A Sublinear-Time Quantum Algorithm for Approximating Partition Functions

Arjan Cornelissen, Yassine Hamoudi

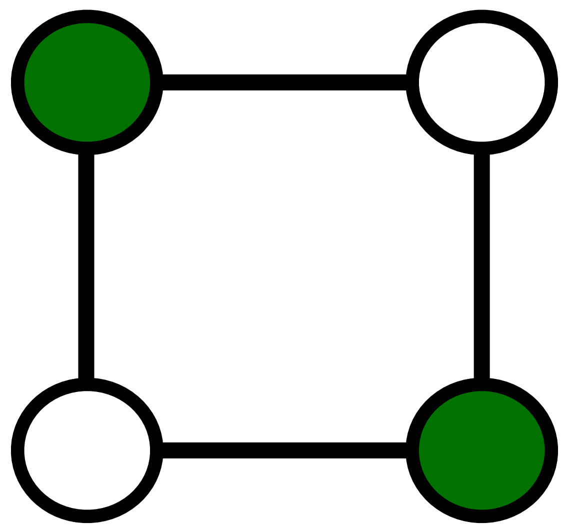
QuSoft UC Berkeley

SODA'23, QIP'23

Independent sets

Independent set

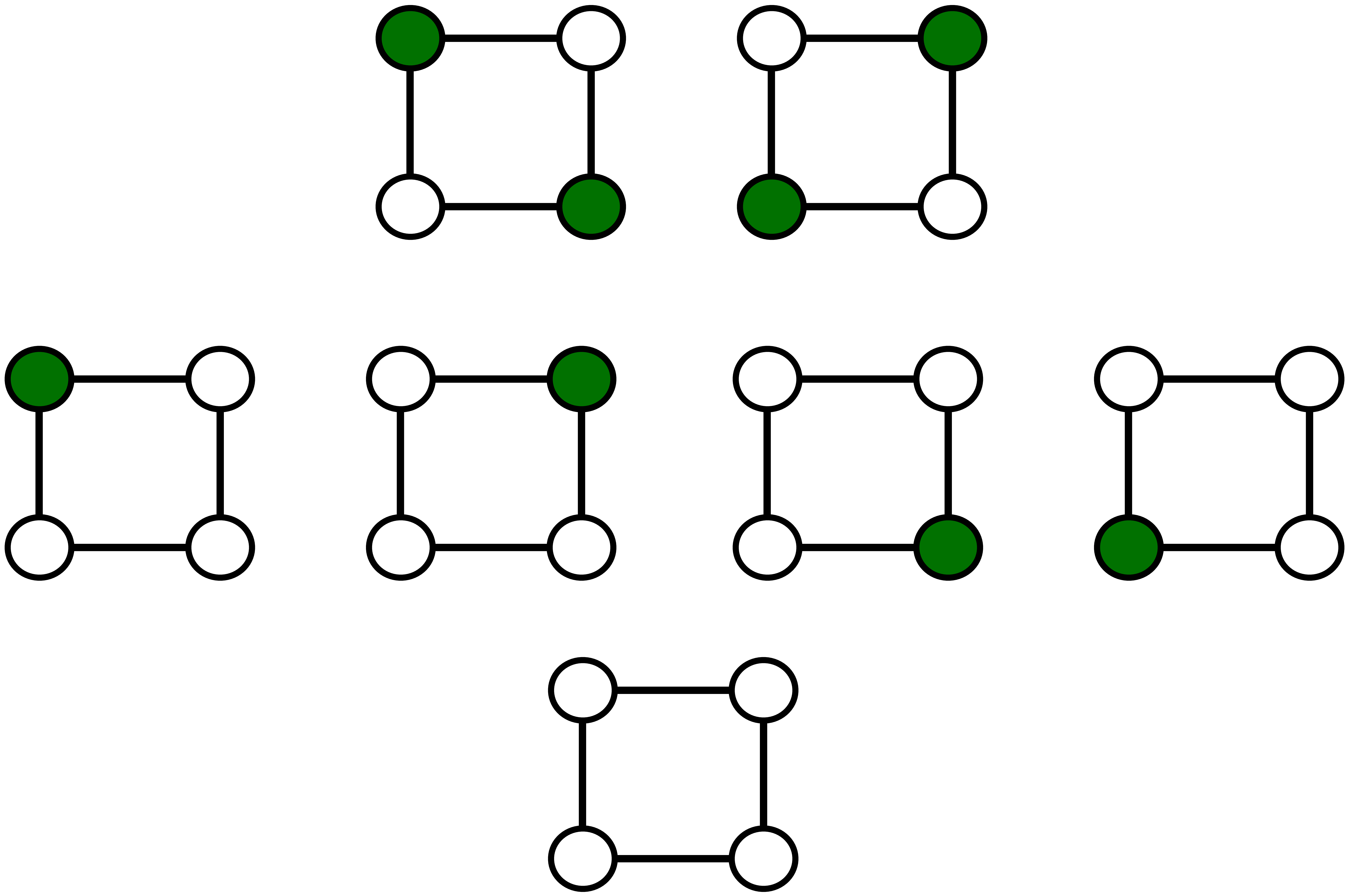
= subset of non-adjacent vertices



= occupied

Hard-core gas model in statistical physics

independent sets = 7



Input: graph G
Output: # independent sets of G

#P-hard in many regimes

Bipartite graphs
[Provan,Ball'83]

3-regular graphs
[Dyer,Greenhill'00]

...

Exact counting ⟶ Approximate counting?

Input: graph G and $\epsilon \in (0,1)$

Output: S s.t. $(1-\epsilon)$ #ind $\leq S \leq (1+\epsilon)$ #ind

$n = $ #vertices

Classical algorithms

$\tilde{O}(n^2/\epsilon^2)$ [Štefankovič,Vempala,Vigoda'09] [Chen,Liu,Vigoda'21]

5

No FPRAS unless NP = RP [Sly'10]

Maximum degree in G

Quantum algorithms

$\tilde{O}(n^2 + n^{3/2}/\epsilon)$ [Montanaro'15]

$\tilde{O}(n^{3/2}/\epsilon)$ [Harrow,Wei'20]

$\tilde{O}(n^{5/4}/\epsilon)$ **Our work**

Weighted independent sets

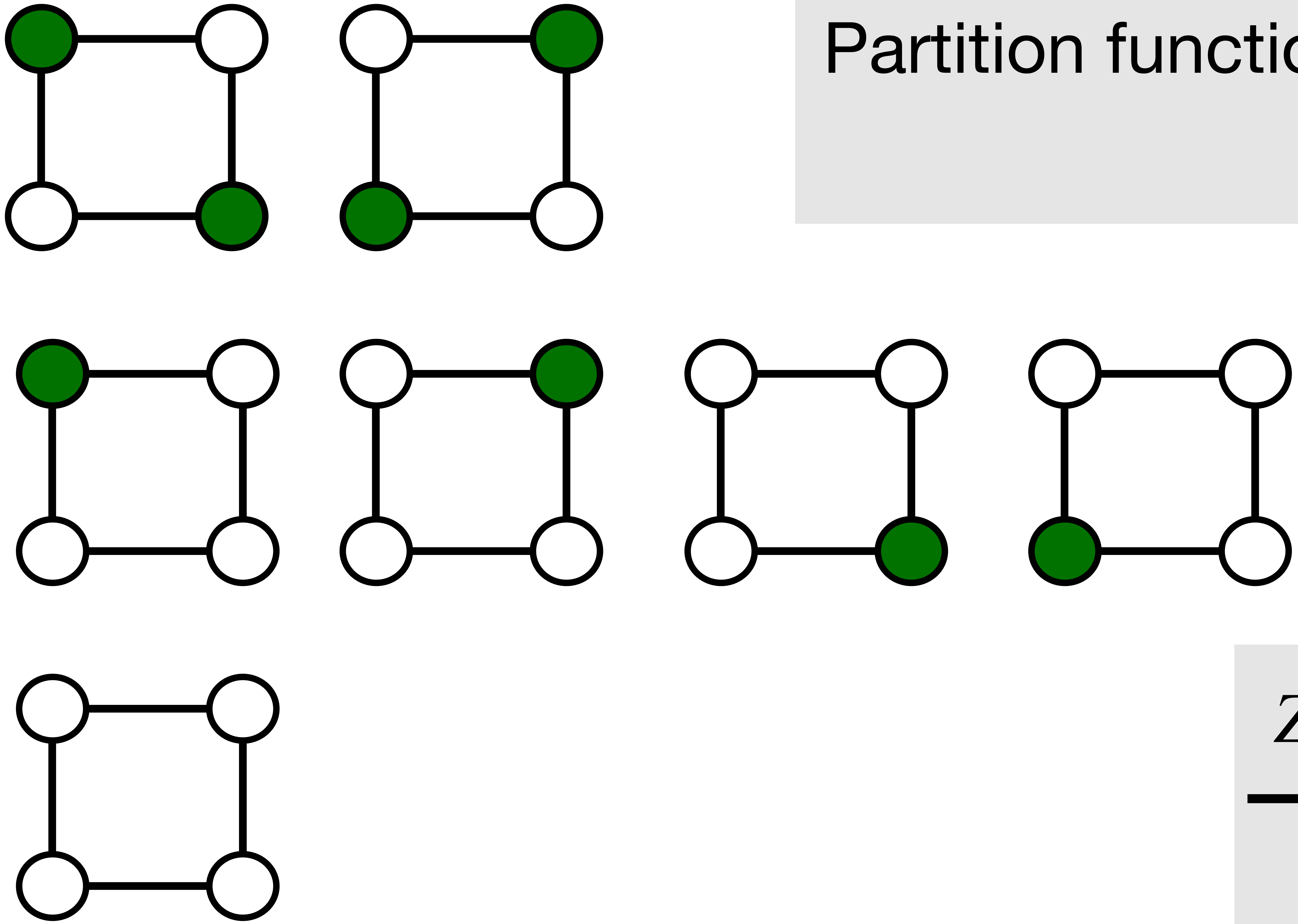
λ = fugacity

Partition function: $Z(\lambda) = \sum_{I \text{ ind. set}} \lambda^{|I|}$

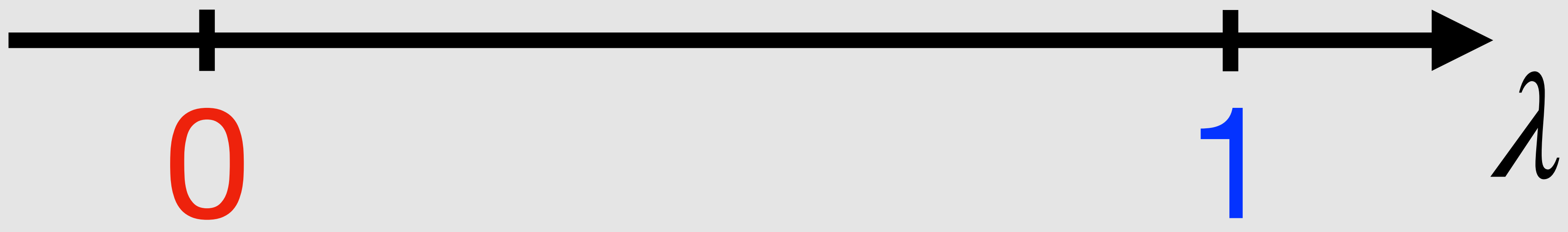
λ^2

λ

1



$Z(\mathbf{0}) = 1$ $Z(\mathbf{1}) =$ # ind



Partition function: $Z(\lambda) = \sum_{I \text{ ind. set}} \lambda^{|I|}$

Gibbs distribution: $\pi(I) = \frac{\lambda^{|I|}}{Z(\lambda)}$

Glauber dynamics

[Chen,Liu,Vigoda'21]

For any $0 \leq \lambda \leq 1$,

we can sample $I \sim \pi$ in time $O(n \log n)$

Sampling $\longrightarrow$ Approximate counting?

Cooling schedule

[Štefankovič,Vempala,Vigoda'09]

$Z(0) = 1$ $\qquad$ $Z(1) = \#$ ind

$\lambda_0 = 0 \quad \lambda_1 \quad \lambda_2 \quad \cdots \quad \lambda_k = 1$

$\pi_0 \approx \pi_1$ $\qquad$ $\pi_1 \approx \pi_2$

$i \to i+1$:

$$Z(\lambda_{i+1}) = \mathrm{E}_{I \sim \pi_i} \left(\frac{\lambda_{i+1}}{\lambda_i} \right)^{|I|} Z(\lambda_i)$$

Highly concentrated: variance ≤ expectation²

schedule length k

sample complexity

cost per sample

Cost = $\sqrt{n} \times \sqrt{n}/\epsilon^2 \times n\log(n)$

Quantum speedup(s)

Glauber dynamics

Sample ind. set $I \sim \pi$ in time $O(n \log n)$

No speedup for sampling?

We can check in time $O(\sqrt{n})$ if a quantum state is equal to:

$$|\pi\rangle = \sum_I \sqrt{\pi(I)}\,|I\rangle$$ (quantum sample)

… without destroying the state!

$|\pi\rangle = \sum_I \sqrt{\pi(I)}\,|I\rangle$

$$\text{Check}(\pi):$$

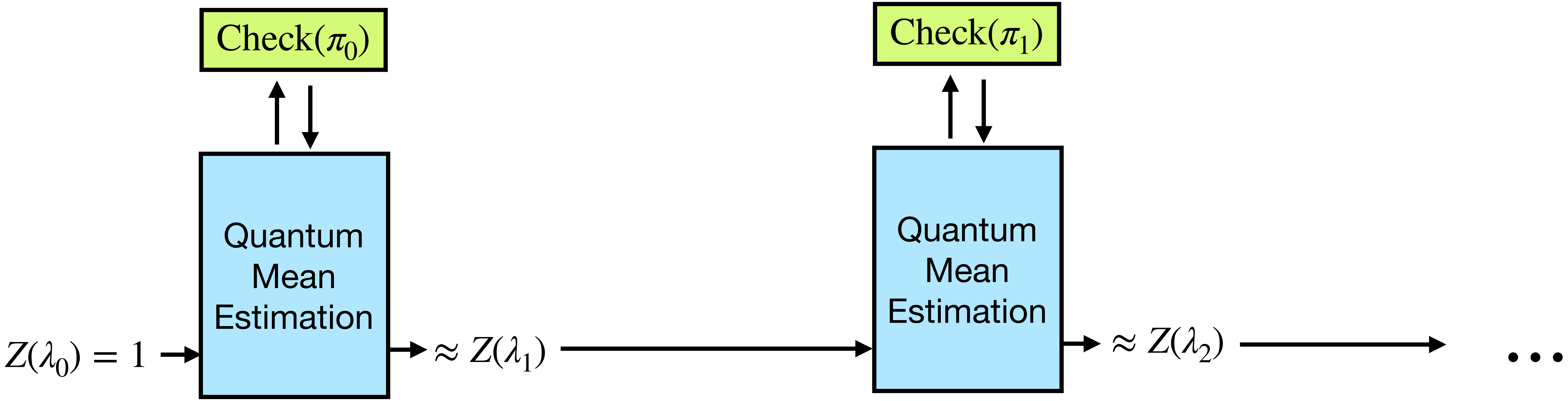
$$|\pi\rangle|0\rangle \mapsto |\pi\rangle|0\rangle$$

$$|\mu\rangle|0\rangle \mapsto |\mu\rangle|1\rangle \text{ if } |\mu\rangle \perp |\pi\rangle$$

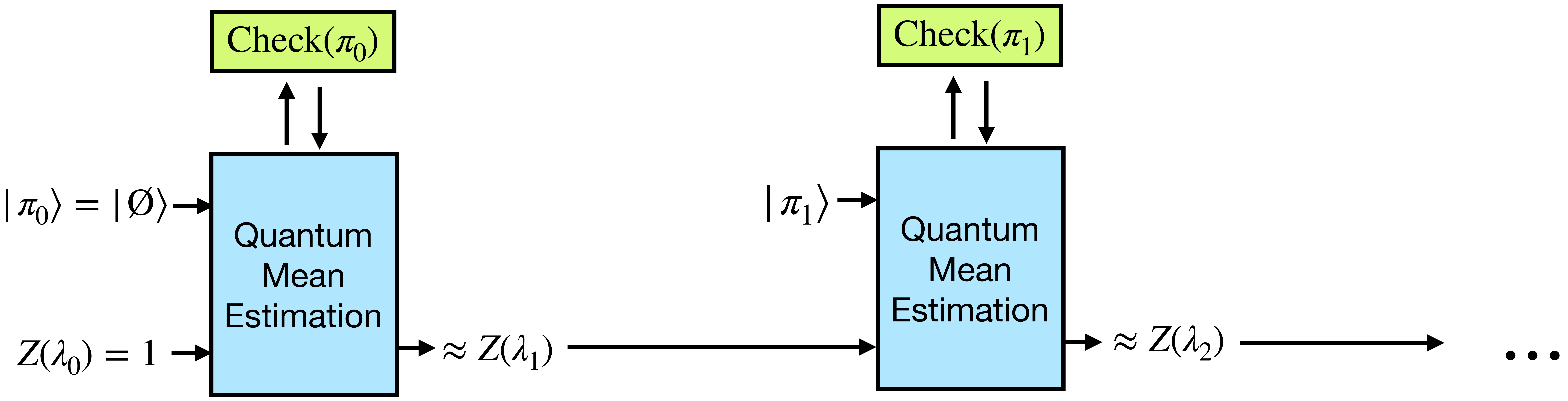
Szegedy Quantum Walk

Markov Chain with stationary distribution π and spectral gap δ $\Rightarrow$ $\text{Check}(\pi)$ in time $\sim 1/\sqrt{\delta}$

Estimate $Z(\lambda_{i+1})$ using $\mathrm{Check}(\pi_i)$?

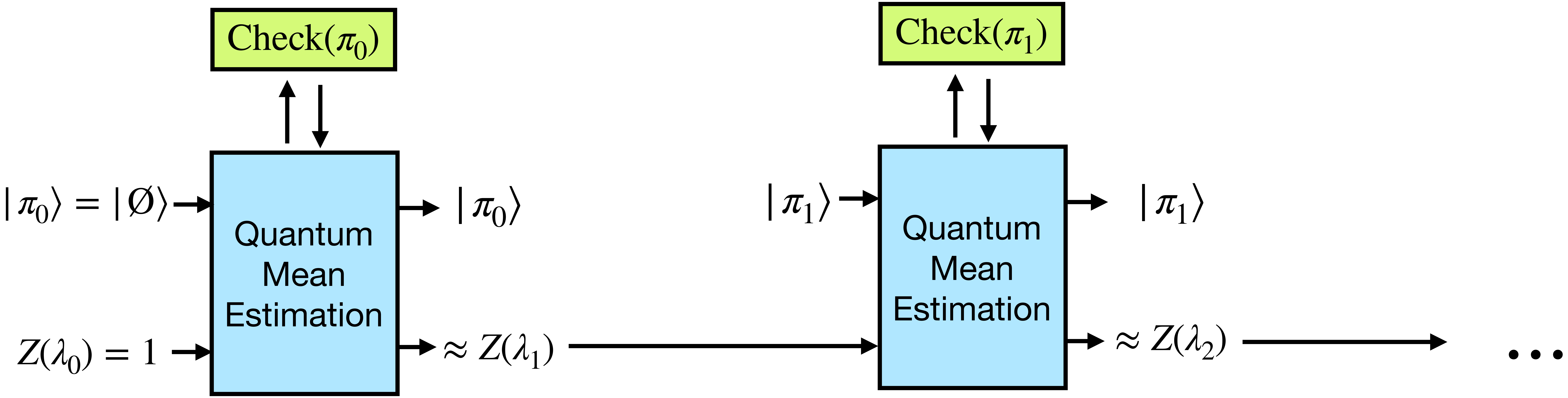


Estimate $Z(\lambda_{i+1})$ using $\mathrm{Check}(\pi_i)$? | ... and 1 copy of $|\pi_i\rangle$



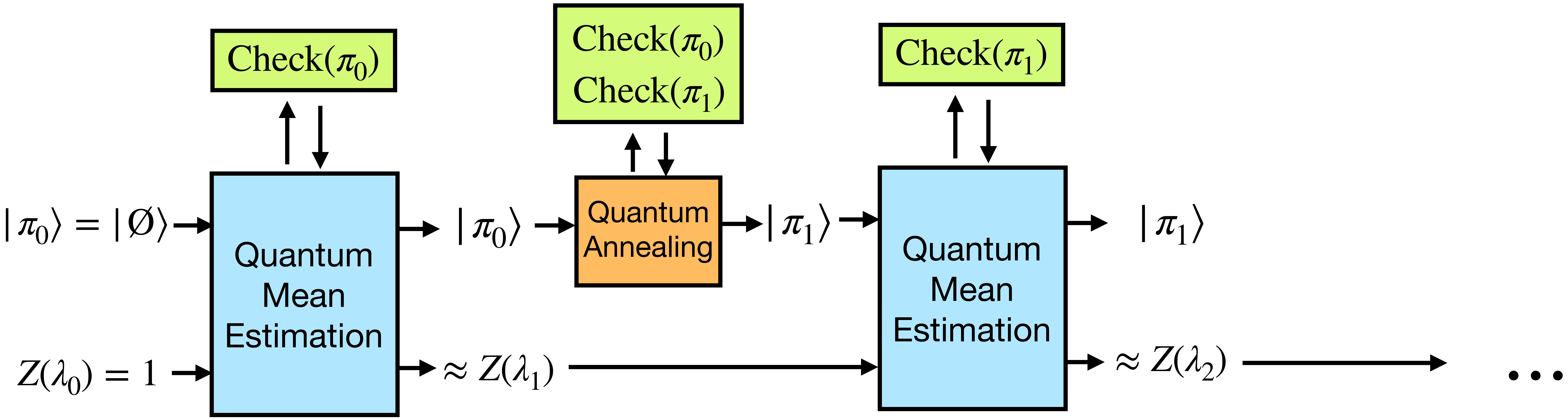
Estimate $Z(\lambda_{i+1})$ using $\mathrm{Check}(\pi_i)$? ... and 1 copy of $|\pi_i\rangle$

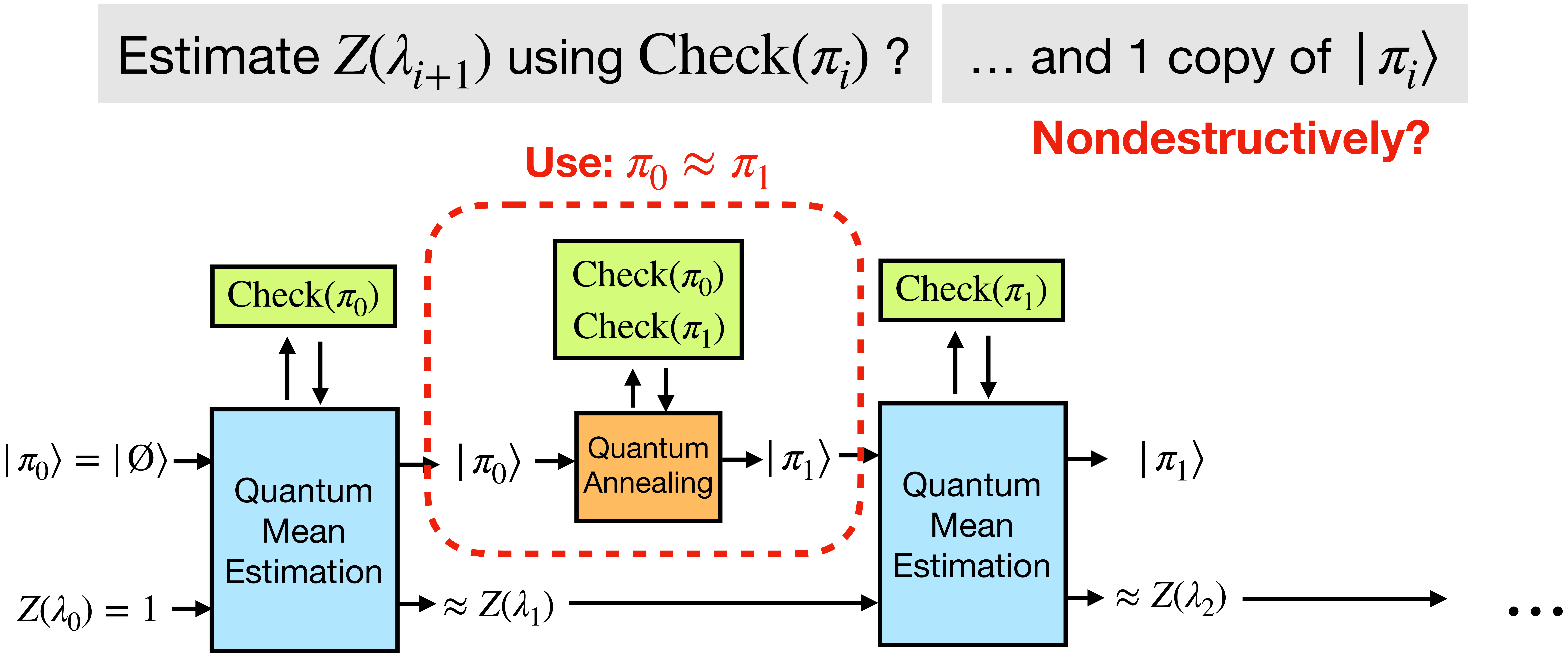
Nondestructively?



Estimate $Z(\lambda_{i+1})$ using $\mathrm{Check}(\pi_i)$? ... and 1 copy of $|\pi_i\rangle$

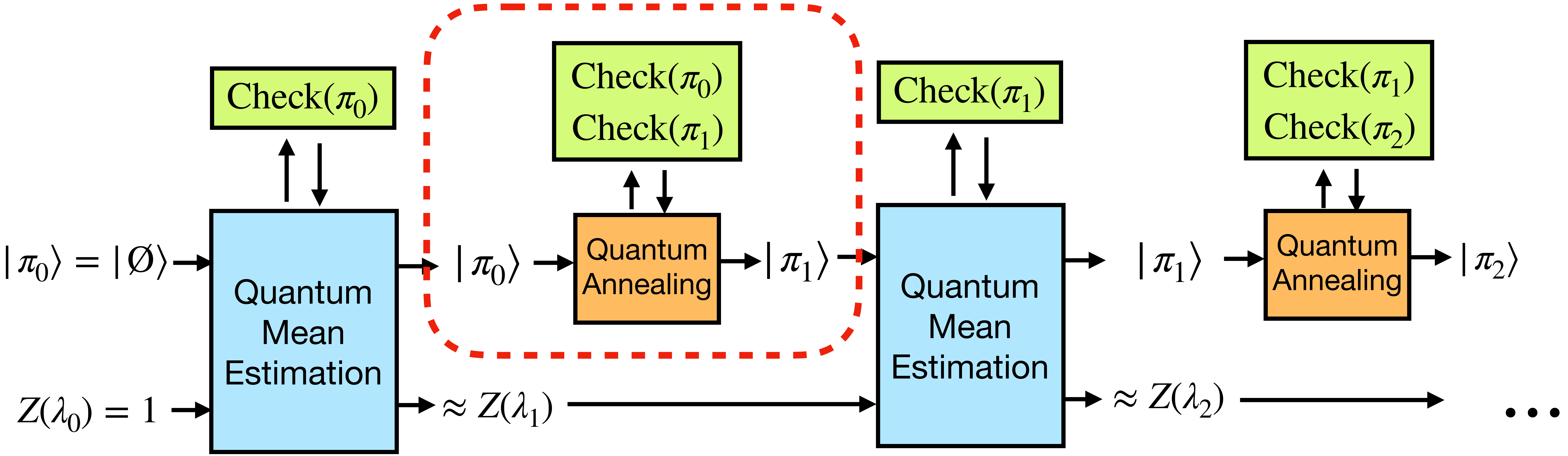
Nondestructively?




Estimate $Z(\lambda_{i+1})$ using $\mathrm{Check}(\pi_i)$?
... and 1 copy of $|\pi_i\rangle$
Nondestructively?
Use: $\pi_0 \approx \pi_1$
$\mathrm{Check}(\pi_0)$
$\mathrm{Check}(\pi_0)$
$\mathrm{Check}(\pi_1)$
$\mathrm{Check}(\pi_1)$
$|\pi_0\rangle = |\emptyset\rangle$
Quantum Mean Estimation
$|\pi_0\rangle$
Quantum Annealing
$|\pi_1\rangle$
Quantum Mean Estimation
$|\pi_1\rangle$
$Z(\lambda_0) = 1$
$\approx Z(\lambda_1)$
$\approx Z(\lambda_2)$
...

Estimate $Z(\lambda_{i+1})$ using $\mathrm{Check}(\pi_i)$? ... and 1 copy of $|\pi_i\rangle$

Nondestructively?

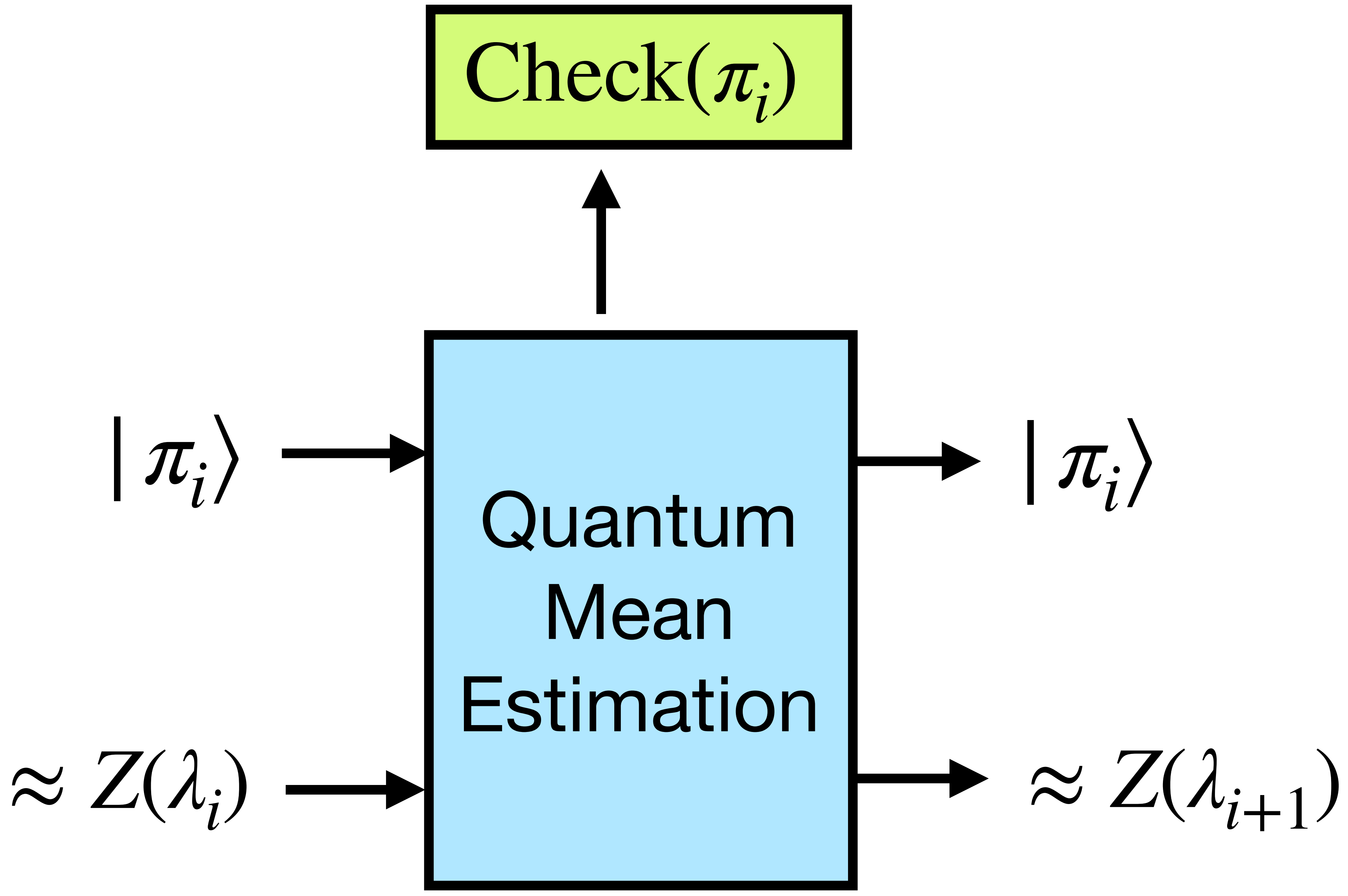


Estimate $Z(\lambda_{i+1})$ using $\mathrm{Check}(\pi_i)$? ... and 1 copy of $|\pi_i\rangle$

Nondestructively?

Unbiased? Otherwise the errors accumulate too quickly!

Framework for Mean Estimation



We want the following properties:

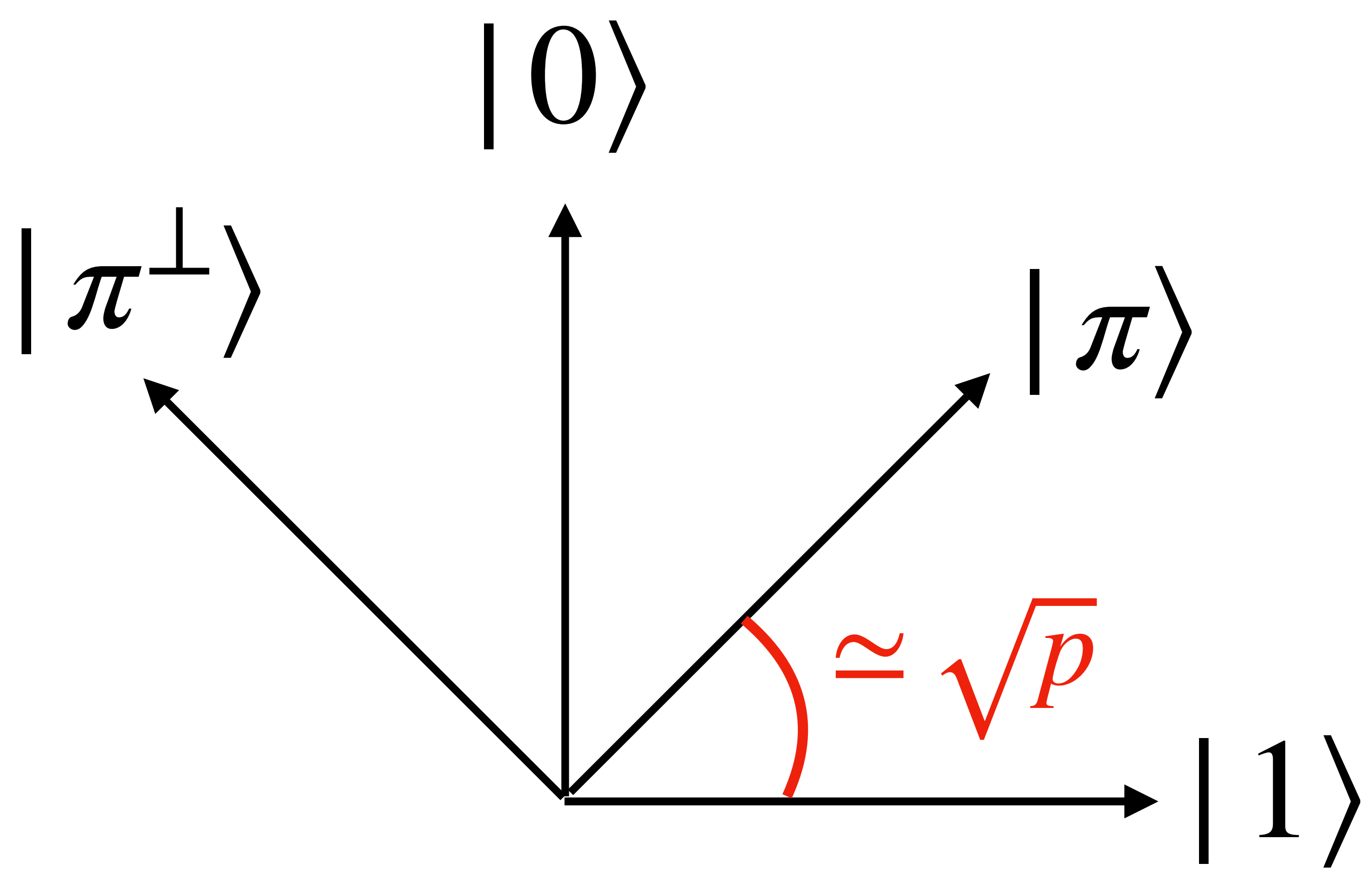
Nondestructive Output a new copy of $|\pi_i\rangle$

Unbiased $\mathrm{E}(\text{output}) = Z(\lambda_{i+1})$

Speedup Faster than the Empirical Mean

The case of Bernoulli distributions

$$|\pi\rangle = \sqrt{1-p}\,|0\rangle + \sqrt{p}\,|1\rangle$$



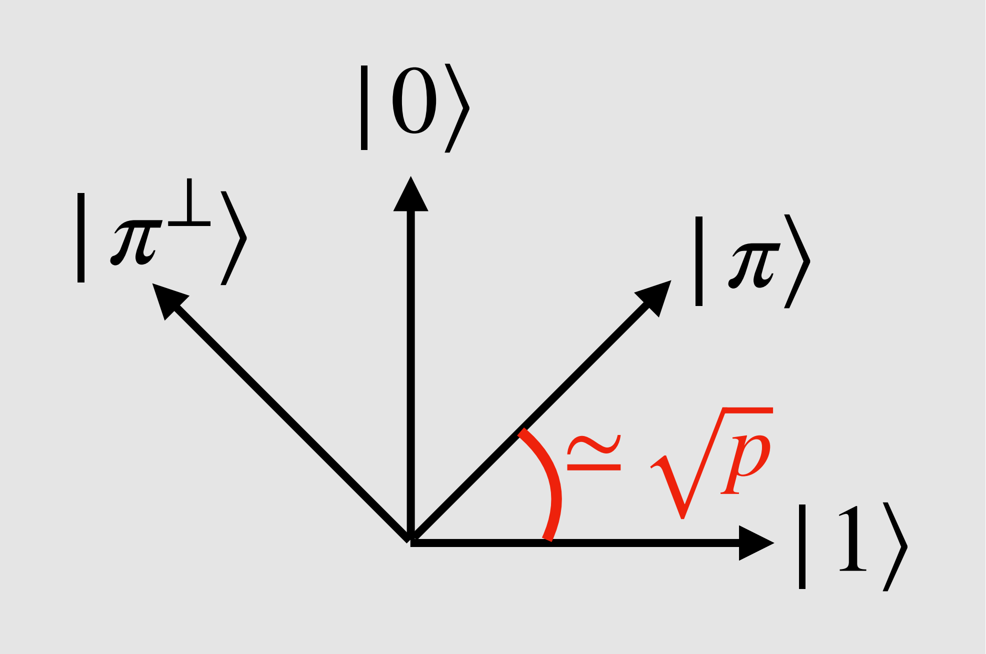
$\mathrm{Check}(\pi)$:

$$|\pi\rangle|0\rangle \mapsto |\pi\rangle|0\rangle$$
$$|\pi^\perp\rangle|0\rangle \mapsto |\pi^\perp\rangle|1\rangle$$

$$\Leftrightarrow$$

$\mathrm{Ref}_{|\pi\rangle}$:

$$|\pi\rangle \mapsto |\pi\rangle$$
$$|\pi^\perp\rangle \mapsto -|\pi^\perp\rangle$$

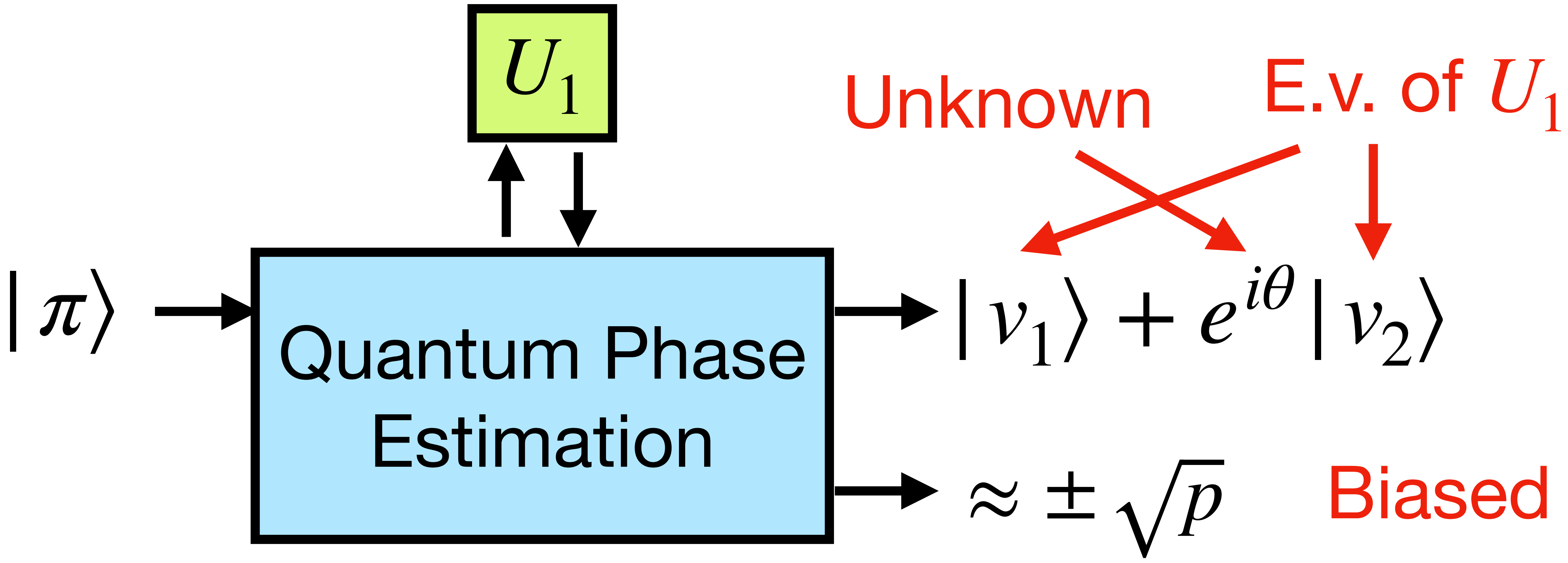


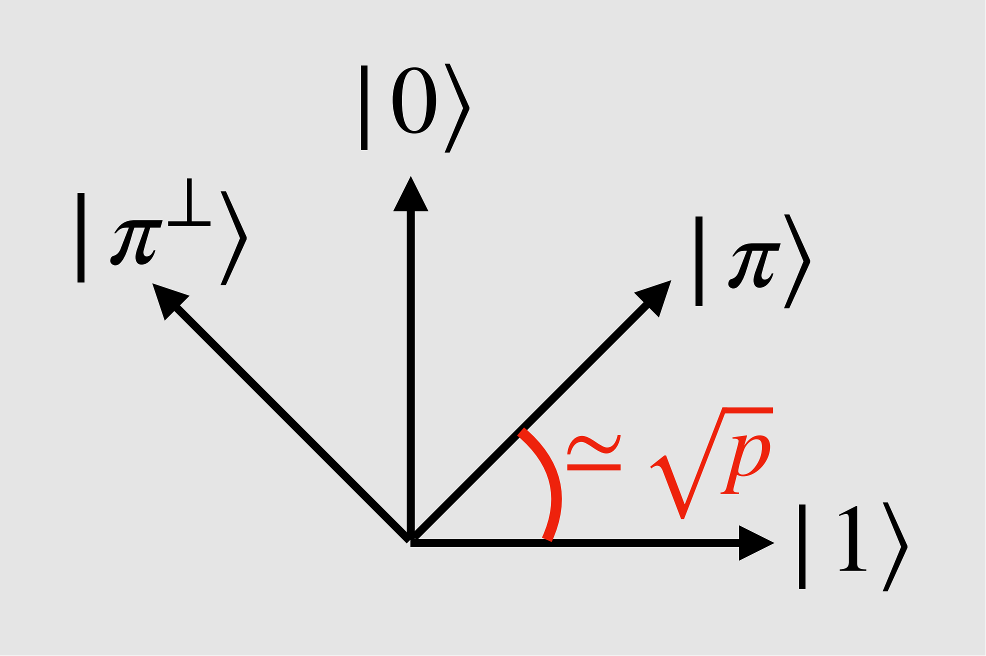
The case of Bernoulli distributions

$$|\pi\rangle = \sqrt{1-p}\,|0\rangle + \sqrt{p}\,|1\rangle$$

$$U_1 = \begin{pmatrix} e^{2i\sqrt{p}} & 0 \\ 0 & e^{-2i\sqrt{p}} \end{pmatrix} \simeq \mathrm{Ref}_{|\pi\rangle} \cdot \mathrm{Ref}_{|1\rangle}$$

(Grover operator)





The case of Bernoulli distributions

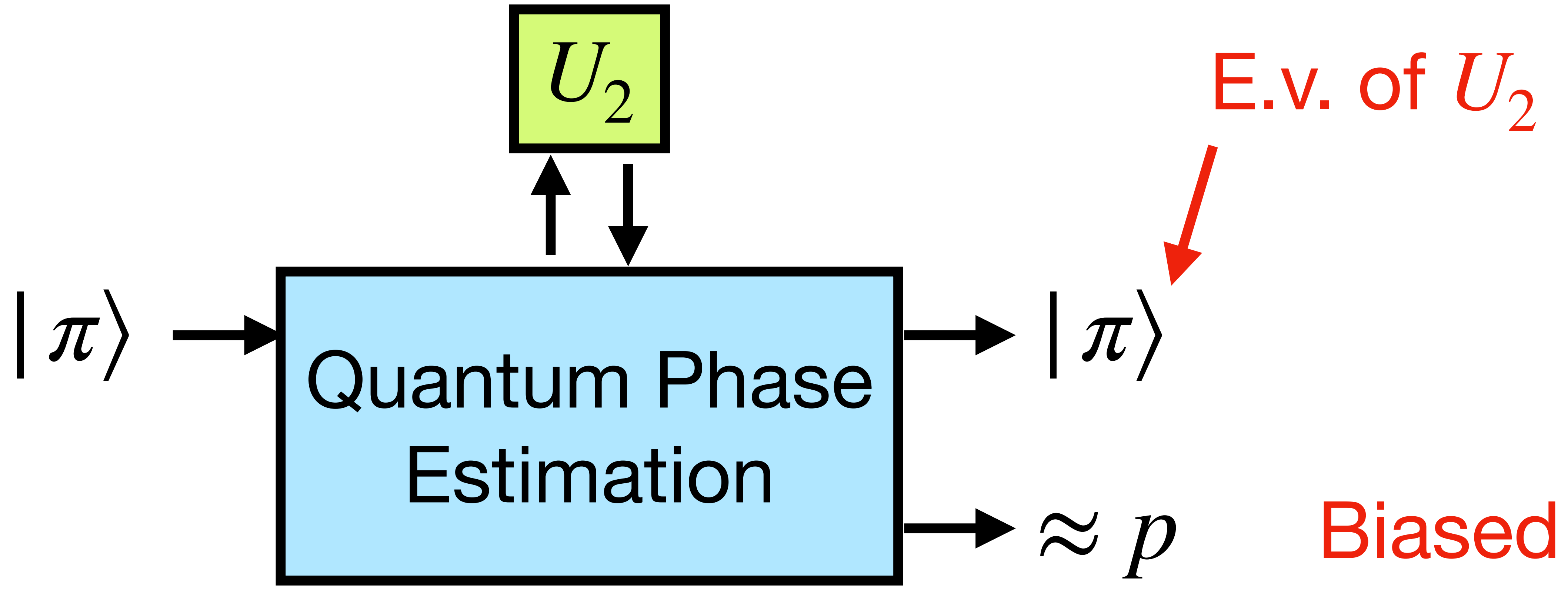
$$|\pi\rangle = \sqrt{1-p}\,|0\rangle + \sqrt{p}\,|1\rangle$$

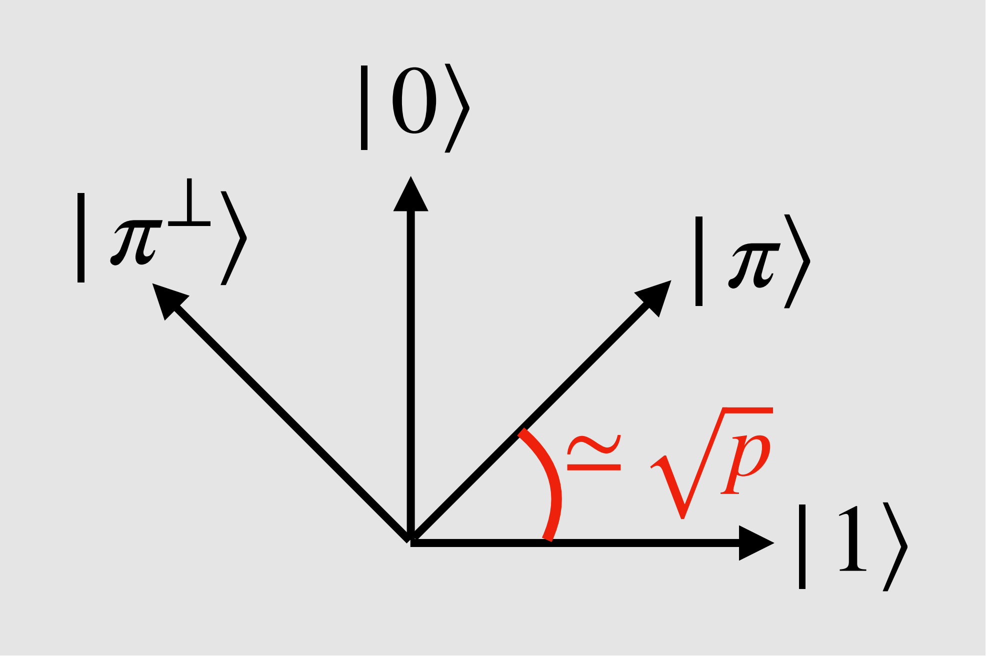
$$U_1 = \begin{pmatrix} e^{2i\sqrt{p}} & 0 \\ 0 & e^{-2i\sqrt{p}} \end{pmatrix} \simeq \mathrm{Ref}_{|\pi\rangle} \cdot \mathrm{Ref}_{|1\rangle}$$

(Grover operator)

$$U_2 = \begin{pmatrix} e^{ip} & 0 \\ 0 & e^{ip} \end{pmatrix} = \sum_{m=-\infty}^{\infty} c_m U_1^m$$

(Taylor expansion + LCU)





The case of Bernoulli distributions

$$|\pi\rangle = \sqrt{1-p}\,|0\rangle + \sqrt{p}\,|1\rangle$$

$$U_1 = \begin{pmatrix} e^{2i\sqrt{p}} & 0 \\ 0 & e^{-2i\sqrt{p}} \end{pmatrix} \simeq \mathrm{Ref}_{|\pi\rangle} \cdot \mathrm{Ref}_{|1\rangle}$$

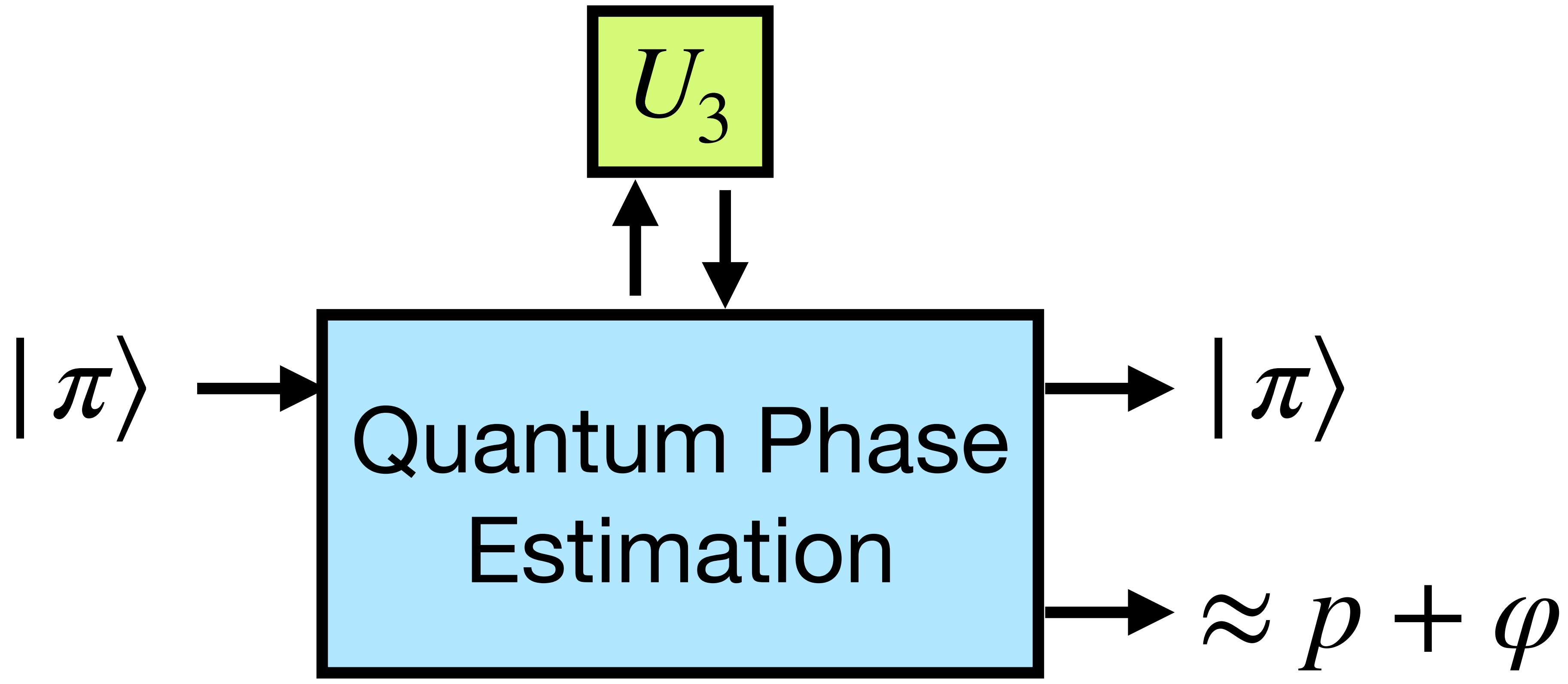
(Grover operator)

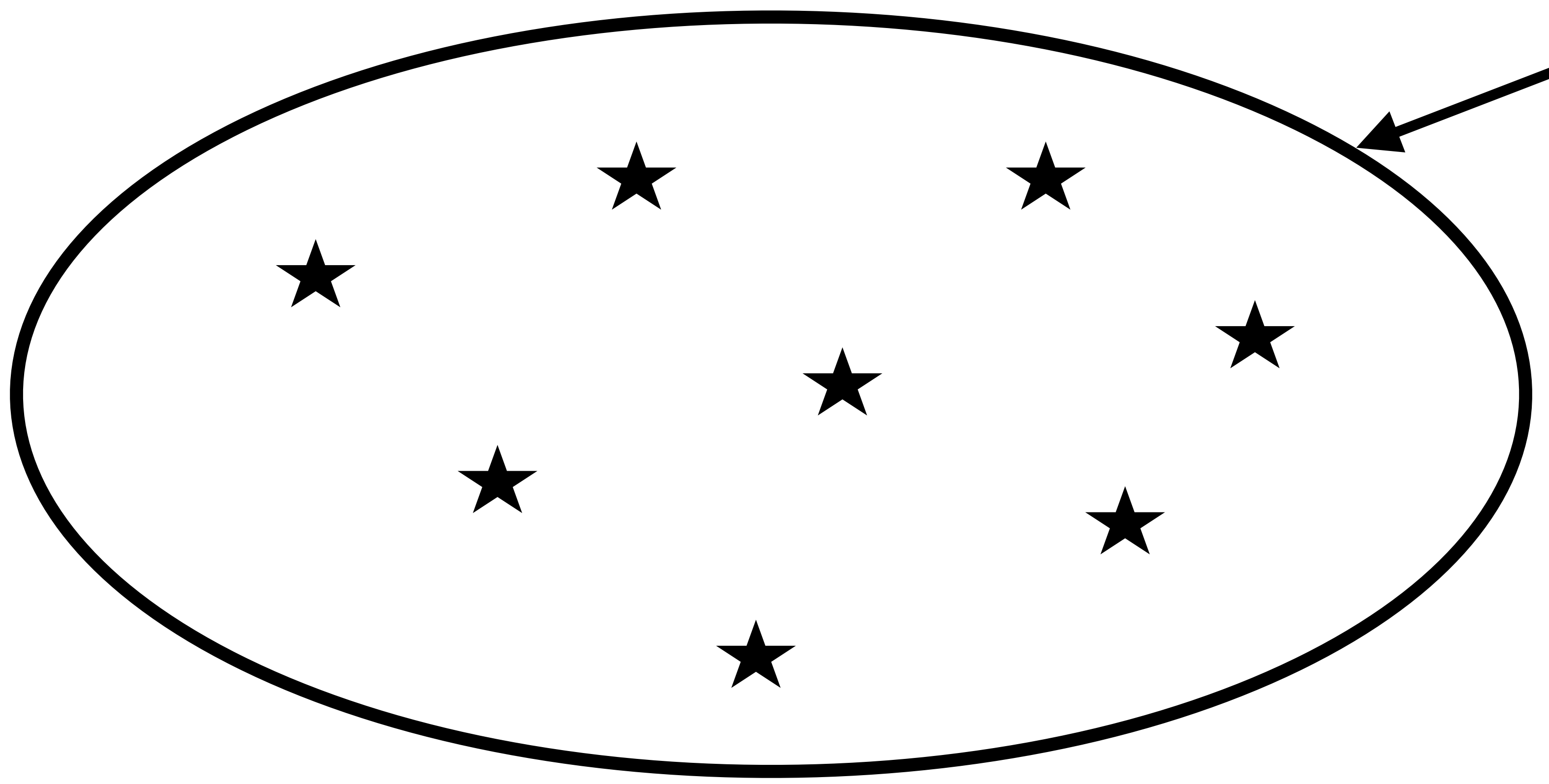
$$U_2 = \begin{pmatrix} e^{ip} & 0 \\ 0 & e^{ip} \end{pmatrix} = \sum_{m=-\infty}^{\infty} c_m U_1^m$$

(Taylor expansion + LCU)

$$U_3 = \begin{pmatrix} e^{i(p+\varphi)} & 0 \\ 0 & e^{i(p+\varphi)} \end{pmatrix}$$

(Random phase shift)



$\Omega =$ 

Examples:

- independent sets
- k-colorings
- matchings
- (volume of convex body)
- (Ising model)
- ...

Approximate the size $|\Omega|$ in time

$$\approx \log^{3/4}|\Omega| \times \sqrt{\text{class. mixing time}}$$

Previous work:

$$\log|\Omega| \times \ldots$$

Open question: $\log^{1/2}|\Omega| \times \sqrt{\text{class. mixing time}}$?