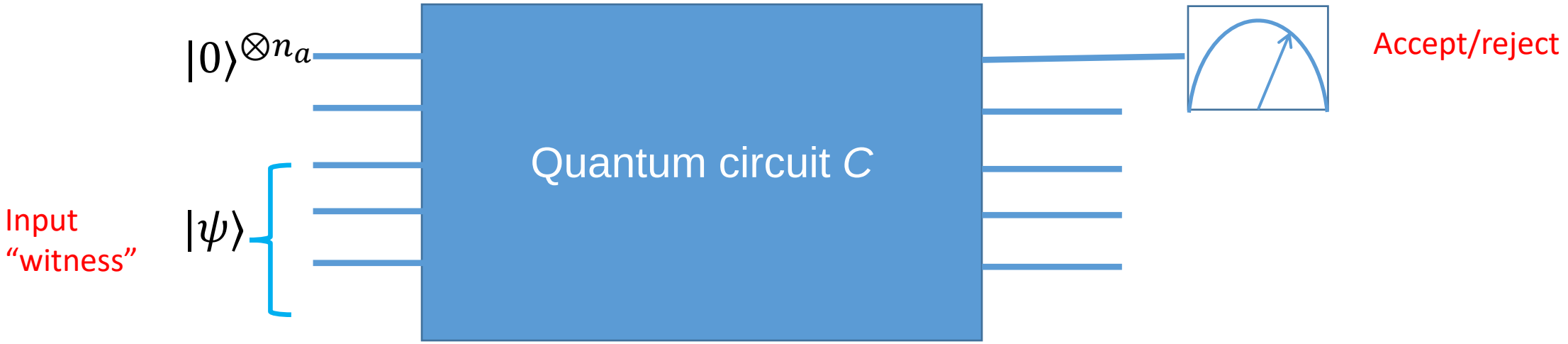


Does it make sense to count the number of accepting witnesses?

Quantum counting problems

[Brown Flammia Schuch 2010] [Shi Zhang 2009]

Consider a quantum (QMA) verifier circuit



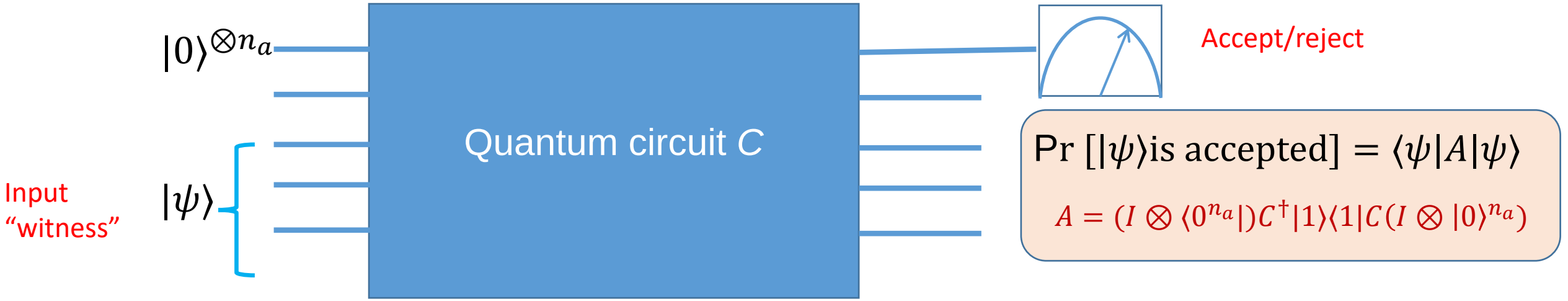
Does it make sense to count the number of accepting witnesses?

Easy special case: If circuit implements a projective measurement. Then we can count the rank of the projector associated with "accept" outcome.

Quantum counting problems

[Brown Flammia Schuch 2010][Shi Zhang 2009]

Consider a quantum (QMA) verifier circuit



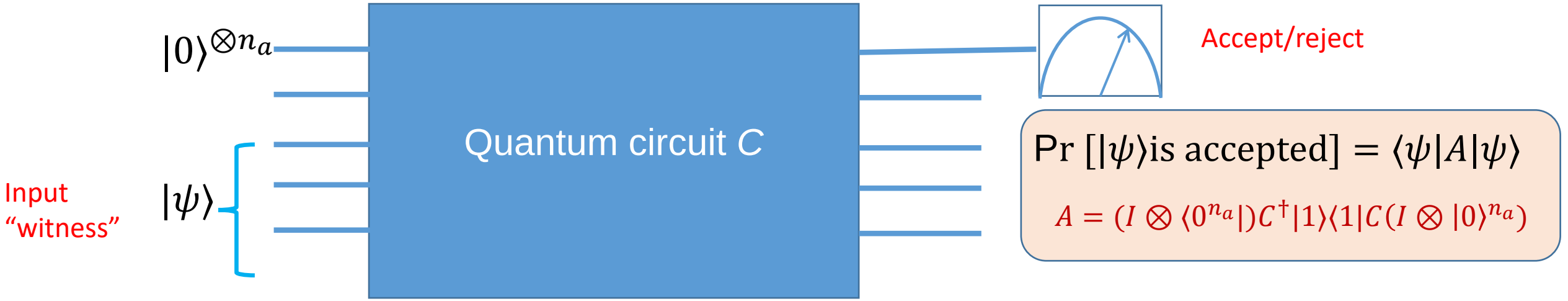
Does it make sense to count the number of accepting witnesses?

More generally: Look at measurement operator *A* corresponding to accept outcome.

Quantum counting problems

[Brown Flammia Schuch 2010][Shi Zhang 2009]

Consider a quantum (QMA) verifier circuit



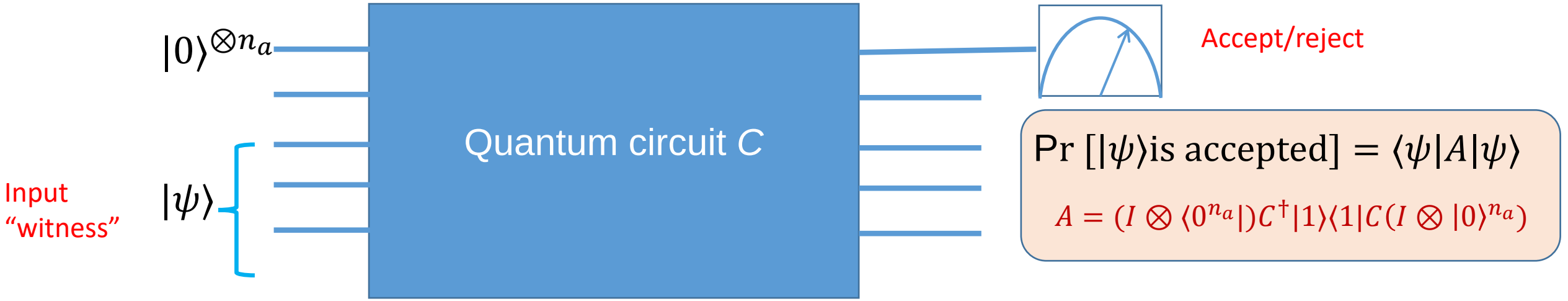
Does it make sense to count the number of accepting witnesses?

More generally: Look at measurement operator *A* corresponding to accept outcome. Count the dimension of the vector space spanned by its eigenvectors with eigenvalues close to 1.

Quantum counting problems

[Brown Flammia Schuch 2010][Shi Zhang 2009]

Consider a quantum (QMA) verifier circuit

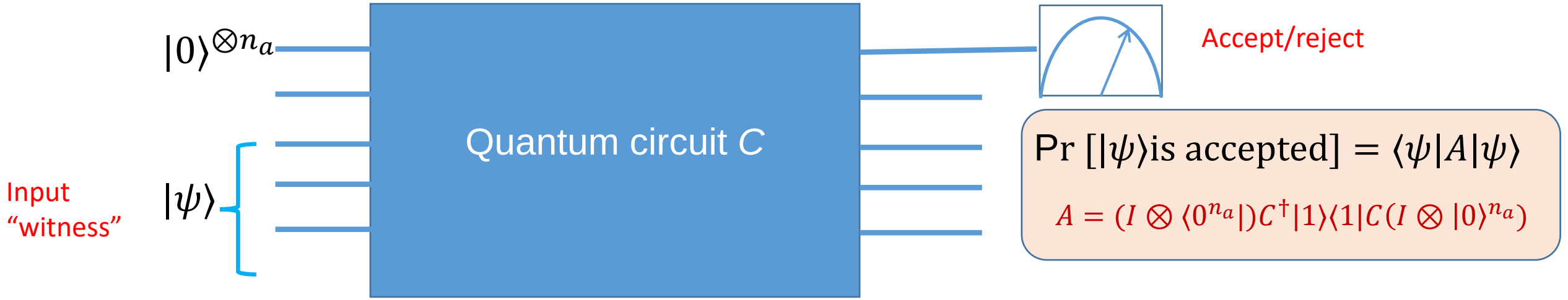


Issue: A witness may be accepted with probability $p \in (0,1)$. How do we decide which ones to count?

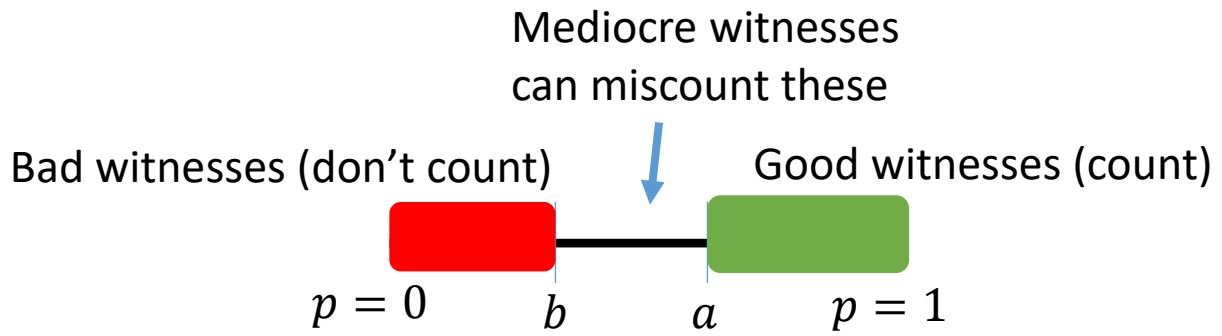
Quantum counting problems

[Brown Flammia Schuch 2010][Shi Zhang 2009]

Consider a quantum (QMA) verifier circuit



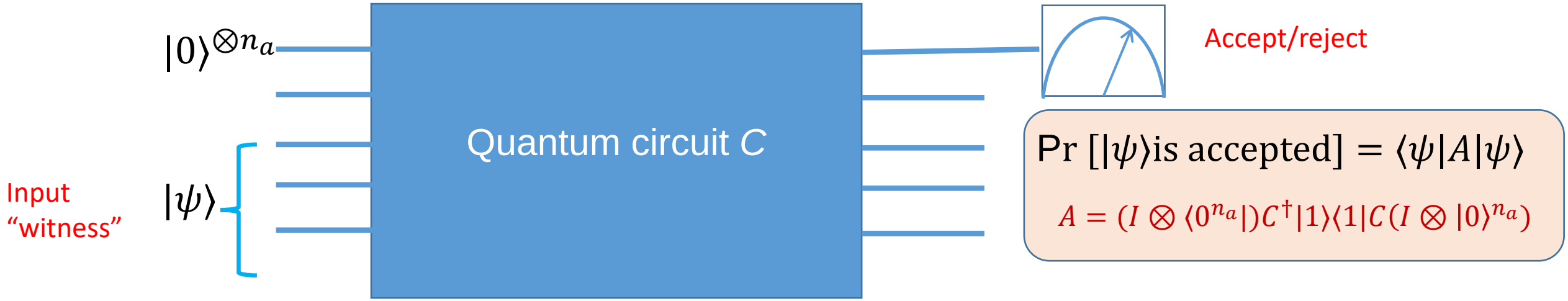
Issue: A witness may be accepted with probability $p \in (0,1)$. How do we decide which ones to count?



Quantum counting problems

[Brown Flammia Schuch 2010][Shi Zhang 2009]

Consider a quantum (QMA) verifier circuit

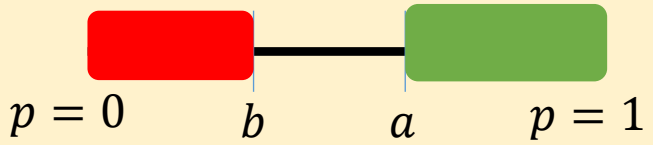


Let N_y be the number of eigenvalues of A with eigenvalue at least y.

#BQP is the class of problems of the form

Input: verifier circuit, two thresholds $0 < b < a \leq 1$

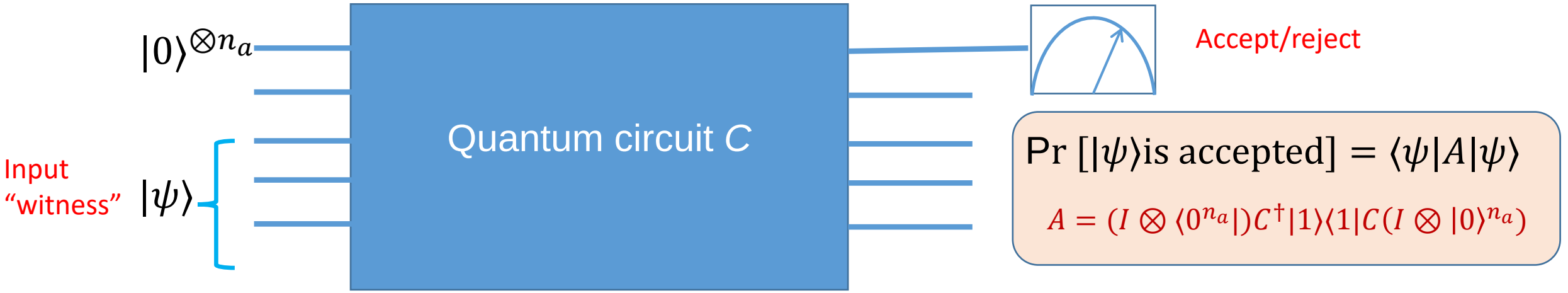
Output: A number M such that $N_a \leq M \leq N_b$



Quantum counting problems

[Brown Flammia Schuch 2010][Shi Zhang 2009]

Consider a quantum (QMA) verifier circuit

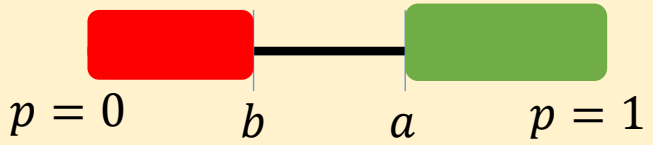


Let N_y be the number of eigenvalues of A with eigenvalue at least y.

#BQP is the class of problems of the form

Input: verifier circuit, two thresholds $0 < b < a \leq 1$

Output: A number M such that $N_a \leq M \leq N_b$



Theorem[Brown Flammia Schuch 2010][Shi Zhang 2009]
#BQP functions are polynomial-time equivalent to #P functions.

Quantum approximate counting

#BQP is the class of problems of the form
Input: verifier circuit, two thresholds $0 < b < \mathrm{a} \leq 1$
Output: A number M such that $N_a \leq M \leq N_b$

QXC is the class of problems of the form
Input: verifier circuit, $\delta \in (0,1)$, two thresholds $0 < b < \mathrm{a} \leq 1$
Output: A number M such that $(1-\delta)N_a \leq M \leq N_b(1+\delta)$

Quantum approximate counting

#BQP is the class of problems of the form
Input: verifier circuit, two thresholds $0 < b < \mathrm{a} \leq 1$
Output: A number M such that $N_a \leq M \leq N_b$

QXC is the class of problems of the form
Input: verifier circuit, $\delta \in (0,1)$, two thresholds $0 < b < \mathrm{a} \leq 1$
Output: A number M such that $(1-\delta)N_a \leq M \leq N_b(1+\delta)$

Possibly controversial statement: QXC is the reason to care about #BQP!

Quantum approximate counting

#BQP is the class of problems of the form
Input: verifier circuit, two thresholds $0 < b < \mathrm{a} \leq 1$
Output: A number M such that $N_a \leq M \leq N_b$

QXC is the class of problems of the form
Input: verifier circuit, $\delta \in (0,1)$, two thresholds $0 < b < \mathrm{a} \leq 1$
Output: A number M such that $(1-\delta)N_a \leq M \leq N_b(1+\delta)$

Possibly controversial statement: QXC is the reason to care about #BQP!

There is no known quantum version of Stockmeyer approximate counting and so complexity of QXC is poorly understood

(cf. lack of quantum Valiant Vazirani theorem [Aharonov Ben-Or Brandao Sattath 2008])

	Function contained in…	Approximation task upper bound…
Classical counting	$\#P$	$FBPP^{NP}$
Quantum counting	$\#BQP$	QXC
Signed counting	gapP	$\#P$-hard

	Function contained in…	Approximation task upper bound…
Classical counting	$\#P$	$FBPP^{NP}$
Quantum counting	$\#BQP$	QXC
Signed counting	gapP	$\#P$-hard

Theorem [Bravyi Chowdhury DG Wocjan 2022]
QPF is QXC-complete.

	Function contained in…	Approximation task upper bound…
Classical counting	$\#P$	$FBPP^{NP}$
Quantum counting	$\#BQP$	QXC
Signed counting	gapP	$\#P$-hard

Theorem [Bravyi Chowdhury DG Wocjan 2022]
QPF is QXC-complete.

Proof ideas (to reduce QXC to 2-local QPF):
Marriott-Watrous error reduction for QMA [Marriott Watrous 2005]
2-local Circuit-to-Hamiltonian mapping [Kempe Kitaev Regev 2004]
Number of QMA witnesses ~number of low energy states

	Exact	Approximate
$Z = \mathrm{Tr}[e^{-\beta H}]$ Classical local Hamiltonian	$\#P$-hard	$FBPP^{NP}$
$Z = \mathrm{Tr}[e^{-\beta H}]$ Quantum local Hamiltonian	$\#P$-hard	QXC
$Z = \mathrm{Tr}[e^{-i\beta H}]$ Local Hamiltonian, imaginary temperature	$\#P$-hard	$\#P$-hard

	Exact	Approximate
$Z = \mathrm{Tr}[e^{-\beta H}]$ Classical local Hamiltonian	$\#P$-hard	$FBPP^{NP}$
$Z = \mathrm{Tr}[e^{-\beta H}]$ Quantum local Hamiltonian	$\#P$-hard	QXC
$Z = \mathrm{Tr}[e^{-i\beta H}]$ Local Hamiltonian, imaginary temperature	$\#P$-hard	$\#P$-hard

Challenging question: how does QXC relate to other known complexity classes?

	Exact	Approximate
$Z = \mathrm{Tr}[e^{-\beta H}]$ Classical local Hamiltonian	$\#P$-hard	$FBPP^{NP}$
$Z = \mathrm{Tr}[e^{-\beta H}]$ Quantum local Hamiltonian	$\#P$-hard	QXC
$Z = \mathrm{Tr}[e^{-i\beta H}]$ Local Hamiltonian, imaginary temperature	$\#P$-hard	$\#P$-hard

Challenging question: how does QXC relate to other known complexity classes?

Easier: **can we find other interesting problems in QXC?...**

Kronecker coefficients

Kronecker coefficients = Clebsch-Gordan coefficients for the symmetric group

Recall that irreps of S_n are labeled by partitions of n or equivalently Young diagrams. For any three partitions $\mu, \nu, \lambda \vdash n$, there is a Kronecker coefficient $g_{\mu\nu\lambda}$ defined as:

$$\rho_\mu \otimes \rho_\nu \simeq \bigoplus_\lambda g_{\mu\nu\lambda} \rho_\lambda$$

Tensor product of two irreps

Multiplicities

Kronecker coefficients

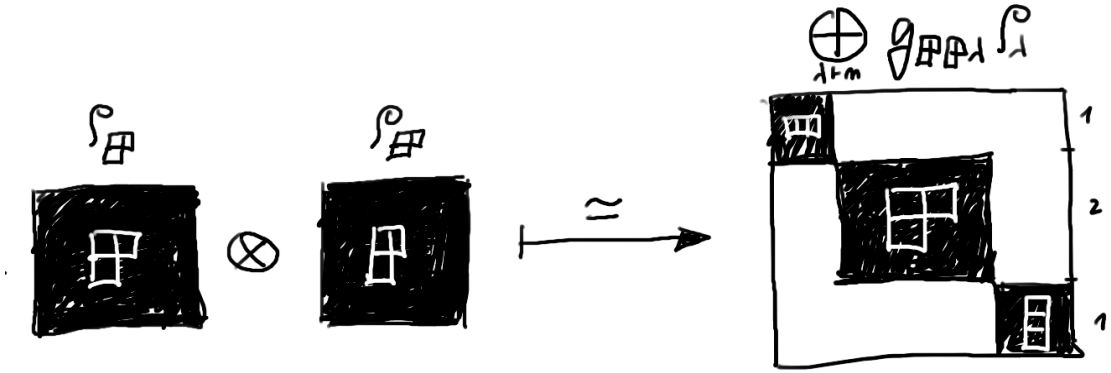
Kronecker coefficients = Clebsch-Gordan coefficients for the symmetric group

Recall that irreps of S_n are labeled by partitions of n or equivalently Young diagrams. For any three partitions $\mu, \nu, \lambda \vdash n$, there is a Kronecker coefficient $g_{\mu\nu\lambda}$ defined as:

$$\rho_\mu \otimes \rho_\nu \simeq \bigoplus_\lambda g_{\mu\nu\lambda} \rho_\lambda$$

Tensor product of two irreps

Multiplicities



Kronecker coefficients

Kronecker coefficients $=$ Clebsch-Gordan coefficients for the symmetric group

Recall that irreps of S_n are labeled by partitions of n or equivalently Young diagrams. For any three partitions $\mu, \nu, \lambda \vdash n$, there is a Kronecker coefficient $g_{\mu\nu\lambda}$ defined as:

$$\rho_\mu \otimes \rho_\nu \simeq \bigoplus_\lambda g_{\mu\nu\lambda} \rho_\lambda$$

Tensor product of two irreps

Multiplicities

Old question: Do the Kronecker coefficients have a combinatorial interpretation?
[Murnaghan 1938] [Stanley 1999]

Kronecker coefficients

Kronecker coefficients = Clebsch-Gordan coefficients for the symmetric group

Recall that irreps of S_n are labeled by partitions of n or equivalently Young diagrams. For any three partitions $\mu, \nu, \lambda \vdash n$, there is a Kronecker coefficient $g_{\mu\nu\lambda}$ defined as:

$$\rho_\mu \otimes \rho_\nu \simeq \bigoplus_\lambda g_{\mu\nu\lambda} \rho_\lambda$$

Tensor product of two irreps

Multiplicities

Old question: Do the Kronecker coefficients have a combinatorial interpretation?
[Murnaghan 1938] [Stanley 1999]

Modern rephrasing: Do the Kronecker coefficients admit a $\#P$ formula?

e.g. [Burgisser Ikenmeyer 2008]
[Pak Panova 2014]
[Ikenmeyer Mulmuley Walter 2017]…

Kronecker coefficients

Kronecker coefficients $=$ Clebsch-Gordan coefficients for the symmetric group

Recall that irreps of S_n are labeled by partitions of n or equivalently Young diagrams. For any three partitions $\mu, \nu, \lambda \vdash n$, there is a Kronecker coefficient $g_{\mu\nu\lambda}$ defined as:

$$\rho_\mu \otimes \rho_\nu \simeq \bigoplus_\lambda g_{\mu\nu\lambda} \rho_\lambda$$

Tensor product of two irreps

Multiplicities

Old question: Do the Kronecker coefficients have a combinatorial interpretation?
[Murnaghan 1938] [Stanley 1999]

Modern rephrasing: Do the Kronecker coefficients admit a $\#P$ formula?

e.g. [Burgisser Ikenmeyer 2008] They admit a gapP formula
[Pak Panova 2014]
[Ikenmeyer Mulmuley Walter 2017] They are #P-hard to compute

Kronecker coefficients are a quantum counting problem

Claim [Bravyi, Anirban Chowdhury, DG, Vojtech Havlicek, Guanyu Zhu 2022]

For any $\mu, \nu, \lambda \vdash n$, there is a quantum circuit of size $poly(n)$ that measures a projector with rank

$$d_\mu d_\nu d_\lambda g_{\mu\nu\lambda}$$

Efficiently computable prefactor

Kronecker coefficients are a quantum counting problem

Claim [Bravyi, Anirban Chowdhury, DG, Vojtech Havlicek, Guanyu Zhu 2022]

For any $\mu, \nu, \lambda \vdash n$, there is a quantum circuit of size $poly(n)$ that measures a projector with rank

$$d_\mu d_\nu d_\lambda g_{\mu\nu\lambda}$$

Efficiently computable prefactor

In other words: $g_{\mu\nu\lambda}$ is proportional to a #BQP function

Also in unpub. work by:
[Christandl, Harrow, Walter]
[Kuperberg, Pak, Panova]

Kronecker coefficients are a quantum counting problem

Claim [Bravyi, Anirban Chowdhury, DG, Vojtech Havlicek, Guanyu Zhu 2022]

For any $\mu, \nu, \lambda \vdash n$, there is a quantum circuit of size $poly(n)$ that measures a projector with rank

$$d_\mu d_\nu d_\lambda g_{\mu\nu\lambda}$$

Efficiently computable prefactor

In other words: $g_{\mu\nu\lambda}$ is proportional to a #BQP function

Also in unpub. work by:
[Christandl, Harrow, Walter]
[Kuperberg, Pak, Panova]

Corollary

The problem of approximating $g_{\mu\nu\lambda}$ to within a given relative error is in QXC.

Kronecker coefficients are a quantum counting problem

Claim [Bravyi, Anirban Chowdhury, DG, Vojtech Havlicek, Guanyu Zhu 2022]

For any $\mu, \nu, \lambda \vdash n$, there is a quantum circuit of size $poly(n)$ that measures a projector with rank

$$d_\mu d_\nu d_\lambda g_{\mu\nu\lambda}$$

Efficiently computable prefactor

Proof sketch

For any representation ρ of S_n we can define a projective quantum measurement with outcomes labeled by partitions $\lambda \vdash n$:

$$\Pi_\lambda = \frac{d_\lambda}{n!} \sum_{g \in S_n} \chi_\lambda(g) \rho(g)$$

Projector for outcome λ

Kronecker coefficients are a quantum counting problem

Claim [Bravyi, Anirban Chowdhury, DG, Vojtech Havlicek, Guanyu Zhu 2022]

For any $\mu, \nu, \lambda \vdash n$, there is a quantum circuit of size $poly(n)$ that measures a projector with rank

$$d_\mu d_\nu d_\lambda g_{\mu\nu\lambda}$$

Efficiently computable prefactor

Proof sketch

For any representation ρ of S_n we can define a projective quantum measurement with outcomes labeled by partitions $\lambda \vdash n$:

$$\Pi_\lambda = \frac{d_\lambda}{n!} \sum_{g \in S_n} \chi_\lambda(g) \rho(g)$$

Projector for outcome λ

This projective measurement can be implemented in time $poly(n)$ on a quantum computer using Beals' quantum Fourier transform over the symmetric group.

[Beals 1997] via generalized phase estimation [Harrow 2005]

Kronecker coefficients are a quantum counting problem

Claim [Bravyi, Anirban Chowdhury, DG, Vojtech Havlicek, Guanyu Zhu 2022]

For any $\mu, \nu, \lambda \vdash n$, there is a quantum circuit of size $poly(n)$ that measures a projector with rank

$$d_\mu d_\nu d_\lambda g_{\mu\nu\lambda}$$

Efficiently computable prefactor

Proof sketch

It is not hard to show that

$$d_\mu d_\nu d_\lambda g_{\mu\nu\lambda} = Tr(\Pi_\mu \otimes \Pi_\nu \otimes \Pi_\lambda P)$$

Defined as on previous slide
with $\rho =$ left regular representation

$$P = \frac{1}{n!} \sum_{g \in S_n} g \otimes g \otimes g$$

This projector can also be measured efficiently on a quantum computer

Kronecker coefficients are a quantum counting problem

Claim [Bravyi, Anirban Chowdhury, DG, Vojtech Havlicek, Guanyu Zhu 2022]

For any $\mu, \nu, \lambda \vdash n$, there is a quantum circuit of size $poly(n)$ that measures a projector with rank

$$d_\mu d_\nu d_\lambda g_{\mu\nu\lambda}$$

Efficiently computable prefactor

Proof sketch

It is not hard to show that

$$d_\mu d_\nu d_\lambda g_{\mu\nu\lambda} = Tr(\Pi_\mu \otimes \Pi_\nu \otimes \Pi_\lambda P)$$

$$P = \frac{1}{n!} \sum_{g \in S_n} g \otimes g \otimes g$$

The quantum circuit first measures $\Pi_\mu \otimes \Pi_\nu \otimes \Pi_\lambda$ and then measures P, and accepts iff both outcomes are +1.

Row sums of the character table

[Bravyi, Anirban Chowdhury, DG, Vojtech Havlicek, Guanyu Zhu 2022]

There is a similar story for the row sums of the character table of the symmetric group, for which a combinatorial representation is also an open question [Stanley 1999]

$$R_\lambda = \sum_{\mu \vdash n} \chi_\lambda(\mu)$$

These can also be expressed as multiplicities and can be shown to be proportional to a #BQP function, and therefore approximable in QXC.

Summary so far

The computational complexity of relative-error approximation of quantum partition functions or equivalently quantum approximate counting is a fascinating mystery.

Some mathematical quantities that are not known to admit combinatorial interpretations may in fact be quantum counting problems. In that case the related approximation tasks are no harder than approximating the partition function of 2-local Hamiltonians.

Can we find efficient algorithms for QPF in special cases?

Mean field theory via convex optimization (but worse approximation guarantee)
Dense Hamiltonians, ground energy problems [Brandao Harrow 2013]
Dense classical Ising partition functions [Risteski 2016]
Dense quantum partition functions [Sergey Bravyi, Anirban Chowdhury, DG, Pawel Wocjan 2022]

Rigorous Monte Carlo methods
Classical ferromagnetic Ising model [Jerrum Sinclair 1993]
Quantum transverse Ising ferromagnet [Bravyi 2014]
Broader class of quantum ferromagnets (e.g. XY model) [Bravyi DG 2017]

High temperature expansion
General local Hamiltonians above a critical temperature [Harrow Mehraban Soleimanifar 2020]

Can we find efficient algorithms for QPF in special cases?

Mean field theory via convex optimization (but worse approximation guarantee)

Dense Hamiltonians, ground energy problems [Brandao Harrow 2013]

Dense classical Ising partition functions [Risteski 2016]

Dense quantum partition functions [Sergey Bravyi, Anirban Chowdhury, DG, Pawel Wocjan 2022]

Rigorous Monte Carlo methods

Classical ferromagnetic Ising model [Jerrum Sinclair 1993]

Quantum transverse Ising ferromagnet [Bravyi 2014]

Broader class of quantum ferromagnets (e.g. XY model) [Bravyi DG 2017]

High temperature expansion

General local Hamiltonians above a critical temperature [Harrow Mehraban Soleimanifar 2020]

Quantum Monte Carlo: a powerful suite of probabilistic classical simulation algorithms for quantum many-body systems.

Can simulate systems orders of magnitude larger than with exact diagonalization…

What's the catch?

Quantum Monte Carlo can only be used to study **stoquastic Hamiltonians**

$$\langle x|H|x\rangle \in \mathbb{R} \qquad \langle y|H|x\rangle \leq 0 \qquad x \neq y$$

Stoquastic
i.e., "sign-problem free"

Examples:

What's the catch?

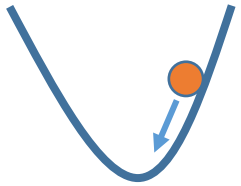
Quantum Monte Carlo can only be used to study **stoquastic Hamiltonians**

$$\langle x|H|x\rangle \in \mathbb{R} \qquad \langle y|H|x\rangle \leq 0 \qquad x \neq y$$

Stoquastic
i.e., "sign-problem free"

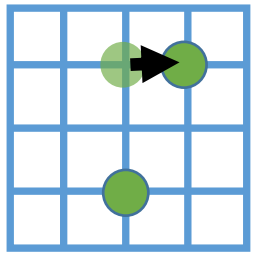
Examples:

Particle in a potential



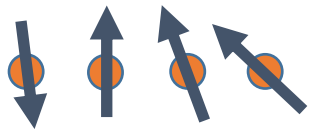
$$H = \frac{p^2}{2m} + V(\vec{x})$$

Hopping and interacting bosons



$$H = -\sum_{<ij>} (a_i^\dagger a_j + a_j^\dagger a_i) + V(\vec{n})$$

Quantum annealing Hamiltonians



$$H = -(1-s)\sum_i X_i + sV\left(\vec{Z}\right)$$

How does it work?

Quantum Monte Carlo is based on a probabilistic representation of the Gibbs state

$$\rho = \frac{e^{-\beta H}}{Z(\beta)} \qquad Z(\beta) = \mathrm{Tr}(e^{-\beta H})$$

A collection of samples from a certain probability distribution associated with ρ are sufficient to evaluate expectation values of observables.

How does it work?

Quantum Monte Carlo is based on a probabilistic representation of the Gibbs state

$$\rho = \frac{e^{-\beta H}}{Z(\beta)} \qquad Z(\beta) = \mathrm{Tr}(e^{-\beta H})$$

A collection of samples from a certain probability distribution associated with ρ are sufficient to evaluate expectation values of observables.

Clever choice of probabilistic representation and sampling method is key.

Can we identify cases where Quantum Monte Carlo is provably efficient?

$$H = \sum_{1 \le i < j \le n} h(i,j)$$

		Approximate Ground energy	Approximate Partition Function
Ferromagnetic Ising model	$h(i,j) = -\lvert\alpha_{ij}\rvert Z_i Z_j$	Trivial	In BPP [Jerrum ,Sinclair 1989]
Ferromagnetic Transverse-field Ising model	$h(i,j) = -\lvert\alpha_{ij}\rvert X_i X_j - \gamma_i Z_i - \gamma_j Z_j$	In BPP [Bravyi 2015]	In BPP [Bravyi 2015]
Ferromagnetic XY model	$h(i,j) = -\lvert\alpha_{ij}\rvert (X_i X_j + Y_i Y_j)$	In BPP [Bravyi DG 2017]	In BPP [Bravyi DG 2017]

The Hamiltonians we consider

$$H = \sum_{i<j} -b_{ij} X_i X_j + c_{ij} Y_i Y_j + \sum_{i=1}^{n} d_i (I + Z_i)$$

Coefficients must satisfy

$$|b_{ij}|, |c_{ij}|, |d_i| \leq 1 \quad \text{(sets energy scale)}$$

$$|c_{ij}| \leq b_{ij} \quad \text{(ensures stoquasticity)}$$

The Hamiltonians we consider

$$H = \sum_{i<j} \underbrace{-b_{ij} X_i X_j + c_{ij} Y_i Y_j} + \sum_{i=1}^{n} d_i (I + Z_i)$$

$$\begin{pmatrix} 0 & 0 & 0 & -b_{ij} - c_{ij} \\ 0 & 0 & c_{ij} - b_{ij} & 0 \\ 0 & c_{ij} - b_{ij} & 0 & 0 \\ -b_{ij} - c_{ij} & 0 & 0 & 0 \end{pmatrix}$$

$$|c_{ij}| \leq b_{ij}$$ (ensures stoquasticity)

The Hamiltonians we consider

$$H = \sum_{i<j} \underbrace{-b_{ij} X_i X_j + c_{ij} Y_i Y_j} + \sum_{i=1}^{n} d_i (I + Z_i)$$

$$p_{ij}\left(Y_i Y_j - X_i X_j\right) + q_{ij}(-Y_i Y_j - X_i X_j) \qquad p_{ij}, q_{ij} \geq 0$$

The Hamiltonians we consider

$$H = \sum_{i<j} -b_{ij} X_i X_j + c_{ij} Y_i Y_j + \sum_{i=1}^{n} d_i (I + Z_i)$$

Special cases:

$d_i = 0$	$c_{ij} = 0$	Classical Ferromagnetic Ising model
	$c_{ij} = 0$	Ferromagnetic transverse-field Ising model
$b_{ij} = 1$	$c_{ij} = -1$	Ferromagnetic XY model
$b_{ij} = 1$	$c_{ij} = 1$	(name?)

Efficient algorithm for the partition function

Theorem [Bravyi DG 2017]
There exists a classical randomized algorithm which, given H, β, and a precision parameter $\epsilon \in (0,1)$ outputs an estimate satisfying $Z \approx^{\epsilon} Z(\beta)$ with high probability.

The runtime of the algorithm is $poly(n, \beta, \epsilon^{-1})$

Efficient algorithm for the partition function

Theorem [Bravyi DG 2017]
There exists a classical randomized algorithm which, given H, β, and a precision parameter $\epsilon \in (0,1)$ outputs an estimate satisfying $Z \approx^{\epsilon} Z(\beta)$ with high probability.

The runtime of the algorithm is $poly(n, \beta, \epsilon^{-1})$

As a corollary we obtain an efficient algorithm to approximate the free energy, the ground energy, and thermal expected values to a given additive error.

Efficient algorithm for the partition function

Theorem [Bravyi DG 2017]
There exists a classical randomized algorithm which, given H, β, and a precision parameter $\epsilon \in (0,1)$ outputs an estimate satisfying $Z \approx^{\epsilon} Z(\beta)$ with high probability.

The runtime of the algorithm is $poly(n, \beta, \epsilon^{-1}) = \tilde{O}\left(n^{115}(1+\beta^{46})\epsilon^{-25}\right)$.

As a corollary we obtain an efficient algorithm to approximate the free energy, the ground energy, and thermal expected values to a given additive error.

The algorithm is not practical.

Efficient algorithm for the partition function

Theorem [Bravyi DG 2017]
There exists a classical randomized algorithm which, given H, β, and a precision parameter $\epsilon \in (0,1)$ outputs an estimate satisfying $Z \approx^{\epsilon} Z(\beta)$ with high probability.

The runtime of the algorithm is $poly(n, \beta, \epsilon^{-1}) = \tilde{O}\left(n^{115}(1+\beta^{46})\epsilon^{-25}\right)$.

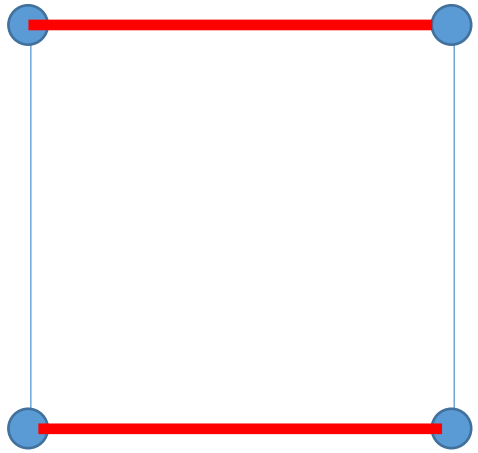
As a corollary we obtain an efficient algorithm to approximate the free energy, the ground energy, and thermal expected values to a given additive error.

The algorithm is not practical.

The proof is based on a reduction to counting perfect matchings…

A **perfect matching** of a graph $G = (V, E)$ is a subset of edges $M \subseteq E$ such that every vertex is incident to exactly one edge in M

Example:



Now suppose the graph has edge weights $\{w_e\}_{e \in E}$. Each perfect matching M is assigned weight

$$\prod_{e \in M} w_e$$

Perfect matching sum:

$$\text{PerfMatch}(G) = \sum_{\text{Perfect matchings M}} \prod_{e \in M} w_e$$

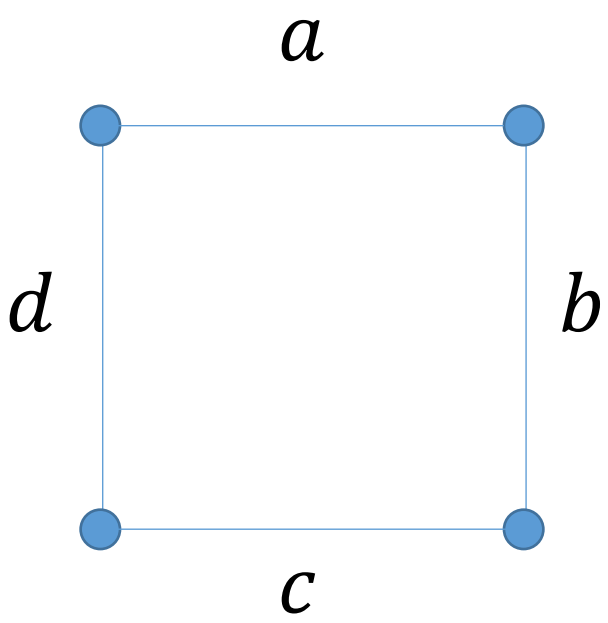
Now suppose the graph has edge weights $\{w_e\}_{e \in E}$. Each perfect matching M is assigned weight

$$\prod_{e \in M} w_e$$

Perfect matching sum:

$$\text{PerfMatch}(G) = \sum_{\text{Perfect matchings M}} \prod_{e \in M} w_e$$

Example:



$$\text{PerfMatch}(G) = ac + bd$$

A **nearly perfect matching** of a graph $G = (V, E)$ is a subset of edges $M \subseteq E$ such that every vertex is incident to exactly one edge in M, **except for 2 vertices which are untouched.**

Nearly perfect matching sum:

$$\text{NearPerfMatch}(G) = \sum_{\substack{\text{Nearly} \\ \text{Perfect matchings M}}} \prod_{e \in M} w_e$$

Suppose G is a graph with nonnegative edge weights.

	Exactly compute $\mathrm{PerfMatch}(G)$	**ϵ-approximation to** $\mathrm{PerfMatch}(G)$
Planar graphs:	**In P** Fisher, Kasteleyn, Temperley algorithm	**In P**
Bipartite graphs: (permanent of nonnegative matrix)	**#*P*-hard** [Valiant 1979]	**In BPP** [Jerrum, Sinclair, Vigoda 2004]
General graphs:	**#*P*-hard**	[Jerrum Sinclair 1989] **Algorithm with runtime** $\mathrm{poly}(\lvert V\rvert, \epsilon,^{-1} R)$ $R = \frac{\mathrm{NearPerfMatch}(G)}{\mathrm{PerfMatch}(G)}$

These partition functions reduce to perfect matchings

Theorem [Bravyi DG 2017]
There is an (efficiently computable) graph G with positive edge weights, such that

$$Z(\beta) \approx^{\epsilon} \mathrm{PerfMatch}(G)$$

and

$$\frac{\mathrm{NearPerfMatch}(G)}{\mathrm{PerfMatch}(G)} = O(\mathrm{poly}(\beta, n, \epsilon^{-1}))$$

We then use [Jerrum, Sinclair 1989] which gives an efficient algorithm for approximating the perfect matching sum.

These partition functions reduce to perfect matchings

Theorem [Bravyi DG 2017]
There is an (efficiently computable) graph G with positive edge weights, such that

$$Z(\beta) \approx^{\epsilon} \mathrm{PerfMatch}(G)$$

and

$$\frac{\mathrm{NearPerfMatch}(G)}{\mathrm{PerfMatch}(G)} = O(\mathrm{poly}(\beta, n, \epsilon^{-1}))$$

We then use [Jerrum, Sinclair 1989] which gives an efficient algorithm for approximating the perfect matching sum.

The spin model can be defined on any graph! The graph G obtained from the reduction always has the desired feature that makes [Jerrum Sinclair 1989] efficient.

Proof sketch:

Start with a Trotter-Suzuki style approximation

$$\mathrm{Tr}\left(e^{-\beta H}\right) \approx^{\epsilon} \mathrm{Tr}(G_J \ldots G_2 G_1) \qquad J = \mathrm{poly}(n, \beta, \epsilon^{-1})$$

Proof sketch:

Start with a Trotter-Suzuki style approximation

$$\mathrm{Tr}\left(e^{-\beta H}\right) \approx^{\epsilon} \mathrm{Tr}(G_J \ldots G_2 G_1) \qquad J = \mathrm{poly}(n, \beta, \epsilon^{-1})$$

Each G_j satisfies

$$G_j = e^{-sh+O(s^2)} \qquad s > 0$$

where h is one of the terms in the Hamiltonian

Proof sketch:

Start with a Trotter-Suzuki style approximation

$$\mathrm{Tr}\left(e^{-\beta H}\right) \approx^{\epsilon} \mathrm{Tr}(G_J \ldots G_2 G_1) \qquad J = \mathrm{poly}(n, \beta, \epsilon^{-1})$$

Each G_j satisfies

$$G_j = e^{-sh+O(s^2)} \qquad s > 0$$

where h is one of the terms in the Hamiltonian

$$h = Y_i Y_j - X_i X_j, \quad \text{or} \quad h = -Y_i Y_j - X_i X_j, \quad \text{or} \quad h = \pm(I + Z_i)$$

Proof sketch:

Start with a Trotter-Suzuki style approximation

$$\mathrm{Tr}\left(e^{-\beta H}\right) \approx^{\epsilon} \mathrm{Tr}(G_J \ldots G_2 G_1) \qquad J = \mathrm{poly}(n, \beta, \epsilon^{-1})$$

Each G_j satisfies

$$G_j = e^{-sh+O(s^2)} \qquad s > 0$$

where h is one of the terms in the Hamiltonian

$$h = Y_i Y_j - X_i X_j, \quad \text{or} \quad h = -Y_i Y_j - X_i X_j, \quad \text{or} \quad h = \pm(I + Z_i)$$

The resulting G_j are very special gates…

Proof sketch:

Start with a Trotter-Suzuki style approximation

$$\mathrm{Tr}\left(e^{-\beta H}\right) \approx^{\epsilon} \mathrm{Tr}(G_J \ldots G_2 G_1) \qquad J = \mathrm{poly}(n, \beta, \epsilon^{-1})$$

Each G_J is from the gate set containing 1-qubit gates

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \begin{pmatrix} t & 0 \\ 0 & 1 \end{pmatrix} \qquad t > 0$$

and two qubit gates

$$\begin{pmatrix} 1+t^2 & 0 & 0 & t \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t & 0 & 0 & 1 \end{pmatrix} \qquad t > 0$$

"Matchgates"

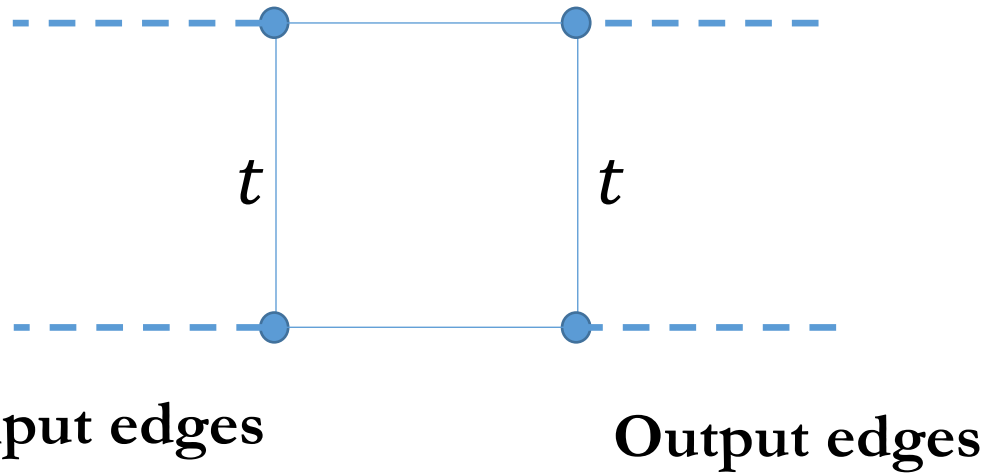
Let Γ be a a weighted graph with special input and output edges (k of each, say)

We say Γ implements a k-qubit operator G if

$$\langle y|G|x\rangle = \mathrm{PerfMatch}(\Gamma_{xy})$$

Γ_{xy} = remove input edges with $x_i = 0$ and output edges with $y_i = 0$. Require that a perfect matching includes the remaining external edges.

Example:



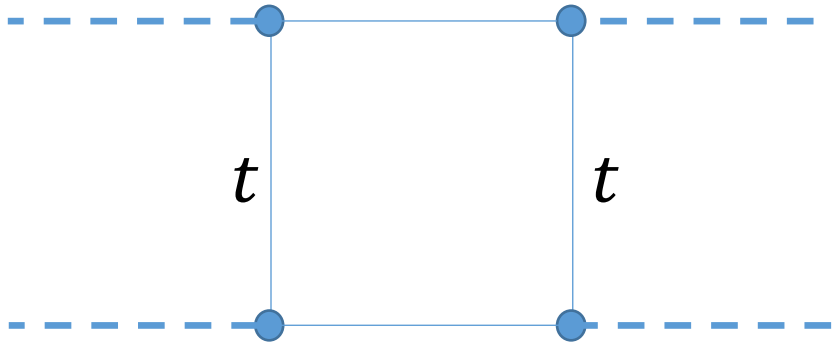
Let Γ be a a weighted graph with special input and output edges (k of each, say)

We say Γ implements a k-qubit operator G if

$$\langle y|G|x\rangle = \mathrm{PerfMatch}(\Gamma_{xy})$$

Γ_{xy} = remove input edges with $x_i = 0$ and output edges with $y_i = 0$. Require that a perfect matching includes the remaining external edges.

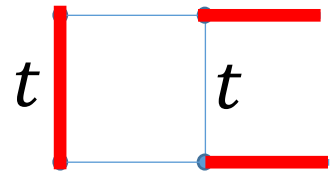
Example:



Input edges

Output edges

$\boldsymbol{x = 00, y = 11}$



$\boldsymbol{\langle 11|G|00\rangle = t}$

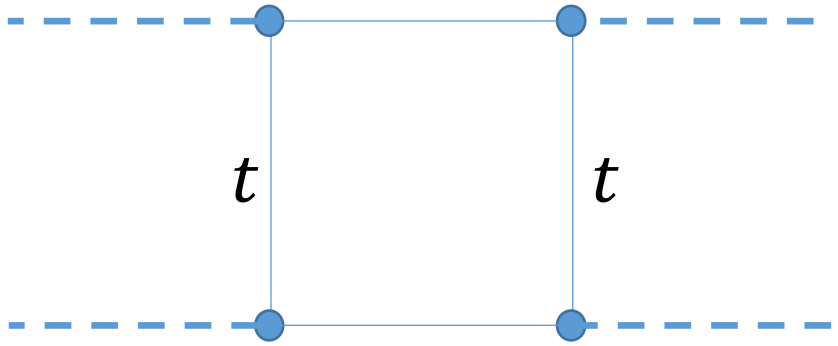
Let Γ be a a weighted graph with special input and output edges (k of each, say)

We say Γ implements a k-qubit operator G if

$$\langle y|G|x\rangle = \mathrm{PerfMatch}(\Gamma_{xy})$$

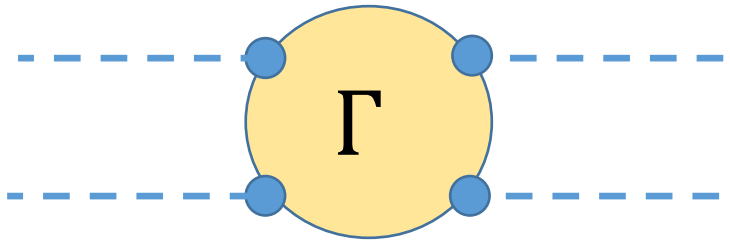
Γ_{xy} = remove input edges with $x_i = 0$ and output edges with $y_i = 0$. Require that a perfect matching includes the remaining external edges.

Example:

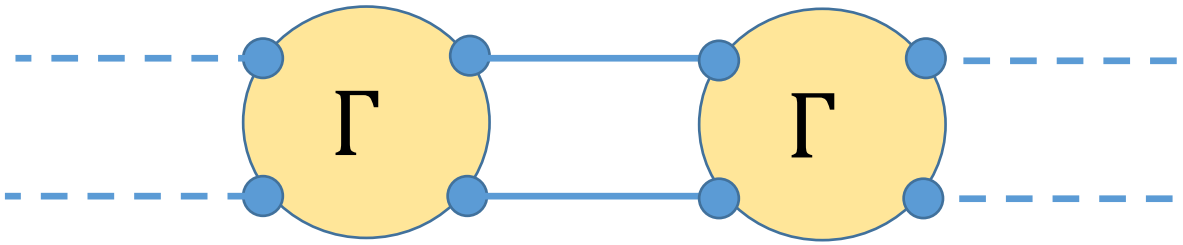


$$G = \begin{pmatrix} 1+t^2 & 0 & 0 & t \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t & 0 & 0 & 1 \end{pmatrix}$$

Matchgates compose nicely

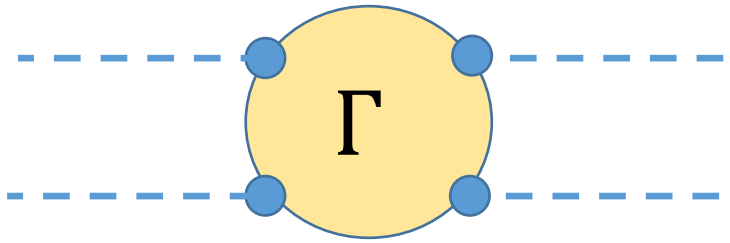


Implements a 2 qubit gate G

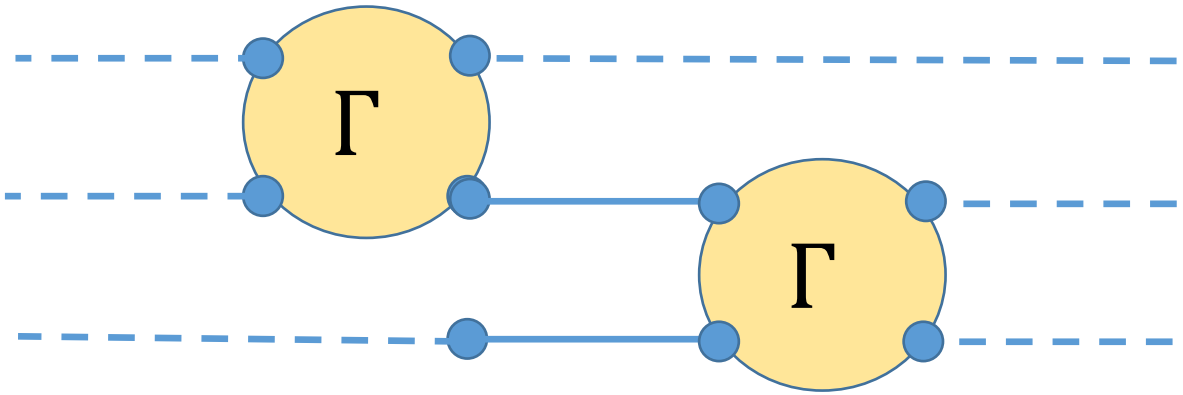


Implements G^2

Matchgates compose nicely

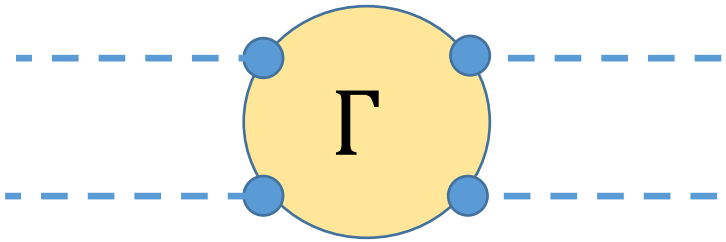


Implements a 2 qubit gate G

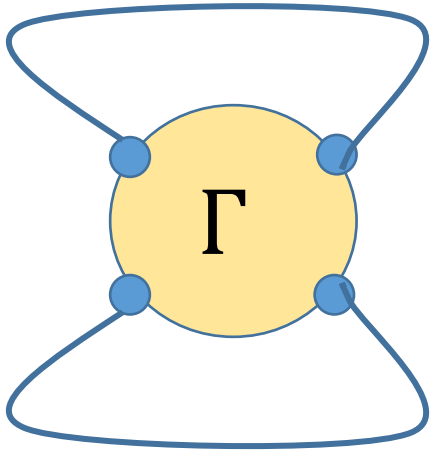


Implements $G_{12}G_{23}$

Matchgates compose nicely



Implements a 2 qubit gate G



Implements $\mathbf{Tr}(G)$

Proof sketch:

Start with a Trotter-Suzuki style approximation

$$\mathrm{Tr}\left(e^{-\beta H}\right) \approx^{\epsilon} \mathrm{Tr}(G_J \ldots G_2 G_1) \qquad J = \mathrm{poly}(n, \beta, \epsilon^{-1})$$

Each G_J is a matchgate.

Proof sketch:

Start with a Trotter-Suzuki style approximation

$$\mathrm{Tr}\left(e^{-\beta H}\right) \approx^{\epsilon} \mathrm{Tr}(G_J \ldots G_2 G_1) \qquad J = \mathrm{poly}(n, \beta, \epsilon^{-1})$$

Matchgate implementable with n input edges and n output edges

Each G_J is a matchgate.

Proof sketch:

Start with a Trotter-Suzuki style approximation

$$\mathrm{Tr}\left(e^{-\beta H}\right) \approx^{\epsilon} \mathrm{Tr}(G_J \ldots G_2 G_1) \qquad J = \mathrm{poly}(n, \beta, \epsilon^{-1})$$

Each G_J is a matchgate.

Matchgate implementable with n input edges and n output edges

Trace is a matchgate with no external edges, i.e., a perfect matching sum.

Proof sketch:

Start with a Trotter-Suzuki style approximation

$$\mathrm{Tr}\left(e^{-\beta H}\right) \approx^{\epsilon} \mathrm{Tr}(G_J \ldots G_2 G_1) \qquad J = \mathrm{poly}(n, \beta, \epsilon^{-1})$$

Each G_J is a matchgate.

Matchgate implementable with n input edges and n output edges

Trace is a matchgate with no external edges, i.e., a perfect matching sum. (nonplanar and nonbipartite)

Proof sketch:

Start with a Trotter-Suzuki style approximation

$$\mathrm{Tr}\left(e^{-\beta H}\right) \approx^{\epsilon} \mathrm{Tr}(G_J \ldots G_2 G_1) \qquad J = \mathrm{poly}(n, \beta, \epsilon^{-1})$$

Each G_J is a matchgate.

Matchgate implementable with n input edges and n output edges

Trace is a matchgate with no external edges, i.e., a perfect matching sum. (nonplanar and nonbipartite)

This gives first part of theorem:

$$Z(\beta) \approx^{\epsilon} \mathrm{PerfMatch}(G)$$

Bounding $\mathrm{NearPerfMatch}(G)$

We need to show:

$$\frac{\mathrm{NearPerfMatch}(G)}{\mathrm{PerfMatch}(G)} = O(\mathrm{poly}(\beta, n, \epsilon^{-1}))$$

Recall that a nearly perfect matching is like a perfect matching but with 2 vertices unmatched.

Bounding $\mathrm{NearPerfMatch}(G)$

We need to show:

$$\frac{\mathrm{NearPerfMatch}(G)}{\mathrm{PerfMatch}(G)} = O(\mathrm{poly}(\beta, n, \epsilon^{-1}))$$

Recall that a nearly perfect matching is like a perfect matching but with 2 vertices unmatched.

$$\mathrm{NearPerfMatch}(G) = \sum_{u,v \in G} \Omega_{u,v} \qquad \Omega_{u,v} = \text{sum of nearly perfect matchings with } u, v \text{ unmatched.}$$

Bounding $\mathrm{NearPerfMatch}(G)$

We need to show:

$$\frac{\mathrm{NearPerfMatch}(G)}{\mathrm{PerfMatch}(G)} = O(\mathrm{poly}(\beta, n, \epsilon^{-1}))$$

Recall that a nearly perfect matching is like a perfect matching but with 2 vertices unmatched.

$$\mathrm{NearPerfMatch}(G) = \sum_{u,v \in G} \Omega_{u,v} \qquad \Omega_{u,v} = \text{sum of nearly perfect matchings with } u, v \text{ unmatched.}$$

To complete the proof we show that

$$\frac{\Omega_{u,v}}{\mathrm{PerfMatch}(G)} \approx \frac{\mathrm{Tr}\left(G_J G_{J-1} \dots G_j O G_{j-1} G_{j-2} \dots G_i P G_{i-1} G_{i-2} \dots G_2 G_1\right)}{\mathrm{Tr}\left(G_J \dots G_2 G_1\right)} = O(1)$$

Imaginary time spin-spin correlation function

Bounding $\mathrm{NearPerfMatch}(G)$

We need to show:

$$\frac{\mathrm{NearPerfMatch}(G)}{\mathrm{PerfMatch}(G)} = O(\mathrm{poly}(\beta, n, \epsilon^{-1}))$$

Recall that a nearly perfect matching is like a perfect matching but with 2 vertices unmatched.

$$\mathrm{NearPerfMatch}(G) = \sum_{u,v \in G} \Omega_{u,v} \qquad \Omega_{u,v} = \text{sum of nearly perfect matchings with } u, v \text{ unmatched.}$$

To complete the proof we show that

$$\frac{\Omega_{u,v}}{\mathrm{PerfMatch}(G)} \approx \frac{\mathrm{Tr}\left(G_J G_{J-1} \dots G_j O G_{j-1} G_{j-2} \dots G_i P G_{i-1} G_{i-2} \dots G_2 G_1\right)}{\mathrm{Tr}\left(G_J \dots G_2 G_1\right)} = O(1)$$

Imaginary time spin-spin correlation function

Summary

The computational complexity of relative-error approximation of quantum partition functions or equivalently quantum approximate counting is a fascinating mystery.

Some mathematical quantities that are not known to admit combinatorial interpretations may in fact be quantum counting problems. In that case the related approximation tasks are no harder than approximating the partition function of 2-local Hamiltonians.

Efficient algorithms reveal exploitable structure and more should be found…