Demystifying Quantum Power Flow: Is It Fast?

Parikshit Pareek
Assistant Professor
Indian Institute of Technology Roorkee

Grid Science Winter School
Santa Fe, New Mexico, USA
9 January 2025

Joint work with — Abhijith Jayakumar, Carleton Coffrin and Sidhant Misra at LANL

LA-UR 24-28593

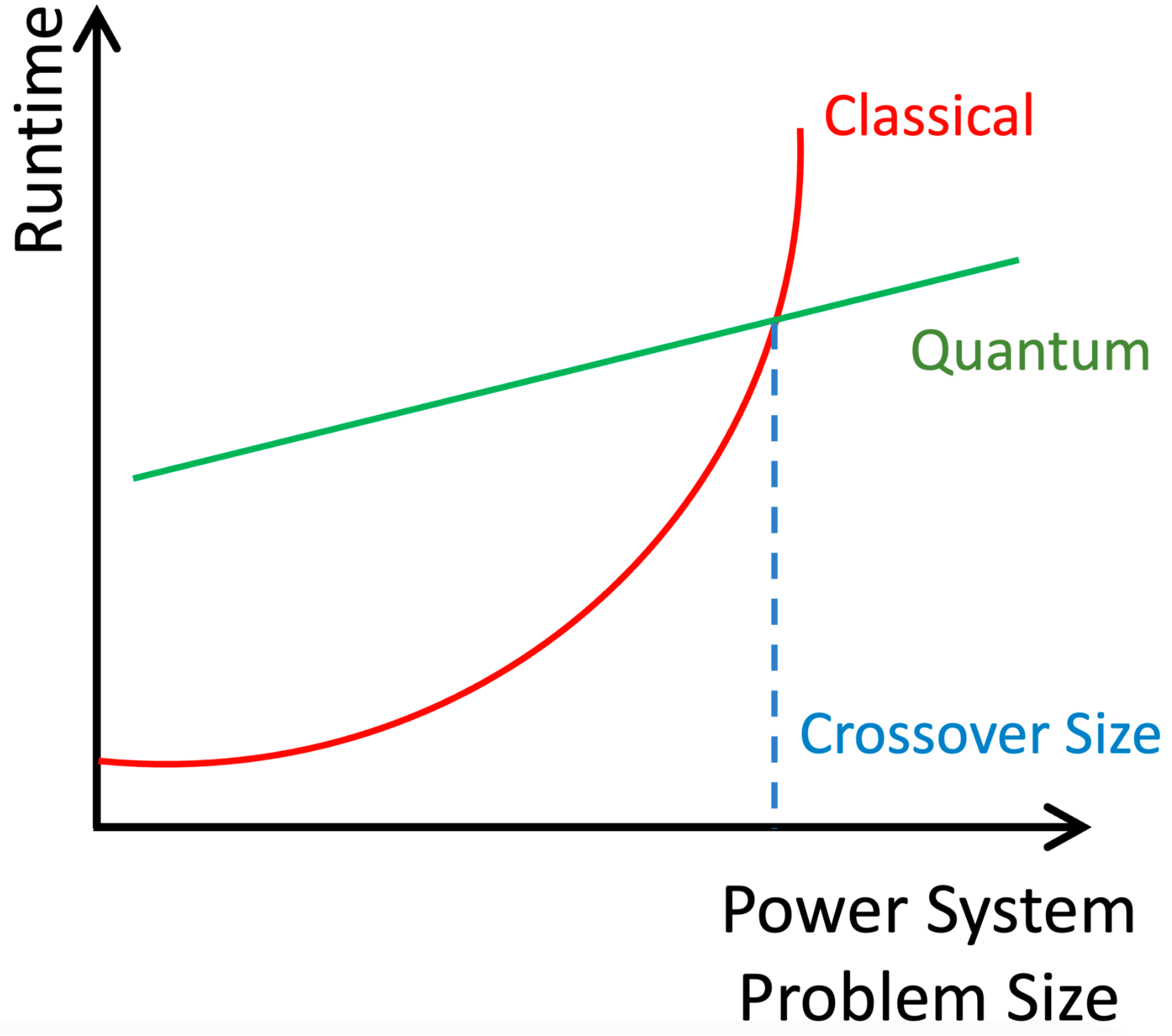
What is this talk about?

What is this talk about?

Trying to find out If there is a **Potential** for Quantum Advantage for **My Favourite Problem**

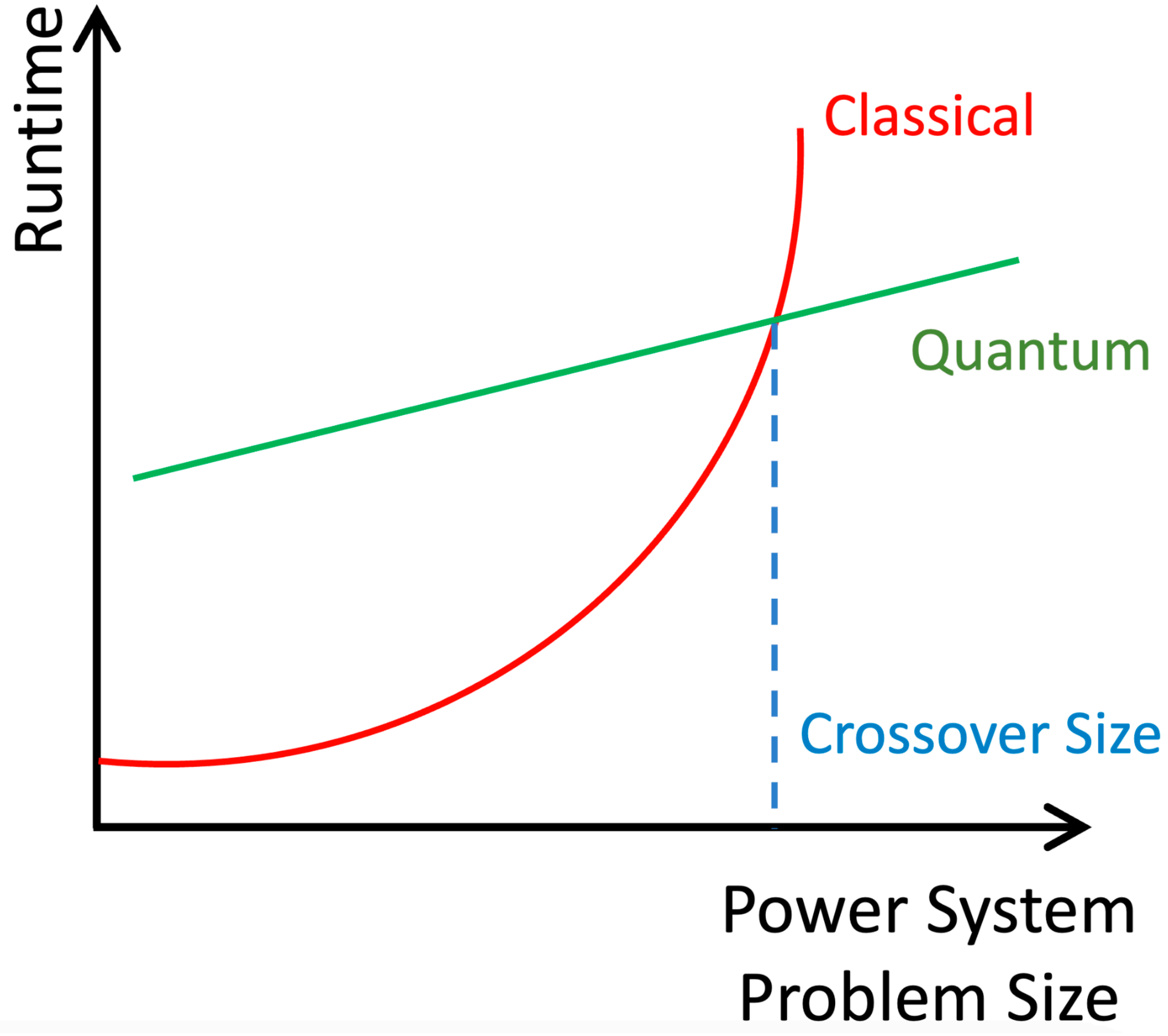
What is this talk about?

Trying to find out If there is a **Potential** for Quantum Advantage for **My Favourite Problem**



What is this talk about?

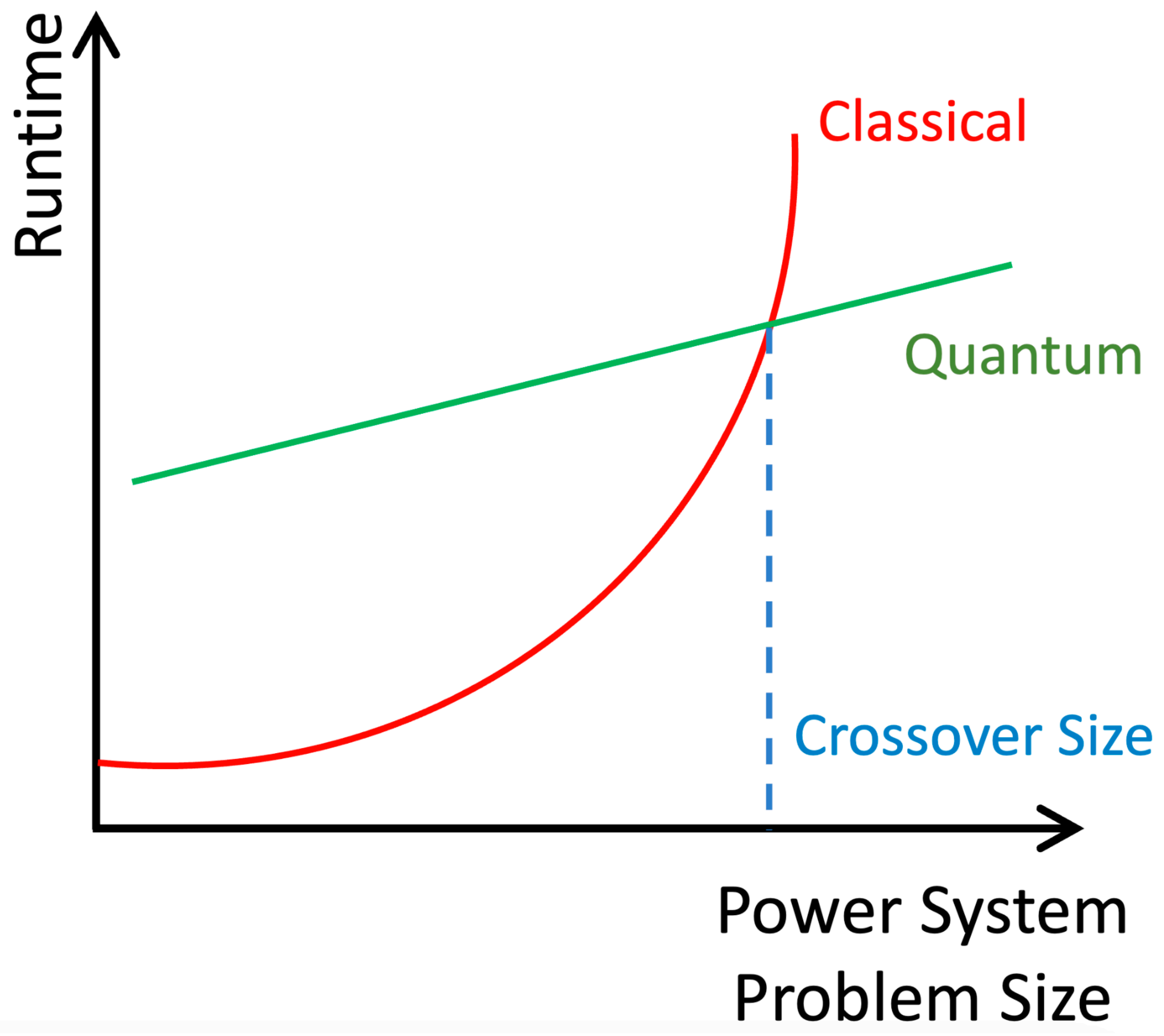
Trying to find out If there is a **Potential** for Quantum Advantage for **My Favourite Problem**



Potential Quantum Advantage

What is this talk about?

Trying to find out If there is a **Potential** for Quantum Advantage for **My Favourite Problem**



Target

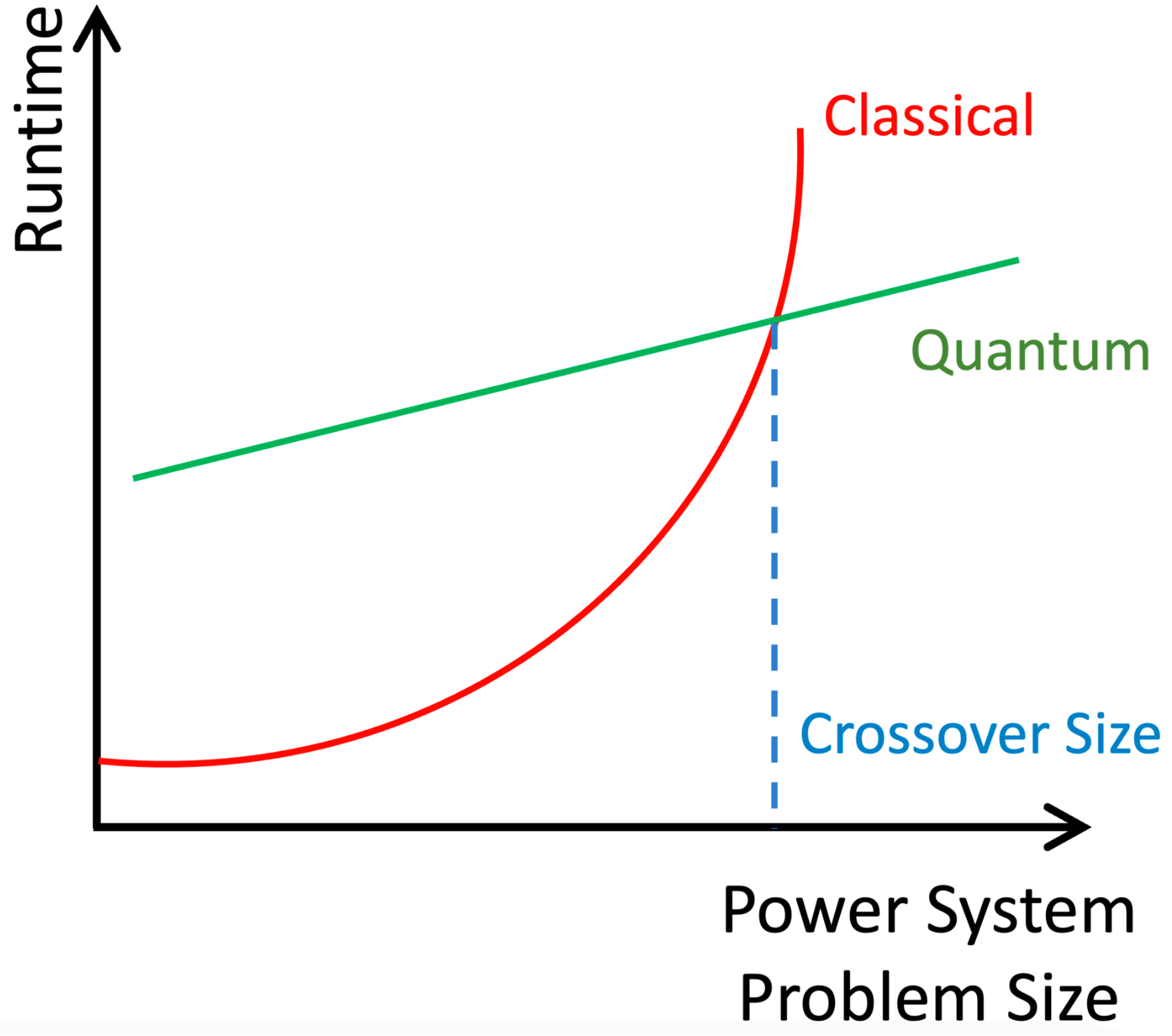
Find Crossover Size (if exist) &

Make this Graph

What is this talk about?

Trying to find out If there is a **Potential** for Quantum Advantage for

My Favourite Problem



Target

Find Crossover Size (if exist) &

Make this Graph

This talk is **NOT** about proposing a 'New' Quantum algorithm & I am **NOT** a Quantum Guy

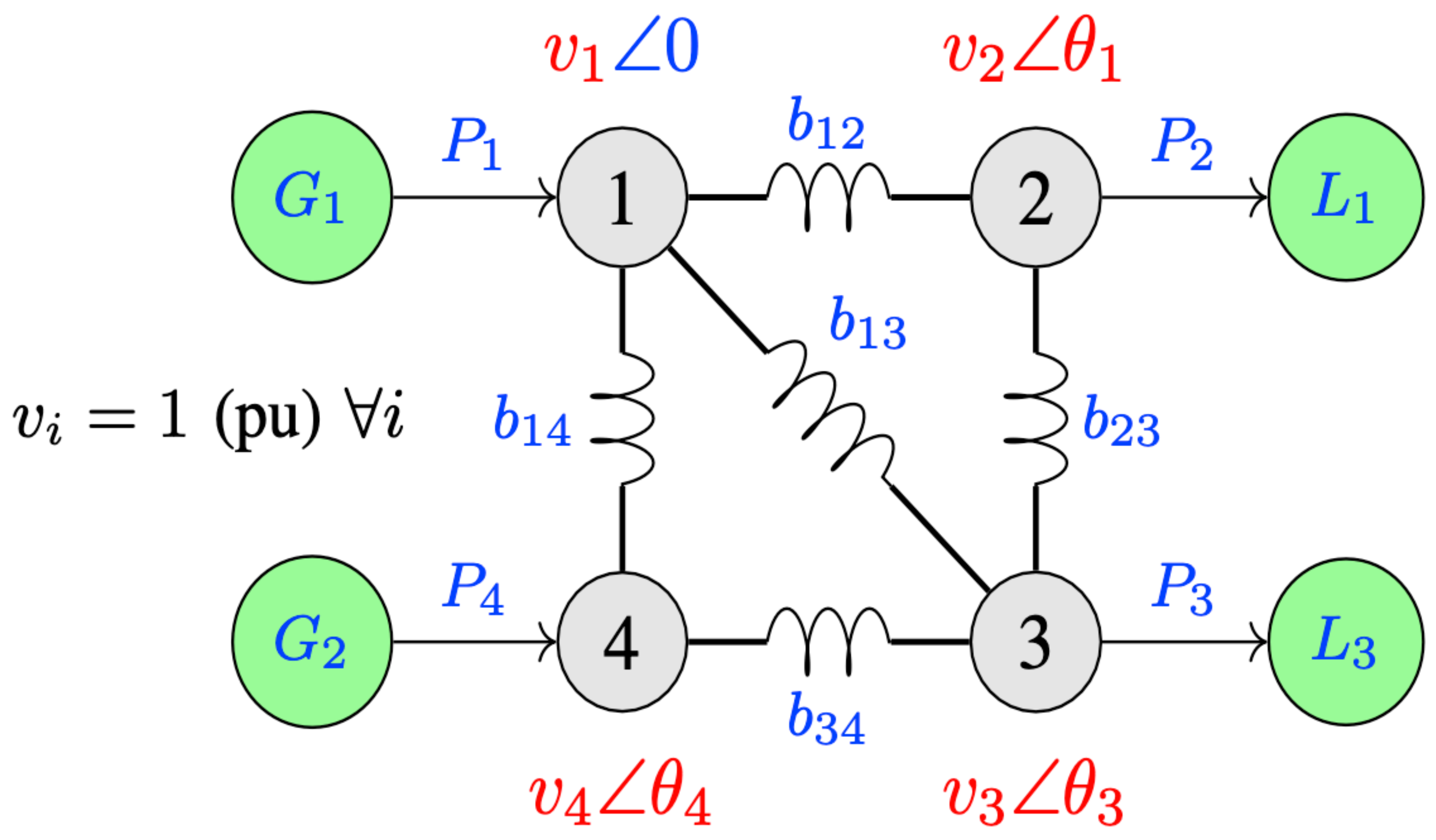
My Favourite Problem: Power Flow

My Favourite Problem: Power Flow

Given a power network and load inputs, find out state of the system i.e. Nodal Parameters

My Favourite Problem: Power Flow

Given a power network and load inputs, find out state of the system i.e. Nodal Parameters



$$\begin{bmatrix} \overline{b_{2j}} & -b_{23} & -b_{24} \\ -b_{32} & \overline{b_{3j}} & -b_{34} \\ -b_{42} & -b_{43} & \overline{b_{4j}} \end{bmatrix} \begin{bmatrix} \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} P_2 \\ P_3 \\ P_4 \end{bmatrix}$$

Angle Variable Vector

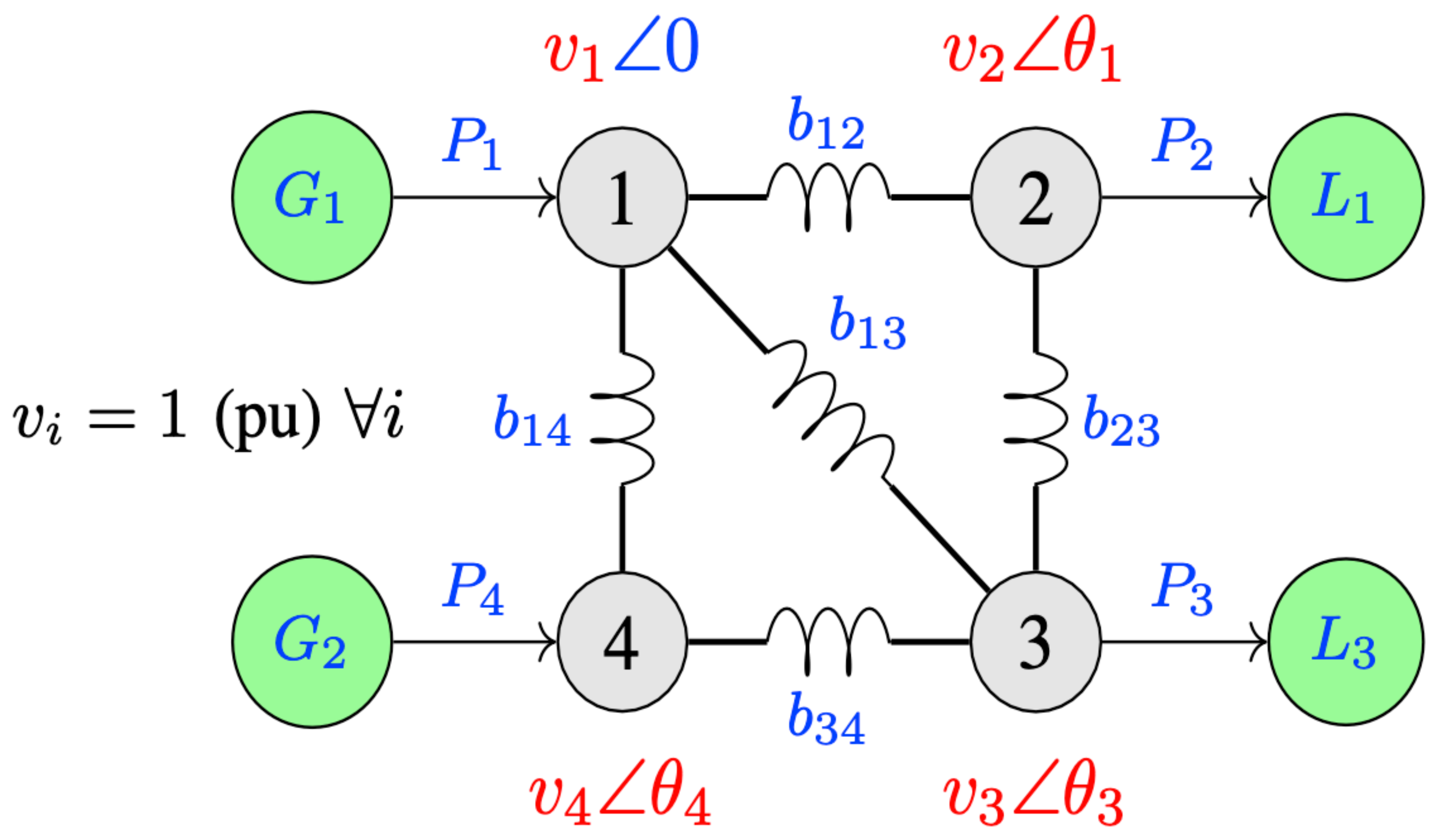
Injection Vector

Susceptance Matrix

$$A\,\mathbf{x} = \mathbf{b}$$

My Favourite Problem: Power Flow

Given a power network and load inputs, find out state of the system i.e. Nodal Parameters



$$\begin{bmatrix} \overline{b_{2j}} & -b_{23} & -b_{24} \\ -b_{32} & \overline{b_{3j}} & -b_{34} \\ -b_{42} & -b_{43} & \overline{b_{4j}} \end{bmatrix} \begin{bmatrix} \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} P_2 \\ P_3 \\ P_4 \end{bmatrix}$$

Angle Variable Vector

Injection Vector

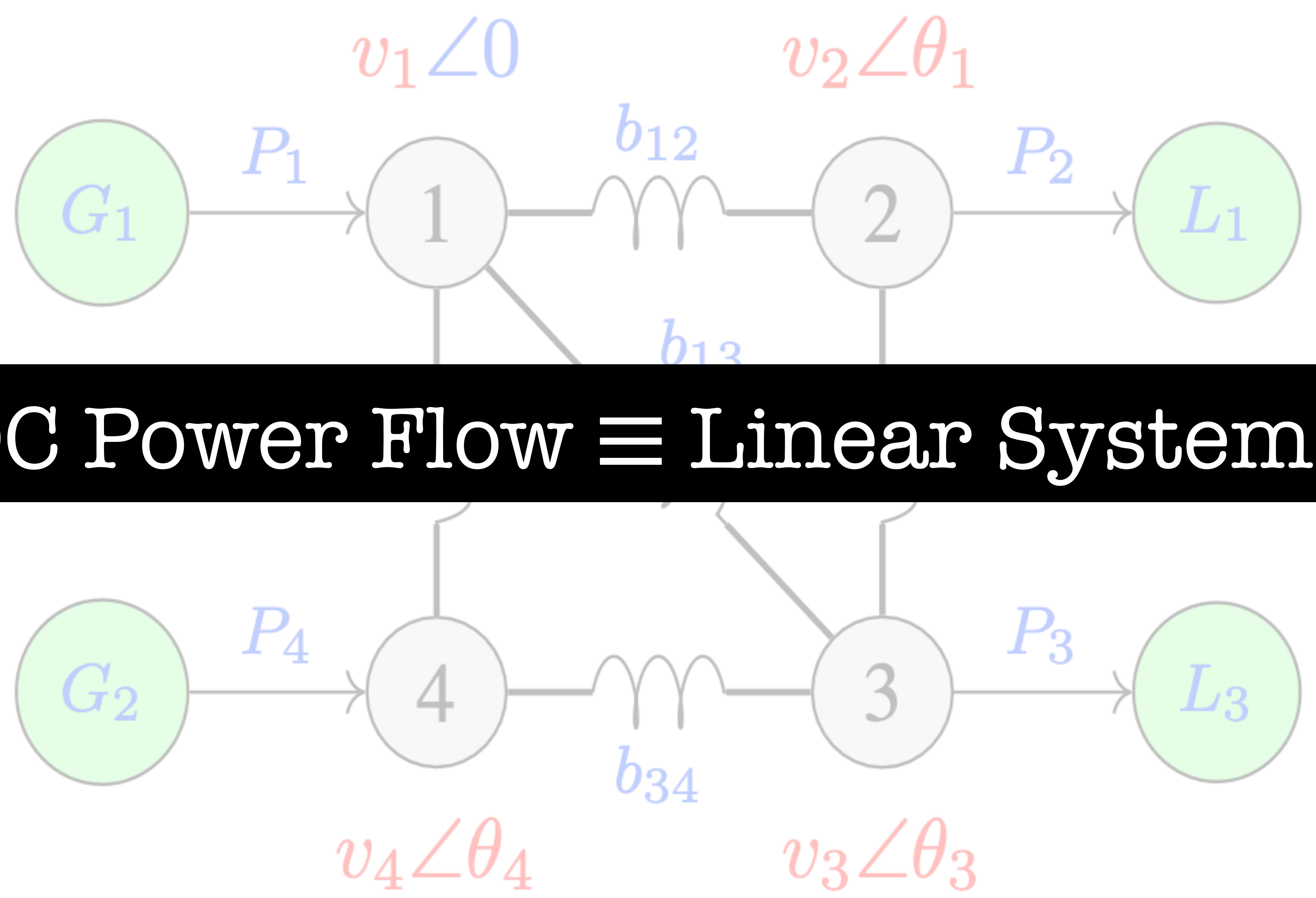
Susceptance Matrix

$$A\,\mathbf{x} = \mathbf{b}$$

So in DC Power Flow formulation we want to solve for voltage angles at each node of the system, with reference set to zero.

My Favourite Problem: Power Flow

Given a power network and load inputs, find out state of the system i.e. Nodal Parameters



DC Power Flow ≡ Linear System Solve

$$\begin{bmatrix} \overline{b_{2j}} & -b_{23} & -b_{24} \\ -b_{32} & \overline{b_{3j}} & -b_{34} \\ -b_{42} & -b_{43} & \overline{b_{4j}} \end{bmatrix} \begin{bmatrix} \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} P_2 \\ P_3 \\ P_4 \end{bmatrix}$$

$$\overset{\text{Susceptance Matrix}}{A}\,\overset{\text{Angle Variable Vector}}{\mathbf{x}} = \overset{\text{Injection Vector}}{\mathbf{b}}$$

So in DC Power Flow formulation we want to solve for voltage angles at each node of the system, with reference set to zero.

So where does Quantum fit in?

So where does Quantum fit in?

Using Quantum linear system solving algorithms for DCPF

So where does Quantum fit in?

Using Quantum linear system solving algorithms for DCPF

Classical State of Art:

Conjugate Gradient (CG) for Linear System Solving which exploits sparsity of network, and terminates at selected precision

So where does Quantum fit in?

Using Quantum linear system solving algorithms for DCPF

Classical State of Art:

Conjugate Gradient (CG) for Linear System Solving which exploits sparsity of network, and terminates at selected precision

$$\mathcal{O}(N \; s \; \sqrt{\kappa} \; \log(1/\varepsilon_c))$$

So where does Quantum fit in?

Using Quantum linear system solving algorithms for DCPF

Classical State of Art:

Conjugate Gradient (CG) for Linear System Solving which exploits sparsity of network, and terminates at selected precision

$$\mathcal{O}(\underbrace{N}_{\text{System Size}} \; s \; \sqrt{\kappa} \; \log(1/\varepsilon_c))$$

So where does Quantum fit in?

Using Quantum linear system solving algorithms for DCPF

Classical State of Art:

Conjugate Gradient (CG) for Linear System Solving which exploits sparsity of network, and terminates at selected precision

$$\mathcal{O}(N \; s \; \sqrt{\kappa} \; \log(1/\varepsilon_c))$$

System Size

Sparsity

So where does Quantum fit in?

Using Quantum linear system solving algorithms for DCPF

Classical State of Art:

Conjugate Gradient (CG) for Linear System Solving which exploits sparsity of network, and terminates at selected precision

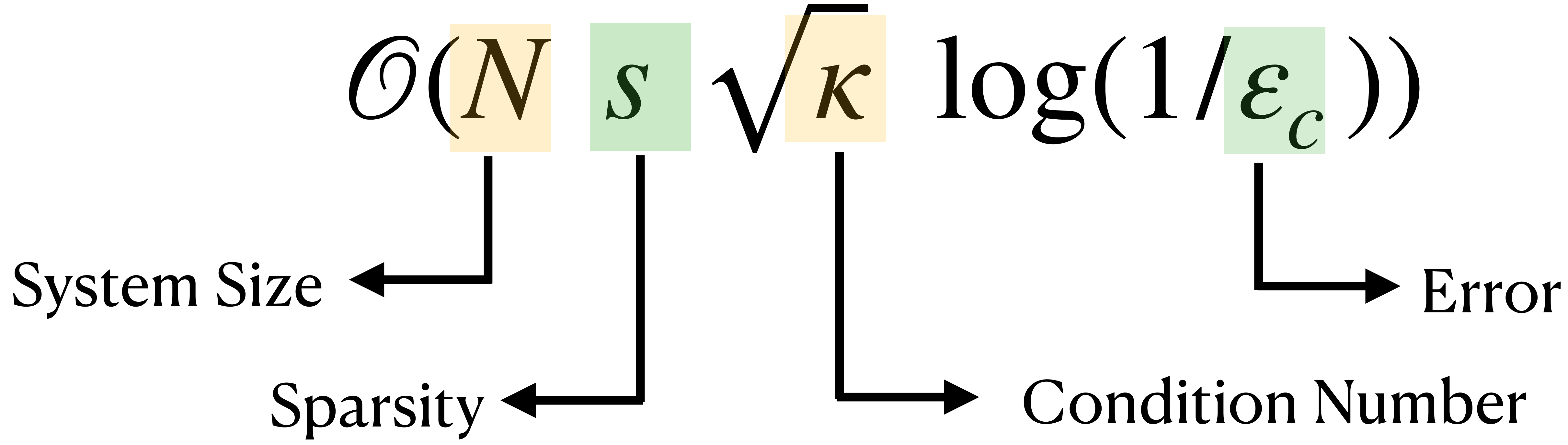
$$\mathcal{O}(\underbrace{N}_{\text{System Size}}\ \underbrace{s}_{\text{Sparsity}}\ \sqrt{\underbrace{\kappa}_{\text{Condition Number}}}\ \log(1/\varepsilon_c))$$

So where does Quantum fit in?

Using Quantum linear system solving algorithms for DCPF

Classical State of Art:

Conjugate Gradient (CG) for Linear System Solving which exploits sparsity of network, and terminates at selected precision



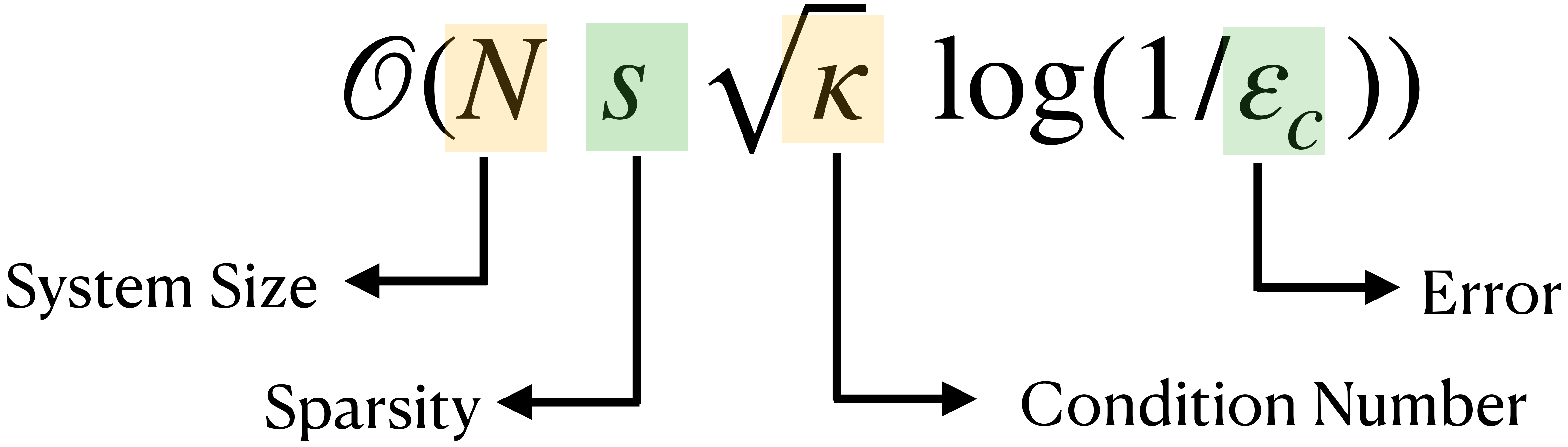
$$\mathcal{O}(N\ s\ \sqrt{\kappa}\ \log(1/\varepsilon_c))$$

So where does Quantum fit in?

Using Quantum linear system solving algorithms for DCPF

Classical State of Art:

Conjugate Gradient (CG) for Linear System Solving which exploits sparsity of network, and terminates at selected precision



$$\mathcal{O}(N\ s\ \sqrt{\kappa}\ \log(1/\varepsilon_c))$$

Quantum Power Flow Claim!

Solving DCPF using Harrow-Hassidim-Lloyd (HHL) algorithm will lead to **Exponential** speed up

So where does Quantum fit in?

Using Quantum linear system solving algorithms for DCPF

Classical State of Art:

Conjugate Gradient (CG) for Linear System Solving which exploits sparsity of network, and terminates at selected precision

$$\mathcal{O}(\underbrace{N}_{\text{System Size}}\ \underbrace{s}_{\text{Sparsity}}\ \sqrt{\underbrace{\kappa}_{\text{Condition Number}}}\ \log(1/\underbrace{\varepsilon_c}_{\text{Error}}))$$

Quantum Power Flow Claim!

Solving DCPF using Harrow-Hassidim-Lloyd (HHL) algorithm will lead to **Exponential** speed up

$$\mathcal{O}(\log N)$$

But there is something missing in $\mathcal{O}(\log N)$

But there is something missing in $\mathcal{O}(\log N)$

Where are the other parameters like condition number and sparsity?

But there is something missing in $\mathcal{O}(\log N)$

Where are the other parameters like condition number and sparsity?

Complete Solve Complexity of HHL $\longrightarrow$ $\mathcal{O}(s^2 \ \kappa^2 \ \log N \ \varepsilon^{-1})$

But there is something missing in $\mathcal{O}(\log N)$

Where are the other parameters like condition number and sparsity?

Complete Solve Complexity of HHL ⟶ $\mathcal{O}(s^2 \;\; \kappa^2 \;\; \log N \; \varepsilon^{-1})$

HHL does scale better in terms of **System Size**, but scales worse in terms of **Condition Number** (& Sparsity)

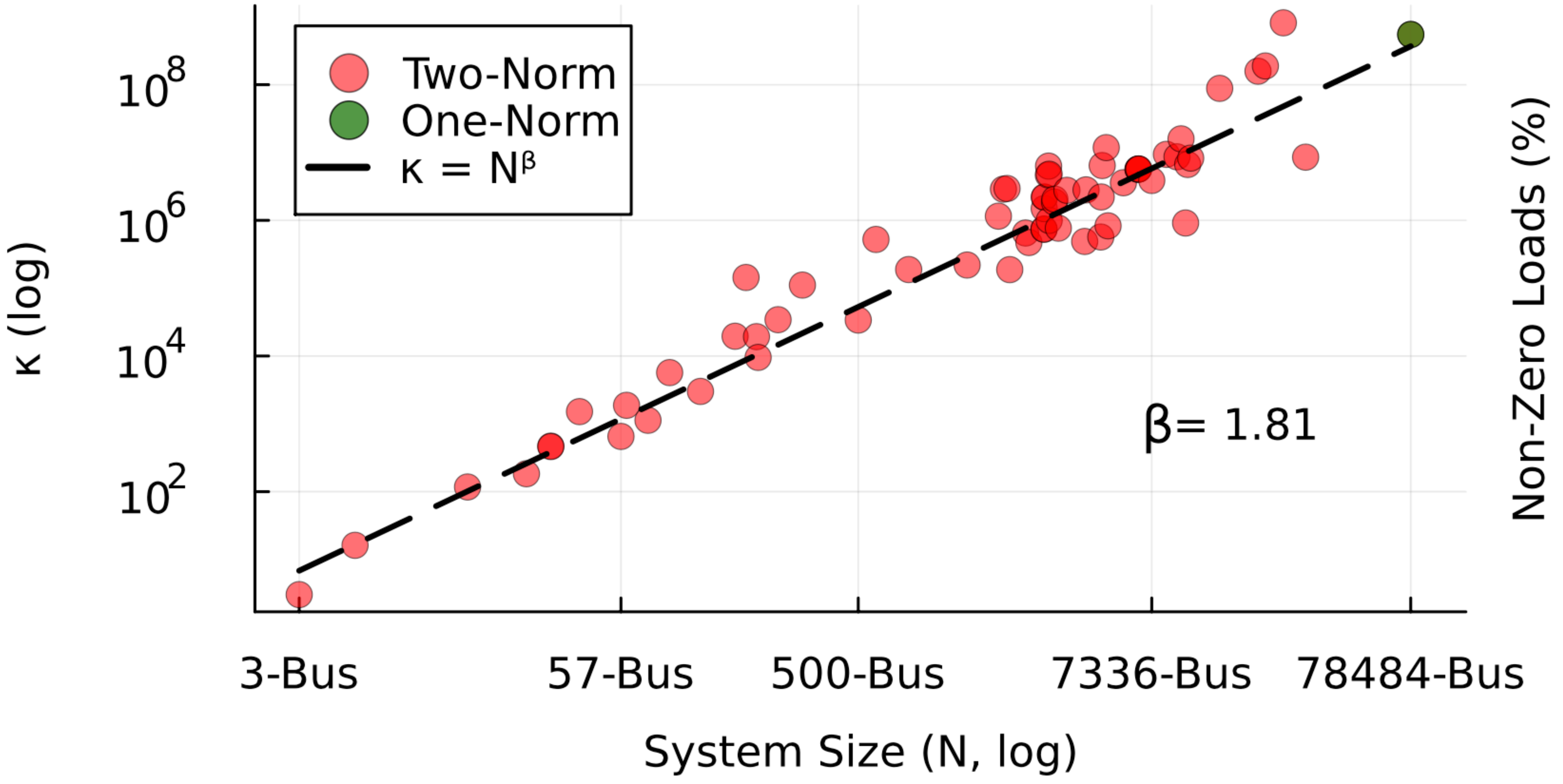
But there is something missing in $\mathcal{O}(\log N)$

Where are the other parameters like condition number and sparsity?

Complete Solve Complexity of HHL $\longrightarrow$ $\mathcal{O}(s^2 \ \kappa^2 \ \log N \ \varepsilon^{-1})$

HHL does scale better in terms of **System Size**, but scales worse in terms of **Condition Number** (& Sparsity)

Scaling of condition number(k) as a function of buses (N) for the PGLib-OPF datasets



But there is something missing in $\mathcal{O}(\log N)$

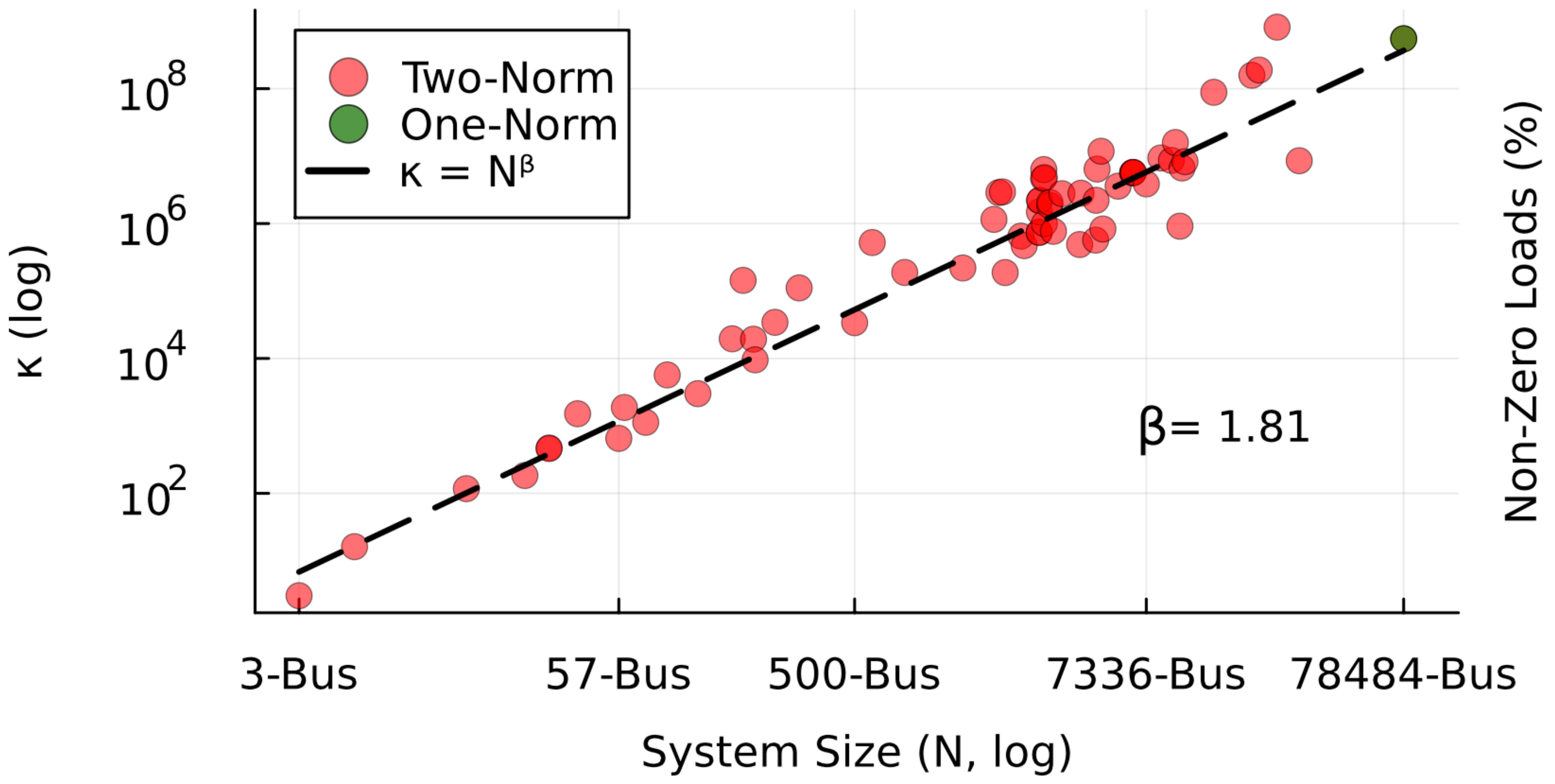
Where are the other parameters like condition number and sparsity?

Complete Solve Complexity of HHL ⟶ $\mathcal{O}(s^2 \ \kappa^2 \ \log N \ \varepsilon^{-1})$

HHL does scale better in terms of **System Size**, but scales worse in terms of **Condition Number** (& Sparsity)

Condition number is scaling worse than N

Algorithms that manage condition number will be better for this application.



Point #1:

During speedup analysis consider runtime complexity with respect to ALL Parameters

Our Favorite Problem might not have so Favorable Parameters

Overheads: Data Loading

Overheads: Data Loading

$$\textsc{Classical Input} \xRightarrow{\textsc{Classical Algorithm}} \textsc{Classical Output}$$

$$\textsc{Classical Input} \xRightarrow{\textsc{State Prep.}} \left|\textsc{Input}\right\rangle \xRightarrow{\textsc{Quantum Algorithm}} \left|\textsc{Output}\right\rangle \xRightarrow{\textsc{Readout}} \textsc{Classical Output}$$

Overheads: Data Loading

$$\begin{tikzcd}
\mathcal{O}(1) & & \\
\textsc{Classical Input} \arrow[u] \arrow[rr, Rightarrow, "\textsc{Classical Algorithm}"] & & \textsc{Classical Output}
\end{tikzcd}$$

$$\textsc{Classical Input} \xRightarrow{\textsc{State Prep.}} |\textsc{Input}\rangle \xRightarrow{\textsc{Quantum Algorithm}} |\textsc{Output}\rangle \xRightarrow{\textsc{Readout}} \textsc{Classical Output}$$

Overheads: Data Loading

$$\begin{tikzcd}
\mathcal{O}(1) & & \mathcal{O}(1) \\
\textsc{Classical Input} \arrow[u] \arrow[rr, Rightarrow, "\textsc{Classical Algorithm}"] & & \textsc{Classical Output} \arrow[u]
\end{tikzcd}$$

$$\textsc{Classical Input} \xRightarrow{\textsc{State Prep.}} |\textsc{Input}\rangle \xRightarrow{\textsc{Quantum Algorithm}} |\textsc{Output}\rangle \xRightarrow{\textsc{Readout}} \textsc{Classical Output}$$

Overheads: Data Loading

$$\begin{tikzcd}
\mathcal{O}(1) & & \mathcal{O}(1) \\
\textsc{Classical Input} \arrow[u] \arrow[rr, Rightarrow, "\textsc{Classical Algorithm}"] & & \textsc{Classical Output} \arrow[u]
\end{tikzcd}$$

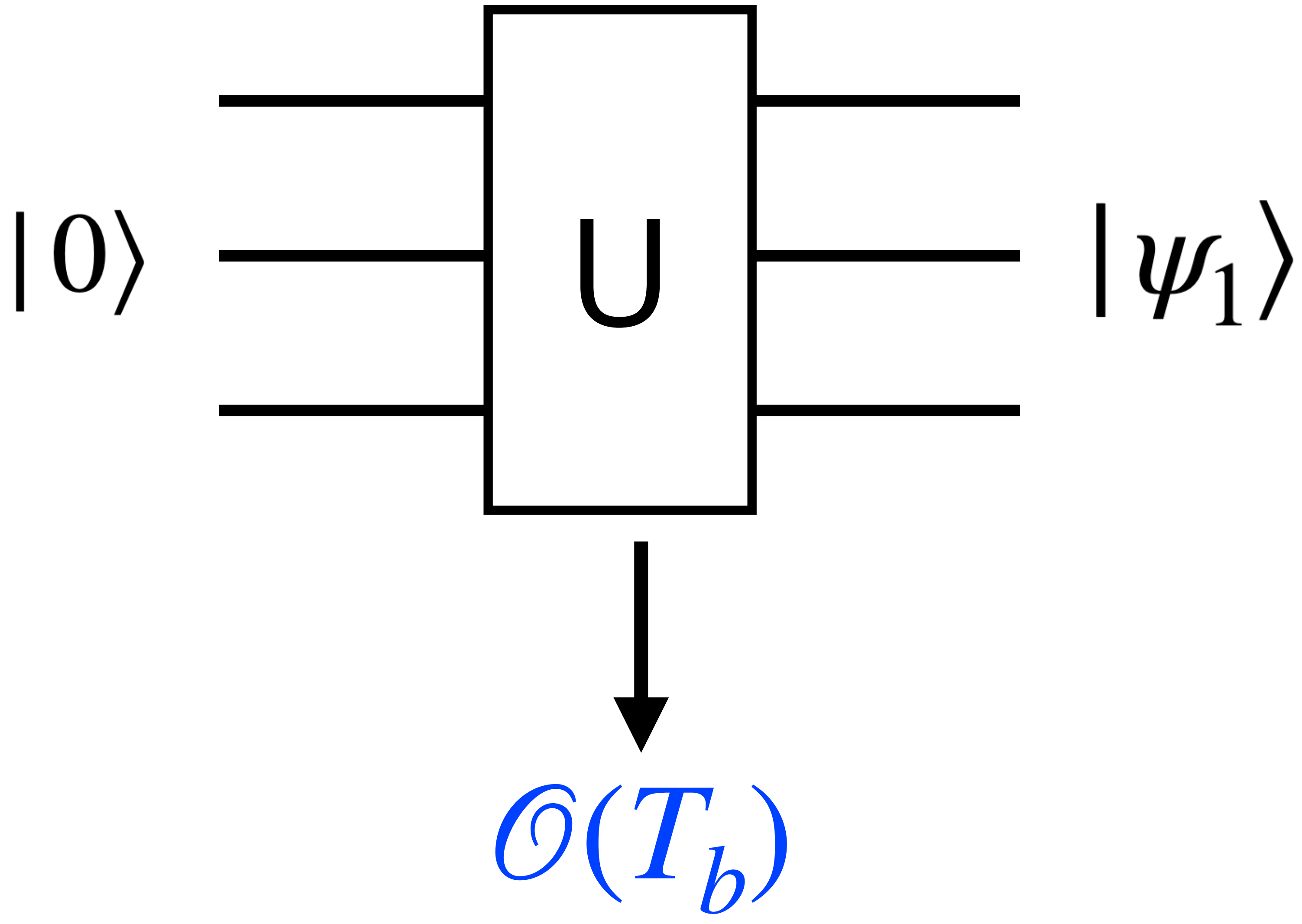
$$\begin{tikzcd}
\textsc{Classical Input} \arrow[rr, Rightarrow, "\textsc{State Prep.}"] & {} \arrow[d] & \lvert \textsc{Input} \rangle \arrow[r, Rightarrow, "\textsc{Quantum Algorithm}"] & \lvert \textsc{Output} \rangle \arrow[rr, Rightarrow, "\textsc{Readout}"] & {} \arrow[d] & \textsc{Classical Output} \\
 & ? & & & ? &
\end{tikzcd}$$

Overheads: Data Loading

$$\begin{tikzcd} \mathcal{O}(1) & & \mathcal{O}(1) \\ \textsc{Classical Input} \arrow[u] \arrow[rr, Rightarrow, "\textsc{Classical Algorithm}"] & & \textsc{Classical Output} \arrow[u] \end{tikzcd}$$

$$\textsc{Classical Input} \xRightarrow{\textsc{State Prep.}} |\textsc{Input}\rangle \xRightarrow{\textsc{Quantum Algorithm}} |\textsc{Output}\rangle \xRightarrow{\textsc{Readout}} \textsc{Classical Output}$$

State Prepration:

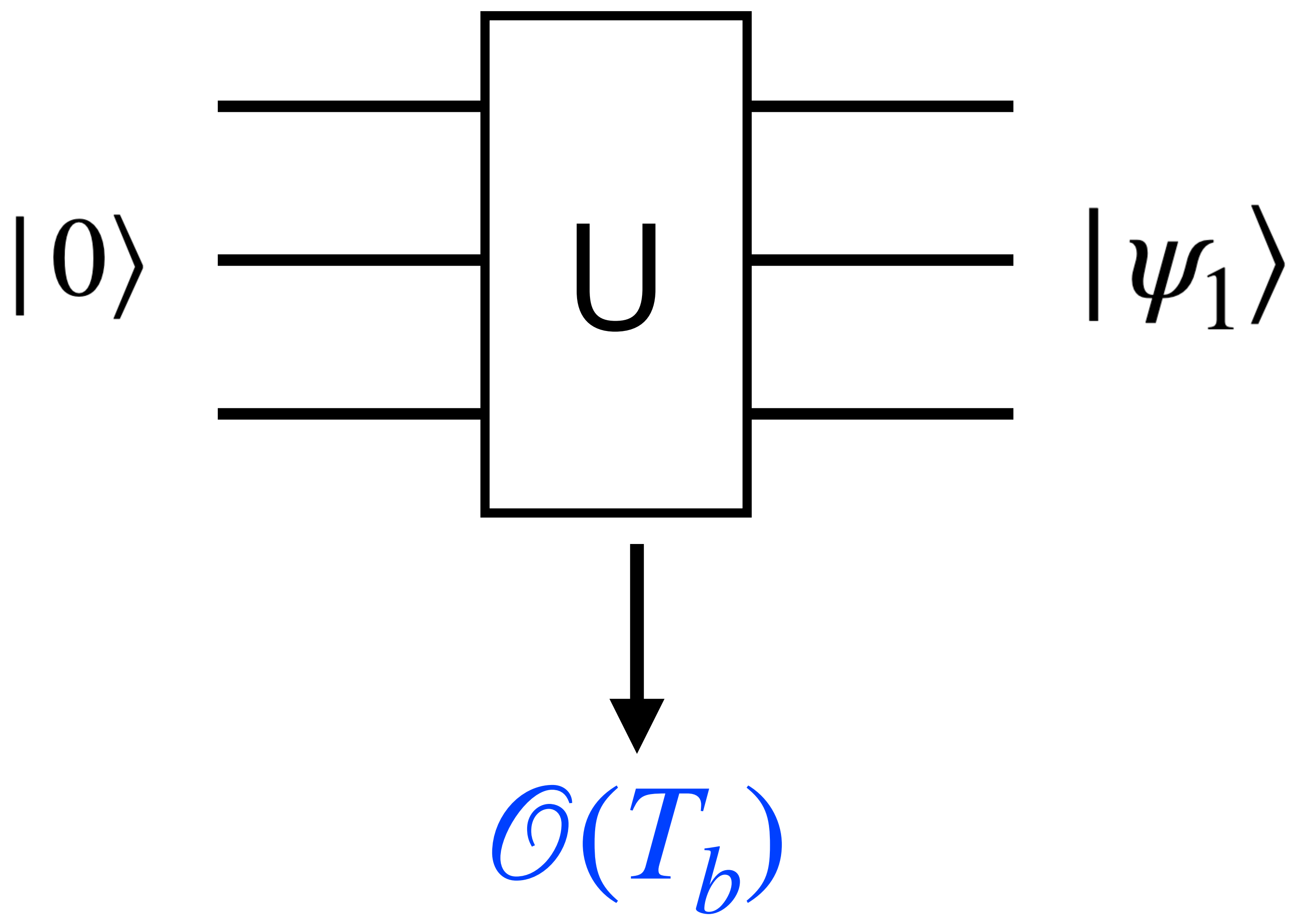


Overheads: Data Loading

$$\begin{tikzcd} \mathcal{O}(1) & & \mathcal{O}(1) \\ \text{Classical Input} \arrow[u] \arrow[rr, Rightarrow, "\text{Classical Algorithm}"] & & \text{Classical Output} \arrow[u] \end{tikzcd}$$

$$\text{Classical Input} \xRightarrow{\text{State Prep.}} |\text{Input}\rangle \xRightarrow{\text{Quantum Algorithm}} |\text{Output}\rangle \xRightarrow{\text{Readout}} \text{Classical Output}$$

State Prepration:



To Load $N-$Dimensional b vector

Amplitude Encoding → $\mathcal{O}(N)$

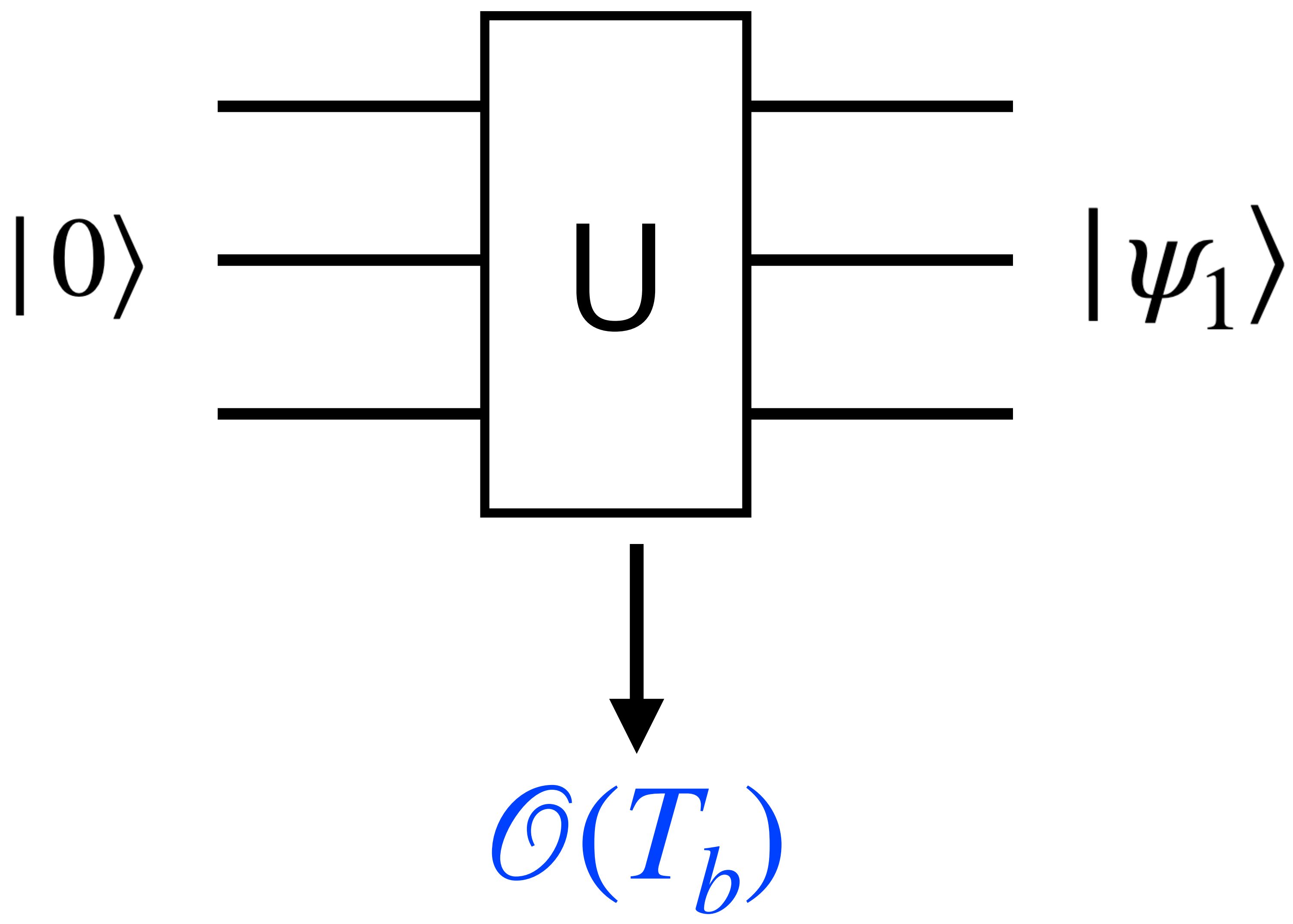
With Quantum RAM → $\mathcal{O}(\log N)$

Overheads: Data Loading

$$\begin{tikzcd}
\mathcal{O}(1) & & \mathcal{O}(1) \\
\textsc{Classical Input} \arrow[u] \arrow[rr, Rightarrow, "\textsc{Classical Algorithm}"] & & \textsc{Classical Output} \arrow[u]
\end{tikzcd}$$

$$\begin{tikzcd}
\textsc{Classical Input} \arrow[r, Rightarrow, "\textsc{State Prep.}"] & \lvert \textsc{Input} \rangle \arrow[r, Rightarrow, "\textsc{Quantum Algorithm}"] & \lvert \textsc{Output} \rangle \arrow[r, Rightarrow, "\textsc{Readout}"] & \textsc{Classical Output}
\end{tikzcd}$$

State Prepration:



To Load $N-$Dimensional b vector

Amplitude Encoding → $\mathcal{O}(N)$

With Quantum RAM → $\mathcal{O}(\log N)$

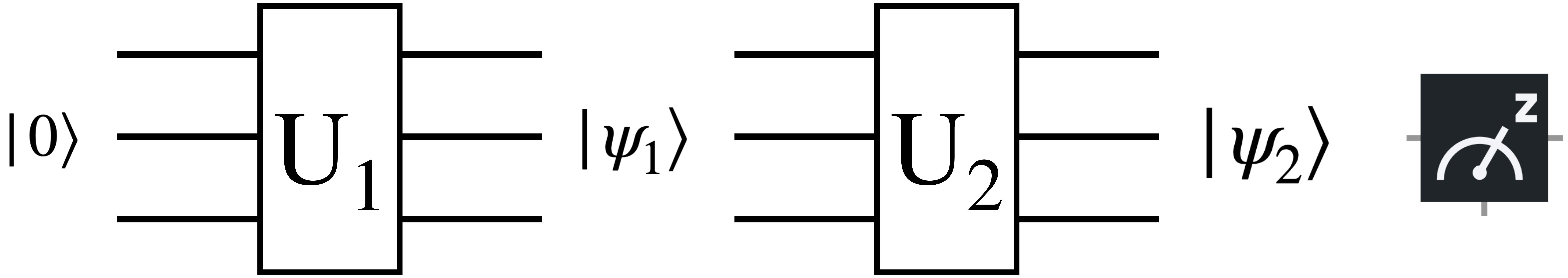
So Optimistically $\mathcal{O}(T_b) \equiv \mathcal{O}(\log N)$

Overheads: Readout

$$\textsc{Classical Input} \xRightarrow{\textsc{State Prep.}} \left|\textsc{Input}\right\rangle \xRightarrow{\textsc{Quantum Algorithm}} \left|\textsc{Output}\right\rangle \xRightarrow{\textsc{Readout}} \textsc{Classical Output}$$

Overheads: Readout

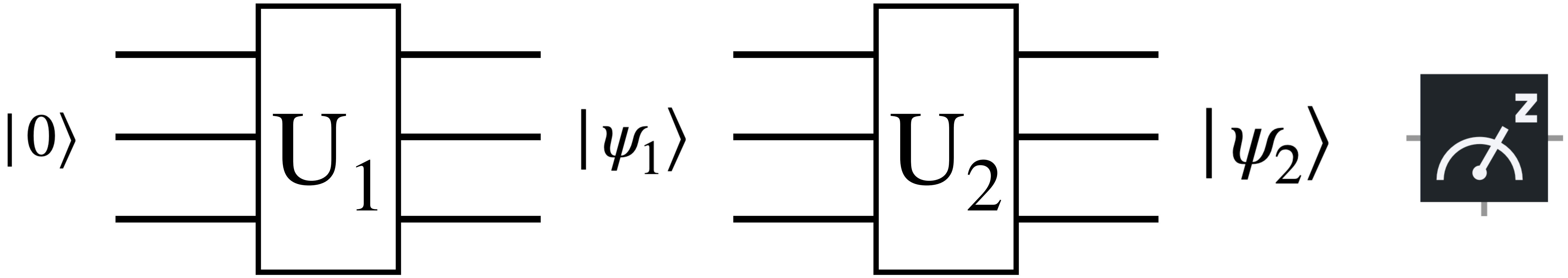
$$\text{Classical Input} \xRightarrow{\text{State Prep.}} |\text{Input}\rangle \xRightarrow{\text{Quantum Algorithm}} |\text{Output}\rangle \xRightarrow{\text{Readout}} \text{Classical Output}$$



Readout Complexity ?

Overheads: Readout

$$\text{Classical Input} \xrightarrow{\text{State Prep.}} |\text{Input}\rangle \xrightarrow{\text{Quantum Algorithm}} |\text{Output}\rangle \xrightarrow{\text{Readout}} \text{Classical Output}$$



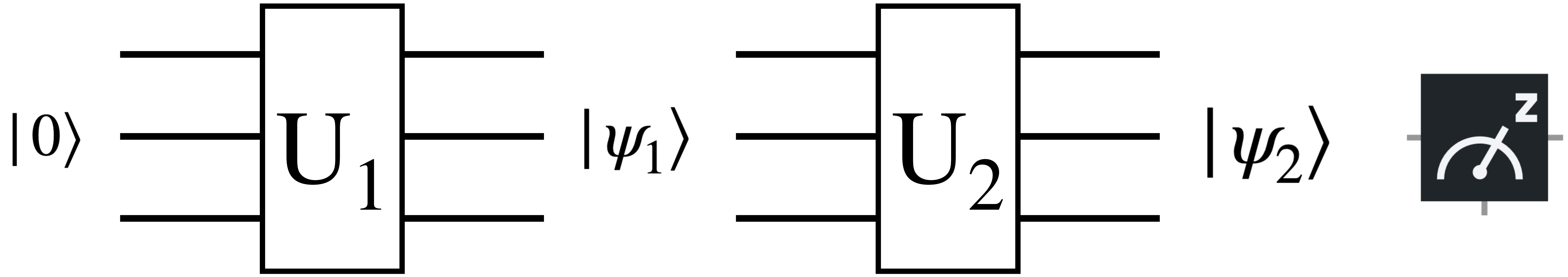
Readout Complexity ?

1. How much effort in reading ?

2. How much to read ?

Overheads: Readout

$$\text{CLASSICAL INPUT} \xRightarrow{\text{STATE PREP.}} |\text{INPUT}\rangle \xRightarrow{\text{QUANTUM ALGORITHM}} |\text{OUTPUT}\rangle \xRightarrow{\text{READOUT}} \text{CLASSICAL OUTPUT}$$

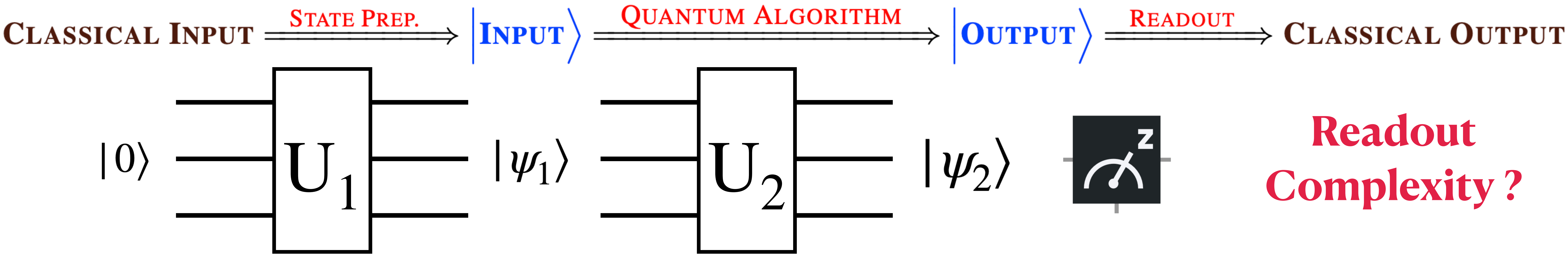


Readout Complexity ?

1. How much effort in reading ?

2. How much to read ? →

Overheads: Readout



Readout Complexity ?

1. How much effort in reading ?
2. How much to read ?

→

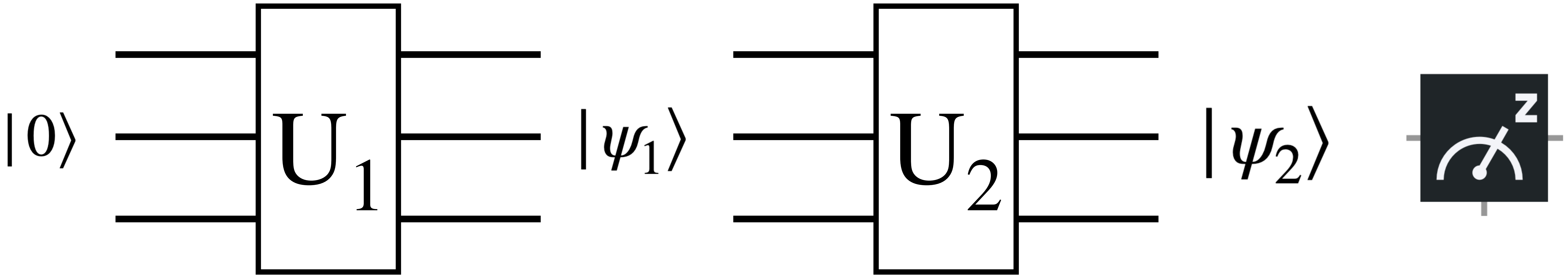
In Power Flow Problem we want to complete state of the system so

$N-$Dimensional vector need to be read

$$\mathbf{x} = A^{-1}\mathbf{b}$$

Overheads: Readout

$$\textsc{Classical Input} \xRightarrow{\textsc{State Prep.}} |\textsc{Input}\rangle \xRightarrow{\textsc{Quantum Algorithm}} |\textsc{Output}\rangle \xRightarrow{\textsc{Readout}} \textsc{Classical Output}$$



Readout Complexity ?

1. How much effort in reading ?
2. How much to read ?

→ In Power Flow Problem we want to complete state of the system so

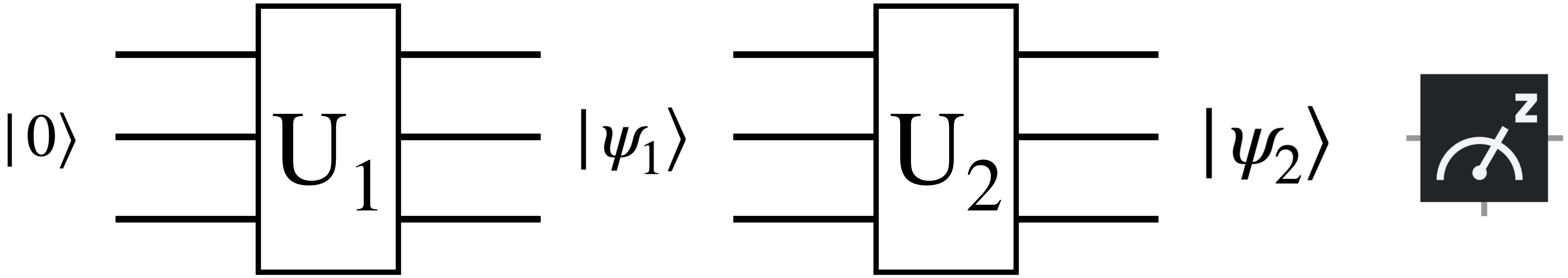
$N-$Dimensional vector need to be read

$$\mathbf{x} = A^{-1}\mathbf{b}$$

Reading amount will depend on **Your Favourite Problem's**

Overheads: Readout

$$\text{CLASSICAL INPUT} \xRightarrow{\text{STATE PREP.}} \left|\text{INPUT}\right\rangle \xRightarrow{\text{QUANTUM ALGORITHM}} \left|\text{OUTPUT}\right\rangle \xRightarrow{\text{READOUT}} \text{CLASSICAL OUTPUT}$$



Readout Complexity ?

1. How much effort in reading ?
2. How much to read ? →

In Power Flow Problem we want to complete state of the system so

$N-$Dimensional vector need to be read

$$\mathbf{x} = A^{-1}\mathbf{b}$$

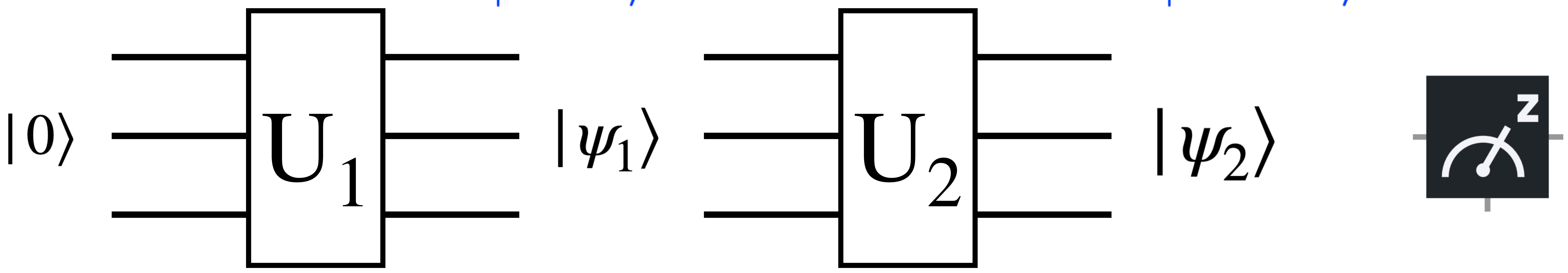
Reading amount will depend on **Your Favourite Problem's** — **Your Favourite Formulation**

Overheads: Readout

$$\text{Classical Input} \xRightarrow{\text{State Prep.}} |\text{Input}\rangle \xRightarrow{\text{Quantum Algorithm}} |\text{Output}\rangle \xRightarrow{\text{Readout}} \text{Classical Output}$$

Overheads: Readout

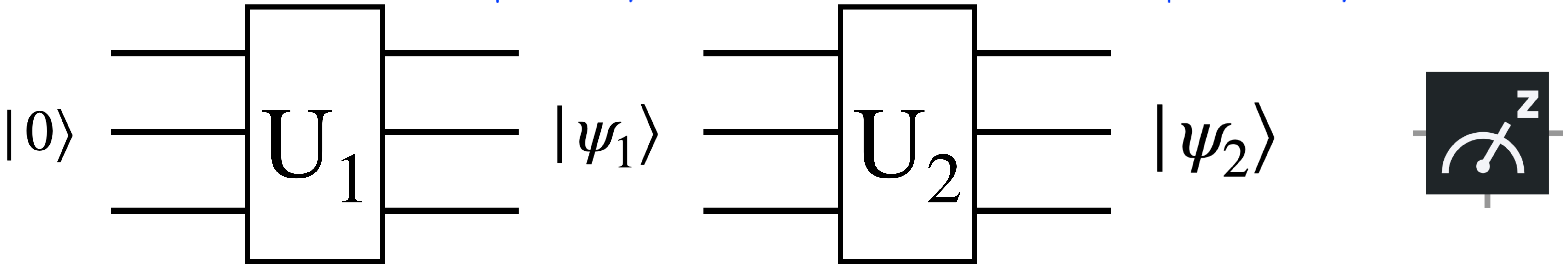
$$\text{Classical Input} \xRightarrow{\text{State Prep.}} |\text{Input}\rangle \xRightarrow{\text{Quantum Algorithm}} |\text{Output}\rangle \xRightarrow{\text{Readout}} \text{Classical Output}$$



Readout Complexity ?

Overheads: Readout

$$\text{Classical Input} \xRightarrow{\text{State Prep.}} \big|\text{Input}\big\rangle \xRightarrow{\text{Quantum Algorithm}} \big|\text{Output}\big\rangle \xRightarrow{\text{Readout}} \text{Classical Output}$$



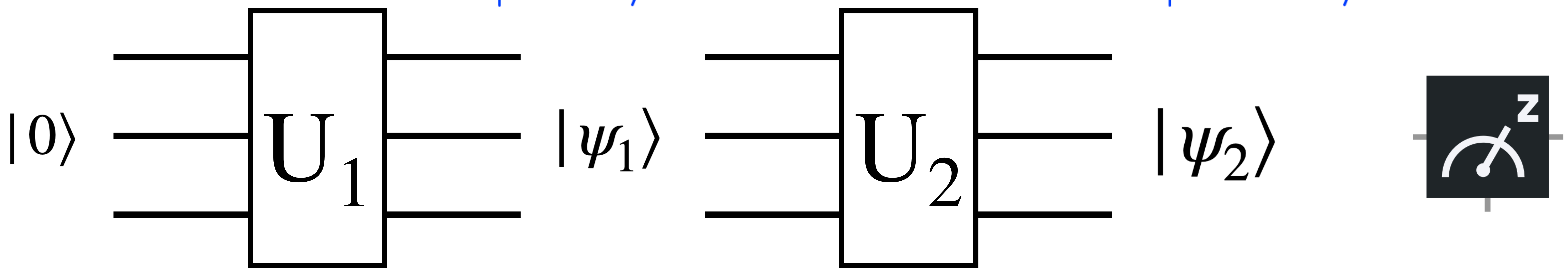
Readout Complexity ?

1. How much effort to read ?

2. How much to read ?

Overheads: Readout

$\text{CLASSICAL INPUT} \xRightarrow{\text{STATE PREP.}} |\text{INPUT}\rangle \xRightarrow{\text{QUANTUM ALGORITHM}} |\text{OUTPUT}\rangle \xRightarrow{\text{READOUT}} \text{CLASSICAL OUTPUT}$



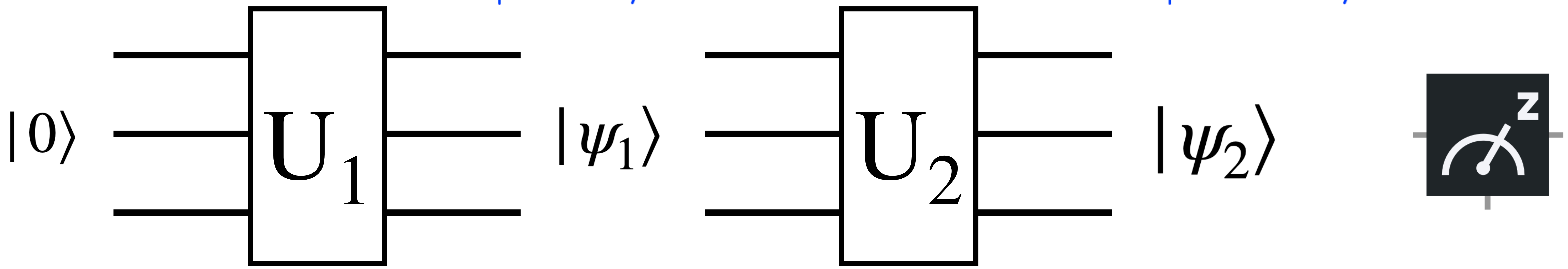
Readout Complexity ?

1. How much effort to read ? $\xrightarrow{|x\rangle = A^{-1}|b\rangle}$

2. How much to read ?

Overheads: Readout

$$\text{Classical Input} \xRightarrow{\text{State Prep.}} |\text{Input}\rangle \xRightarrow{\text{Quantum Algorithm}} |\text{Output}\rangle \xRightarrow{\text{Readout}} \text{Classical Output}$$



Readout Complexity ?

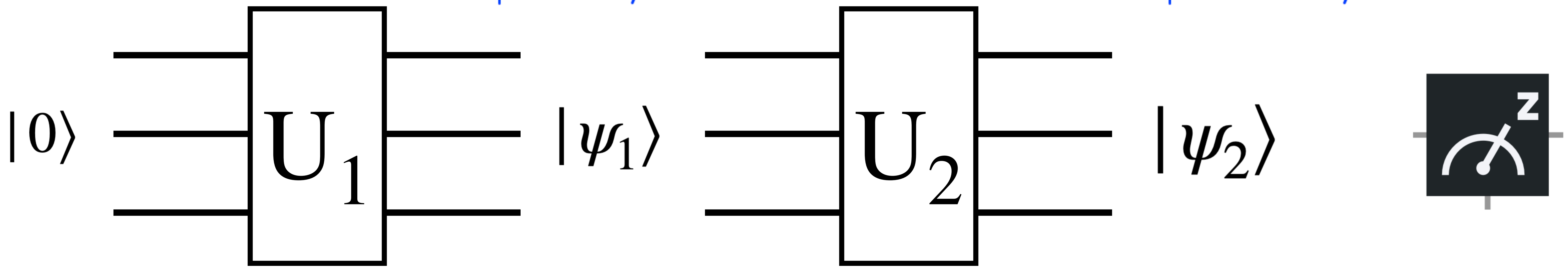
1. How much effort to read ?
2. How much to read ?

$$\xrightarrow{|x\rangle = A^{-1}|b\rangle}$$

To perform **quantum tomography** to get bus angle vector estimate, we need **multiple copies** of HHL solution $|x\rangle$.

Overheads: Readout

$\text{Classical Input} \xRightarrow{\text{State Prep.}} |\text{Input}\rangle \xRightarrow{\text{Quantum Algorithm}} |\text{Output}\rangle \xRightarrow{\text{Readout}} \text{Classical Output}$



Readout Complexity ?

1. How much effort to read ?

2. How much to read ?

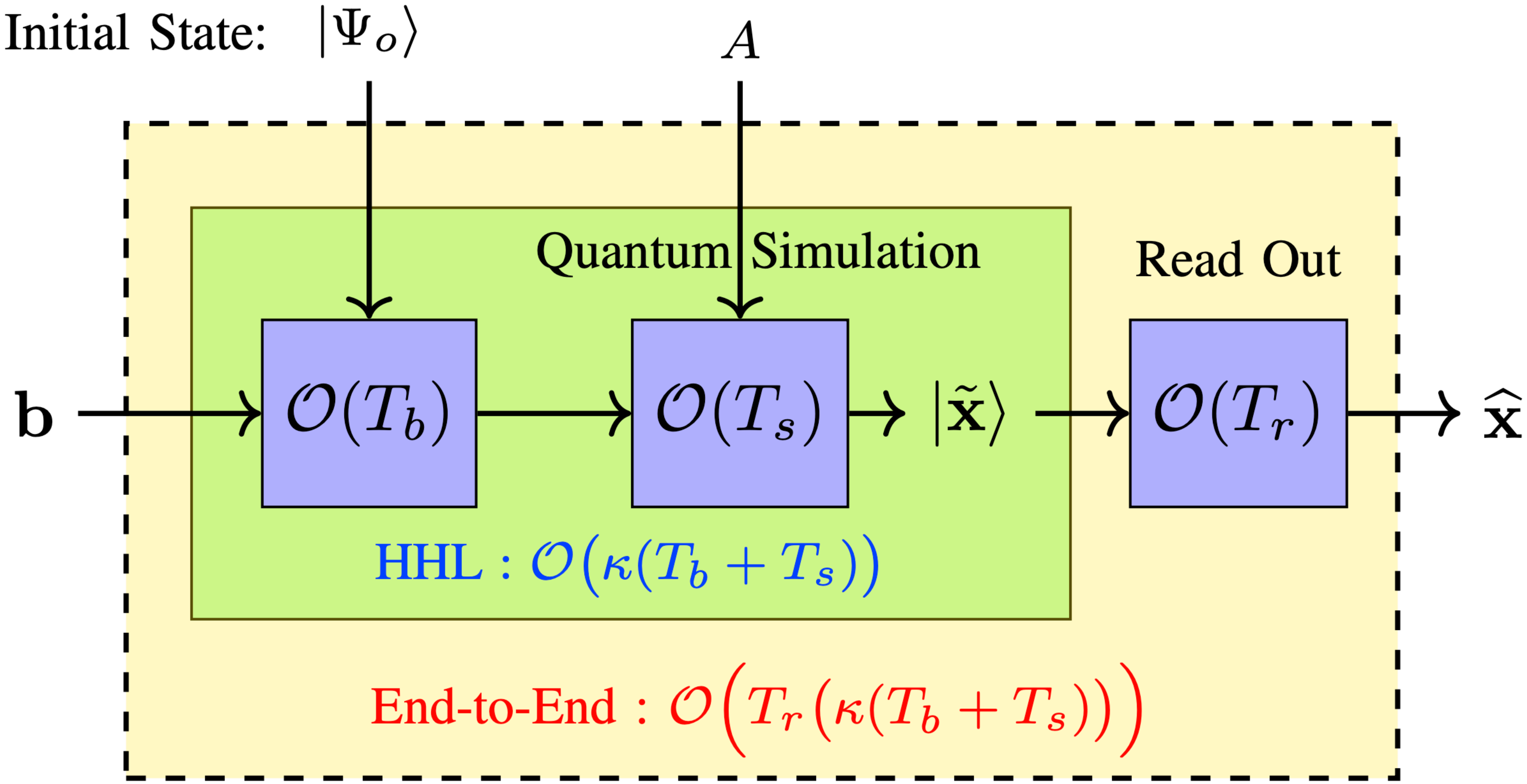
$\xrightarrow{|x\rangle = A^{-1}|b\rangle}$

To perform **quantum tomography** to get bus angle vector estimate, we need **multiple copies** of HHL solution $|x\rangle$.

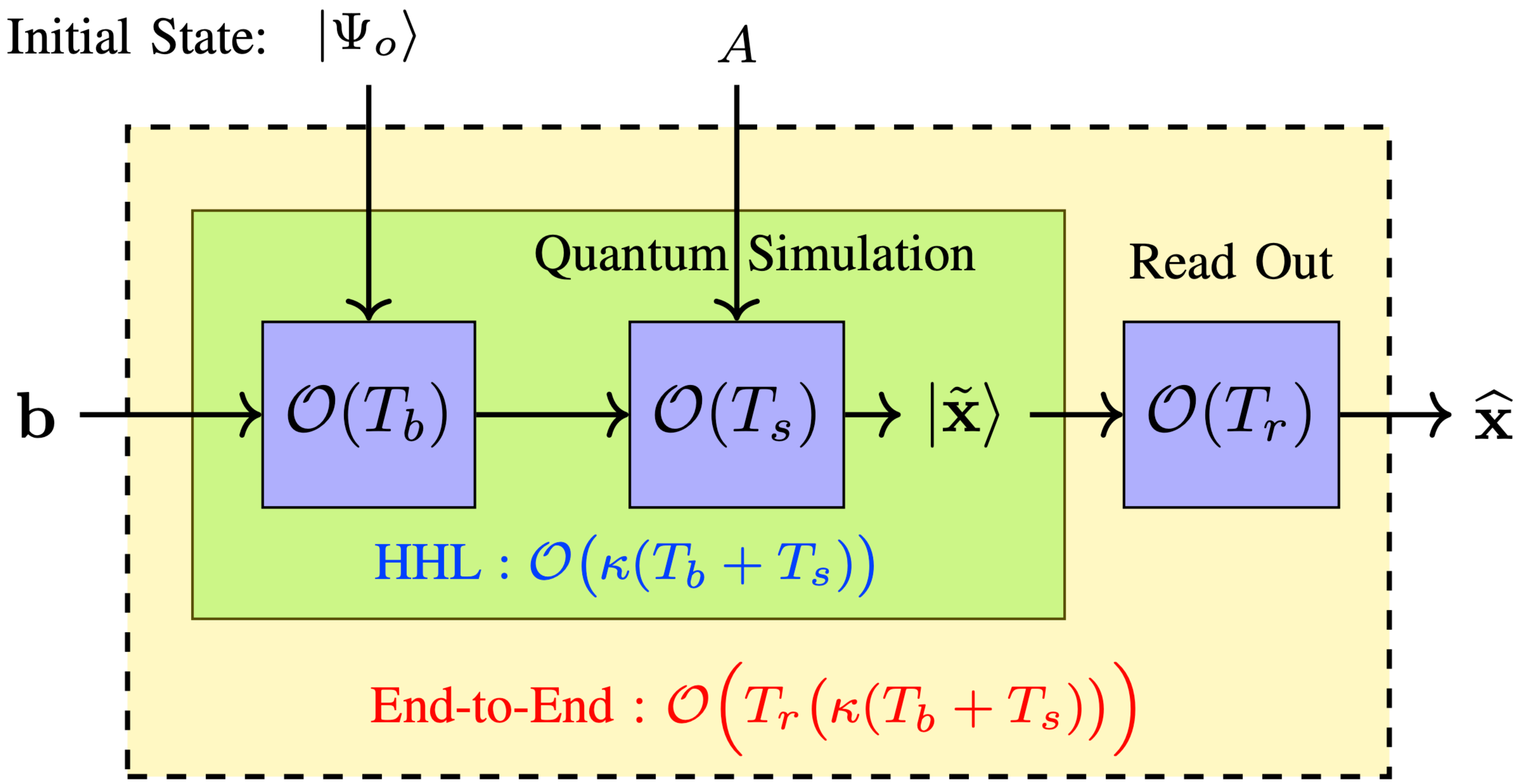
How many copies ? $\longrightarrow$ $\Theta(poly(N)/\varepsilon)$

J. van Apeldoorn, A. Cornelissen, A. Gilye´n, and G. Nannicini, "Quantum tomography using state-preparation unitaries," in *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2023, pp. 1265–1318.

Complete Quantum Picture: End-to-End


Initial State: $|\Psi_o\rangle$
A
Quantum Simulation
Read Out
$\mathbf{b}$
$\mathcal{O}(T_b)$
$\mathcal{O}(T_s)$
$|\tilde{\mathbf{x}}\rangle$
$\mathcal{O}(T_r)$
$\widehat{\mathbf{x}}$
HHL : $\mathcal{O}\big(\kappa(T_b + T_s)\big)$
End-to-End : $\mathcal{O}\Big(T_r\big(\kappa(T_b + T_s)\big)\Big)$

Complete Quantum Picture: End-to-End


Initial State: $|\Psi_o\rangle$
A
Quantum Simulation
Read Out
$\mathbf{b}$
$\mathcal{O}(T_b)$
$\mathcal{O}(T_s)$
$|\tilde{\mathbf{x}}\rangle$
$\mathcal{O}(T_r)$
$\widehat{\mathbf{x}}$
HHL : $\mathcal{O}\big(\kappa(T_b + T_s)\big)$
End-to-End : $\mathcal{O}\Big(T_r\big(\kappa(T_b + T_s)\big)\Big)$

Evaluating End-to-End Complexity of Solving Linear Power Flows using Quantum Linear System Solving Algorithms (HHL Family)

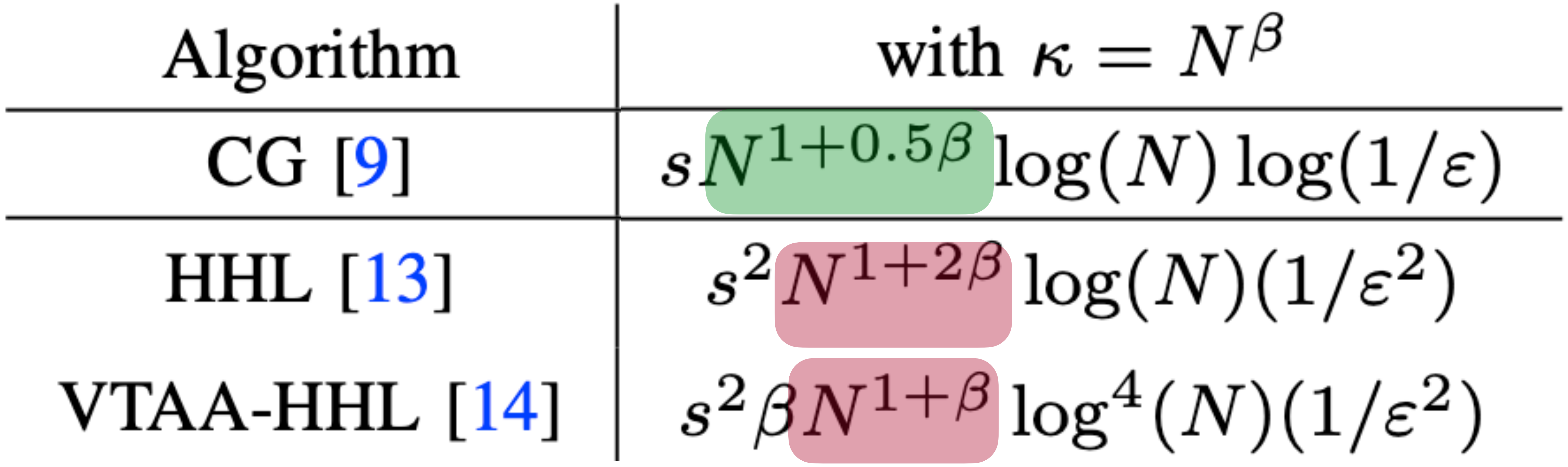
Point #2:

During speedup analysis consider End-to-End runtime complexity

Readout alone is enough to Kill any advantage in general Power Flow setting

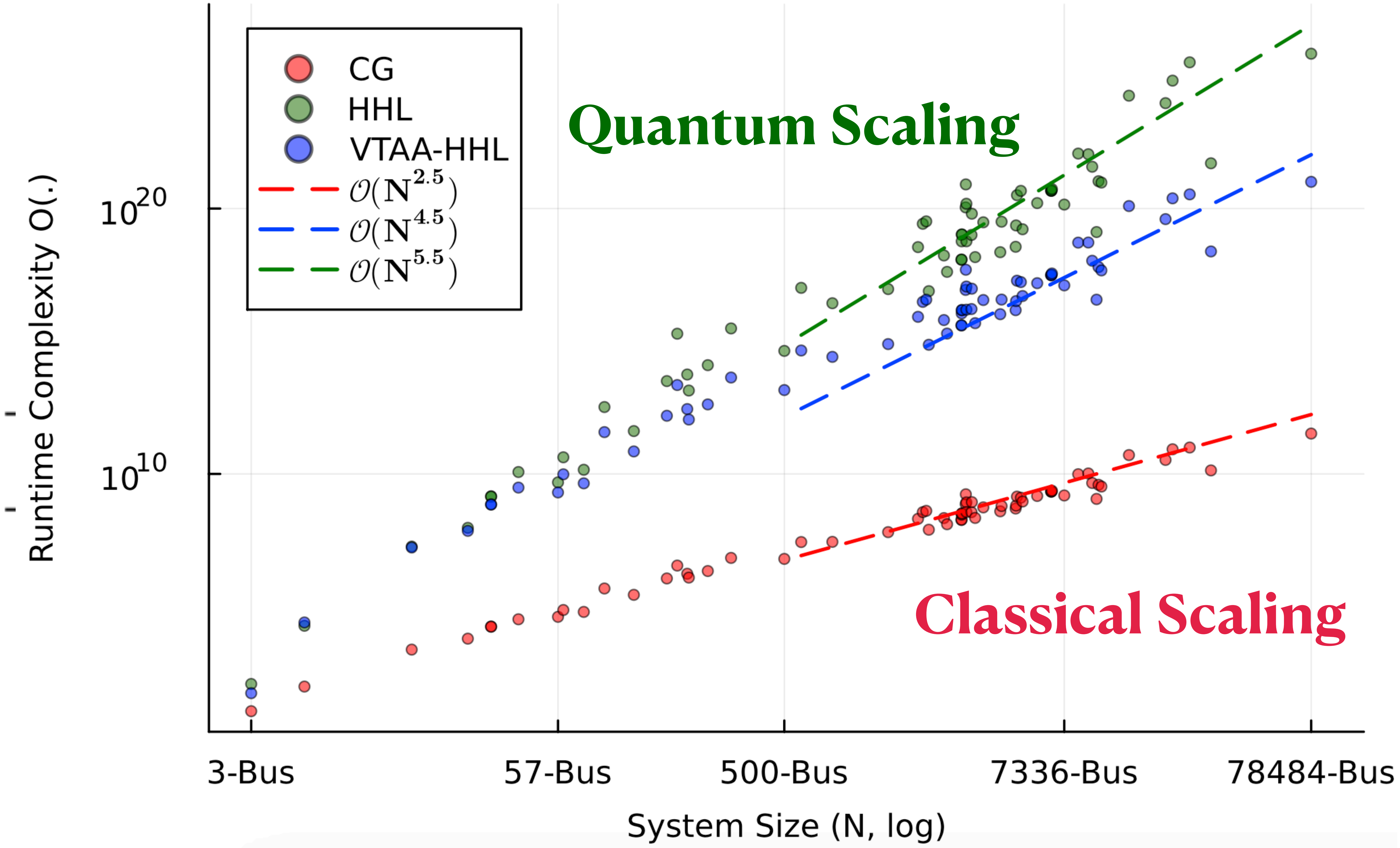
Comparative Picture

Comparative Picture

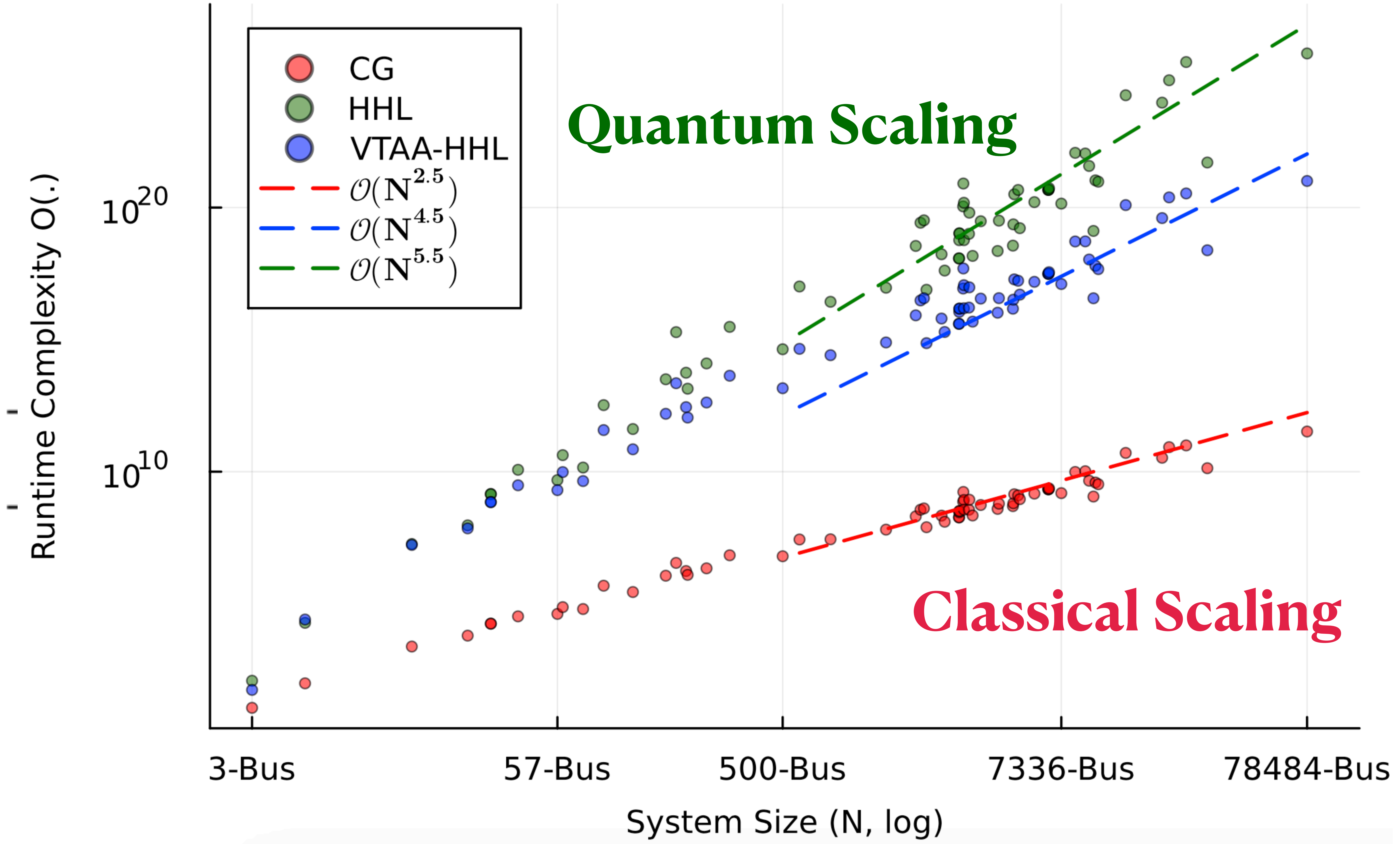


Algorithm	with $\kappa = N^{\beta}$
CG [9]	$sN^{1+0.5\beta}\log(N)\log(1/\varepsilon)$
HHL [13]	$s^2N^{1+2\beta}\log(N)(1/\varepsilon^2)$
VTAA-HHL [14]	$s^2\beta N^{1+\beta}\log^4(N)(1/\varepsilon^2)$

Comparative Picture



Comparative Picture



Current Quantum Linear Solving Algorithms offer **No Advantage** in solving Power Flow in Standard Formulations

What is needed for ‘Potential’ Speedup?

What is needed for ‘Potential’ Speedup?

$$s^2\,\kappa^2\,N\log(N)(1/\varepsilon^2)$$

What is needed for ‘Potential’ Speedup?

$$s^2\,\kappa^2\,N\log(N)(1/\varepsilon^2)$$

Pre-conditioning to
Suppress Condition Number

What is needed for ‘Potential’ Speedup?

$$s^2 \kappa^2 N \log(N)(1/\varepsilon^2)$$

Pre-conditioning to
Suppress Condition Number

Reading Partial Output/
Lower Readout

What is needed for 'Potential' Speedup?

$$s^2\kappa^2 N\log(N)(1/\varepsilon^2)$$

Pre-conditioning to Suppress Condition Number

Reading Partial Output/ Lower Readout

$$s^2\kappa_r^2 D\log(N)(1/\varepsilon^2)$$

Readout level

Reduced Condition-Number

What is needed for ‘Potential’ Speedup?

$$s^2 \kappa^2 N \log(N)(1/\varepsilon^2)$$

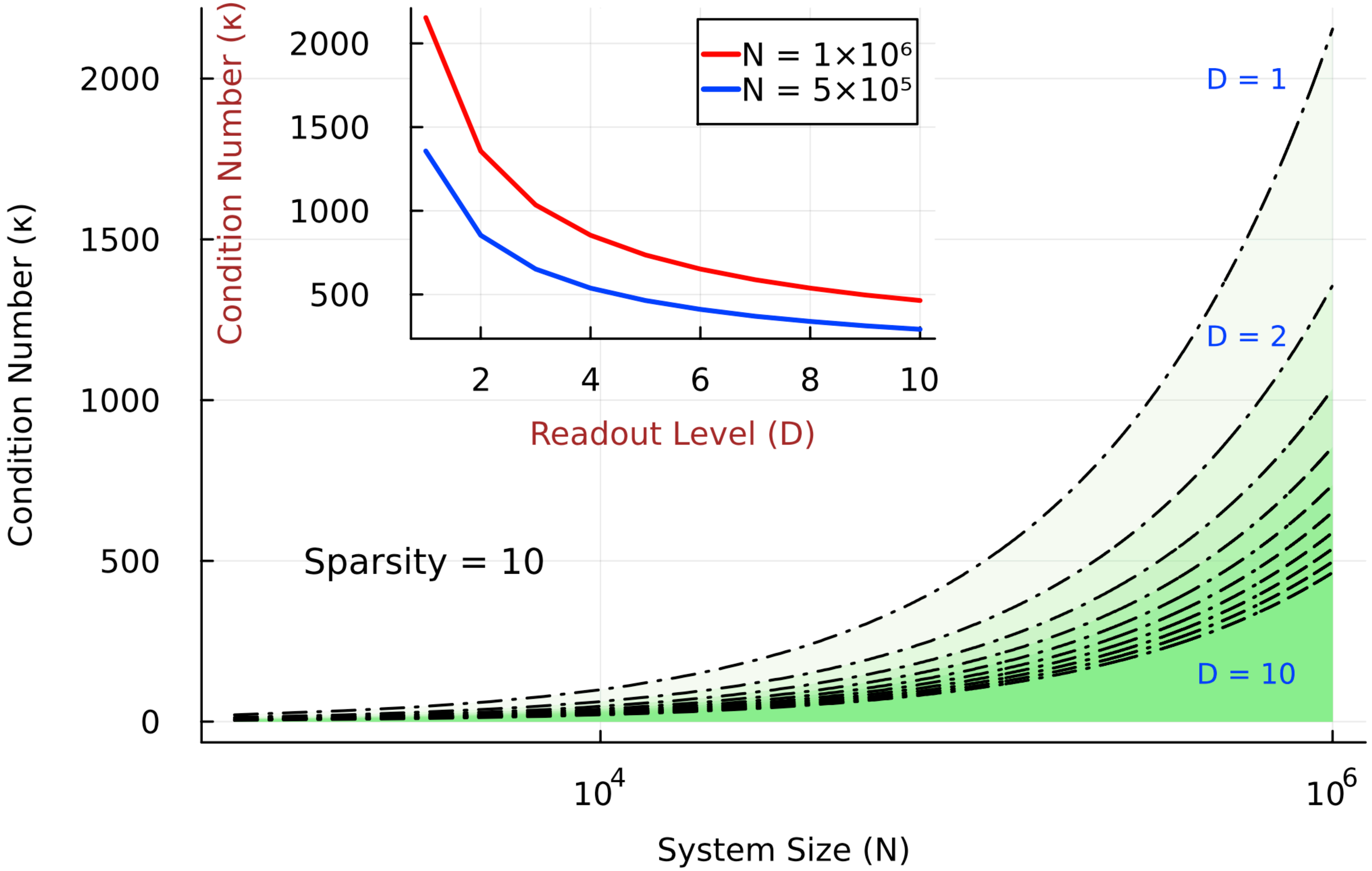
Pre-conditioning to Suppress Condition Number

Reading Partial Output/ Lower Readout

$$s^2 \kappa_r^2 D \log(N)(1/\varepsilon^2)$$

Readout level

Reduced Condition-Number



What is needed for ‘Potential’ Speedup?

$$s^2 \kappa^2 N \log(N)(1/\varepsilon^2)$$

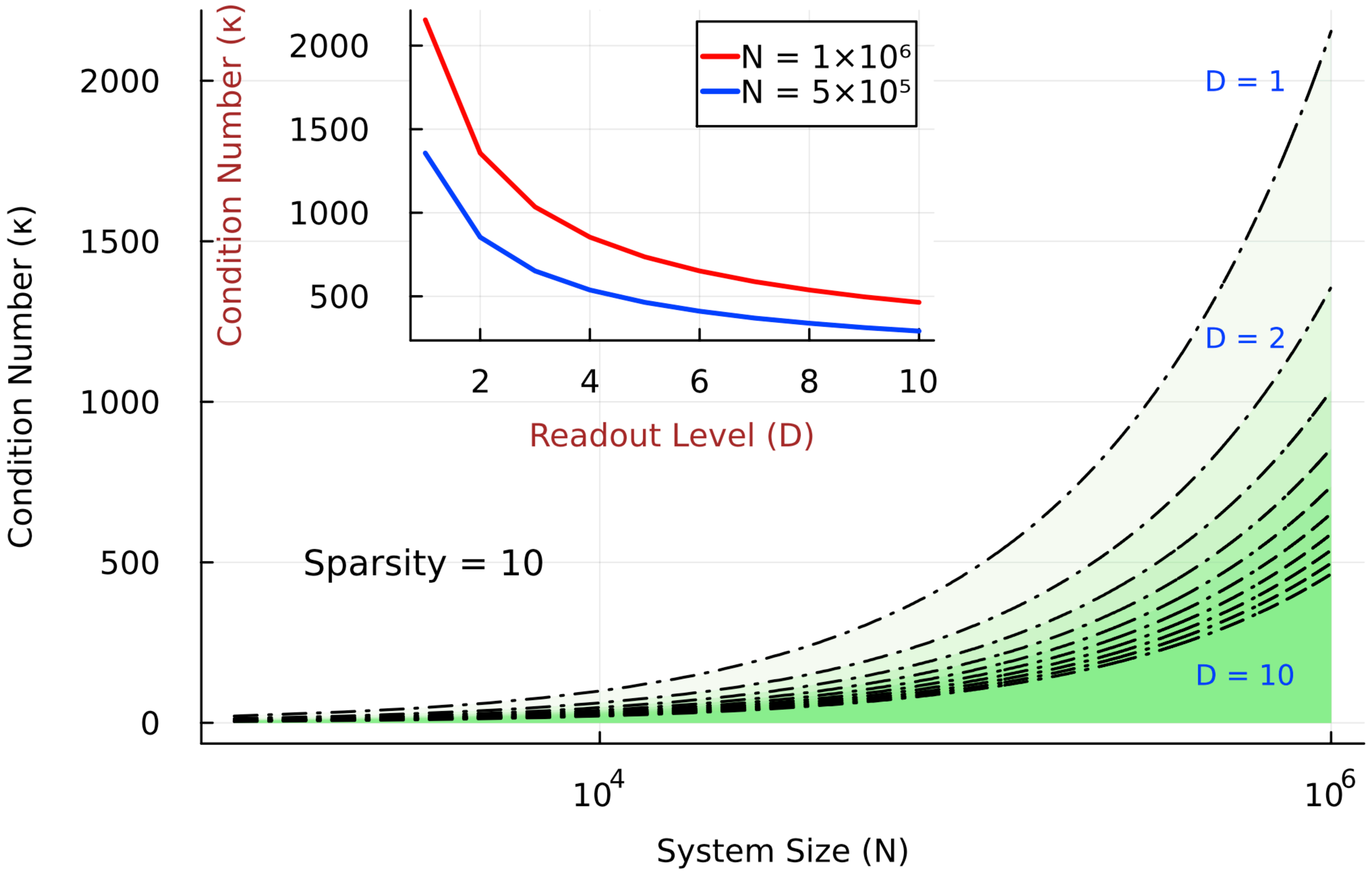
Pre-conditioning to Suppress Condition Number

Reading Partial Output/ Lower Readout

$$s^2 \kappa_r^2 D \log(N)(1/\varepsilon^2)$$

Readout level

Reduced Condition-Number



Pre-Condition & Read Less

Ok! Forget DC, What About AC Power Flow?

Ok! Forget DC, What About AC Power Flow?

Newton Raphson Load Flow

Ok! Forget DC, What About AC Power Flow?

Newton Raphson Load Flow → Solving Linear Systems of Equations with Jacobian Matrix Multiple Times

Ok! Forget DC, What About AC Power Flow?

Newton Raphson Load Flow → Solving Linear Systems of Equations with Jacobian Matrix Multiple Times

$$\mathcal{O}(K\ \ Ns\sqrt{\kappa}\log(1/\varepsilon_c))$$

Ok! Forget DC, What About AC Power Flow?

Newton Raphson Load Flow → Solving Linear Systems of Equations with Jacobian Matrix Multiple Times

$$\mathcal{O}(K\ Ns\sqrt{\kappa}\log(1/\varepsilon_c))$$

Number of NRLF Iterations

Ok! Forget DC, What About AC Power Flow?

Newton Raphson Load Flow → Solving Linear Systems of Equations with Jacobian Matrix Multiple Times

$$\mathcal{O}(\underbrace{K}_{\text{Number of NRLF Iterations}}\ \underbrace{Ns\sqrt{\kappa}\log(1/\varepsilon_c)}_{\text{Conjugate Gradient}})$$

Ok! Forget DC, What About AC Power Flow?

Newton Raphson Load Flow → Solving Linear Systems of Equations with Jacobian Matrix Multiple Times

$$\mathcal{O}(\underbrace{K}_{\text{Number of NRLF Iterations}}\ \underbrace{Ns\sqrt{\kappa}\log(1/\varepsilon_c)}_{\text{Conjugate Gradient}})$$

We already saw that

One iteration of Linear System Solve is slower using Quantum!

Ok! Forget DC, What About AC Power Flow?

Newton Raphson Load Flow → Solving Linear Systems of Equations with Jacobian Matrix Multiple Times

$$\mathcal{O}(\underbrace{K}_{\text{Number of NRLF Iterations}}\ \underbrace{Ns\sqrt{\kappa}\log(1/\varepsilon_c)}_{\text{Conjugate Gradient}})$$

We already saw that

One iteration of Linear System Solve is slower using Quantum!

So.....

As long as K is same for Quantum & Classical, We have Less Hopes![†]

Ok! Forget DC, What About AC Power Flow?

Newton Raphson Load Flow → Solving Linear Systems of Equations with Jacobian Matrix Multiple Times

$$\mathcal{O}(\underbrace{K}_{\text{Number of NRLF Iterations}}\ \underbrace{Ns\sqrt{\kappa}\log(1/\varepsilon_c)}_{\text{Conjugate Gradient}})$$

We already saw that

One iteration of Linear System Solve is slower using Quantum!

So.....

As long as K is same for Quantum & Classical, We have Less Hopes![†]

[†] Note that exact Quantum Complexity will depends on how the proposed algorithm handles error propagation within Quantum.

Another Issue

Another Issue — Jacobian’s Condition Number is Not Constant

Another Issue — Jacobian's Condition Number is Not Constant

κ = N^β
β = 1.8
10^2
10^4
10^6
10^8
κ (log)
30-Bus
1803-Bus
78484-Bus
System Size (N, log)
10^6
10^7
10^8
κ (log)
2869-bus
4601-bus
9241-bus
1
2
3
4
5
6
7
#Iterations
19402-bus
24464-bus
78484-bus
10^10
10^20
κ (log)
5
10
15
20
#Iterations

Another Issue — Jacobian's Condition Number is Not Constant

κ = N^β
β = 1.8
κ (log)
10^2
10^4
10^6
10^8
30-Bus
1803-Bus
78484-Bus
System Size (N, log)
κ (log)
10^6
10^7
10^8
2869-bus
4601-bus
9241-bus
1
2
3
4
5
6
7
#Iterations
κ (log)
10^10
10^20
19402-bus
24464-bus
78484-bus
5
10
15
20
#Iterations

Will need Adaptive Preconditioning

Another Issue — Jacobian's Condition Number is Not Constant

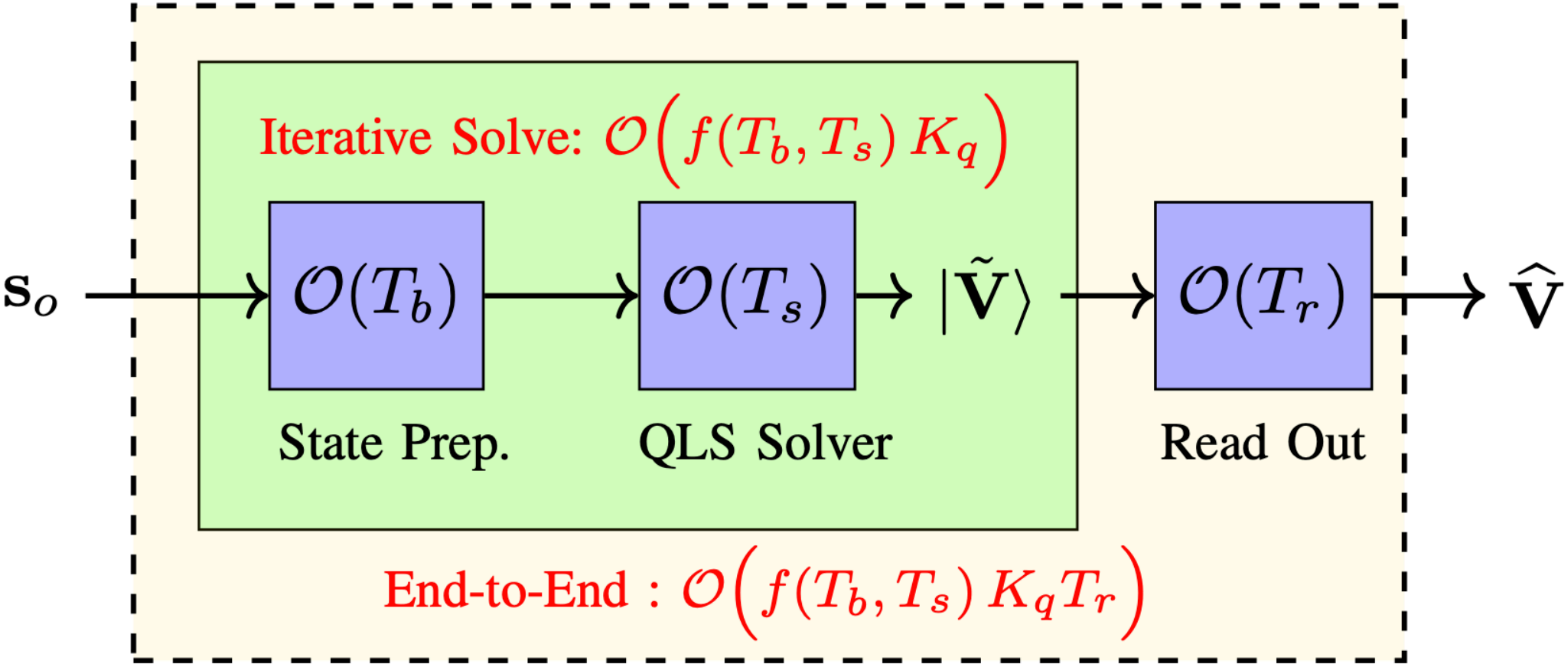
κ = N^β
β = 1.8
κ (log)
10^2
10^4
10^6
10^8
30-Bus
1803-Bus
78484-Bus
System Size (N, log)
κ (log)
10^6
10^7
10^8
2869-bus
4601-bus
9241-bus
1
2
3
4
5
6
7
#Iterations
κ (log)
10^10
10^20
19402-bus
24464-bus
78484-bus
5
10
15
20
#Iterations

Will need Adaptive Preconditioning

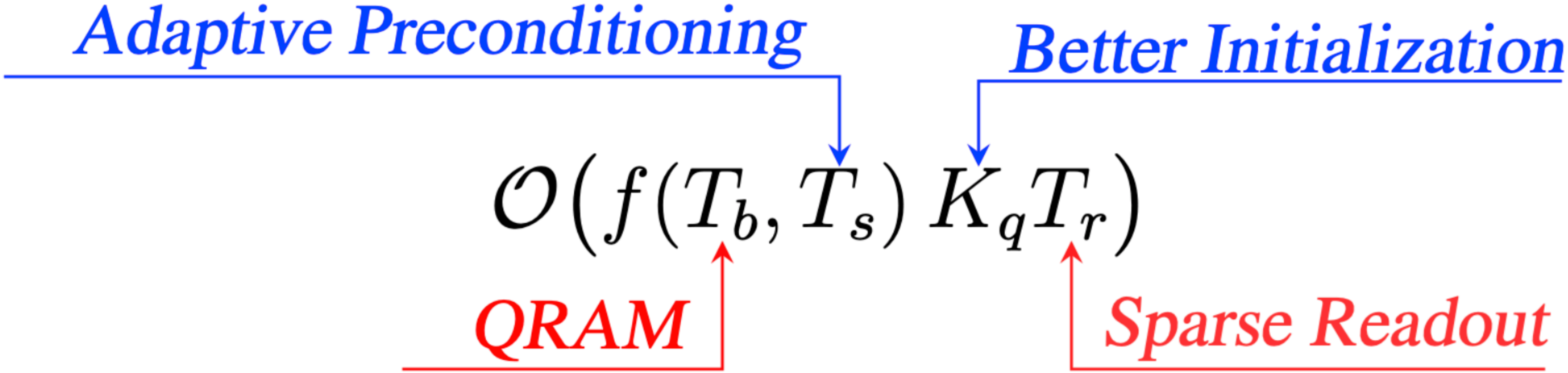
Tamas Terlaky and his group from Lehigh University have some work on it, in the context of Interior Point Methods

Overall— What will it take to have Hope?

Overall— What will it take to have Hope?


Iterative Solve: $\mathcal{O}\Big(f(T_b, T_s)\, K_q\Big)$
$\mathbf{s}_o$
$\mathcal{O}(T_b)$
$\mathcal{O}(T_s)$
$|\tilde{\mathbf{V}}\rangle$
$\mathcal{O}(T_r)$
$\widehat{\mathbf{V}}$
State Prep.
QLS Solver
Read Out
End-to-End : $\mathcal{O}\Big(f(T_b, T_s)\, K_q T_r\Big)$

Overall— What will it take to have Hope?



Adaptive Preconditioning

Better Initialization

$$\mathcal{O}\left(f(T_b, T_s)\, K_q T_r\right)$$

QRAM

Sparse Readout

By the way! What best can be achieved by doing all of this?

By the way! What best can be achieved by doing all of this?

$$\mathcal{O}\left(log(N)\frac{N}{\varepsilon} \right)$$

By the way! What best can be achieved by doing all of this?

$$\mathcal{O}\left(\boxed{log(N)}\,\frac{N}{\varepsilon}\right)$$

State Prepration

Putting Problem into Quantum Computer

By the way! What best can be achieved by doing all of this?

$$\mathcal{O}\left(\log(N)\,\frac{N}{\varepsilon}\right)$$

State Prepration

Putting Problem into Quantum Computer

Readout

Reading the Solution

By the way! What best can be achieved by doing all of this?

$$\mathcal{O}\left(log(N)\frac{N}{\varepsilon}\right)$$

State Prepration

Putting Problem into Quantum Computer

Readout

Reading the Solution

So......

By the way! What best can be achieved by doing all of this?

$$\mathcal{O}\left(log(N)\frac{N}{\varepsilon}\right)$$

State Prepration

Putting Problem into Quantum Computer

Readout

Reading the Solution

So……

Is All of This Worth it?

Or

Is it Watt We are Looking for?

Conclusion

End-to-End Complexity based Potential Quantum Speedup Analysis must be done for Your Favorite Problem....

..... Before starting to solve it.

Demystifying Quantum Power Flow:
Unveiling the Limits of Practical Quantum Advantage

Parikshit Pareek, Abhijith Jayakumar, Carleton Coffrin, and Sidhant Misra
Los Alamos National Laboratory, NM, USA.
pareek@lanl.gov; abhijithj@lanl.gov; cjc@lanl.gov, sidhant@lanl.gov.

arXiv:2402.08617

pareek@ee.iitr.ac.in

https://psquare-lab.github.io/

Indian Institute of Technology Roorkee

Asia's Oldest Technical Institute— Founded in 1847

Indian Institute of Technology Roorkee

Asia's Oldest Technical Institute— Founded in 1847

If Need someone who asks Very Stupid Questions in your research group meetings related to an interesting problem on

ML + Power or **Quantum + Power**

Let me know at: *pareek@ee.iitr.ac.in*