Optimal universal programming of unitary gates

Yuxiang Yang

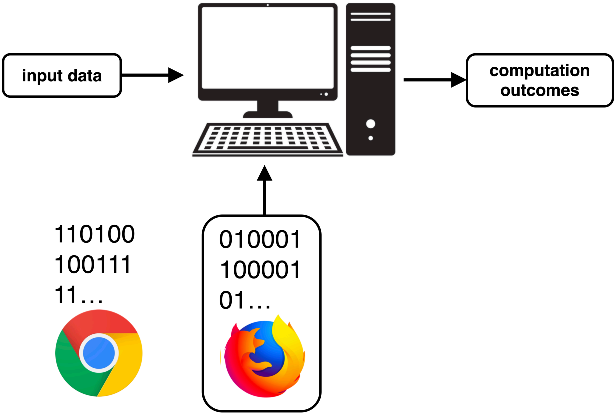
Institute for Theoretical Physics, ETH Zürich

Joint work with Renato Renner and Giulio Chiribella

PRL **125,** 210501 (Editors' suggestion; arXiv 2007.10363)

QIP 2021, Munich

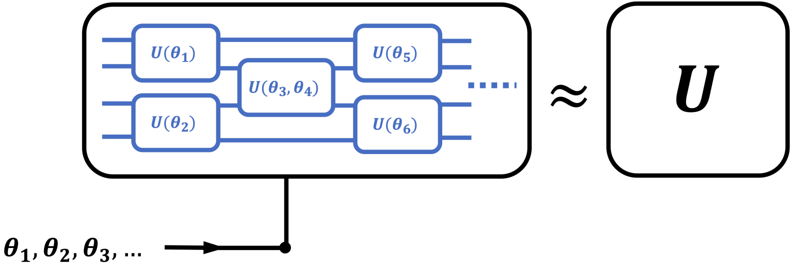
Programming



"010001..." → programs of functions.

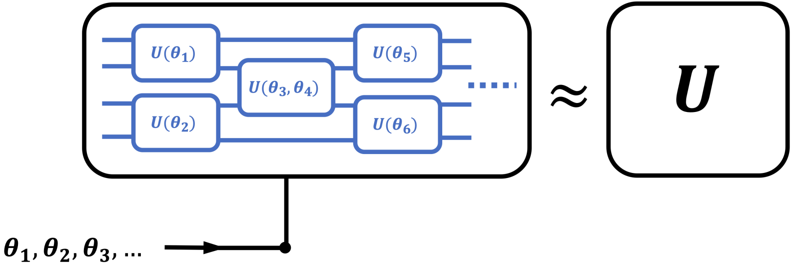
Programming quantum computers

Example: variational approach.



Programming quantum computers

Example: variational approach.



Programming cost: how many bits do we need for the program $(\theta_1, \theta_2, \theta_3, \ldots)$ to achieve desired accuracy?

Programming quantum computers

Solovay-Kitaev algorithm [Kitaev'97;Dawson-Nielson'05]

For a universal gate set **G** and any $\epsilon > 0$, any $U \in \mathsf{SU}(d)$ can be ϵ-approximated by a sequence of gates in **G** of length $\sim (\log_2(1/\epsilon))^c$. Here $c \approx 3.97$ is admissible.

Programming quantum computers

Solovay-Kitaev algorithm [Kitaev'97;Dawson-Nielson'05]

For a universal gate set **G** and any $\epsilon > 0$, any $U \in \mathrm{SU}(d)$ can be ϵ-approximated by a sequence of gates in **G** of length $\sim (\log_2(1/\epsilon))^c$. Here $c \approx 3.97$ is admissible.

- Programming from universal gate set (synthesis) requires bit strings of length $\rightarrow \log_2 |\mathbf{G}| \cdot (\log_2(1/\epsilon))^c$ (prefactor could be smaller).

Programming quantum computers

Solovay-Kitaev algorithm [Kitaev'97;Dawson-Nielson'05]

For a universal gate set **G** and any $\epsilon > 0$, any $U \in \mathrm{SU}(d)$ can be ϵ-approximated by a sequence of gates in **G** of length $\sim (\log_2(1/\epsilon))^c$. Here $c \approx 3.97$ is admissible.

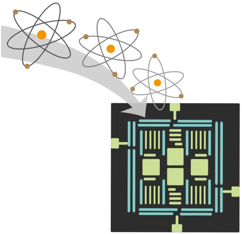
- Programming from universal gate set (synthesis) requires bit strings of length $\rightarrow \log_2 |\mathbf{G}| \cdot (\log_2(1/\epsilon))^c$ (prefactor could be smaller).
- Exact ($\epsilon = 0$) programming? Smaller cost?

Programming quantum computers

Solovay-Kitaev algorithm [Kitaev'97;Dawson-Nielson'05]

For a universal gate set **G** and any $\epsilon > 0$, any $U \in \mathrm{SU}(d)$ can be ϵ-approximated by a sequence of gates in **G** of length $\sim (\log_2(1/\epsilon))^c$. Here $c \approx 3.97$ is admissible.

- Programming from universal gate set (synthesis) requires bit strings of length $\rightarrow \log_2 |\mathbf{G}| \cdot (\log_2(1/\epsilon))^c$ (prefactor could be smaller).
- Exact ($\epsilon = 0$) programming?
  Smaller cost?
  Quantum programs?

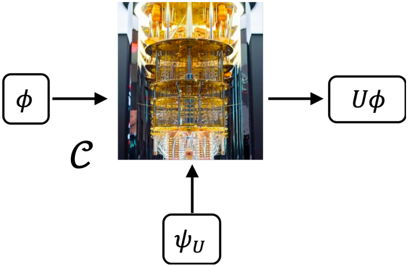


Quantum universal programmability

- Implement any unitary U of dim d in a programmable way.

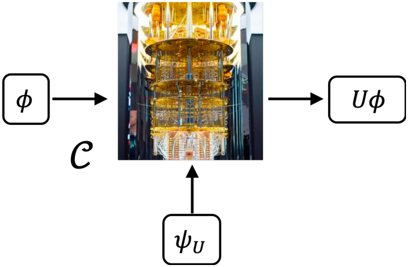
Quantum universal programmability

- Implement any unitary U of dim d in a programmable way.
- Task: to design $(\mathcal{C}, \{|\psi_U\rangle\})$.



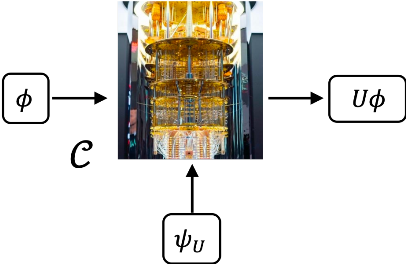
Quantum universal programmability

- Implement any unitary U of dim d in a programmable way.
- Task: to design $(\mathcal{C}, \{|\psi_U\rangle\})$.
- Input to the universal processor arbitrary $|\phi\rangle$ and a program state $|\psi_U\rangle$; output $U|\phi\rangle$.



Quantum universal programmability

- Implement any unitary U of dim d in a programmable way.
- Task: to design $(\mathcal{C}, \{|\psi_U\rangle\})$.
- Input to the universal processor arbitrary $|\phi\rangle$ and a program state $|\psi_U\rangle$; output $U|\phi\rangle$.



No programming (Nielsen and Chuang '97)

For any $d \geq 2$, to allow for programming arbitrary $U \in \mathrm{SU}(d)$, $|\psi_U\rangle$ needs to be infinite dimensional.

Approximate programmability

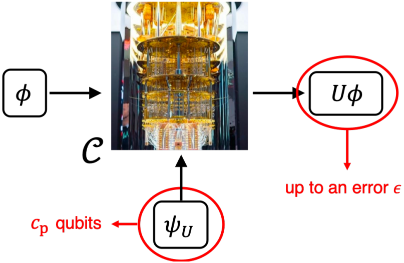
- Allow an error $\leq \epsilon$ (diamond norm) in the output.

Approximate programmability

- Allow an error $\leq \epsilon$ (diamond norm) in the output.
- Programming cost (# qubits needed to describe U)

$$c_P := \log_2 |\mathrm{span}\{|\psi_U\rangle\langle\psi_U|\}_U|$$

can be finite (via, e.g., Solovay-Kitaev).

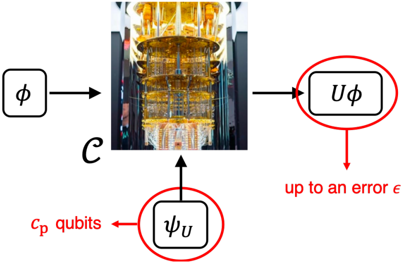


Approximate programmability

- Allow an error $\leq \epsilon$ (diamond norm) in the output.
- Programming cost (# qubits needed to describe U)

  $$c_{\mathsf{P}} := \log_2 |\mathrm{span}\{|\psi_U\rangle\langle\psi_U|\}_U|$$

  can be finite (via, e.g., Solovay-Kitaev).
- Allowing a chance of failure also works [Vidal-Masanes-Cirac-2002-PRL].



Optimal programming

The optimal programming problem

Consider programming of unitary gates of dimension d.
What is the optimal tradeoff between ϵ and c_{P}?

Optimal programming

The optimal programming problem

Consider programming of unitary gates of dimension d.
What is the optimal tradeoff between ϵ and c_{P}?

- An open problem for 23 years.

  Nielsen and Chuang, Phys. Rev. Lett. 79.321 (1997);
  Kim et al., Phys. Rev. A 65, 012302 (2001)
  Ishizaka and Hiroshima, Phys. Rev. Lett. 101, 240501 (2008);
  Kubicki, Palazuelos, and Perez-Garcıa, Phys. Rev. Lett. 122, 080505 (2019); (QIP'19)
  Sedlak, Bisio, and Ziman, Phys. Rev. Lett. 122, 170502 (2019)

  ...

Optimal programming

The optimal programming problem

Consider programming of unitary gates of dimension d.
What is the optimal tradeoff between ϵ and c_{P}?

- An open problem for 23 years.

  Nielsen and Chuang, Phys. Rev. Lett. 79.321 (1997);
  Kim et al., Phys. Rev. A 65, 012302 (2001)
  Ishizaka and Hiroshima, Phys. Rev. Lett. 101, 240501 (2008);
  Kubicki, Palazuelos, and Perez-Garcıa, Phys. Rev. Lett. 122, 080505 (2019); (QIP'19)
  Sedlak, Bisio, and Ziman, Phys. Rev. Lett. 122, 170502 (2019)

  ...

- Resource quantification of implementing quantum computing. Also a benchmark for any practical approach.

Optimal programming

The optimal programming problem

Consider programming of unitary gates of dimension d. What is the optimal tradeoff between ϵ and c_{P}?

- Leads to port-based teleportation [Ishizaka-Hiroshima-2008-PRL] see also [Studziński-Mozrzymas-Kopszak-Horodecki-QIP'21], which has applications in cryptography [Christandl et al.-2020-CMP] and computing [Beigi-König-2011-NJP].

Optimal programming

The optimal programming problem

Consider programming of unitary gates of dimension d. What is the optimal tradeoff between ϵ and c_{P}?

- Leads to port-based teleportation [Ishizaka-Hiroshima-2008-PRL] see also [Studziński-Mozrzymas-Kopszak-Horodecki-QIP'21], which has applications in cryptography [Christandl et al.-2020-CMP] and computing [Beigi-König-2011-NJP].
- Programmable measurements [Dušek-Bužek-2002-PRA; Fiurášek-Dušek-Filip-2002-PRL; D'Ariano-Perinotti-2005-PRL].

The main message of our work

- The optimal cost-accuracy[1] tradeoff identified to be

$$c_P \to \frac{d^2 - 1}{2} \log_2 \left(\frac{1}{\epsilon} \right).$$

[1] ϵ: diamond norm error

The main message of our work

- The optimal cost-accuracy[1] tradeoff identified to be

$$c_P \to \frac{d^2-1}{2}\log_2\left(\frac{1}{\epsilon}\right).$$

- Optimal programming uses genuine quantum states as programs.

[1] ϵ: diamond norm error

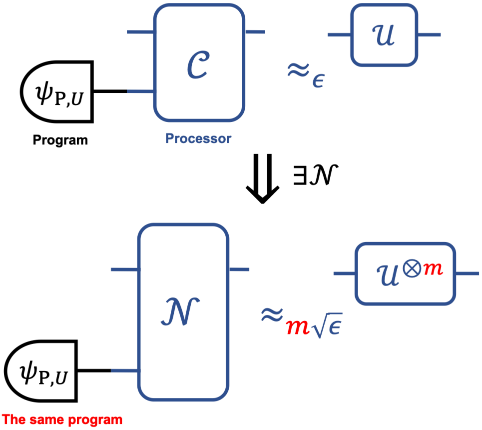
Comparison with previous results

c_P in the leading order of $1/\epsilon$.

	Upper bounds	Lower bounds
Prior works	$d^2 \log_2(1/\epsilon)$(QIP'19) [Kubicki-Palazuelos-Pérez-García-2019-PRL] $(4d^2 \log_2 d)/\epsilon^2$ port-based teleportation	$\propto (1-\epsilon)d$ [Kubicki-Palazuelos-Pérez-García-2019-PRL] $\left(\frac{d-1}{2}\right)\log_2(1/\epsilon)$ [Pérez-García-2006-PRA]
This work	$\left(\frac{d^2-1}{2}\right)\log_2(1/\epsilon)$	$\left(\frac{d^2-1}{2}\right)\log_2(1/\epsilon)^*$

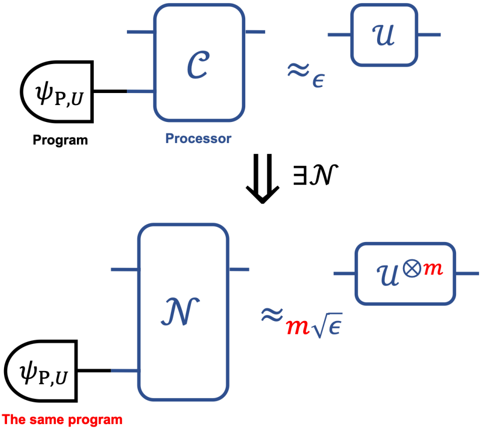
$*$: the exact scaling is $\alpha \log_2(1/\epsilon)$ for any constant $\alpha < (d^2-1)/2$.

Proof idea of lower bound on c_P (Step 1)



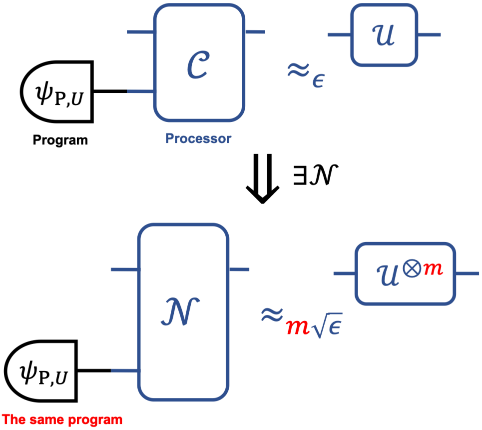
- **QR recycling Lemma:**
  [Chiribella-Yang-Renner-2019-QIP'20]
  Resource to implement a unitary can be recycled for up to $1/\sqrt{\epsilon}$ times.

Proof idea of lower bound on c_P (Step 1)



- **QR recycling Lemma:** [Chiribella-Yang-Renner-2019-QIP'20] **Resource** to implement a unitary can be recycled for up to $1/\sqrt{\epsilon}$ times.
- A quantum feature.

Proof idea of lower bound on c_P (Step 1)



- **QR recycling Lemma:** [Chiribella-Yang-Renner-2019-QIP'20]
  Resource to implement a unitary can be recycled for up to $1/\sqrt{\epsilon}$ times.
- A quantum feature.
- Can be applied to general resource theories (see [Chiribella-Yang-Renner-2019]).

Proof idea of lower bound on c_{P} (Step 2)

- $|\psi_U\rangle$ programs U to precision $1/\epsilon$
  $\Rightarrow |\psi_U\rangle$ programs $1/\sqrt{\epsilon}$ copies of U to precision $O(1)$.

Proof idea of lower bound on c_P (Step 2)

- $|\psi_U\rangle$ programs U to precision $1/\epsilon$
  $\Rightarrow |\psi_U\rangle$ programs $1/\sqrt{\epsilon}$ copies of U to precision $O(1)$.

- Intuitively, one needs a large enough ($\frac{1}{\epsilon}$-dependent) system to hold that much information.

Proof idea of lower bound on c_P (Step 2)

- $|\psi_U\rangle$ programs U to precision $1/\epsilon$
  $\Rightarrow |\psi_U\rangle$ programs $1/\sqrt{\epsilon}$ copies of U to precision $O(1)$.

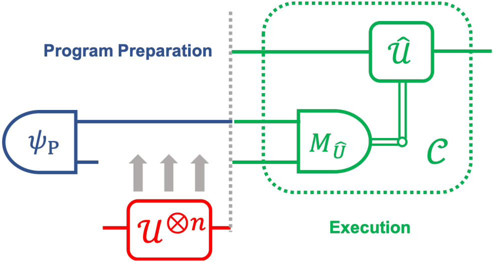
- Intuitively, one needs a large enough ($\frac{1}{\epsilon}$-dependent) system to hold that much information.

- Quantification:
  $U^{\otimes m}$ has $\approx (d^2 - 1) \log m$ qubits of information about U

  (measured by the Holevo information χ [Holevo-1973])

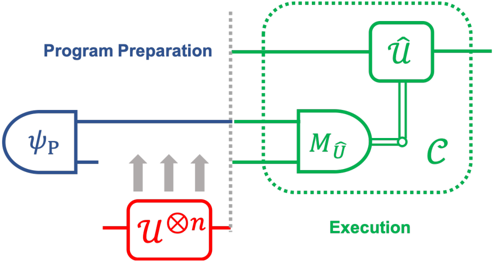
$$\Rightarrow c_P \gtrapprox \frac{d^2 - 1}{2} \log \left(\frac{1}{\epsilon} \right).$$

Upper bound: designing an optimal programmer



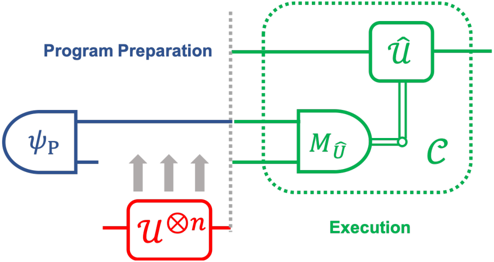
- Programs prepared by "learning" U [Bisio et al.-'10-PRA] via $n \gg 1$ instances.

Upper bound: designing an optimal programmer



- Programs prepared by "learning" U [Bisio et al.-'10-PRA] via $n \gg 1$ instances.
- Processor executes U via measure-and-operate.

Upper bound: designing an optimal programmer



- Programs prepared by "learning" U [Bisio et al.-'10-PRA] via $n \gg 1$ instances.
- Processor executes U via measure-and-operate.
- Optimal programming = Learning via Metrology
  Programs are a new class of quantum states with high performance in metrology (see SI of the paper).

Upper bound

Theorem

The program cost c_{P} of the metrological protocol is upper bounded as

$$c_{\mathsf{P}} \leq \left(\frac{d^2-1}{2}\right)\log_2\left(\frac{162\pi^2 d^2}{\epsilon}\right).$$

Upper bound

Theorem

The program cost c_{P} of the metrological protocol is upper bounded as

$$c_{\mathsf{P}} \leq \left(\frac{d^2-1}{2}\right)\log_2\left(\frac{162\pi^2 d^2}{\epsilon}\right).$$

$$\text{Upper bound} \xrightarrow{\epsilon\to 0} \text{Lower bound}$$

Conclusion & Open questions

- The optimal cost-accuracy tradeoff in programming:

$$c_{\mathsf{P}} \to \frac{d^2 - 1}{2} \log_2 \left(\frac{1}{\epsilon} \right).$$

Conclusion & Open questions

- The optimal cost-accuracy tradeoff in programming:

$$c_P \to \frac{d^2 - 1}{2} \log_2 \left(\frac{1}{\epsilon} \right).$$

- Programming a (large) quantum computer: c_P vs. d?

Conclusion & Open questions

- The optimal cost-accuracy tradeoff in programming:

$$c_{\mathsf{P}} \to \frac{d^2-1}{2}\log_2\left(\frac{1}{\epsilon}\right).$$

- Programming a (large) quantum computer: c_{P} vs. d?
- Observation: $d^2 - 1 = \#$ real parameters for $\mathrm{SU}(d)$.

Conclusion & Open questions

- The optimal cost-accuracy tradeoff in programming:

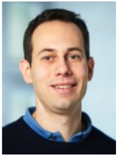
$$c_P \to \frac{d^2-1}{2} \log_2\left(\frac{1}{\epsilon}\right).$$

- Programming a (large) quantum computer: c_P vs. d?
- Observation: $d^2 - 1 = \#$ real parameters for $\mathrm{SU}(d)$.

Conjecture

To program $U \in \mathbf{S}$ with $\mathbf{S}$ parametrized by ν parameters, $c_P \to (\nu/2)\log(1/\epsilon)$.

Acknowledgement & Advertisement



Renato Renner

Giulio Chiribella

- I'm joining the University of Hong Kong as an AP in the summer of 2021.
- Looking for PhD students & Postdocs in quantum metrology and quantum information!
- Contact me: yxyang@hku.hk