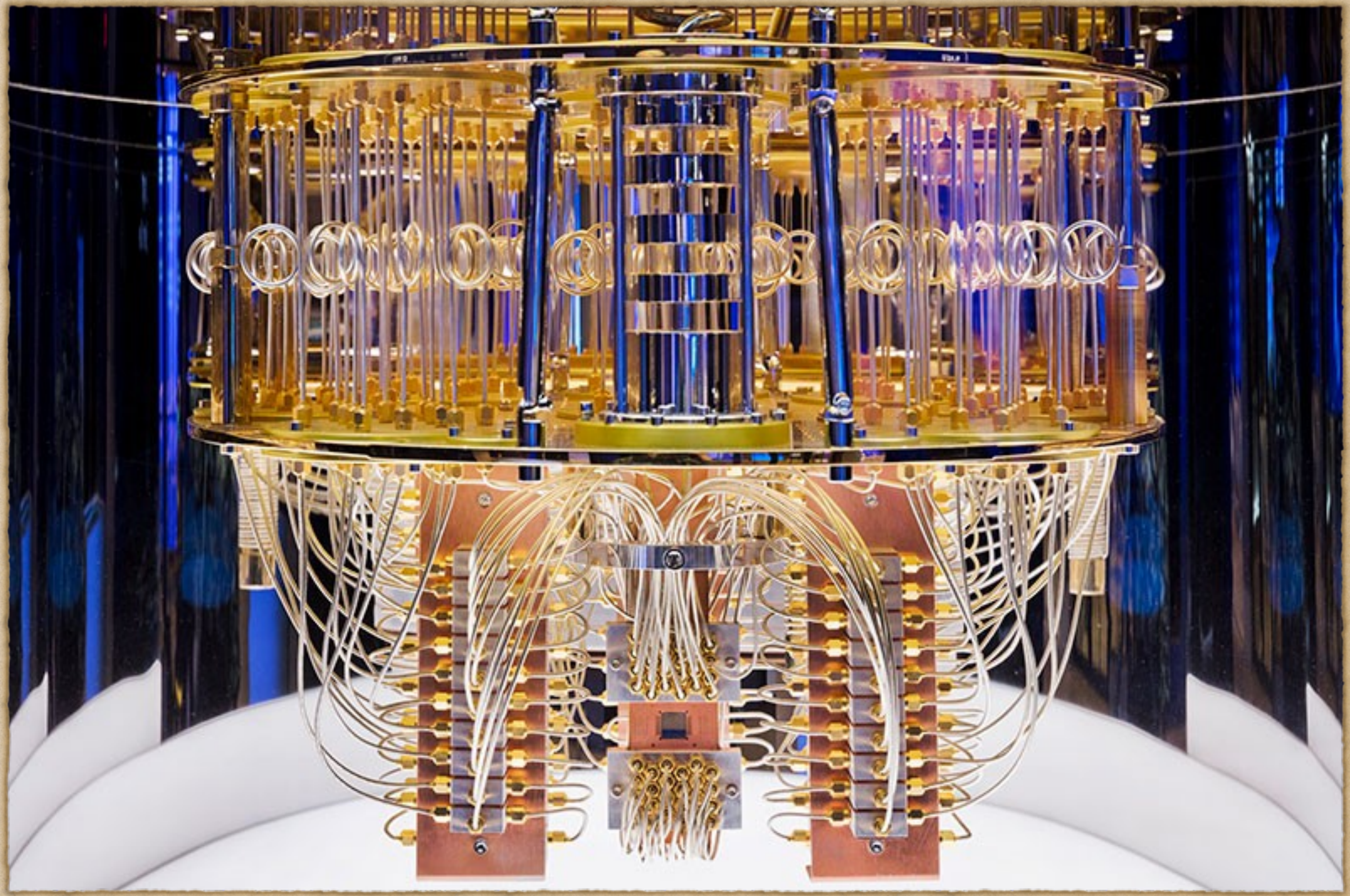
Applications of Variational Quantum Algorithms

Mark M. Wilde
School of Electrical and Computer Engineering, Cornell University

4th Annual International Quantum Information Science Workshop
Innovare Advancement Center, Rome, New York

Motivation

- We are in exciting times, with basic quantum computers available (~ 100 qubits), from IBM, IonQ, Rigetti, etc.
- Current era is called NISQ (noisy intermediate-scale)



Programming Existing Quantum Computers

- Programming quantum computers is becoming commonplace, and some universities are offering freshman courses on this topic

- What can we do with existing quantum computers?

Outline

- Background on variational quantum algorithms
- Application to semidefinite programming arXiv:2108.08406 (w/ Patel, Coles)
- Background on quantum computational complexity theory
- Other applications:
  - Estimating distinguishability measures arXiv:2108.08406 (w/ Rethinasamy, Agarwal, Sharma)
  - Symmetry testing arXiv:2105.12758, arXiv:2203.10017 (w/ LaBorde)

Collaborations with Students

Rochisha Agarwal

Margarite LaBorde

Dhrumil Patel

Soorya Rethinasamy

Overview of Variational Quantum Algorithms

Variational Principle

- The variational principle in quantum mechanics:

$$\langle \psi(\theta) | H | \psi(\theta) \rangle \geq E_0 \equiv \min_{|\psi\rangle} \langle \psi | H | \psi \rangle$$

  where $|\psi(\theta)\rangle$ is a trial wavefunction, H is a Hamiltonian, & E_0 is the ground-state energy

- Variational principle has played an important role in physics calculations for many years

Variational Quantum Algorithms (VQAs)

- Proposed as a method for reducing quantum computing resources, while still doing something presumably difficult classically

nature COMMUNICATIONS

ARTICLE

Received 9 Dec 2013 | Accepted 27 May 2014 | Published 23 Jul 2014 DOI: 10.1038/ncomms5213 OPEN

A variational eigenvalue solver on a photonic quantum processor

Alberto Peruzzo[1,*,†], Jarrod McClean[2,*], Peter Shadbolt[1], Man-Hong Yung[2,3], Xiao-Qi Zhou[1], Peter J. Love[4], Alán Aspuru-Guzik[2] & Jeremy L. O'Brien[1]

- Use the quantum computer for essentially one task! Estimate $\langle \psi | O | \psi \rangle$, i.e., expectation value of observable O

VQAs: How do they work?

- Consider example of Variational Quantum Eigensolver
- Goal: find the ground-state energy E_0 of an n-qubit Hamiltonian H
- Typical assumption: H decomposes as a sum of $p(n) \equiv \text{poly}(n)$ efficiently measurable observables

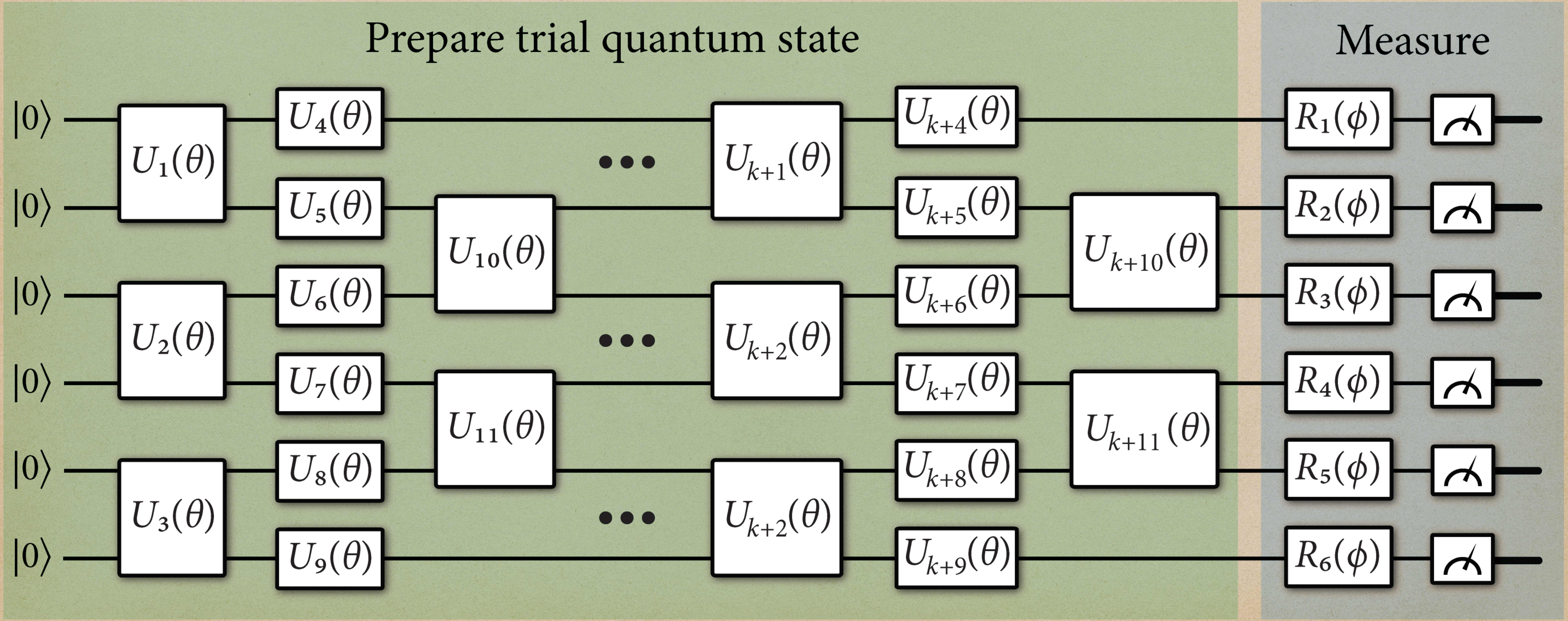
$$H = \sum_{i=1}^{p(n)} c_i O_i$$

where $c_i \in \mathbb{R}$ and O_i is an efficiently measurable observable

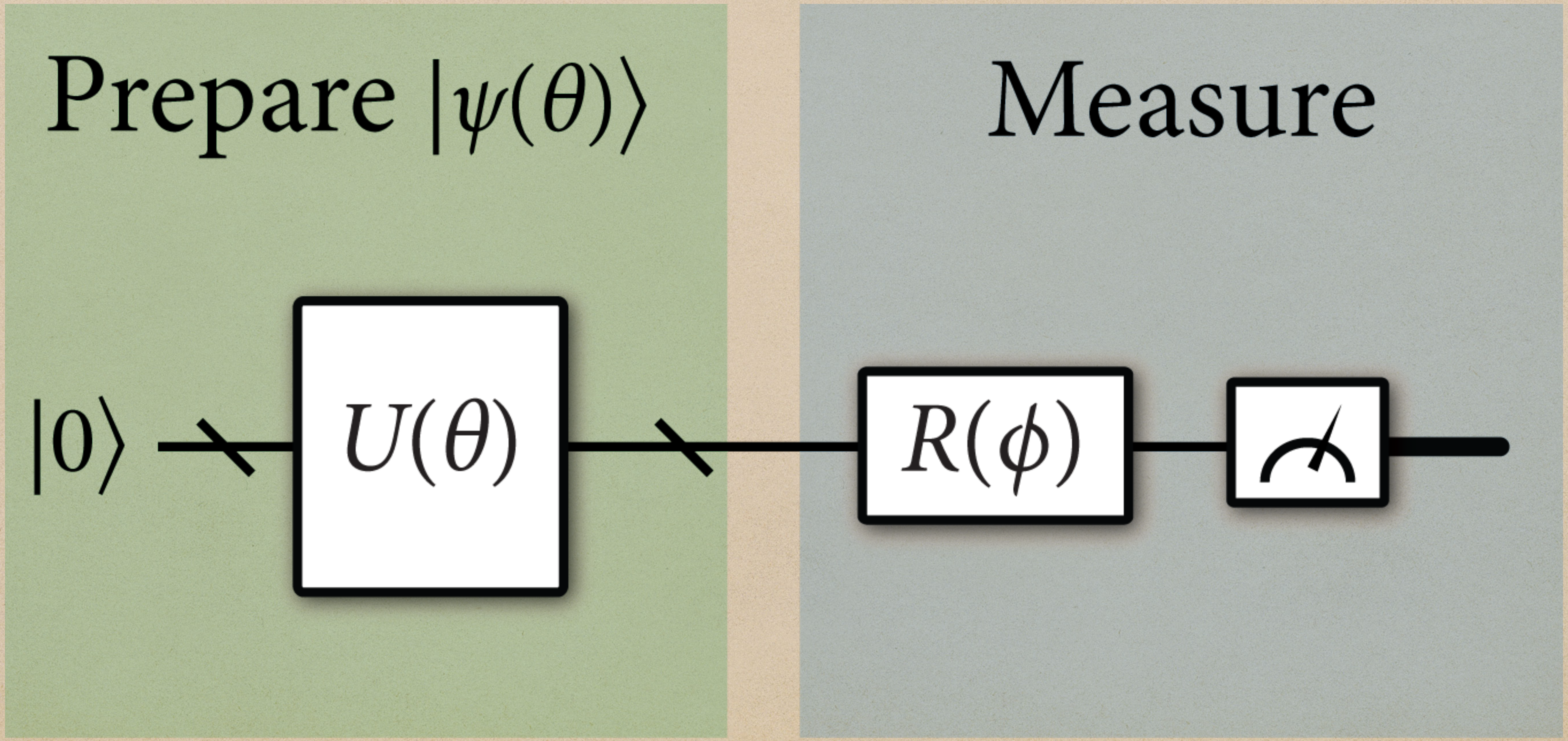
Quantum Part of VQAs

- Use quantum computer for this one thing:
- Execute parameterized circuit to prepare trial state $|\psi(\theta)\rangle$ and then estimate $\langle\psi(\theta)|O_i|\psi(\theta)\rangle$ for all i, through sampling / repetition
- Let $\widetilde{o}_i$ denote the estimate of $\langle\psi(\theta)|O_i|\psi(\theta)\rangle$

VQAs: Depiction of Quantum Part



Abbreviated Depiction



First Classical Part of VQAs

- Calculate $\sum_{i=1}^{p(n)} c_i \widetilde{o}_i$ as guess for ground-state energy
- To estimate $\langle \psi(\theta) | H | \psi(\theta) \rangle$ with ε-error and success probability $1 - \delta$, $O\left(\frac{C^2}{\varepsilon^2} \log \frac{1}{\delta} \right)$ circuit executions are required, where $C \equiv \sum_{i=1}^{p(n)} |c_i| \, \| O_i \|$ (consequence of Hoeffding bound)

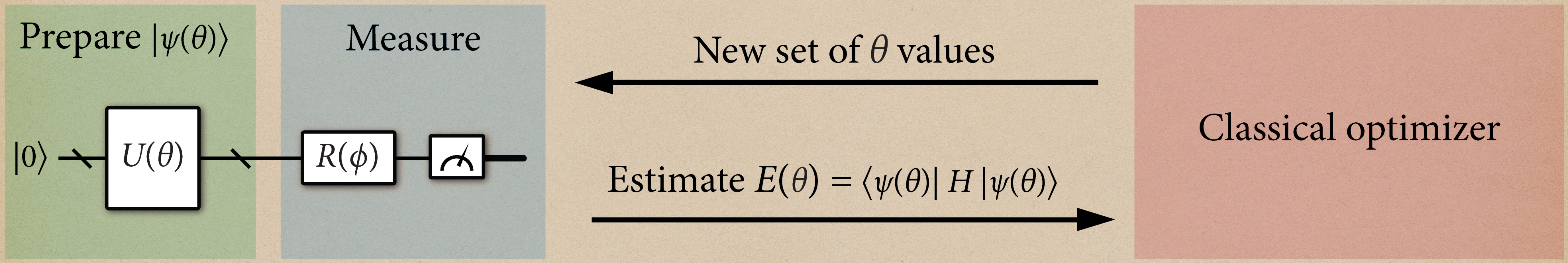
Second Classical Part of VQAs

- With $\langle\psi(\theta)|H|\psi(\theta)\rangle$ estimated, use a classical optimizer to compute next values of parameter θ (gradient descent or related method)
- Goal is to minimize cost function $\langle\psi(\theta)|H|\psi(\theta)\rangle$
- Variational principle guarantees that

$$\min_{\theta}\langle\psi(\theta)|H|\psi(\theta)\rangle \geq E_0$$

and the hope is to saturate the inequality

VQAs: Hybrid Quantum-Classical Optimization



- Outsource parameter optimization to a classical optimizer
- Use the quantum computer only to estimate expectations of observables

Evaluating Gradients: Parameter Shift Rule

- Use the parameter shift rule to evaluate gradients on quantum computers
- Applies to parameterized circuits of the form

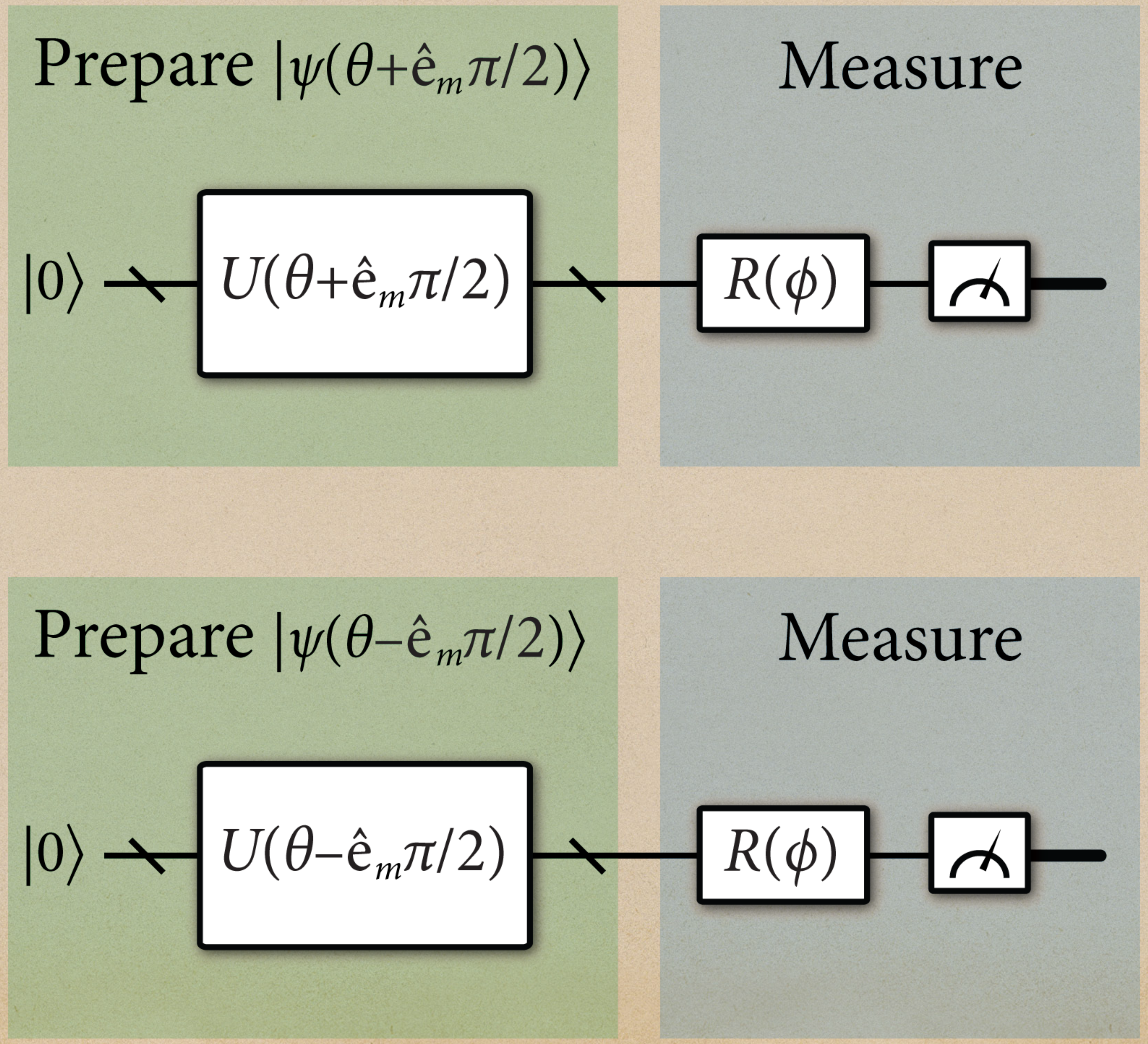
$$U(\theta) = \prod_m \exp(-i\theta_m H_m) W_m$$

where H_m is Hermitian w/ 2 eigenvalues & W_m is unparameterized unitary

- Gradient can be evaluated analytically as

$$\nabla_{\theta_m} \langle H \rangle_\theta = \frac{1}{2} \left(\langle H \rangle_{\theta + \frac{\pi}{2}\hat{e}_m} - \langle H \rangle_{\theta - \frac{\pi}{2}\hat{e}_m} \right)$$

Quantum Circuits for Evaluating Gradients



Issues with VQAs

- Runtime - depends on circuit depth of ansatz, number of iterations needed to find global optimum, and shots needed to estimate cost and gradient
- Barren plateau problem - can happen that magnitude of gradient exponentially vanishes with system size, requiring exponential precision to escape a barren plateau (where cost landscape is flat)
- Noise - Try to use shallow depth parameterized quantum circuits to mitigate the effects of noise

VQAs for Semidefinite Programming

Review of Semidefinite Programming

- A semidefinite program (SDP) is an optimization problem, having applications in operations research, combinatorial optimization, etc.
- Standard form: $\sup_{X \geq 0} \{\mathrm{Tr}[CX] : \mathrm{Tr}[A_i X] = b_i \quad \forall i \in [M]\}$
- Defining $\Phi(X) \equiv (\mathrm{Tr}[A_1 X], \ldots, \mathrm{Tr}[A_M X])$ and $b \equiv (b_1, \ldots, b_M)$, can abbreviate as $\sup_{X \geq 0} \{\mathrm{Tr}[CX] : \Phi(X) = b\}$

Lagrangian of an SDP

- For $c > 0$ and $y \in \mathbb{R}^M$, define the augmented Lagrangian:

$$\mathscr{L}(X, y) \equiv \mathrm{Tr}[CX] + y^T(b - \Phi(X)) - \frac{c}{2} \parallel b - \Phi(X) \parallel_2^2$$

- Since $X \geq 0$, can substitute with $X = \lambda\rho$, where ρ is a quantum state and $\lambda \geq 0$ is a scalar:

$$\mathscr{L}(\lambda\rho, y) \equiv \lambda\,\mathrm{Tr}[C\rho] + y^T(b - \lambda\Phi(\rho)) - \frac{c}{2} \parallel b - \lambda\Phi(\rho) \parallel_2^2$$

- Can cast optimization as $p^* \equiv \sup_{\rho \in \mathrm{States}, \lambda \geq 0} \inf_{y \in \mathbb{R}^M} \mathscr{L}(\lambda\rho, y)$

Rewriting an SDP as a VQA

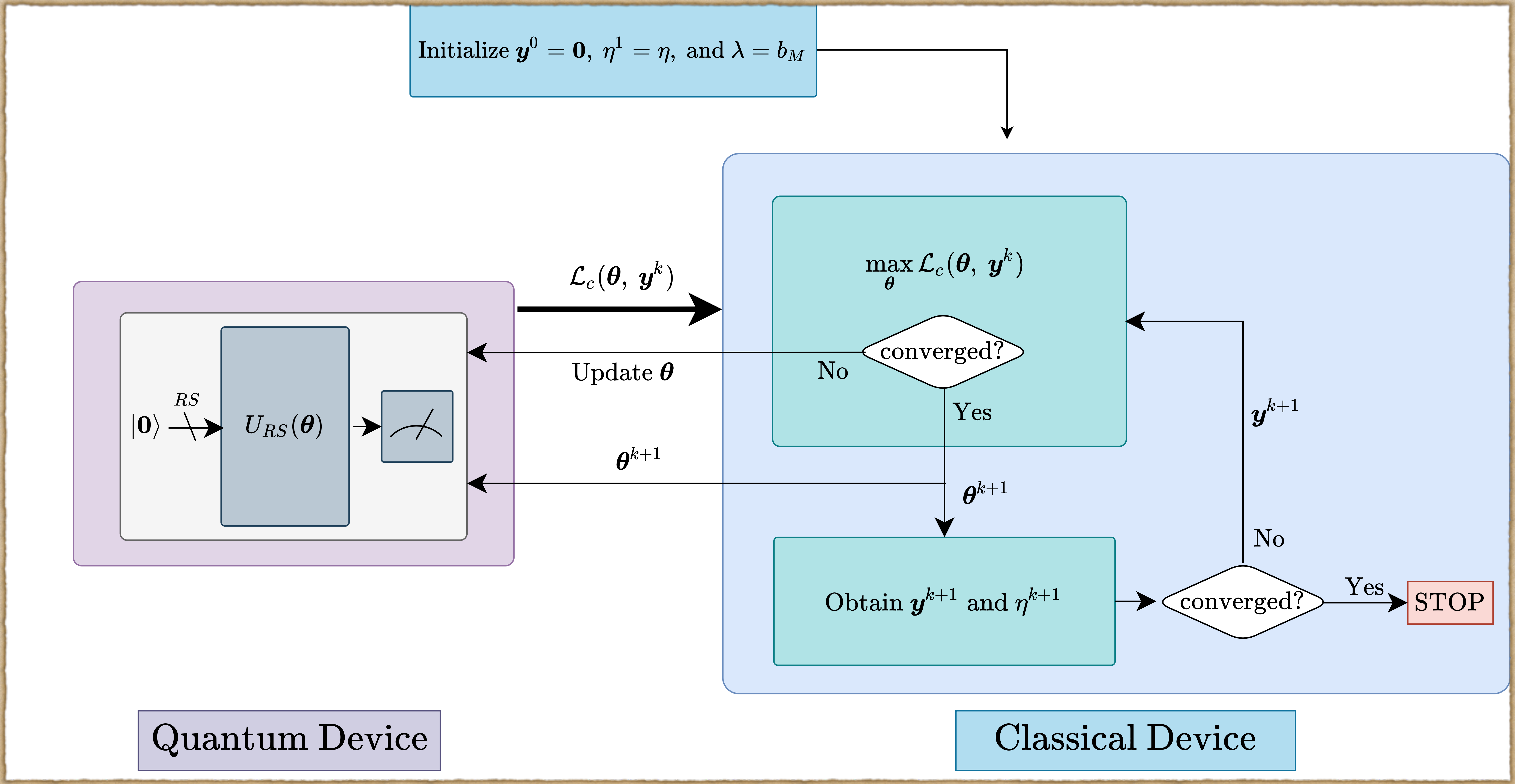
- With the last rewrite, we can replace the optimization over all states with an optimization over a parameterized family

$$p^* \geq \sup_{\theta \in [0,2\pi]^r, \lambda \geq 0} \inf_{y \in \mathbb{R}^M} \mathscr{L}(\lambda \rho(\theta), y)$$

- The optimization problem involves estimating $\mathrm{Tr}[C\rho(\theta)], \mathrm{Tr}[A_1\rho(\theta)], \ldots, \mathrm{Tr}[A_M\rho(\theta)]$, as well as their gradients, each of which we evaluate using the quantum computer

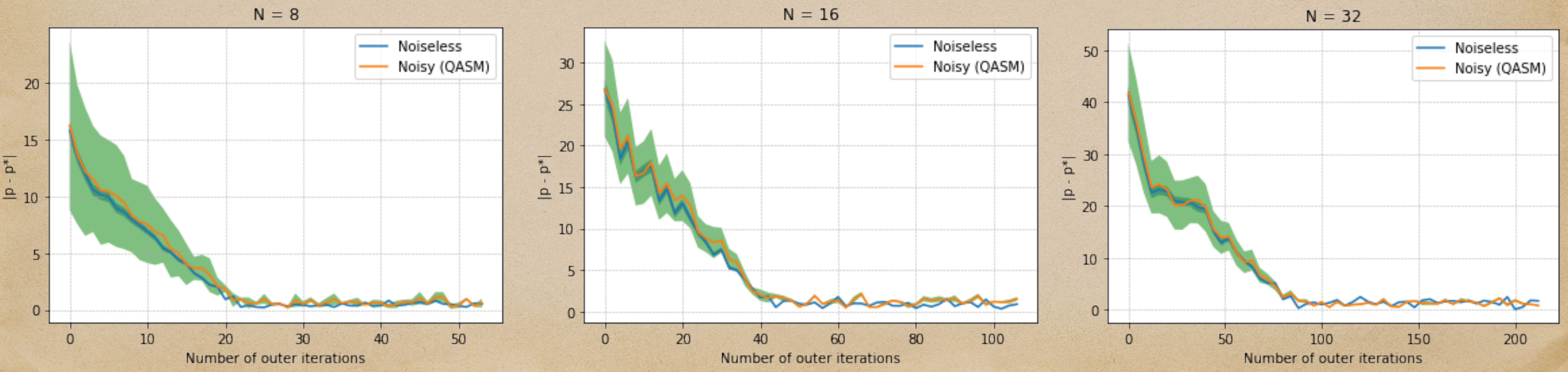
- Following the VQA principle, everything else is classical processing

Schematic of VQA for SDPs



Example of Performance

- Executed performance of the algorithm for randomly generated feasible SDPs with size of the matrices ≫ number of constraints



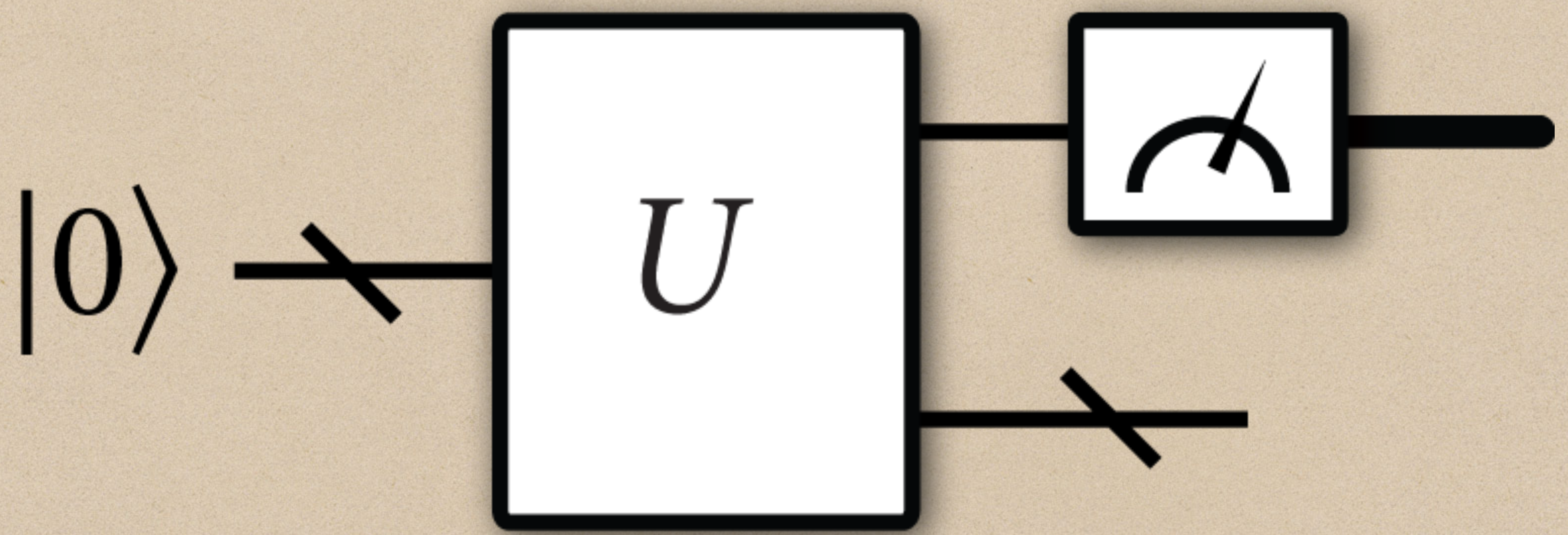
Quantum Computational Complexity Theory

Quantum Computational Complexity Theory

- For understanding & classifying difficulty of computational problems
- Most important complexity classes for quantum computation are BQP, QMA, QIP(2), QIP(3)
- These classes generalize P, NP, IP(2), IP(3), respectively

BQP in a Nutshell

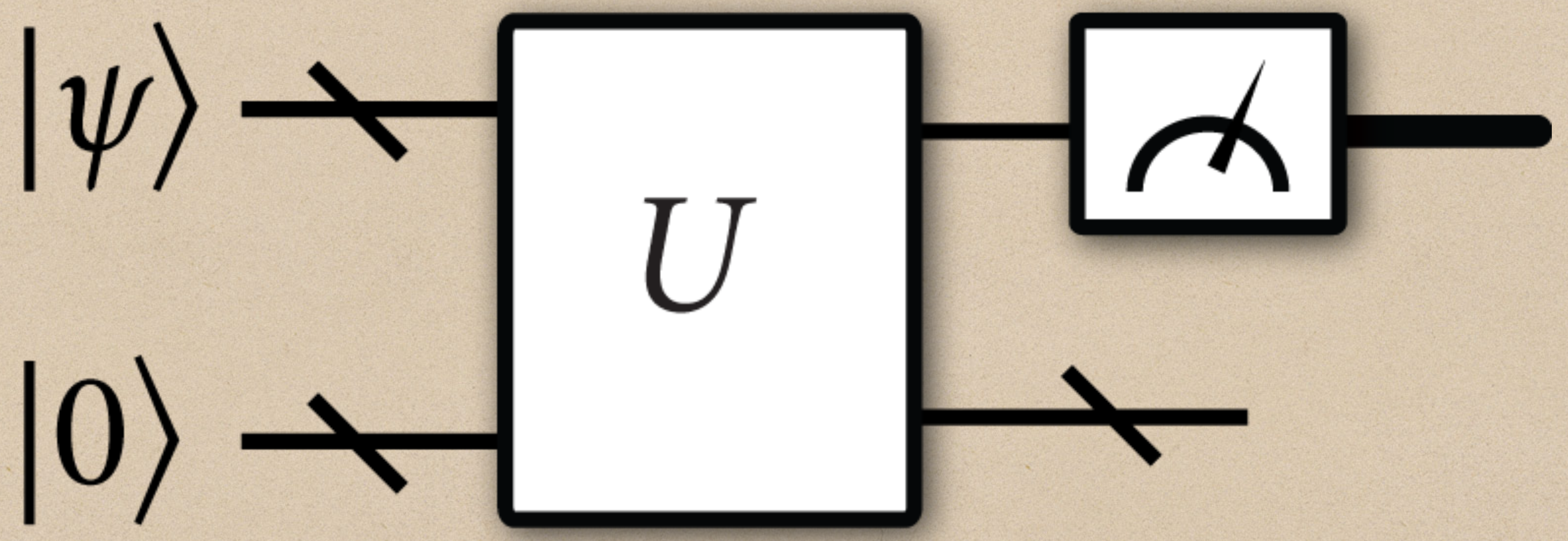
- BQP stands for "bounded error quantum polynomial time"
- Problems that are efficiently decidable by a quantum computer



- Funding agencies have spent $$$ based on the P ⊊ BQP belief

QMA in a Nutshell

- QMA stands for "quantum Merlin Arthur"
- Problems believed to be hard for a quantum computer to decide



- Model is that $|\psi\rangle$ is a state that is difficult to prepare on a quantum computer
- Assumption: quantum prover with unbounded computational resources prepares $|\psi\rangle$

Canonical QMA-Complete Problem

- A problem is called QMA-complete if it is in QMA and if it is as computationally difficult to solve as every problem in QMA
- Canonical QMA-complete problem is k-local Hamiltonian: Given a Hamiltonian $H = \sum_{i=1}^{n} H_i$, where each H_i acts on no more than k qubits, decide if its ground-state energy is $\geq a$ or $\leq b$

k-Local Hamiltonian

- To show that it is in QMA, quantum prover prepares ground state, sends it to verifier, who then picks H_i at random, and performs a measurement related to it and accepts based on the outcome of the measurement
- Acceptance probability is related to the ground-state energy

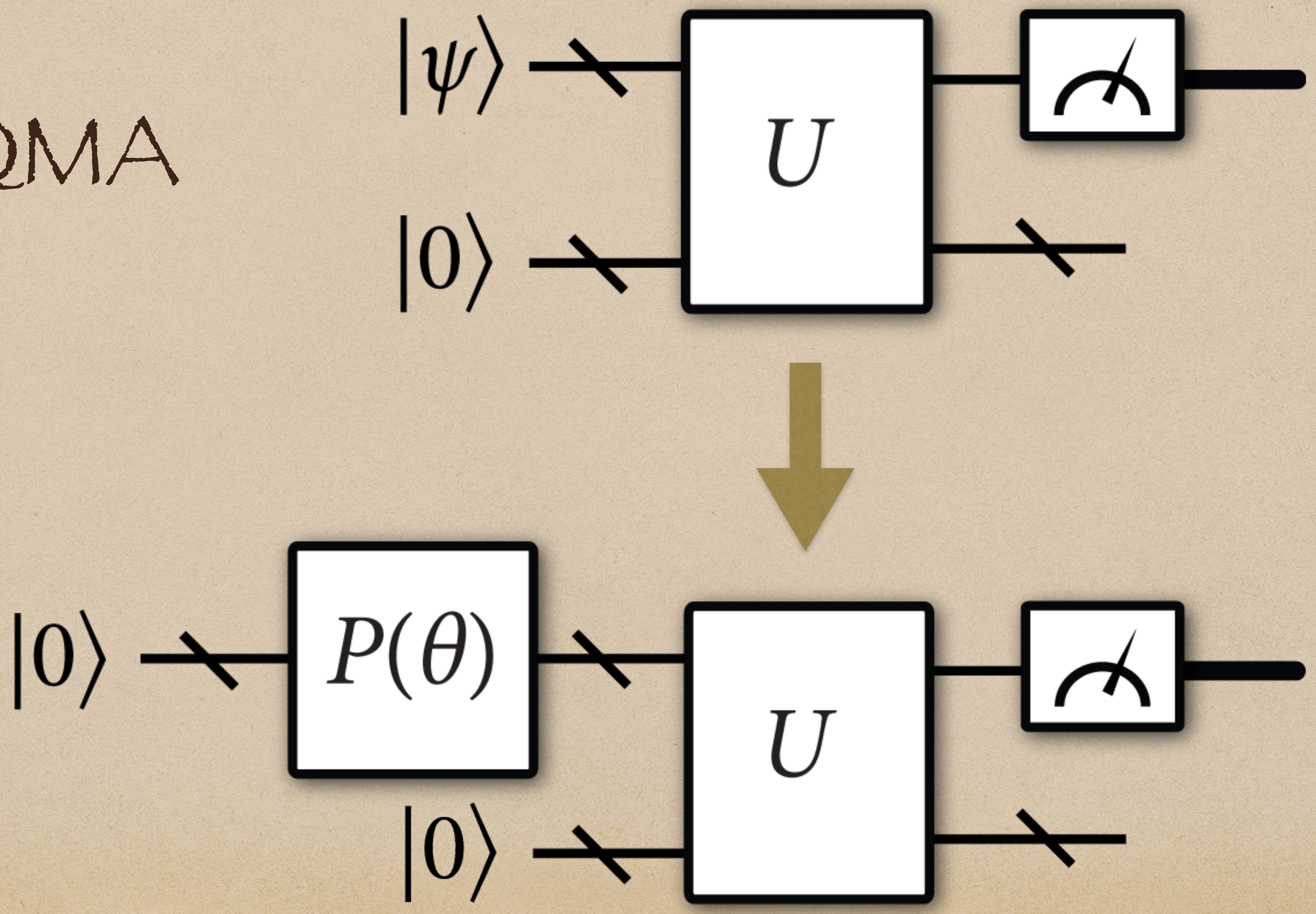
Variational Q. Eigensolver and *k*-Local Hamiltonian

- There is a direct link between VQE and *k*-Local Hamiltonian!
- VQE is trying to solve a QMA-complete problem
- By our beliefs in quantum complexity theory, it should not be possible to do so in the worst case
- However, ∃ evidence that VQE works well in practice, much like there are heuristics for trying to solve NP-complete problems

QMA and VQAs

QMA

VQA



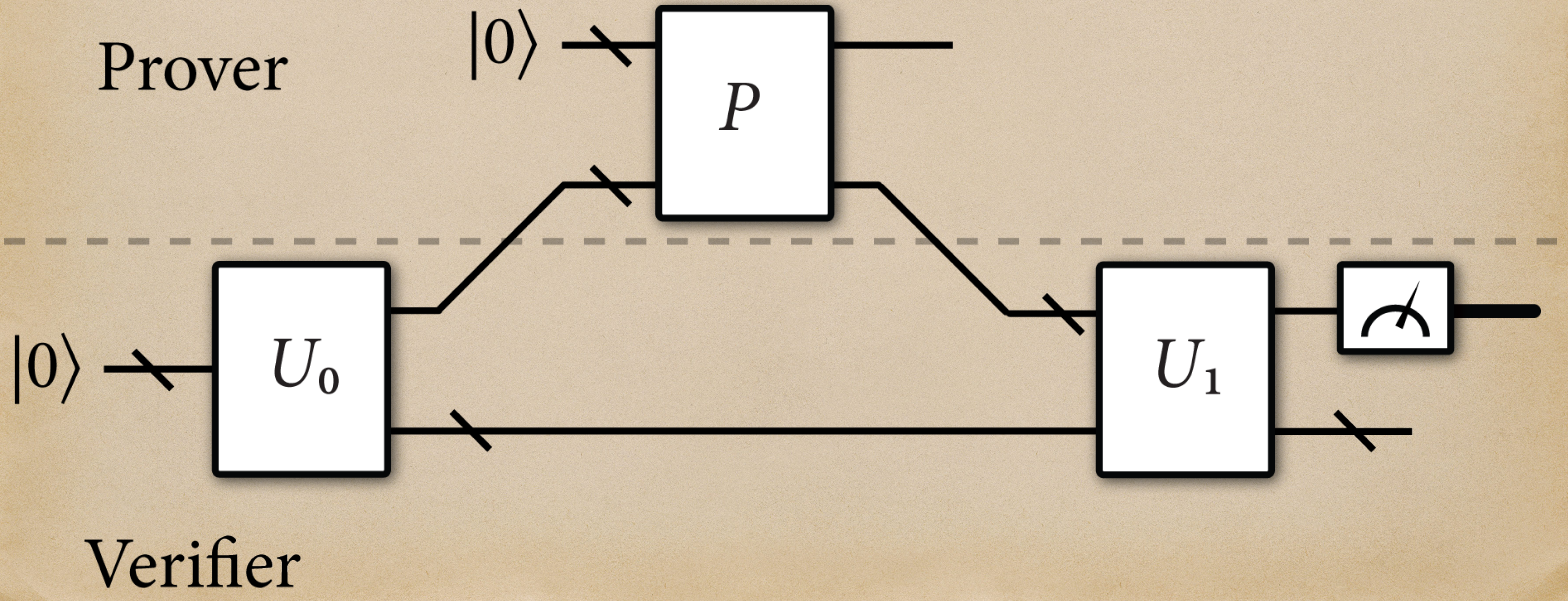
QMA and VQAs

- Basic idea is to replace the prover with a parameterized circuit and set the reward function (for maximization) equal to the acceptance probability in the original QMA problem

Quantum Interactive Proofs (QIP)

- We can view QMA as a communication protocol in which the prover sends a quantum message to the verifier
- BQP involves no messages sent from the prover to the verifier
- Taking this concept further, allow for prover and verifier to exchange more messages (called "quantum interactive proof")
- Idea is that interaction can allow for solving more difficult problems, like interacting with an omniscient teacher

QIP(2) - Two Messages Exchanged



QIP(2)-Complete Problem

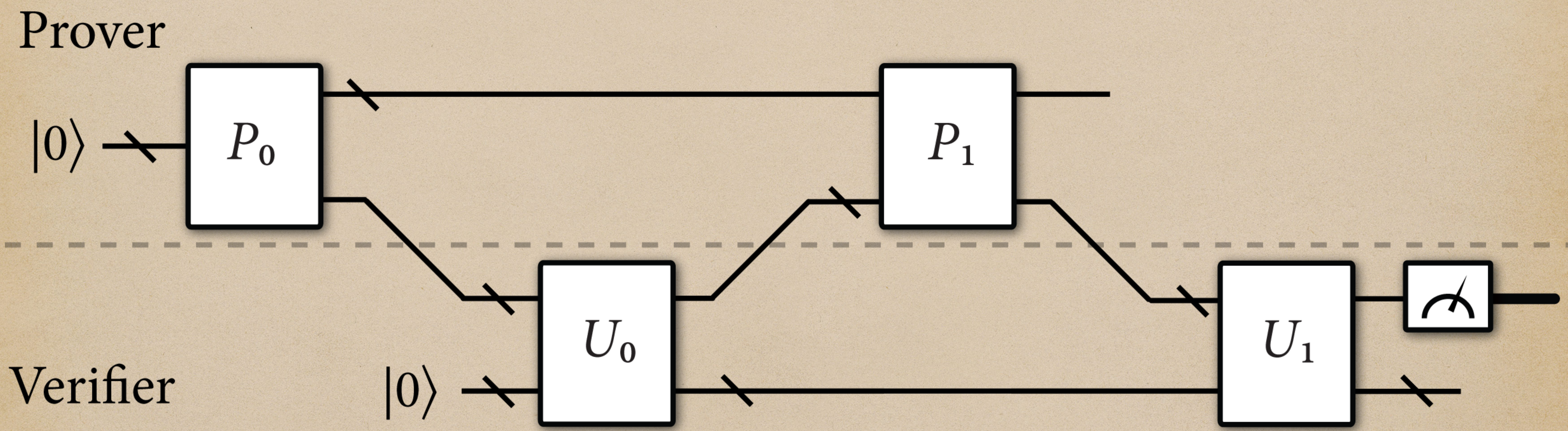
- Given a quantum channel $\mathcal{N}$ and a state ρ, estimate

$$\max_{\sigma \in \text{States}} F(\rho, \mathcal{N}(\sigma))$$

where the fidelity is defined as

$$F(\omega, \tau) \equiv \left\| \sqrt{\omega}\sqrt{\tau} \right\|_1^2$$

QIP(3) - Three Messages Exchanged



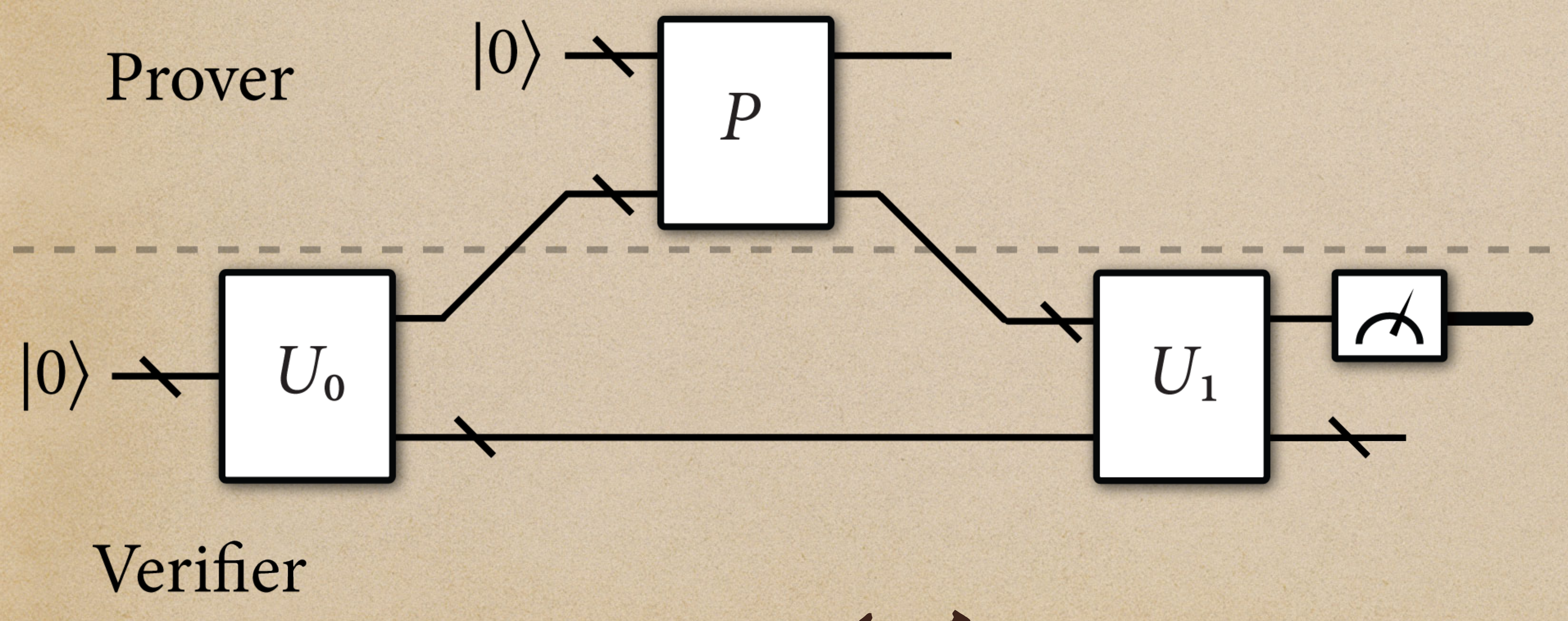
QIP(3)-Complete Problem

- Given quantum channels $\mathcal{N}$ and $\mathcal{M}$, estimate

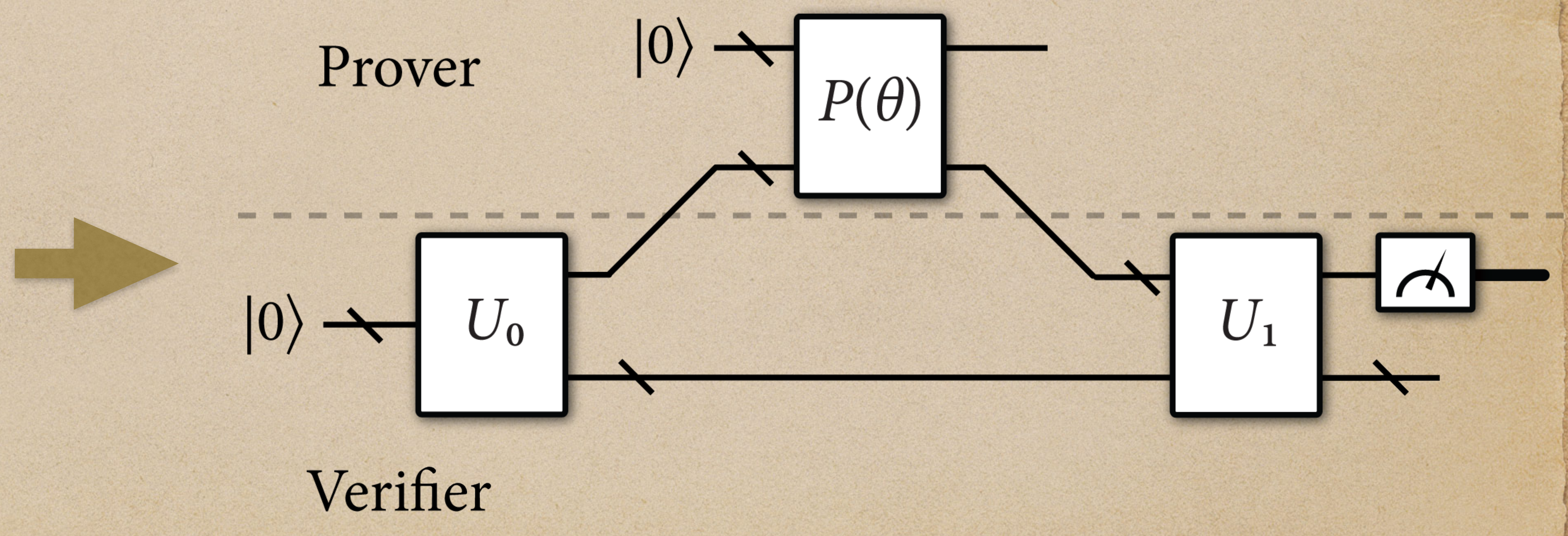
$$\max_{\rho,\sigma \in \text{States}} F(\mathcal{N}(\rho), \mathcal{M}(\sigma))$$

Quantum Interactive Proofs and VQAs

- Follow same reasoning as before & replace actions of prover with parameterized circuits, & set acceptance probability as reward function



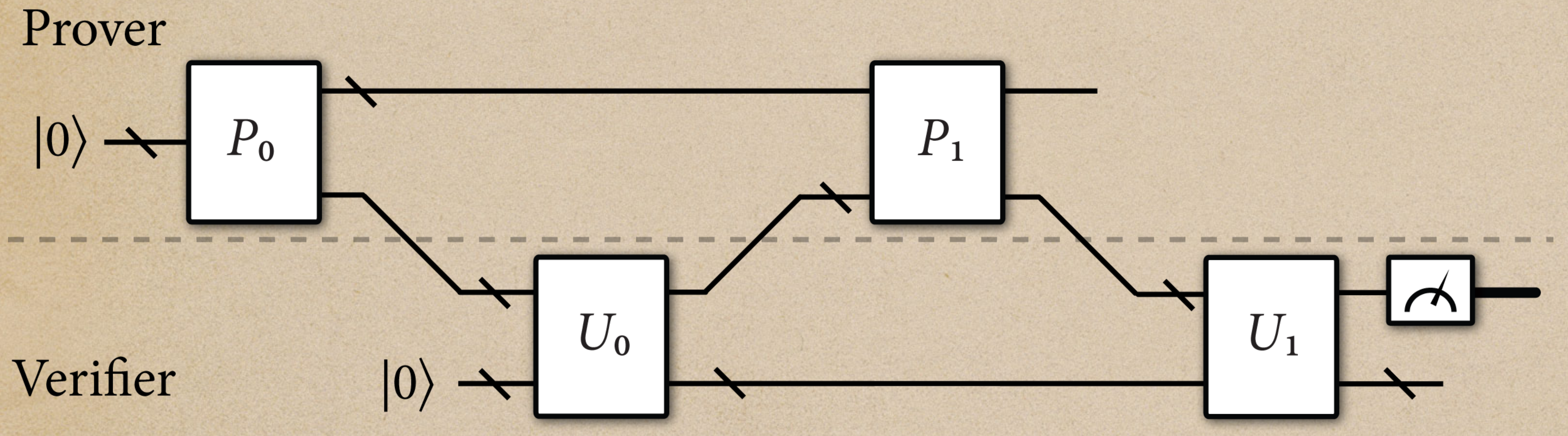
QIP(2)



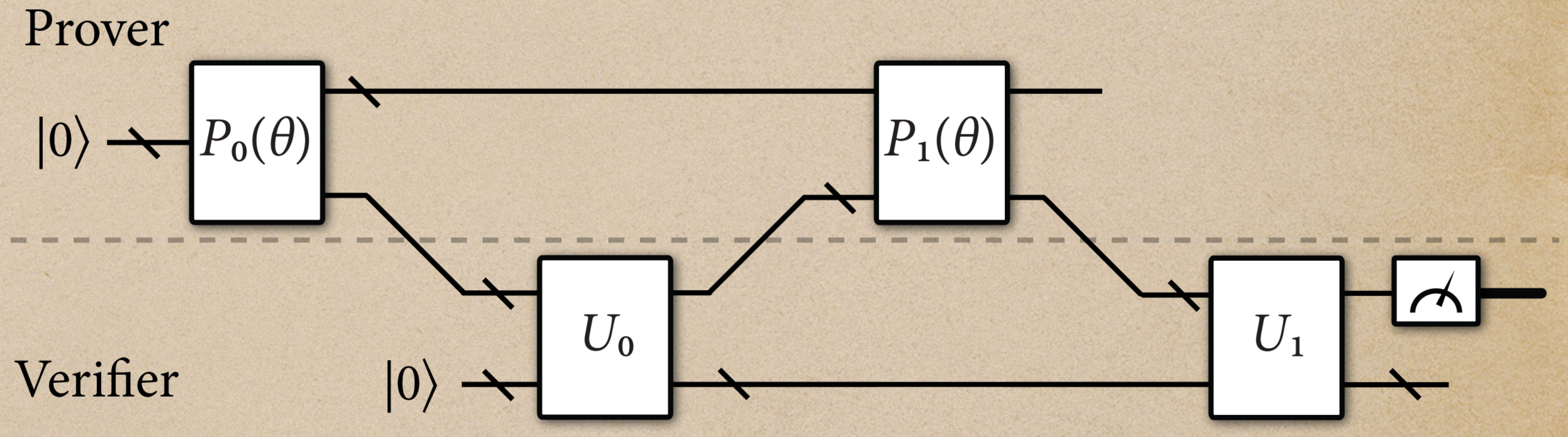
VQA

Quantum Interactive Proofs and VQAs

- Can do the same for QIP(3)



QIP(3)



VQA

Other Applications of VQAs

- Estimating distinguishability measures arXiv:2108.08406 (w/ Rethinasamy, Agarwal, Sharma)
- Symmetry testing arXiv:2105.12758, arXiv:2203.10017 (w/ LaBorde)

VQAs for Estimating Distinguishability Measures

State Distinguishability Measures

- Trace distance:

$$\| \rho - \sigma \|_1$$

for states ρ and σ, where $\| A \|_1 \equiv \mathrm{Tr}\left[\sqrt{A^\dagger A}\right]$

- Fidelity:

$$F(\rho, \sigma) \equiv \left\| \sqrt{\rho}\sqrt{\sigma} \right\|_1^2$$

- These measures give a sense of how close or far two states are
- Used all throughout quantum information science

Distinguishability Measures as Optimizations

- Can write both of these measures as optimizations:
- $\frac{1}{2} \| \rho - \sigma \|_1 = \max_{\Lambda : 0 \leq \Lambda \leq I} \mathrm{Tr}[\Lambda(\rho - \sigma)]$
- $F(\rho, \sigma) = \max_U \left| \langle \psi^\rho | U \otimes I | \psi^\sigma \rangle \right|^2$, where ψ^ρ and ψ^σ purify ρ and σ
- This suggests using VQAs to evaluate them for unknown states

VQA for estimating trace distance

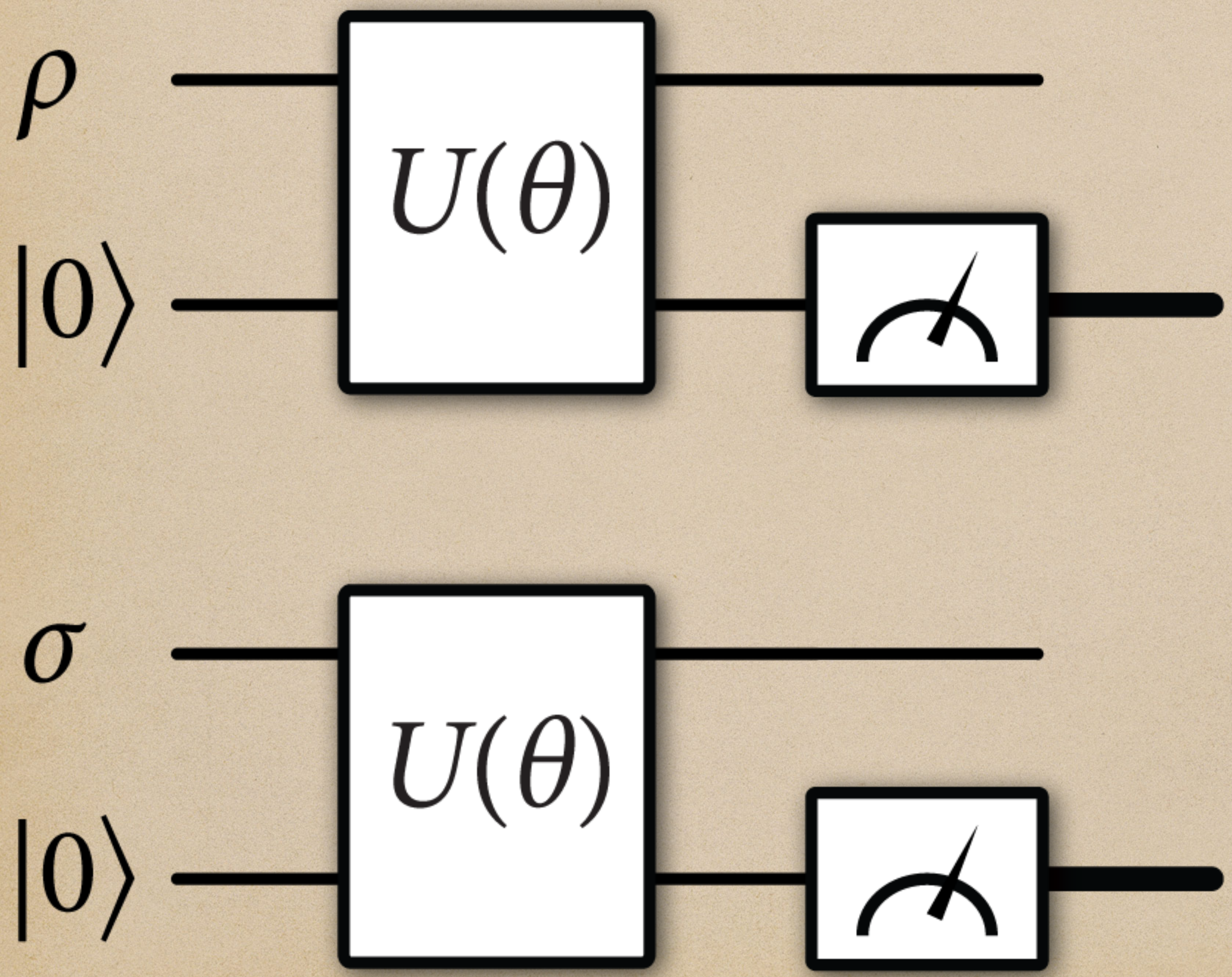
- Naimark extension theorem states that for every measurement operator Λ, $\exists$ a unitary U acting on a larger Hilbert space such that

$$\text{Tr}[\Lambda\rho] = \text{Tr}\left[(I \otimes |0\rangle\langle 0|)U(\rho \otimes |0\rangle\langle 0|)U^\dagger\right]$$

- Use this idea to formulate a VQA for estimating trace distance

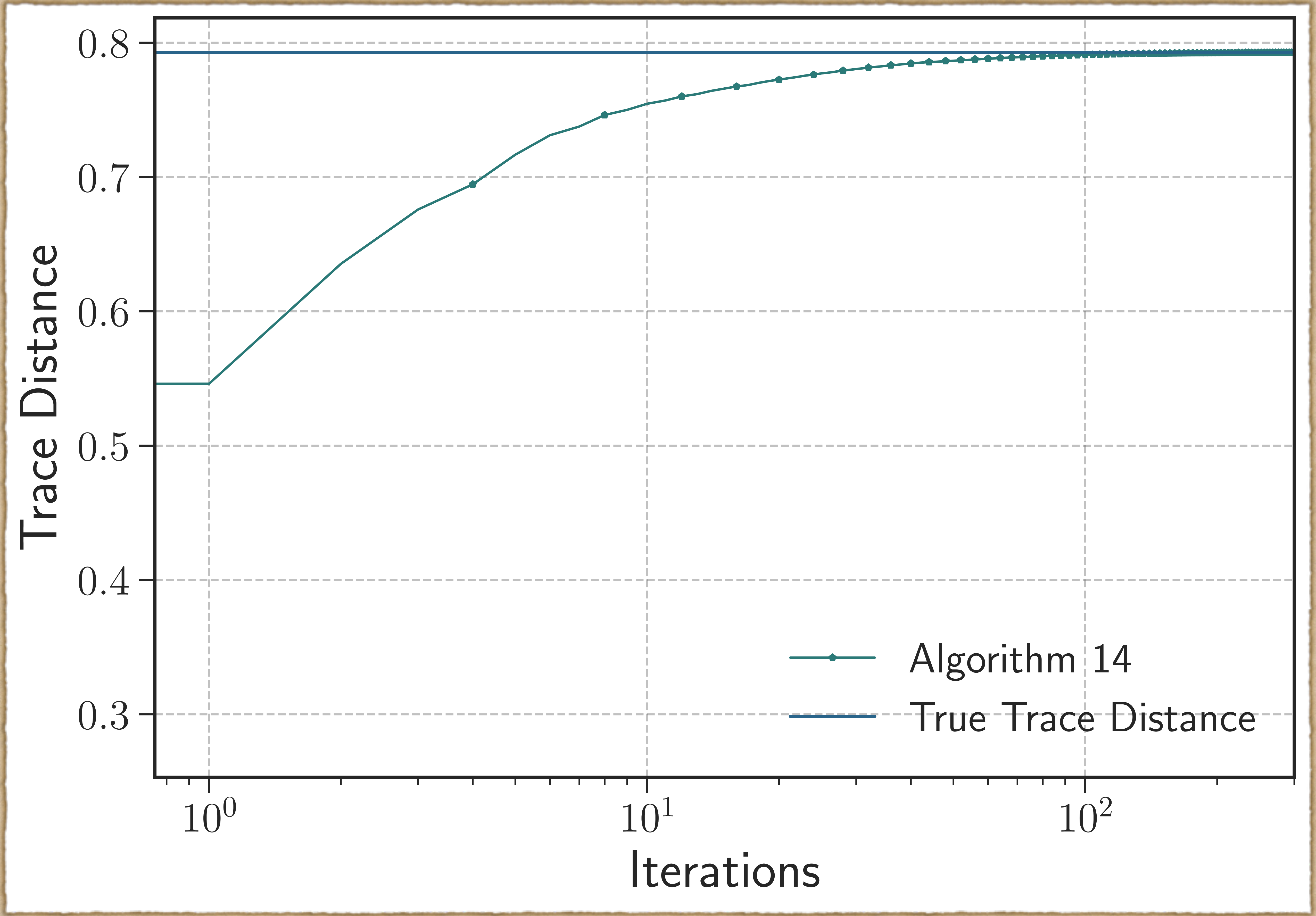
- Rather than optimize over states, optimize over measurement operators

VQA for estimating trace distance



- First circuit estimates $\mathrm{Tr}[\Lambda\rho]$ and second estimates $\mathrm{Tr}[\Lambda\sigma]$
- Reward function is $\mathrm{Tr}[\Lambda(\rho - \sigma)]$
- Can use in a VQA to estimate trace distance

Performance of trace distance estimation



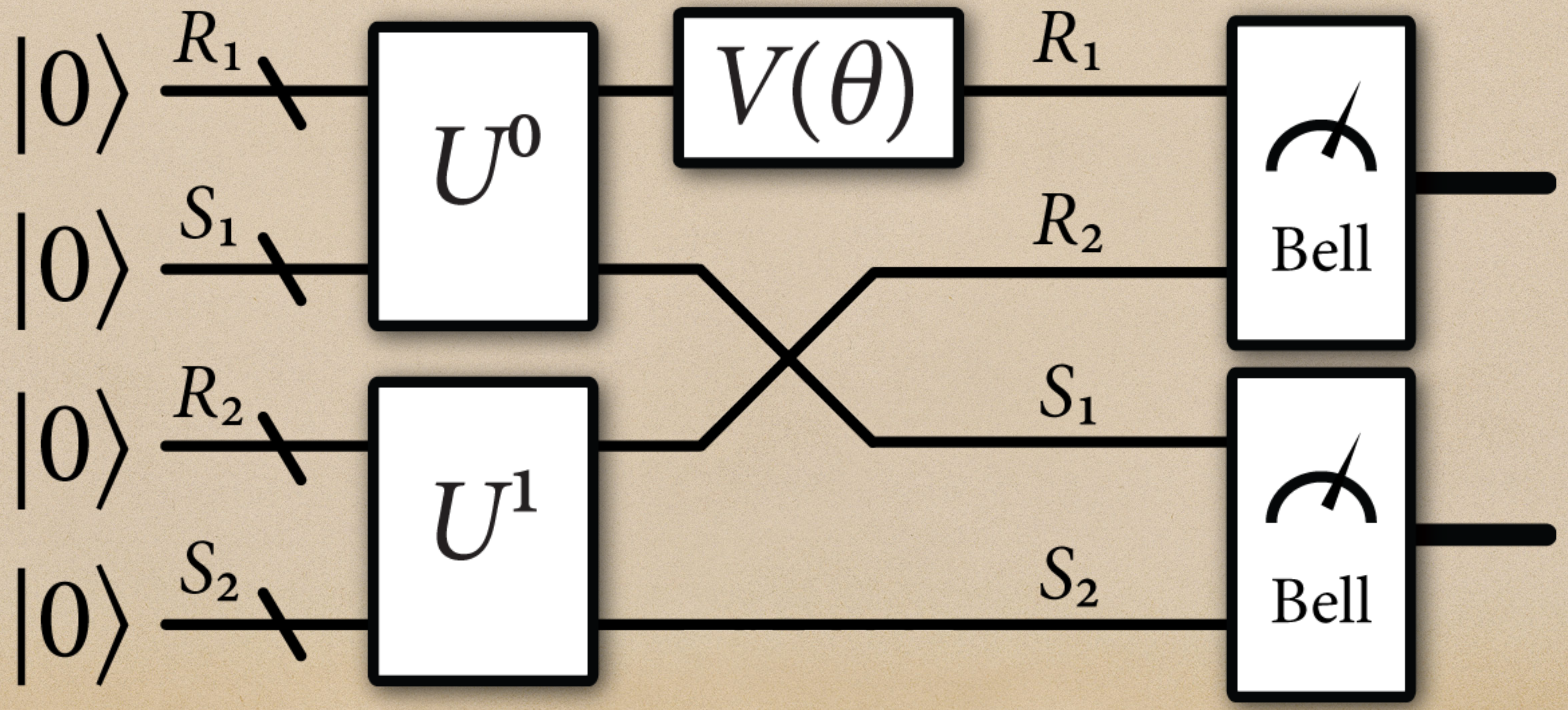
3-qubit states generated randomly using hardware efficient ansatz

VQAs for Estimating State Fidelity

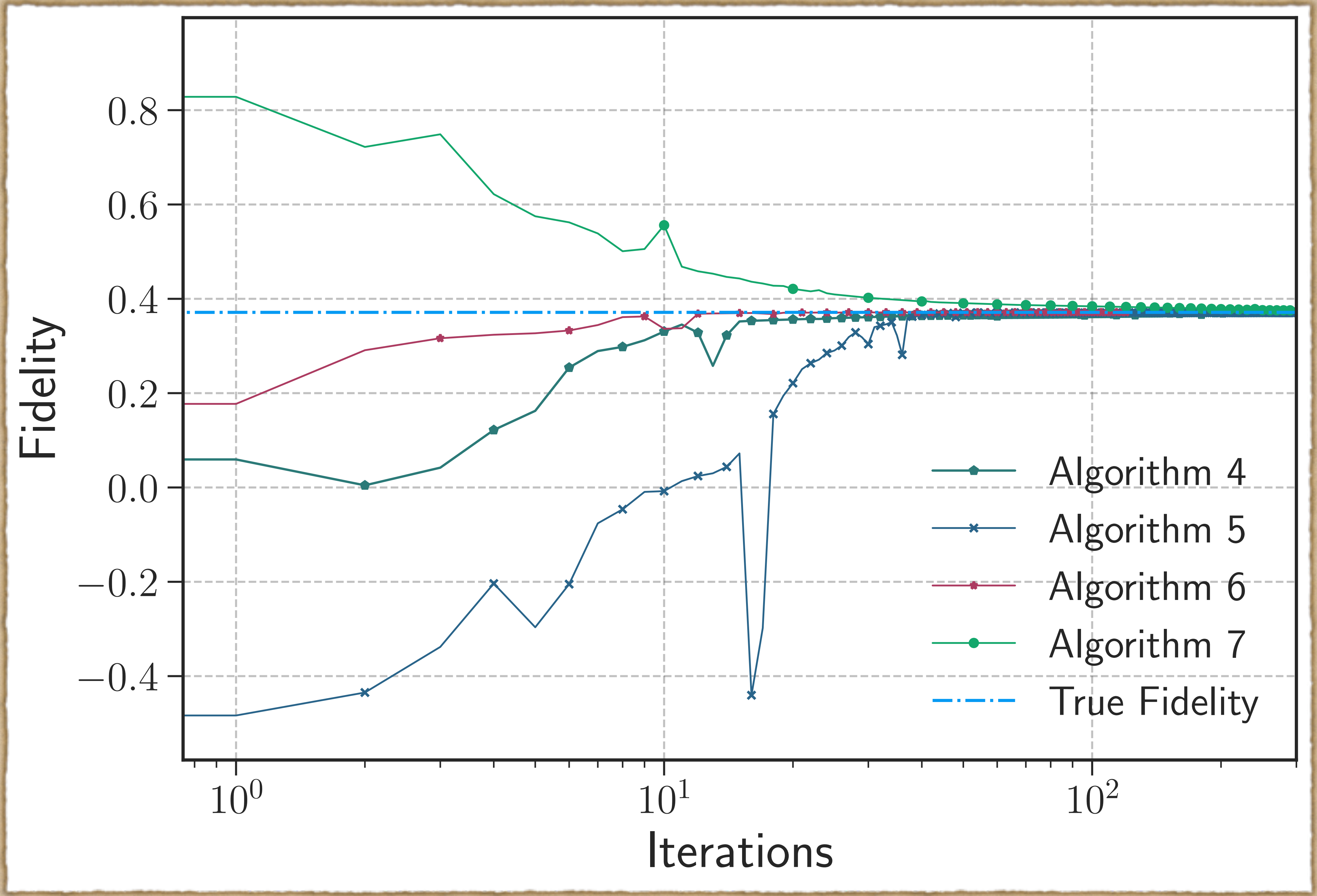
- We proposed many VQAs for estimating state fidelity
- Let us discuss the approach that gives the best performance
- $F(\psi, \phi) = |\langle \psi | \phi \rangle|^2$ for pure states ψ and ϕ
- Consider that $|\langle \psi | \phi \rangle|^2 = \mathrm{Tr}[\mathrm{SWAP}(|\psi\rangle\langle\psi| \otimes |\phi\rangle\langle\phi|)]$
  and $\mathrm{SWAP} = |\Phi^+\rangle\langle\Phi^+| + |\Phi^-\rangle\langle\Phi^-| + |\Psi^+\rangle\langle\Psi^+| - |\Psi^-\rangle\langle\Psi^-|$
- Can then estimate pure-state fidelity by repeatedly performing Bell measurements on $|\psi\rangle\langle\psi| \otimes |\phi\rangle\langle\phi|$

VQA for Estimating State Fidelity

- Can use the optimization formula for fidelity along with SWAP observation to propose VQA for estimating fidelity:



Performance of state fidelity estimation



- Algorithm 6 is the one we discussed
- All estimated using noiseless simulator
- 3-qubit states generated randomly using hardware efficient ansatz

Channel Distinguishability Measures

- Concepts can be generalized to channel fidelity, diamond distance, and multiple state discrimination:

$$F(\mathcal{N}, \mathcal{M}) \equiv \min_{\rho_{RA}} F((\mathrm{id}_R \otimes \mathcal{N})(\rho_{RA}), (\mathrm{id}_R \otimes \mathcal{M})(\rho_{RA}))$$

$$\| \mathcal{N} - \mathcal{M} \|_{\diamond} \equiv \max_{\rho_{RA}} \| (\mathrm{id}_R \otimes \mathcal{N})(\rho_{RA}) - (\mathrm{id}_R \otimes \mathcal{M})(\rho_{RA}) \|_1$$

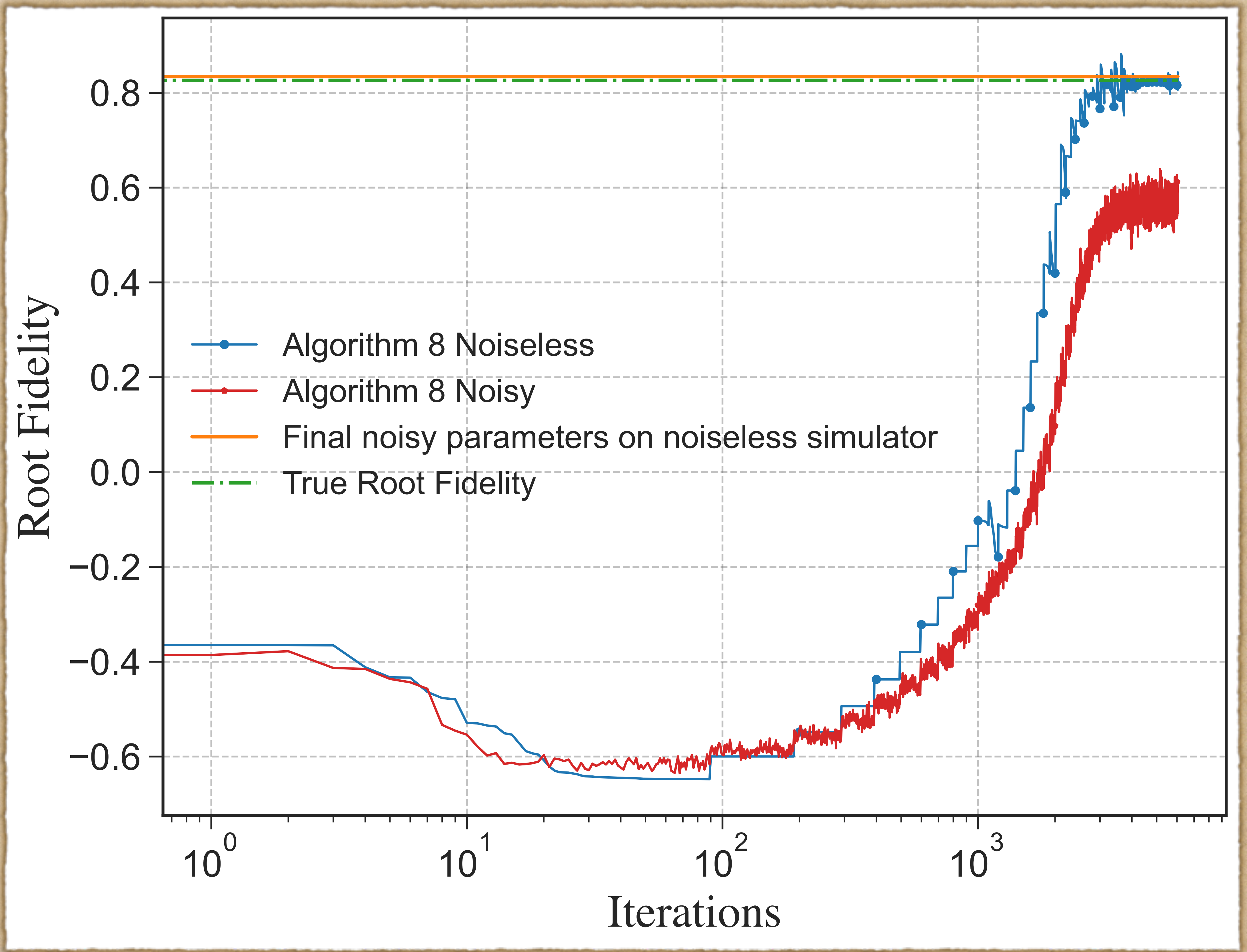
$$p_{\mathrm{succ}}((p(x), \rho_x)_x) \equiv \max_{(\Lambda_x)_x} \sum_x p(x) \mathrm{Tr}[\Lambda_x \rho_x]$$

- Can also build VQAs for these tasks

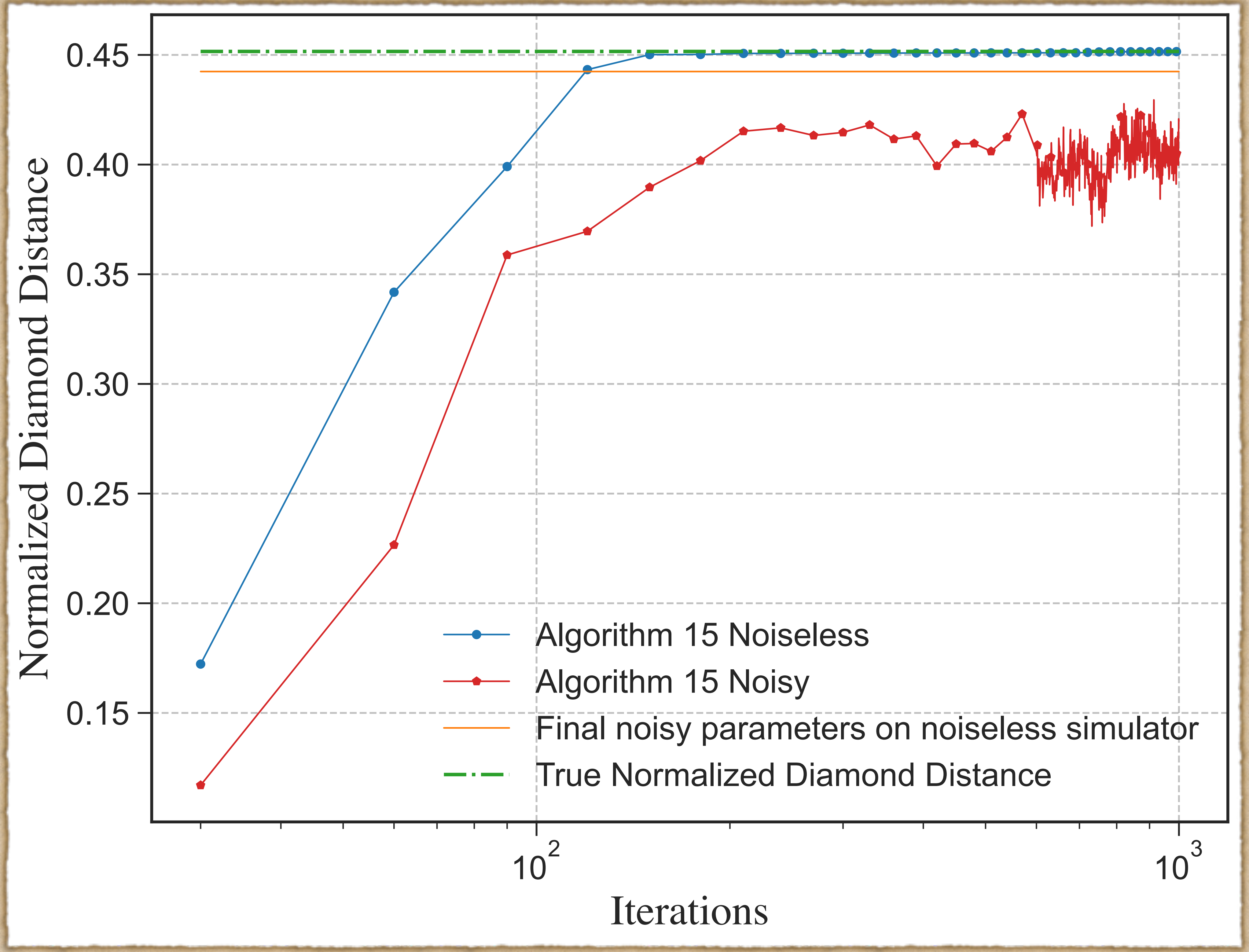
Performance of channel fidelity estimation

- Unitary dilations of two-qubit channels generated randomly using hardware efficient ansatz

- Noise resilience - training still occurs in the presence of noise

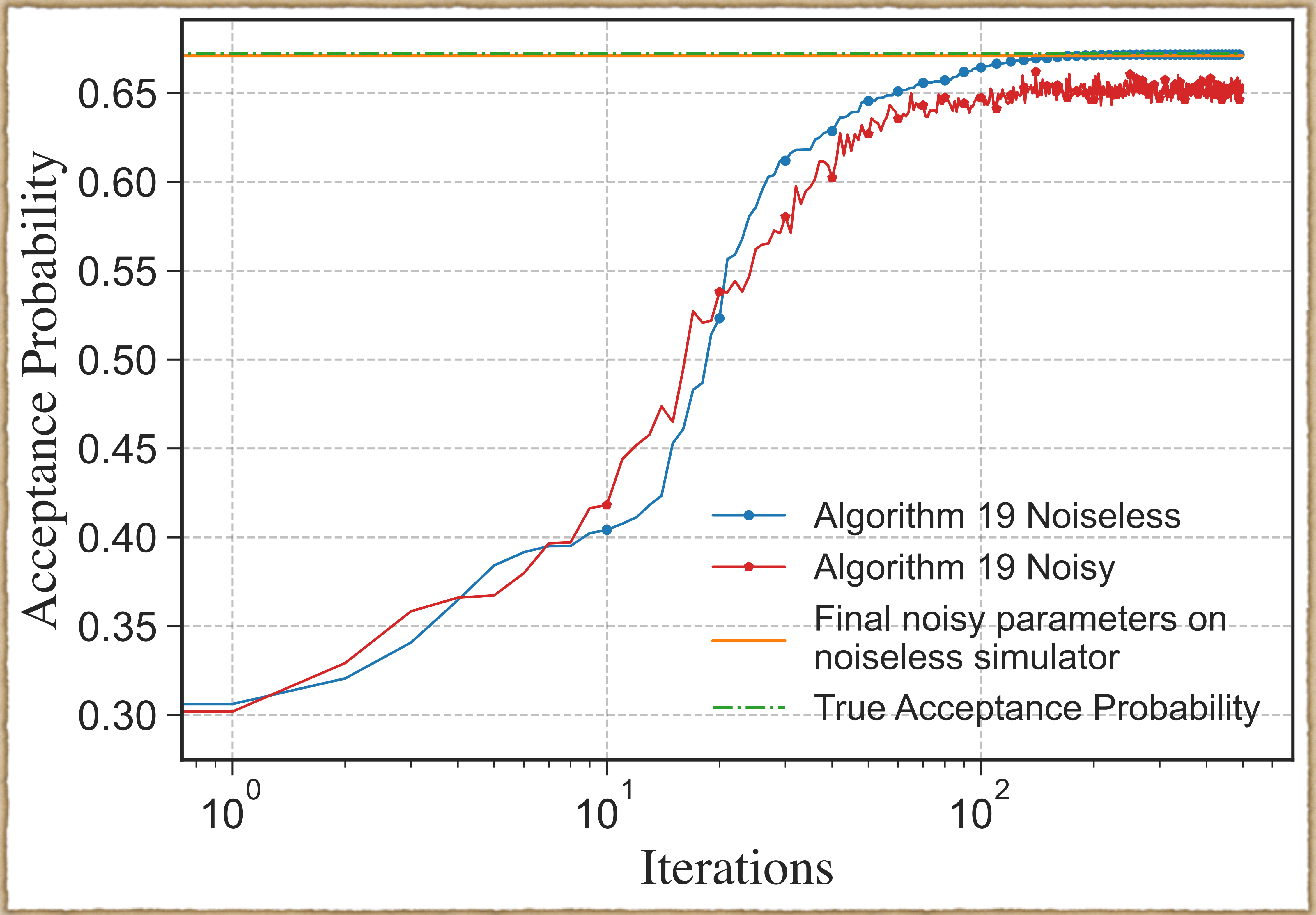


Performance of diamond distance estimation



Unitary dilations of one-qubit channels generated randomly using hardware efficient ansatz

Performance of multiple state discrimination



Three one-qubit mixed states generated randomly using hardware efficient ansatz

VQAs for Symmetry Testing

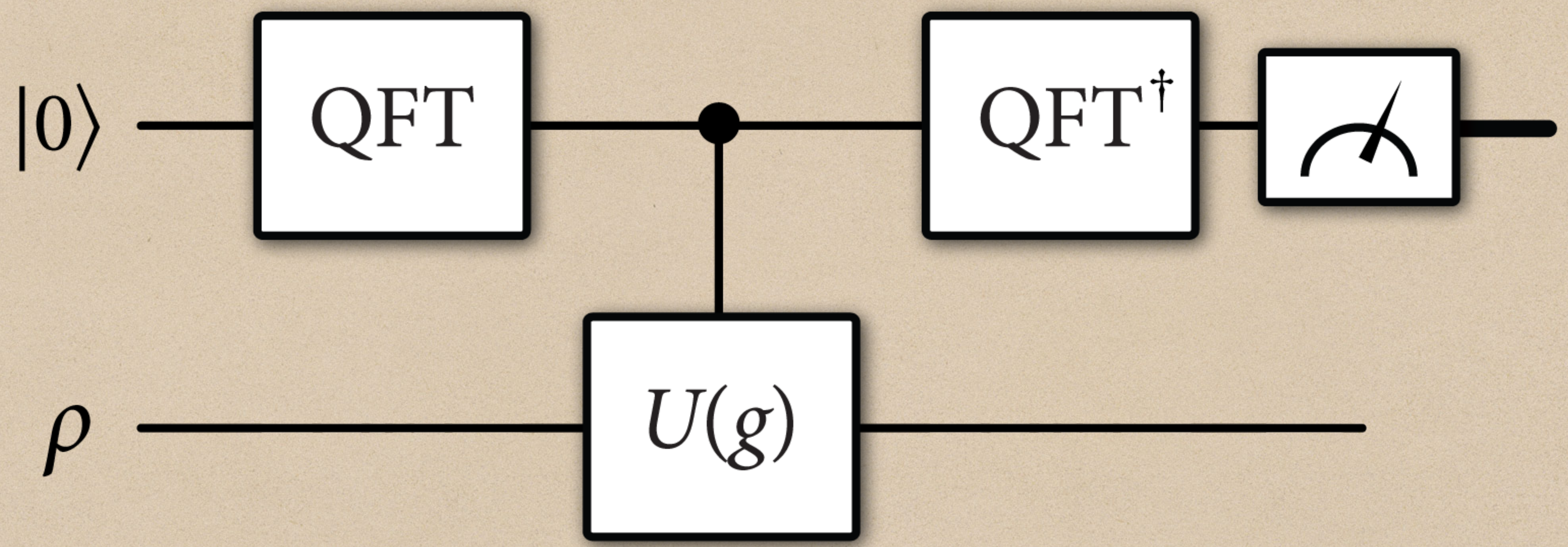
Notions of Symmetry

- Let $\{U(g)\}_{g \in G}$ denote a unitary representation of a group G
- Let $\Pi_G \equiv \frac{1}{|G|} \sum_{g \in G} U(g)$ denote the group projection
- A state ρ is G-Bose symmetric if $\rho = \Pi_G \rho \Pi_G$
- A state ρ is G-symmetric if $[U(g), \rho] = 0$ for all $g \in G$
- A Hamiltonian H is G-symmetric if $[U(g), H] = 0$ for all $g \in G$

Efficient Algorithm for G-Bose Symmetry Testing

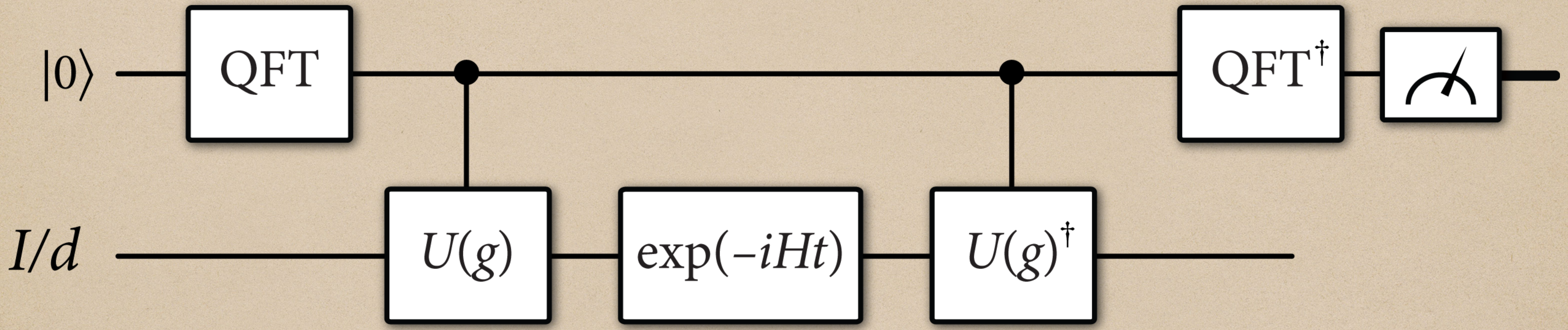
- Assumption: $\exists$ efficient circuit implementing $U(g)$ for all $g \in G$

Efficient Algorithm for G-Bose Symmetry Testing



- Accept if measurement gives all zeros outcome
- Algorithm's acceptance probability $= \mathrm{Tr}[\Pi_G \rho]$
- $\mathrm{Tr}[\Pi_G \rho] = 1$ if and only if ρ is G-Bose symmetric

Efficient Algorithm for Hamiltonian Symmetry Testing



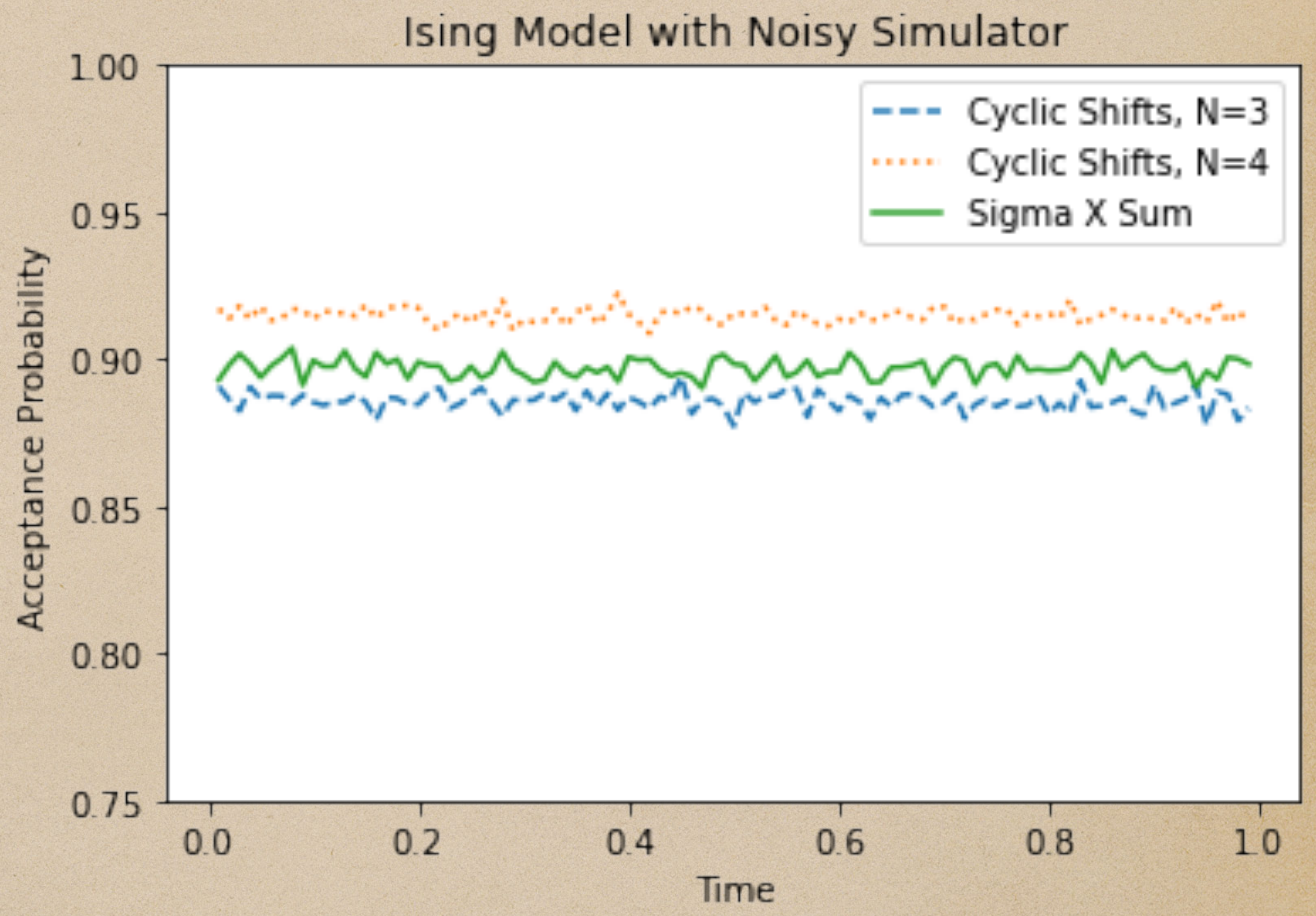
- Acceptance probability $= 1 - \frac{t^2}{2d|G|}\sum_{g\in G} \| [U(g), H] \|_2^2 + O(t^4)$
- Algorithm accepts with certainty if and only if H is G-symmetric

Example: Transverse-Field Ising Model

- Transverse-field Ising model w/ periodic boundary condition:

$$H_{\text{TFIM}} \equiv \sigma_N^Z \otimes \sigma_1^Z + \sum_{i=1}^{N-1} \sigma_i^Z \otimes \sigma_{i+1}^Z + \sum_{i=1}^{N} \sigma_i^X$$

- Symmetries: $[H_{\text{TFIM}}, (\sigma^X)^{\otimes N}] = 0$ and $[H_{\text{TFIM}}, W^\pi] = 0 \quad \forall \pi \in S_N$



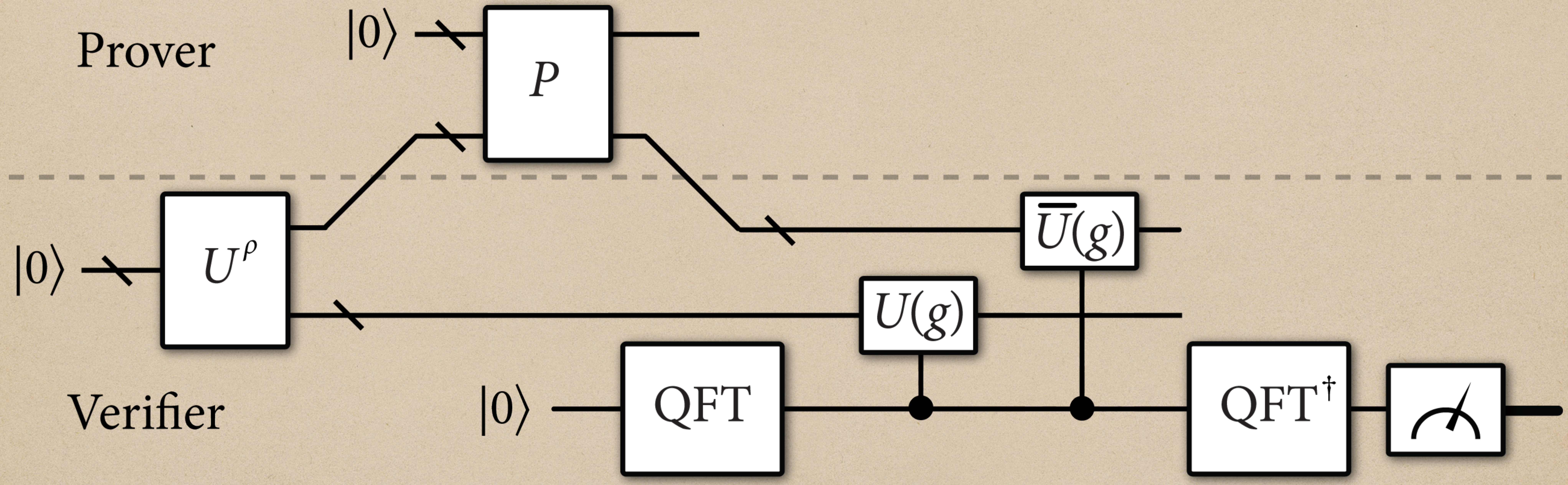
G-Symmetry Testing of States

- Suppose $\exists$ a quantum circuit that prepares a purification ψ^ρ of ρ
- Theorem: If ρ is G-symmetric, $\exists\ \psi^\rho_G$ a purification that is G-Bose symmetric, i.e., satisfying

$$|\psi^\rho_G\rangle = \overline{U}(g) \otimes U(g)|\psi^\rho_G\rangle \quad \forall g \in G$$

- Thus, if ρ is G-symmetric, $\exists$ unitary P such that $|\psi^\rho_G\rangle = P \otimes I|\psi^\rho\rangle$
- Idea: Send the purifying system to the prover, and then do a test to check for G-Bose symmetry of the resulting state

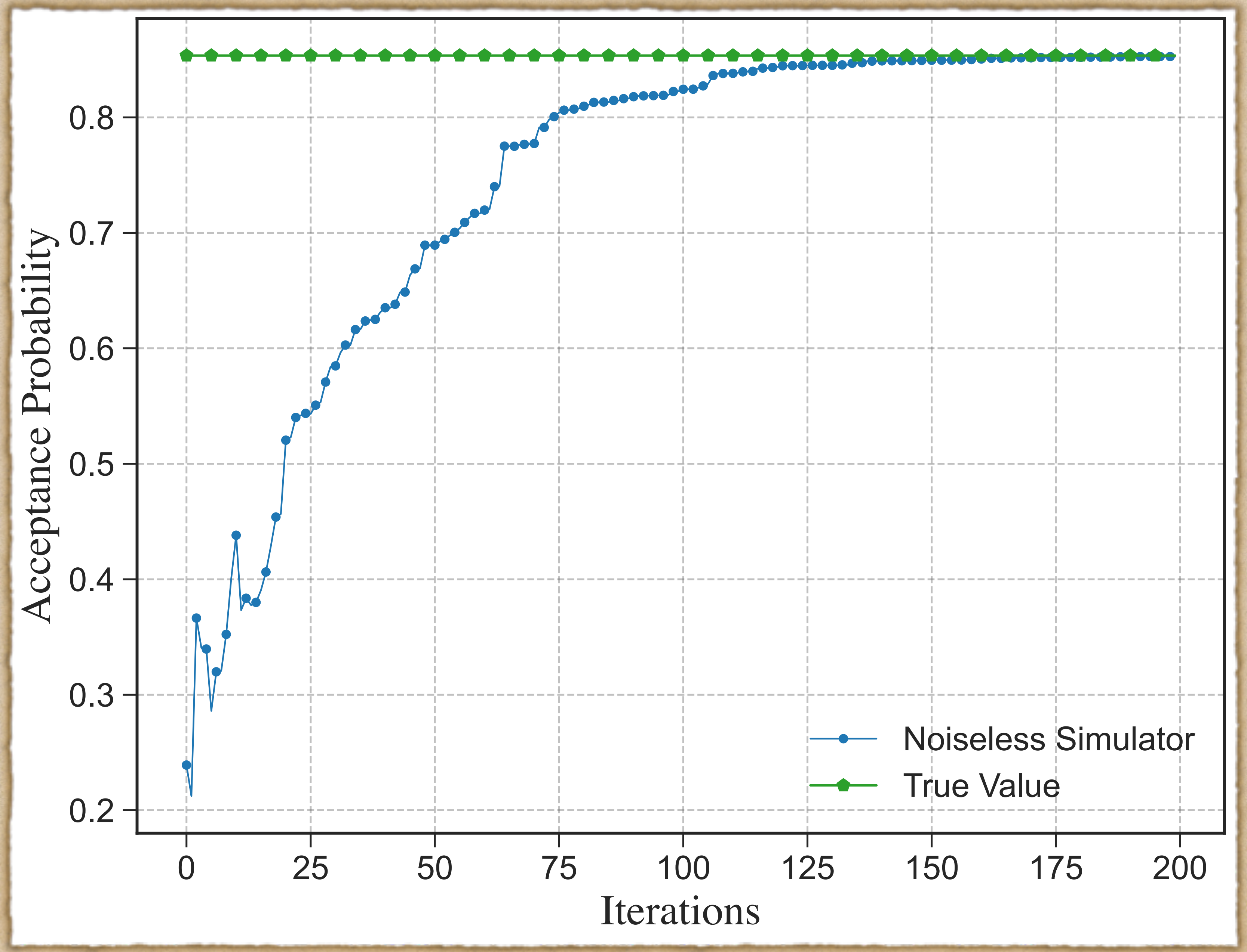
VQA for State G-Symmetry Testing



- Acceptance probability $= \max_{\sigma \in \text{Sym}} F(\rho, \sigma)$, where Sym denotes the set of G-symmetric states

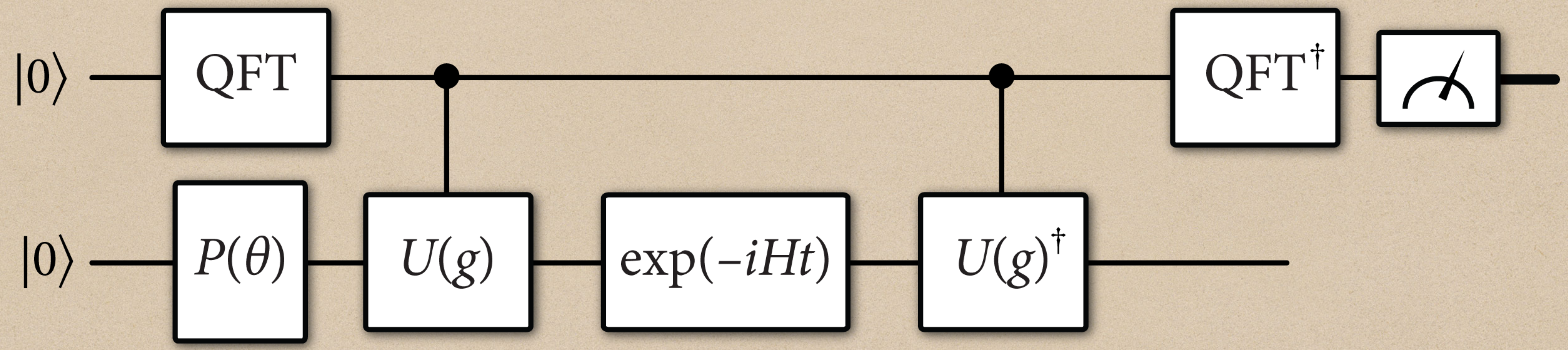
Example: Testing Invariance under Collective Rotations

- Rotationally invariant state ρ satisfies $[\rho, R_Z(\phi) \otimes R_Z(\phi)] = 0 \quad \forall \phi \in [0,2\pi]$
- Plot shows result of test on two-qubit state randomly generated using hardware efficient ansatz



VQA for Hamiltonian Symmetry Testing

- By modifying the previous Hamiltonian symmetry testing algorithm to optimize over input states, we get a VQA for this task:



- Acceptance probability

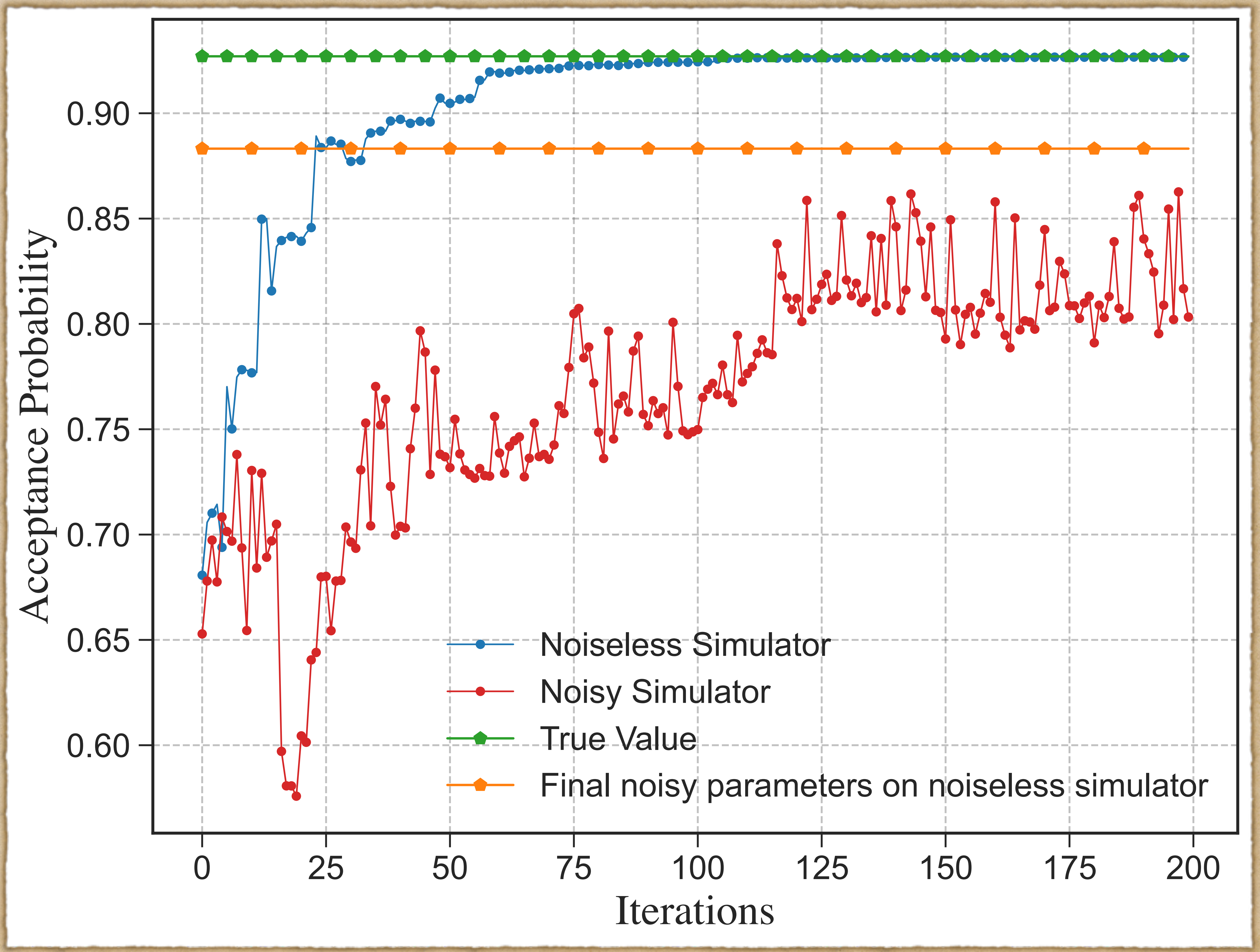
$$= \left\| \frac{1}{|G|} \sum_{g \in G} U(g) e^{-iHt} U(g)^\dagger \right\|_\infty^2 \geq 1 - \frac{2t}{|G|} \sum_{g \in G} \left\| \left[U(g), H \right] \right\|_\infty - O(t^2)$$

Example of Separability Testing

- We can use state symmetry testing for entanglement detection
- A bipartite state ρ_{AB} is k-extendible if 1) $\exists \omega_{AB_1 \cdots B_k}$ such that $\mathrm{Tr}_{B_2 \cdots B_k}[\omega] = \rho$ and 2) $[\omega_{AB_1 \cdots B_k}, I_A \otimes W^{\pi}_{B_1 \cdots B_k}] = 0 \quad \forall \pi \in S_k$
- Symmetry group in this case is the symmetric group
- Every separable (unentangled) state is k-extendible for all k

Example of Separability Testing

- Testing for 2-extendibility
- Two-qubit state generated randomly using hardware efficient ansatz



All Python code freely available

- SDPs - https://github.com/Dhrumil2910/Variational-Quantum-Algorithms-for-Semidefinite-Programming
- Symmetry testing - https://github.com/mlabo15/Hamiltonian-Symmetry
- Estimating distinguishability measures - https://arxiv.org/src/2108.08406v2/anc

Summary

- VQAs constitute an optimization method for the NISQ era (but can also be used in the fault-tolerant era)
- Main idea is to use quantum computer for the simple task of estimating expectations of observables
- Leave all other calculations to classical computers
- We discussed applications to semidefinite programming, estimating distinguishability measures, and symmetry testing

Outlook

- How will these algorithms scale as the NISQ era proceeds?
- Is there a way to give a guaranteed runtime?
- Can we prove that these algorithms will give a quantum advantage?
- Can we prove that certain problem instances can be solved in BQP?
- What other problems can we solve using the VQA paradigm?
- How do different q. computing platforms compare when running these algorithms?