Title: On nonlinear transformations in quantum computation

Speakers: Zoe Holmes

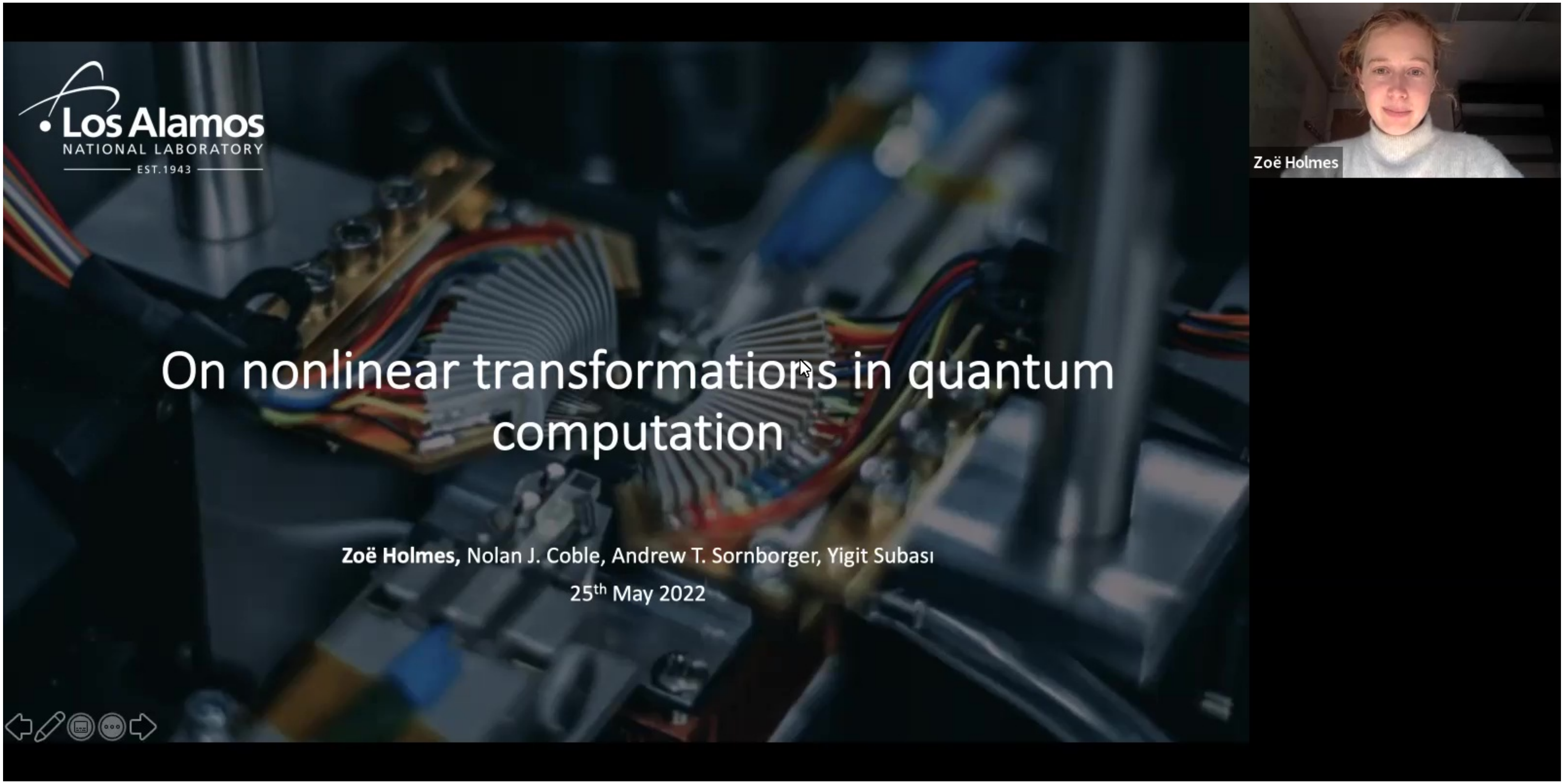
Series: Perimeter Institute Quantum Discussions

Date: May 25, 2022 - 11:00 AM

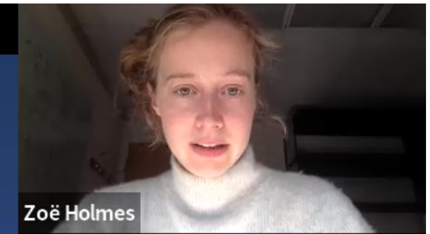
URL: https://pirsa.org/22050067

Abstract: While quantum computers are naturally well-suited to implementing linear operations, it is less clear how to implement nonlinear operations on quantum computers. However, nonlinear subroutines may prove key to a range of applications of quantum computing from solving nonlinear equations to data processing and quantum machine learning. Here we develop algorithms for implementing nonlinear transformations of input quantum states. Our algorithms are framed around the concept of a weighted state, a mathematical entity describing the output of an operational procedure involving both quantum circuits and classical post-processing.

Zoom Link: https://pitp.zoom.us/j/92831825506?pwd=T2VUQ2M2QlZERmRmUHZ0T1VOelkzZz09


Los Alamos
NATIONAL LABORATORY
EST. 1943
On nonlinear transformations in quantum computation
Zoë Holmes, Nolan J. Coble, Andrew T. Sornborger, Yigit Subası
25th May 2022
Zoë Holmes

The Challenge

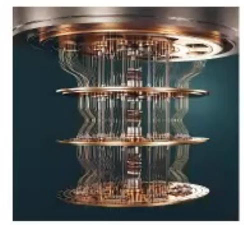


Quantum mechanics is inherently linear:

- Schroedinger's equation is linear $$i\hbar\frac{\partial\psi}{\partial t} = H\psi$$

- Unitary operations (which govern the evolution of quantum states) are linear

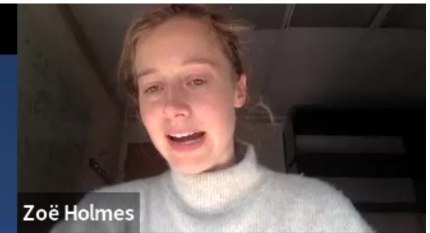
$$U(\alpha|0\rangle + \beta|1\rangle) = \alpha U|0\rangle + \beta U|1\rangle$$



Therefore quantum computers are naturally adept at implementing linear operations

But to exploit the full power of quantum computing we may need to implement non-linear operations...

The Challenge



Here we will present algorithms for implementing operations of the form:

$$|\psi\rangle = \sum_i \psi_i |i\rangle \rightarrow |f(\psi)\rangle = \sum_i f(\psi_i)|i\rangle$$

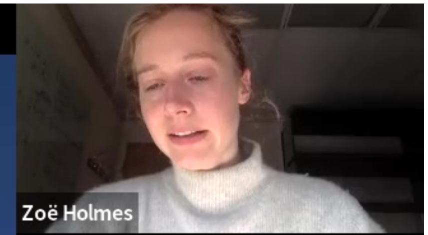
Where f may be a non-linear function.

In particular, we will assume that f can be expanded as a power series, e.g. $f(x) = \sum_j \alpha_j x^j$

Also consider functions of multiple pure states: $|\psi^{\text{in}}\rangle = |\psi^{(0)}\rangle ... |\psi^{(k)}\rangle \rightarrow |\tau\rangle = \sum_j h(\vec{v}_{\psi_{\text{in}}})_j |j\rangle$

Vector containing all amplitudes of $|\psi^{\text{in}}\rangle$

The Challenge



Here we will present algorithms for implementing operations of the form:

$$|\psi\rangle = \sum_i \psi_i |i\rangle \rightarrow |f(\psi)\rangle = \sum_i f(\psi_i)|i\rangle$$

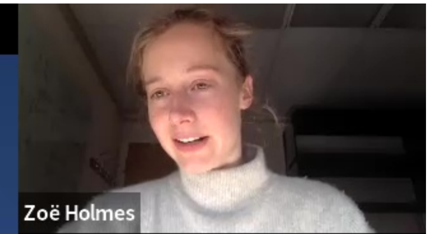
Where f may be a non-linear function.

In particular, we will assume that f can be expanded as a power series, e.g. $f(x) = \sum_j \alpha_j x^j$

Also consider functions of elements of mixed states: $\rho \rightarrow \tau = f(\rho) = \sum_{i,j=0} f(\vec{v}_\rho)_{i,j} |i\rangle\langle j|$

Vector containing all entries of ρ

The Challenge



Here we will present algorithms for implementing operations of the form:

$$|\psi\rangle = \sum_i \psi_i |i\rangle \rightarrow |f(\psi)\rangle = \sum_i f(\psi_i)|i\rangle$$

Where f may be a non-linear function.

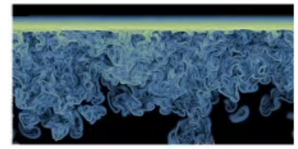
In particular, we will assume that f can be expanded as a power series, e.g. $f(x) = \sum_j \alpha_j x^j$

& functions of elements of multiple mixed states: $\rho^{\text{in}} = \rho^{(0)} \otimes ... \otimes \rho^{(k)} \rightarrow \tau = \sum_j g(\vec{v}_{\rho_{\text{in}}})_{ij} |i\rangle\langle j|$

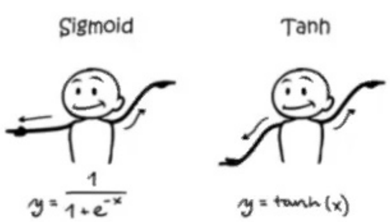
Vector containing all entries of ρ^{in}

Applications?

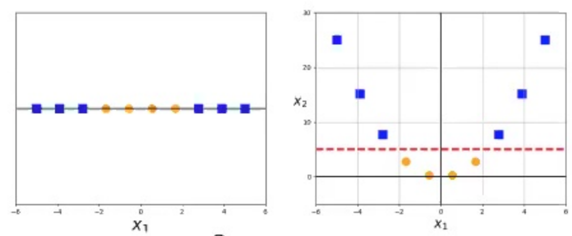
- Solving non-linear partial differential equations
(with applications from fluid simulations to finance)



- Quantum Machine Learning
- Implementing non-linear activation functions
- Implementing non-linear encodings for kernel methods, e.g. $|\psi\rangle \to |\psi^2\rangle$

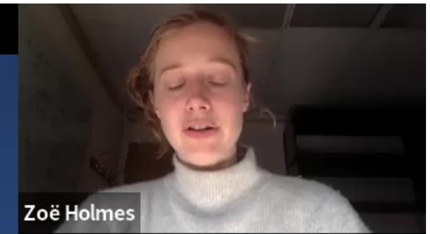


Activation functions

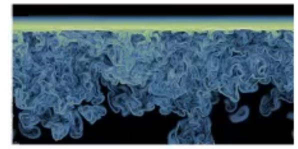


$x \to x^2$ Non-linear encoding

Applications?



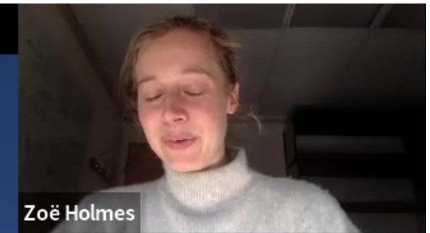
- Solving non-linear partial differential equations (with applications from fluid simulations to finance)



- Quantum Machine Learning
  - Implementing non-linear activation functions
  - Implementing non-linear encodings for kernel methods

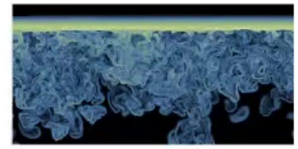
- Error mitigation e.g. amplify a signal $\rho \to \rho^k$

Applications?



- Solving non-linear partial differential equations

(with applications from fluid simulations to finance)



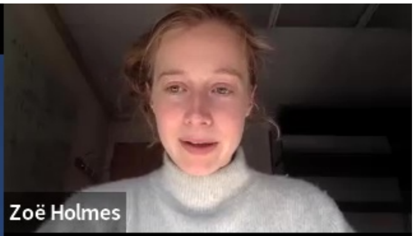
- Quantum Machine Learning

- Implementing non-linear activation functions
- Implementing non-linear encodings for kernel methods

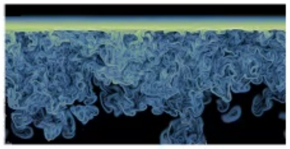
- Error mitigation e.g. amplify a signal $\rho \rightarrow \rho^k$

- Filtering scheme e.g. highlighting unlikely outcomes $|\psi\rangle \rightarrow |\psi^{-1}\rangle$

Applications?



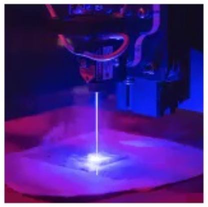
- Solving non-linear partial differential equations
(with applications from fluid simulations to finance)



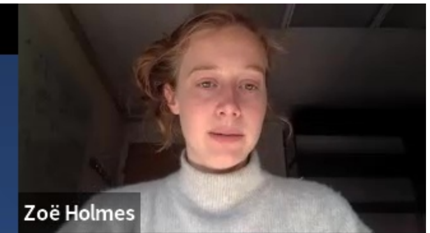
- Quantum Machine Learning
- Implementing non-linear activation functions
- Implementing non-linear encodings for kernel methods

"A solution searching for a problem?"

- Error mitigation e.g. amplify a signal $\rho \to \rho^k$
- Filtering scheme e.g. highlighting unlikely outcomes $|\psi\rangle \to |\psi^{-1}\rangle$



Outline of rest of talk



1) Introduce the weighted state framework

2) Introduce three primitives for implementing non-linear operations
   - Quantum Hadamard Product
   - Generalized Transpose
   - Quantum State Polynomial

3) Sampling complexity analysis

On nonlinear transformations in quantum computation

Zoë Holmes,* Nolan Coble,† Andrew T. Sornborger, and Yiğit Subaşı‡
Information Sciences, Los Alamos National Laboratory, Los Alamos, NM, USA.
(Dated: December 24, 2021)

arXiv:2112.12307

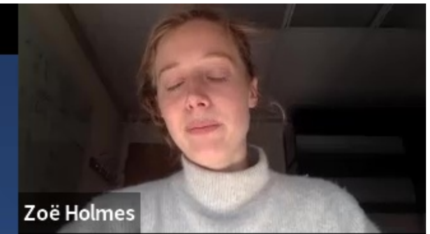
Framework

Note that $|\tau\rangle = |f(\psi)\rangle$ (or more generally $\tau = f(\rho)$) may not be a genuine quantum state.

We only ever have access to quantum states through measurement outcomes

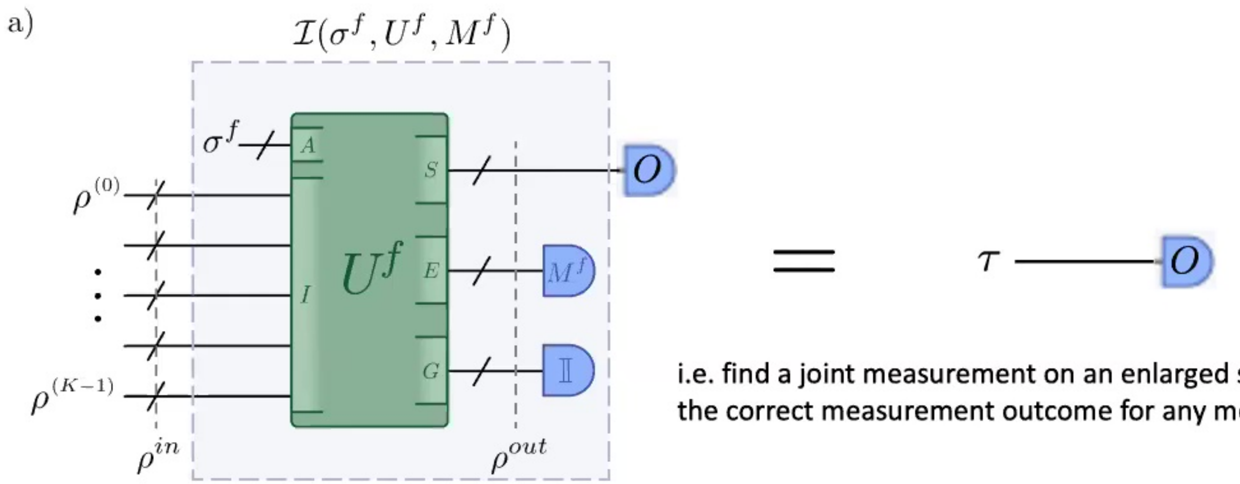
So we don't need to (and often won't be able to) prepare τ on a quantum register…

Rather, we just need to find an operational strategy to compute
$\langle O\rangle_\tau = \mathrm{Tr}[\tau\, O]$ (or more generally compute $\mathrm{Tr}[U(\tau \otimes \rho_{\mathrm{other}})U^\dagger\, O]$)



Weighted states

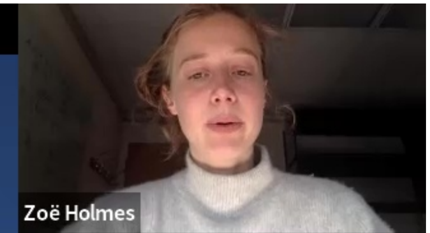
Key idea: Find ρ_{out} and M such that: $\mathrm{Tr}[\tau\, O] = \mathrm{Tr}[\rho_{\text{out}}\, (O_S \otimes M_E \otimes \mathbb{I}_G)]\ \ \forall\, O$



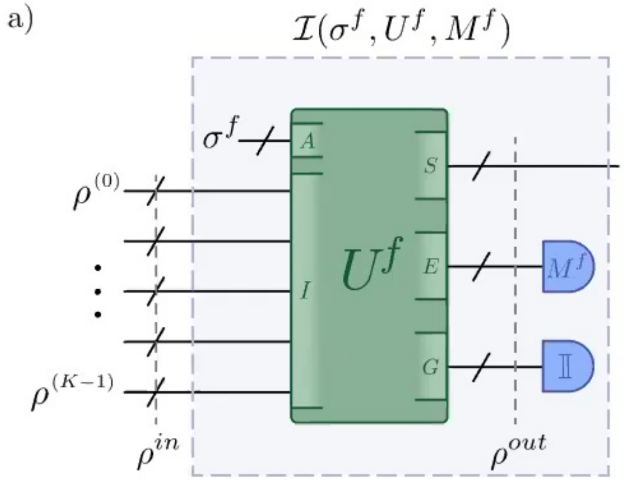
i.e. find a joint measurement on an enlarged system that gives the correct measurement outcome for any measurement O

Zoë Holmes

Weighted states



Key idea: Find ρ_{out} and M such that: $\text{Tr}[\tau\, O] = \text{Tr}[\rho_{\text{out}}\,(O_S \otimes M_E \otimes \mathbb{I}_G)]\ \ \forall\, O$



Since this holds for any observable O we can write

$$\tau = \text{Tr}_{MG}[\rho_{\text{out}}\,(\mathbb{I}_S \otimes M_E \otimes \mathbb{I}_G)]$$

Can use this framework to implement non-linear operations if we feed in multiple copies of the same input

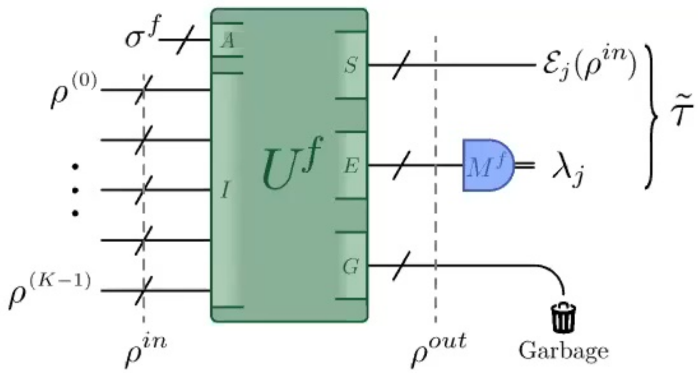
e.g. for $|\psi\rangle \to |\psi^k\rangle$ need $\rho^{(0)} = ... = \rho^{(k)} = \psi$

$\left(|f(\psi)\rangle = \sum_i f(\psi_i)|i\rangle\right)$

Weighted states (alternative perspective)

τ is a 'weighted state'

b)



"re-weight output states by obtained eigenvalues"

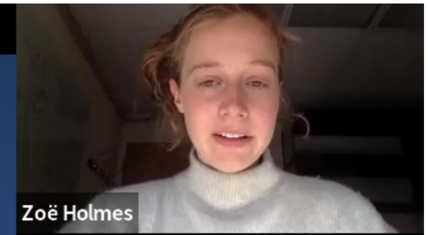
$$\tau = \sum_j \lambda_j p_j \frac{\mathcal{E}_j(\rho^{\mathrm{in}})}{p_j}$$

eigenvalue

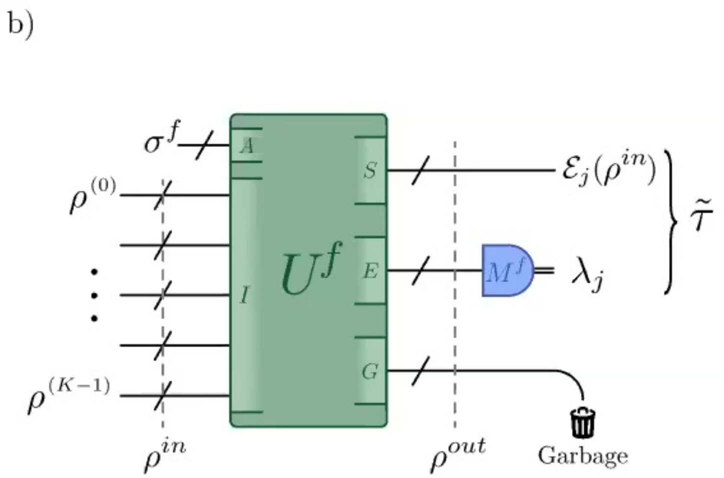
Probability of getting it

Conditional output

Weighted states (alternative perspective)



τ is a 'weighted state'



On obtaining outcome λ_j the conditional outcome on the system is:

$$\mathcal{E}_j(\rho^{\text{in}})/p_j$$

Where $\mathcal{E}_j(\rho_{\text{in}}) = \text{Tr}_{EG}(U^f(\sigma^f_A \otimes \rho_{\text{in}})U^{f\dagger}(\mathbb{I}_S \otimes |\lambda_j\rangle\langle\lambda_j|_E \otimes \mathbb{I}_G))$

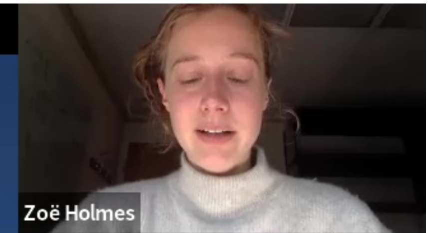
$p_j = \text{Tr}[\mathcal{E}_j(\rho^{\text{in}})]$

$$\tau = \sum_j \lambda_j p_j \frac{\mathcal{E}_j(\rho^{\text{in}})}{p_j}$$

eigenvalue

Probability of getting it

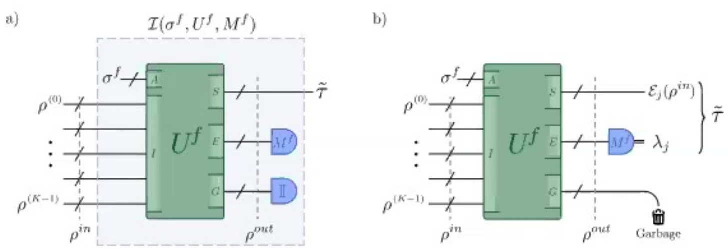
Conditional output



Weighted states (alternative perspective)

τ is a 'weighted state'

Same output as before…
the two pictures are equivalent



On obtaining outcome λ_j the conditional outcome on the system is:

$$\mathcal{E}_j(\rho^{\text{in}})/p_j$$

Where $\mathcal{E}_j(\rho_{\text{in}}) = \text{Tr}_{EG}(U^f(\sigma_A^f \otimes \rho_{\text{in}})U^{f\dagger}(\mathbb{I}_S \otimes |\lambda_j\rangle\langle\lambda_j|_E \otimes \mathbb{I}_G))$

$$p_j = \text{Tr}[\mathcal{E}_j(\rho^{\text{in}})]$$

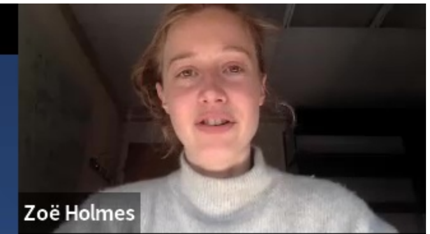
This definition agrees with previous definition:

$$\tau = \sum_j \lambda_j p_j \frac{\mathcal{E}_j(\rho^{\text{in}})}{p_j} = \sum_j \lambda_j \mathcal{E}_j(\rho^{\text{in}})$$

$$= \sum_j \lambda_j \text{Tr}_{EG}(U^f(\sigma_A^f \otimes \rho_{\text{in}})U^{f\dagger}(\mathbb{I}_S \otimes |\lambda_j\rangle\langle\lambda_j|_E \otimes \mathbb{I}_G))$$

$$= \text{Tr}_{EG}(U^f(\sigma_A^f \otimes \rho_{\text{in}})U^{f\dagger}(\mathbb{I}_S \otimes M_E \otimes \mathbb{I}_G))$$

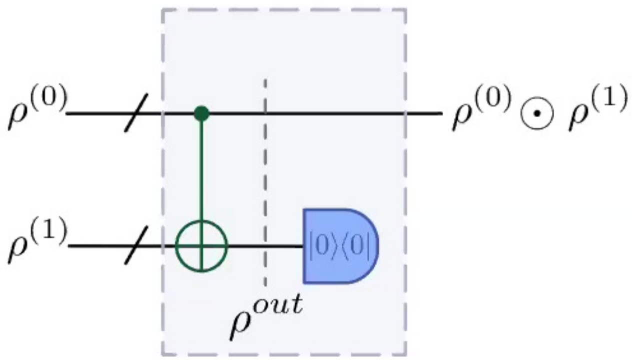
Quantum Hadamard Product



Definition:

$$|\psi^{(0)}\rangle \odot |\psi^{(1)}\rangle \equiv \sum_i \psi_i^{(0)} \psi_i^{(1)} |i\rangle$$

Circuit:



Claim:

$$\tau = \mathrm{Tr}_2[|0\rangle\langle 0|\rho_{\mathrm{out}}] = |\psi^{(0)} \odot \psi^{(1)}\rangle\langle\psi^{(0)} \odot \psi^{(1)}|$$

Proof:

$$|\psi^{(0)}\rangle|\psi^{(1)}\rangle = \sum_{ij} \psi_i^{(0)} \psi_j^{(1)} |ij\rangle$$

CNOT $\rightarrow \rho_{\mathrm{out}} = \sum_{ij} \psi_i^{(0)} \psi_j^{(1)} |ij \oplus i\rangle \times$ c.c.

Measure 0 $\rightarrow |\tau\rangle = \sum_j \psi_j^{(0)} \psi_j^{(1)} |j\rangle = |\psi^{(0)} \odot \psi^{(1)}\rangle$

Quantum Hadamard Product

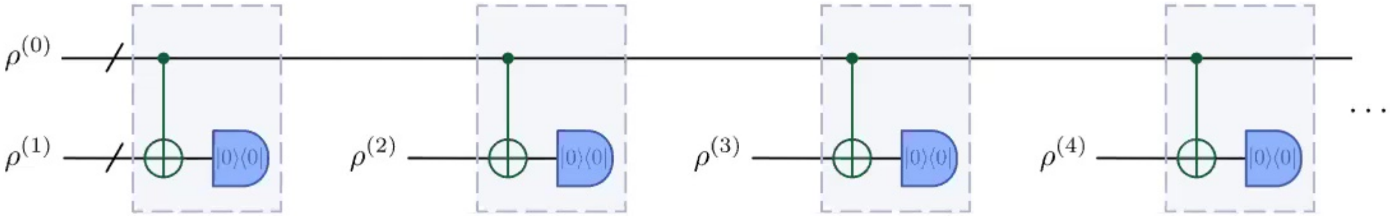
Definition:

$$|\psi^{(0)}\rangle \odot |\psi^{(1)}\rangle \equiv \sum_i \psi_i^{(0)} \psi_i^{(1)} |i\rangle$$

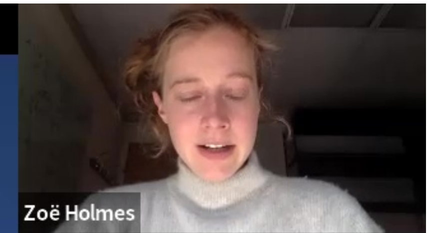
Or more generally $\rho^{(0)} \odot \rho^{(1)} = \sum_{ij} \rho_{ij}^{(0)} \rho_{ij}^{(1)} |i\rangle\langle j|$

Circuit: Repeated use implements powers of states

$$|\psi \odot \psi \odot \cdots \odot \psi\rangle \equiv |\psi^p\rangle \equiv \sum_i \psi_i^p |i\rangle$$



Generalized Transpose Operation

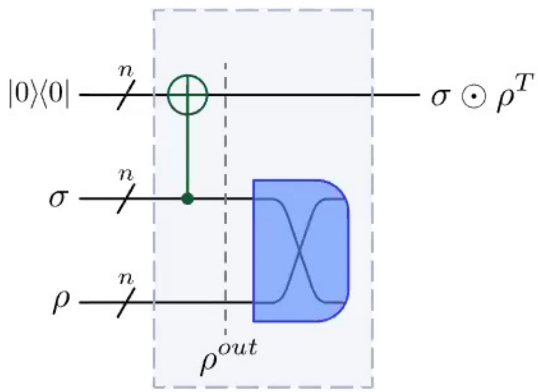


Definition:

$$\rho_{\sigma}^{(T)} = \sigma \odot \rho^{T}$$

Note that $\rho_{|+\rangle\langle+|}^{(T)} = \frac{1}{d}\rho^{T}$

Circuit:

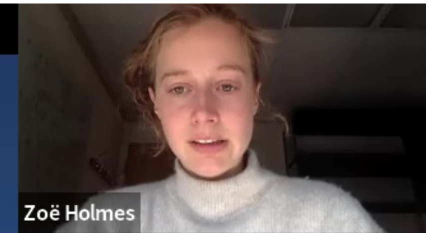


Proof:

$$\sum_{ii'jj'} \sigma_{ij}\rho_{i'j'}|0ii'\rangle\langle 0jj'| \xrightarrow{\text{CNOT}} \rho_{\text{out}} = \sum_{ii'jj'} \sigma_{ij}\rho_{i'j'}|iii'\rangle\langle jjj'|$$

$$\tau = \text{Tr}_{2,3}(\rho^{\text{out}}(\mathbb{I} \otimes \text{SWAP}))$$

Generalized Transpose Operation

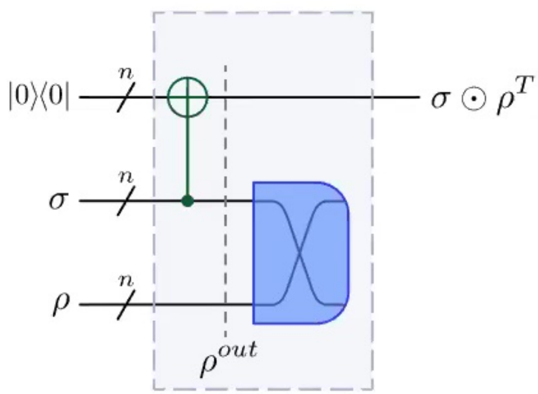


Definition:

$$\rho_{\sigma}^{(T)} = \sigma \odot \rho^{T}$$

Note that $\rho^{(T)}_{|+\rangle\langle+|} = \frac{1}{d}\rho^{T}$

Circuit:



Proof:

$$\sum_{ii'jj'} \sigma_{ij}\rho_{i'j'}|0ii'\rangle\langle 0jj'| \xrightarrow{\text{CNOT}} \rho_{\text{out}} = \sum_{ii'jj'} \sigma_{ij}\rho_{i'j'}|iii'\rangle\langle jjj'|$$

$$\tau = \text{Tr}_{2,3}(\rho^{\text{out}}(\mathbb{I}\otimes\text{SWAP}))$$
$$= \sum_{iji'j'} \sigma_{ij}\rho_{i'j'}\langle ii'|\text{SWAP}|jj'\rangle|i\rangle\langle j|$$
$$= \sum_{iji'j'} \sigma_{ij}\rho_{i'j'}\delta_{ij'}\delta_{ji'}|i\rangle\langle j|$$

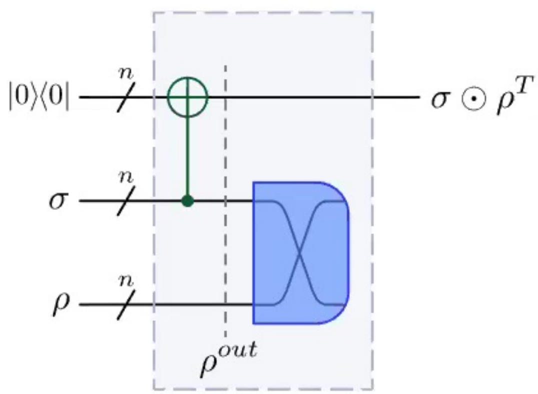
Generalized Transpose Operation

Definition:

$$\rho_{\sigma}^{(T)} = \sigma \odot \rho^{T}$$

Note that $\rho_{|+\rangle\langle+|}^{(T)} = \frac{1}{d}\rho^{T}$

Circuit:



Proof:

$$\sum_{ii'jj'} \sigma_{ij}\rho_{i'j'}|0ii'\rangle\langle 0jj'| \xrightarrow{\text{CNOT}} \rho_{\text{out}} = \sum_{ii'jj'} \sigma_{ij}\rho_{i'j'}|iii'\rangle\langle jjj'|$$

$$\begin{aligned}\tau &= \mathrm{Tr}_{2,3}(\rho^{\text{out}}(\mathbb{I}\otimes \text{SWAP})) \\ &= \sum_{iji'j'} \sigma_{ij}\rho_{i'j'}\langle ii'|\text{SWAP}|jj'\rangle|i\rangle\langle j| \\ &= \sum_{iji'j'} \sigma_{ij}\rho_{i'j'}\delta_{ij'}\delta_{ji'}|i\rangle\langle j| \\ &= \sum_{ij} \sigma_{ij}\rho_{ji}|i\rangle\langle j| = \sigma \odot \rho^{T}\end{aligned}$$

Generalized Transpose Operation

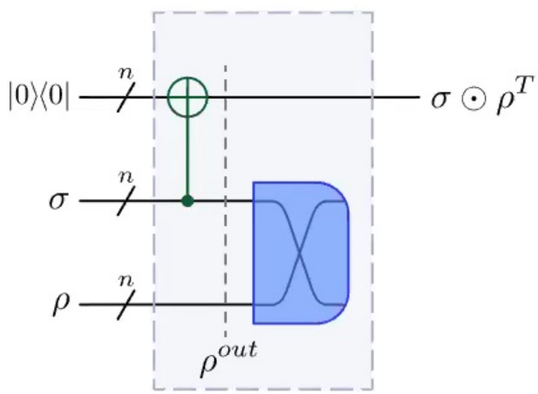
Definition:

$$\rho_\sigma^{(T)} = \sigma \odot \rho^T$$

Note that $\rho^{(T)}_{|+\rangle\langle+|} = \frac{1}{d}\rho^T$

And if the algorithm is only applied to a subsystem can implement the partial transpose – useful for witnessing entanglement!

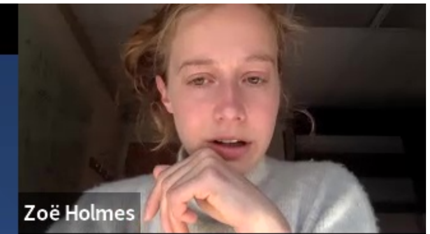
Circuit:



Proof:

$$\sum_{ii'jj'} \sigma_{ij}\rho_{i'j'}|0ii'\rangle\langle 0jj'| \xrightarrow{\text{CNOT}} \rho_{\text{out}} = \sum_{ii'jj'} \sigma_{ij}\rho_{i'j'}|iii'\rangle\langle jjj'|$$

$$\begin{aligned}
\tau &= \text{Tr}_{2,3}(\rho^{\text{out}}(\mathbb{I} \otimes \text{SWAP})) \\
&= \sum_{iji'j'} \sigma_{ij}\rho_{i'j'}\langle ii'|\text{SWAP}|jj'\rangle|i\rangle\langle j| \\
&= \sum_{iji'j'} \sigma_{ij}\rho_{i'j'}\delta_{ij'}\delta_{ji'}|i\rangle\langle j| \\
&= \sum_{ij} \sigma_{ij}\rho_{ji}|i\rangle\langle j| = \sigma \odot \rho^T
\end{aligned}$$



Generalized Transpose Operation

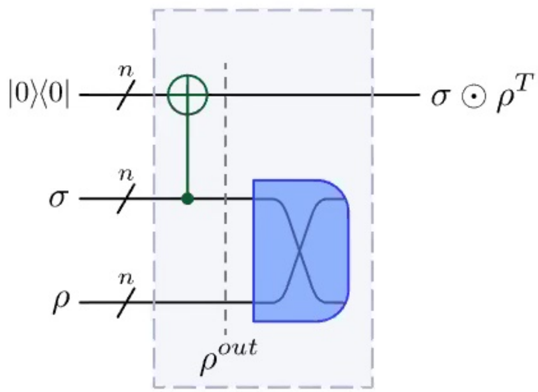
Definition:

$$\rho_\sigma^{(T)} = \sigma \odot \rho^T$$

Note that $\rho_{|+\rangle\langle+|}^{(T)} = \frac{1}{d}\rho^T$

And if the algorithm is only applied to a subsystem can implement the partial transpose – useful for witnessing entanglement!

Circuit:



Proof:

$$\sum_{ii'jj'} \sigma_{ij}\rho_{i'j'}|0ii'\rangle\langle 0jj'| \xrightarrow{\text{CNOT}} \rho_{\text{out}} = \sum_{ii'jj'} \sigma_{ij}\rho_{i'j'}|iii'\rangle\langle jjj'|$$

$$\begin{aligned}\tau &= \text{Tr}_{2,3}(\rho^{\text{out}}(\mathbb{I} \otimes \text{SWAP})) \\ &= \sum_{iji'j'} \sigma_{ij}\rho_{i'j'}\langle ii'|\text{SWAP}|jj'\rangle|i\rangle\langle j| \\ &= \sum_{iji'j'} \sigma_{ij}\rho_{i'j'}\delta_{ij'}\delta_{ji'}|i\rangle\langle j| \\ &= \sum_{ij} \sigma_{ij}\rho_{ji}|i\rangle\langle j| = \sigma \odot \rho^T\end{aligned}$$

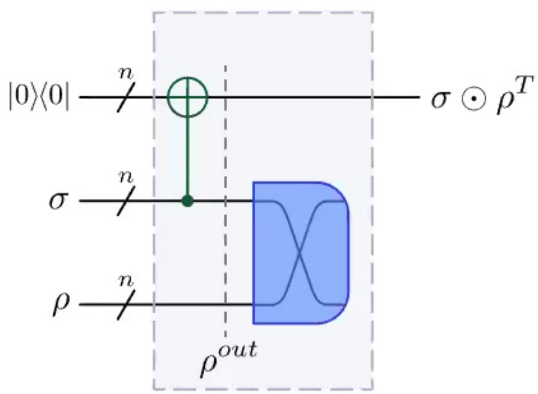
Generalized Transpose Operation

Definition:

$$\rho_{\sigma}^{(T)} = \sigma \odot \rho^{T}$$

Note that $\rho^{(T)}_{|+\rangle\langle+|} = \frac{1}{d}\rho^{T}$

Circuit:



Proof:

$$\sum_{ii'jj'} \sigma_{ij}\rho_{i'j'}|0ii'\rangle\langle 0jj'|$$

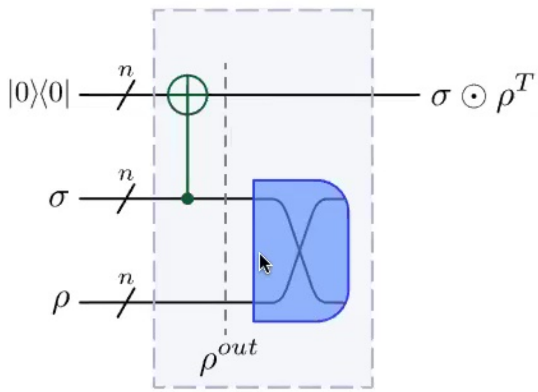
Generalized Transpose Operation

Definition:

$$\rho_\sigma^{(T)} = \sigma \odot \rho^T$$

Note that $\rho^{(T)}_{|+\rangle\langle+|} = \frac{1}{d}\rho^T$

Circuit:

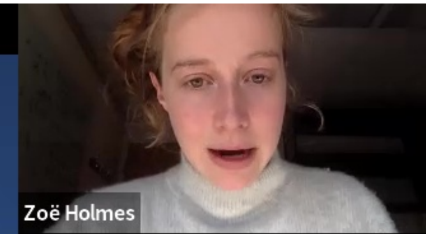


Proof:

$$\sum_{ii'jj'} \sigma_{ij}\rho_{i'j'}|0ii'\rangle\langle 0jj'| \xrightarrow{\text{CNOT}} \rho_{\text{out}} = \sum_{ii'jj'} \sigma_{ij}\rho_{i'j'}|iii'\rangle\langle jjj'|$$

$$\begin{aligned}\tau &= \text{Tr}_{2,3}(\rho^{\text{out}}(\mathbb{I}\otimes \text{SWAP})) \\ &= \sum_{iji'j'} \sigma_{ij}\rho_{i'j'}\langle ii'|\text{SWAP}|jj'\rangle|i\rangle\langle j| \\ &= \sum_{iji'j'} \sigma_{ij}\rho_{i'j'}\delta_{ij'}\delta_{ji'}|i\rangle\langle j| \\ &= \sum_{ij} \sigma_{ij}\rho_{ji}|i\rangle\langle j| = \sigma \odot \rho^T\end{aligned}$$

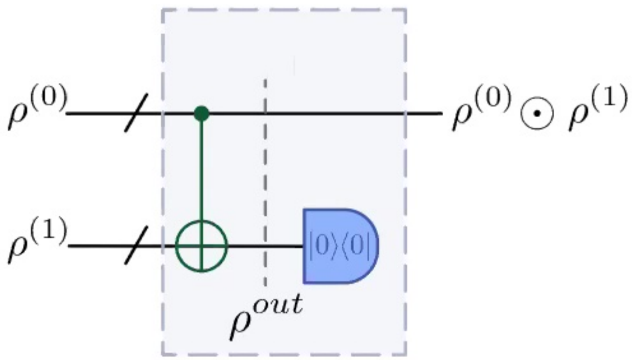
Quantum Hadamard Product



Definition:

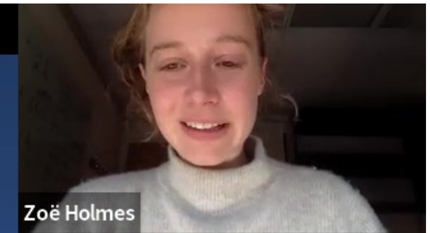
$$|\psi^{(0)}\rangle \odot |\psi^{(1)}\rangle \equiv \sum_i \psi_i^{(0)} \psi_i^{(1)} |i\rangle$$

Circuit:



Claim:

$$\tau = \mathrm{Tr}_2[|0\rangle\langle 0|\rho_{\mathrm{out}}] = |\psi^{(0)} \odot \psi^{(1)}\rangle\langle \psi^{(0)} \odot \psi^{(1)}|$$



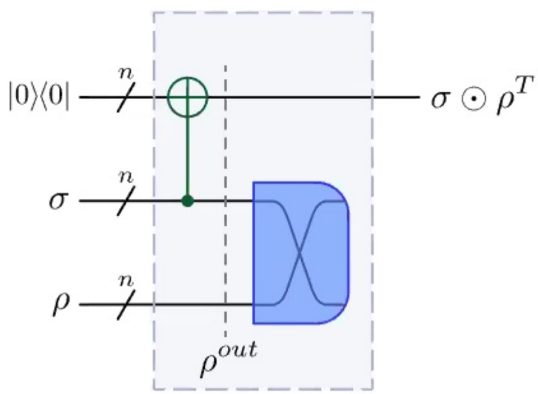
Generalized Transpose Operation

Definition:

$$\rho_\sigma^{(T)} = \sigma \odot \rho^T$$

Note that $\rho^{(T)}_{|+\rangle\langle+|} = \frac{1}{d}\rho^T$

Circuit:

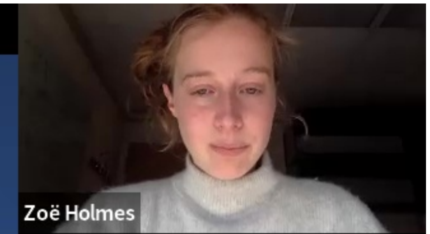


Proof:

$$\sum_{ii'jj'} \sigma_{ij}\rho_{i'j'}|0ii'\rangle\langle 0jj'| \xrightarrow{\text{CNOT}} \rho_{\text{out}} = \sum_{ii'jj'} \sigma_{ij}\rho_{i'j'}|iii'\rangle\langle jjj'|$$

$$\begin{aligned}\tau &= \text{Tr}_{2,3}(\rho^{\text{out}}(\mathbb{I}\otimes \text{SWAP})) \\ &= \sum_{iji'j'} \sigma_{ij}\rho_{i'j'}\langle ii'|\text{SWAP}|jj'\rangle|i\rangle\langle j| \\ &= \sum_{iji'j'} \sigma_{ij}\rho_{i'j'}\delta_{ij'}\delta_{ji'}|i\rangle\langle j| \\ &= \sum_{ij} \sigma_{ij}\rho_{ji}|i\rangle\langle j| = \sigma \odot \rho^T\end{aligned}$$

Quantum State Polynomial



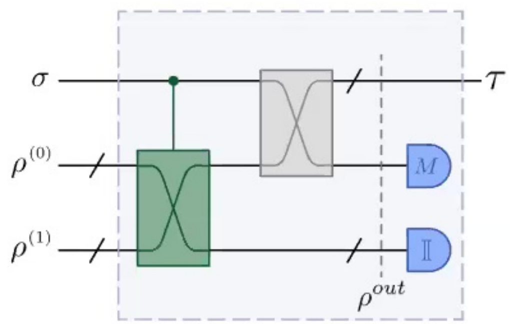
Think about alpha coefficients as forming a matrix

$$\alpha = \begin{pmatrix} \alpha_{00} & \alpha_{01} \\ \alpha_{10} & \alpha_{11} \end{pmatrix}$$

Definition:

$$\tau = \alpha_{00}\rho^{(0)} + \alpha_{11}\rho^{(1)} + \alpha_{01}\rho^{(0)}\rho^{(1)} + \alpha_{10}\rho^{(1)}\rho^{(0)}$$

Circuit:

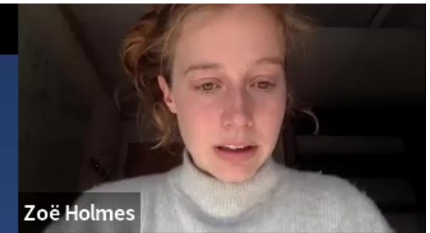


You'll have to believe me on this.

Simple enough to show… but I thought I'd spare you a slide full of algebra.

Pick σ and M such that $\alpha = \sigma \odot M^T$

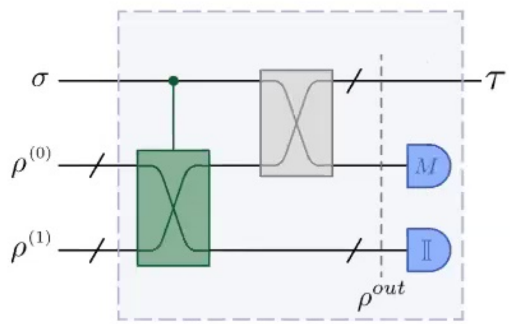
Quantum State Polynomial



Definition:

$$\tau = \alpha_{00}\rho^{(0)} + \alpha_{11}\rho^{(1)} + \alpha_{01}\rho^{(0)}\rho^{(1)} + \alpha_{10}\rho^{(1)}\rho^{(0)}$$

Circuit:



Pick σ and M such that $\alpha = \sigma \odot M^T$

Uses:

1. Mixtures: $\tau = p\rho^{(0)} + (1-p)\rho^{(1)}$

$$\alpha = \begin{pmatrix} p & 0 \\ 0 & 1-p \end{pmatrix}$$

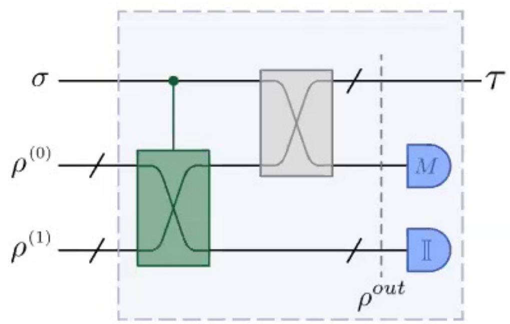
Could use $\sigma = \begin{pmatrix} p & 0 \\ 0 & 1-p \end{pmatrix}$ $\quad M = \mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

Quantum State Polynomial

Definition:

$$\tau = \alpha_{00}\rho^{(0)} + \alpha_{11}\rho^{(1)} + \alpha_{01}\rho^{(0)}\rho^{(1)} + \alpha_{10}\rho^{(1)}\rho^{(0)}$$

Circuit:



Pick σ and M such that $\alpha = \sigma \odot M^T$

Uses:

1. Mixtures: $\tau = p\rho^{(0)} + (1-p)\rho^{(1)}$
2. Anti-commutator: $\tau = \rho^{(0)}\rho^{(1)} + \rho^{(1)}\rho^{(0)}$

$$\alpha = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Could use $\sigma = |+\rangle\langle+| = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$

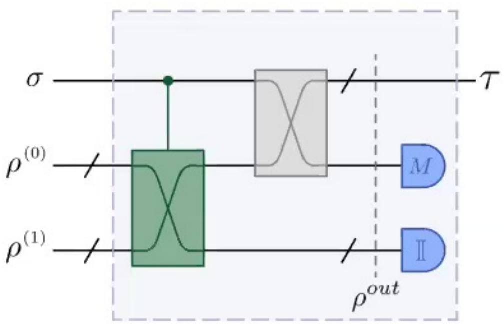
$$M = 2X = 2\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Quantum State Polynomial

Definition:

$$\tau = \alpha_{00}\rho^{(0)} + \alpha_{11}\rho^{(1)} + \alpha_{01}\rho^{(0)}\rho^{(1)} + \alpha_{10}\rho^{(1)}\rho^{(0)}$$

Circuit:

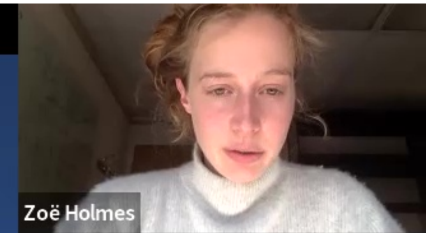


Pick σ and M such that $\alpha = \sigma \odot M^T$

Uses:

1. Mixtures: $\tau = p\rho^{(0)} + (1-p)\rho^{(1)}$
2. Anti-commutator: $\tau = \rho^{(0)}\rho^{(1)} + \rho^{(1)}\rho^{(0)}$
3. Commutator: $\tau = \rho^{(0)}\rho^{(1)} - \rho^{(1)}\rho^{(0)}$
4. Linear Combinations of Pure States:

$$|\psi\rangle = \alpha_0|\psi^{(0)}\rangle + \alpha_1|\psi^{(1)}\rangle$$



Quantum State Polynomial

Definition:

$$\tau = \alpha_{00}\rho^{(0)} + \alpha_{11}\rho^{(1)} + \alpha_{01}\rho^{(0)}\rho^{(1)} + \alpha_{10}\rho^{(1)}\rho^{(0)}$$

$|\psi^{(0)}\rangle\langle\psi^{(0)}|$ $\quad$ $|\psi^{(1)}\rangle\langle\psi^{(1)}|$

Pick σ and M such that $\alpha = \sigma \odot M^T$

4. Linear Combinations of Pure States: $|\psi\rangle = \alpha_0|\psi^{(0)}\rangle + \alpha_1|\psi^{(1)}\rangle$

$$\tau = |\psi\rangle\langle\psi| = |\alpha_0|^2|\psi^{(0)}\rangle\langle\psi^{(0)}| + |\alpha_1|^2|\psi^{(1)}\rangle\langle\psi^{(1)}| + \alpha_0\alpha_1^*|\psi^{(0)}\rangle\langle\psi^{(1)}| + \alpha_1\alpha_0^*|\psi^{(1)}\rangle\langle\psi^{(0)}|$$

$$= \alpha_{00}|\psi^{(0)}\rangle\langle\psi^{(0)}| + \alpha_{11}|\psi^{(0)}\rangle\langle\psi^{(0)}| + \alpha_{01}|\psi^{(0)}\rangle\langle\psi^{(0)}|\psi^{(1)}\rangle\langle\psi^{(1)}| + \alpha_{10}|\psi^{(1)}\rangle\langle\psi^{(1)}|\psi^{(0)}\rangle\langle\psi^{(0)}|$$

Quantum State Polynomial

Definition:

$$\tau = \alpha_{00}\rho^{(0)} + \alpha_{11}\rho^{(1)} + \alpha_{01}\rho^{(0)}\rho^{(1)} + \alpha_{10}\rho^{(1)}\rho^{(0)}$$

Pick σ and M such that $\alpha = \sigma \odot M^T$

4. Linear Combinations of Pure States: $|\psi\rangle = \alpha_0|\psi^{(0)}\rangle + \alpha_1|\psi^{(1)}\rangle$

Comparing two expressions we can see that

$$\tau = |\psi\rangle\langle\psi| = |\alpha_0|^2|\psi^{(0)}\rangle\langle\psi^{(0)}| + |\alpha_1|^2|\psi^{(1)}\rangle\langle\psi^{(1)}| + \alpha_0\alpha_1^*|\psi^{(0)}\rangle\langle\psi^{(1)}| + \alpha_1\alpha_0^*|\psi^{(1)}\rangle\langle\psi^{(0)}|$$

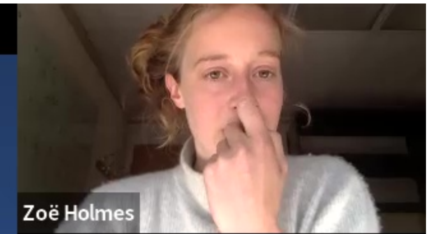
$$= \alpha_{00}|\psi^{(0)}\rangle\langle\psi^{(0)}| + \alpha_{11}|\psi^{(0)}\rangle\langle\psi^{(0)}| + \alpha_{01}|\psi^{(0)}\rangle\langle\psi^{(0)}|\psi^{(1)}\rangle\langle\psi^{(1)}| + \alpha_{10}|\psi^{(1)}\rangle\langle\psi^{(1)}|\psi^{(0)}\rangle\langle\psi^{(0)}|$$

$$\alpha = \begin{pmatrix} |\alpha_0|^2 & \frac{\alpha_0\alpha_1^*}{\langle\psi_0|\psi_1\rangle} \\ \frac{\alpha_1\alpha_0^*}{\langle\psi_1|\psi_0\rangle} & |\alpha_1|^2 \end{pmatrix}$$

Can use $\sigma = \begin{pmatrix} |\beta_0|^2 & \beta_0\beta_1^* \\ \beta_1\beta_0^* & |\beta_1|^2 \end{pmatrix}$ $M = \begin{pmatrix} \frac{|\alpha_0|^2}{|\beta_0|^2} & \frac{\alpha_1\alpha_0^*}{\beta_1\beta_0^*\langle\psi_1|\psi_0\rangle} \\ \frac{\alpha_0\alpha_1^*}{\beta_0\beta_1^*\langle\psi_0|\psi_1\rangle} & \frac{|\alpha_1|^2}{|\beta_1|^2} \end{pmatrix}$

i.e. $\sigma = |\beta\rangle\langle\beta|$ with $|\beta\rangle = \beta_0|0\rangle + \beta_1|1\rangle$

Concatenation: Arbitrary Polynomials



Use Quantum Hadamard Product to implement powers of states

$$|\psi \odot \psi \odot \cdots \odot \psi\rangle \equiv |\psi^p\rangle \equiv \sum_i \psi_i^p |i\rangle$$

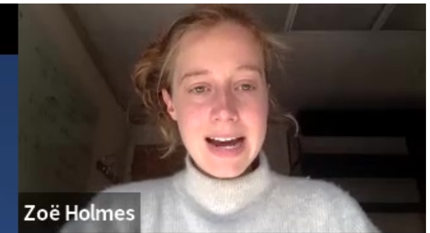
Use Quantum State Polynomial to take linear combinations of states

$$\alpha_0|\psi\rangle + \alpha_1|\psi^2\rangle + \dots$$

Combine the two to implement arbitrary functions (power series) of states

$$|\psi\rangle \rightarrow |f(\psi)\rangle = \sum_i f(\psi_i)|i\rangle \text{ where } f(\psi_i) = \sum_j \alpha_j \psi_i^j$$

Sampling complexity analysis



Sampling complexity scales inversely with variance of circuit measurement outcome:

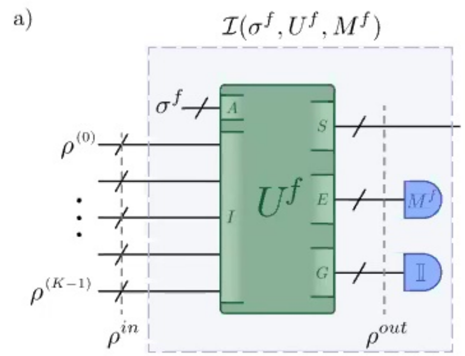
Sampling complexity ~ 1/Var

If we had access to τ as a real quantum state then

$$\mathrm{Var} = \frac{1}{s}\left(\mathrm{Tr}[\tau O^2] - \mathrm{Tr}[\tau O]^2\right)$$

But we don't....

No. of shots used



Sampling complexity analysis

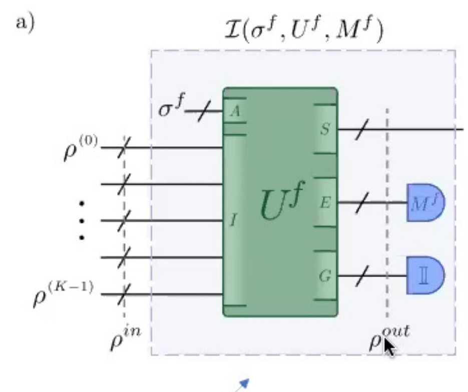
Sampling complexity scales inversely with variance of circuit measurement outcome:

Sampling complexity = 1/Var

If we had access to τ as a real quantum state then

$$\mathrm{Var} = \frac{1}{s}\left(\mathrm{Tr}[\tau O^2] - \mathrm{Tr}[\tau O]^2\right)$$

But we don't.... rather we run



Sampling complexity analysis

Sampling complexity scales inversely with variance of circuit measurement outcome:

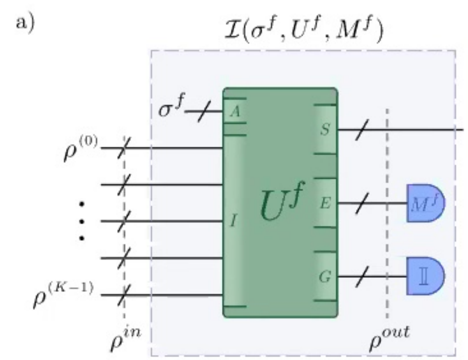
Sampling complexity = 1/Var

$\mathrm{Tr}[\tau O]$

Variance of weighted state output:

$$\mathrm{Var} = \frac{1}{s}\left(\mathrm{Tr}[\rho^{\mathrm{out}}(O \otimes M \otimes \mathbb{I})^2] - \mathrm{Tr}[\rho^{\mathrm{out}}(O \otimes M \otimes \mathbb{I})]^2\right)$$

(The variance associated with performing the measurements O and M on the state ρ^{out})

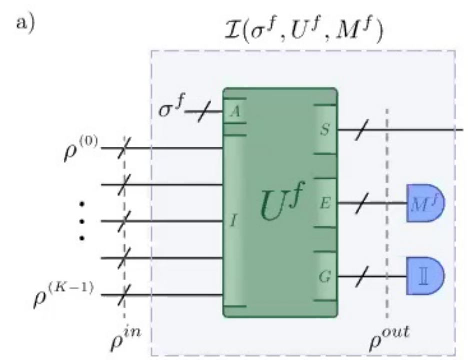


Sampling complexity analysis

Sampling complexity scales inversely with variance of circuit measurement outcome:

Sampling complexity = 1/Var

If freedom- chose M to minimize sampling complexity.



Variance of weighted state output:

$$\mathrm{Var} = \frac{1}{s}\left(\mathrm{Tr}[\rho^{\mathrm{out}}(O \otimes M \otimes \mathbb{I})^2] - \mathrm{Tr}[\tau O]^2\right)$$

(The variance associated with performing the measurements O and M on the state ρ^{out})

Bounding the variance

Variance of weighted state output:

$$\mathrm{Var} = \frac{1}{s}\left(\mathrm{Tr}[\rho^{\mathrm{out}}(O \otimes M \otimes \mathbb{I})^2] - \mathrm{Tr}[\tau O]^2\right)$$

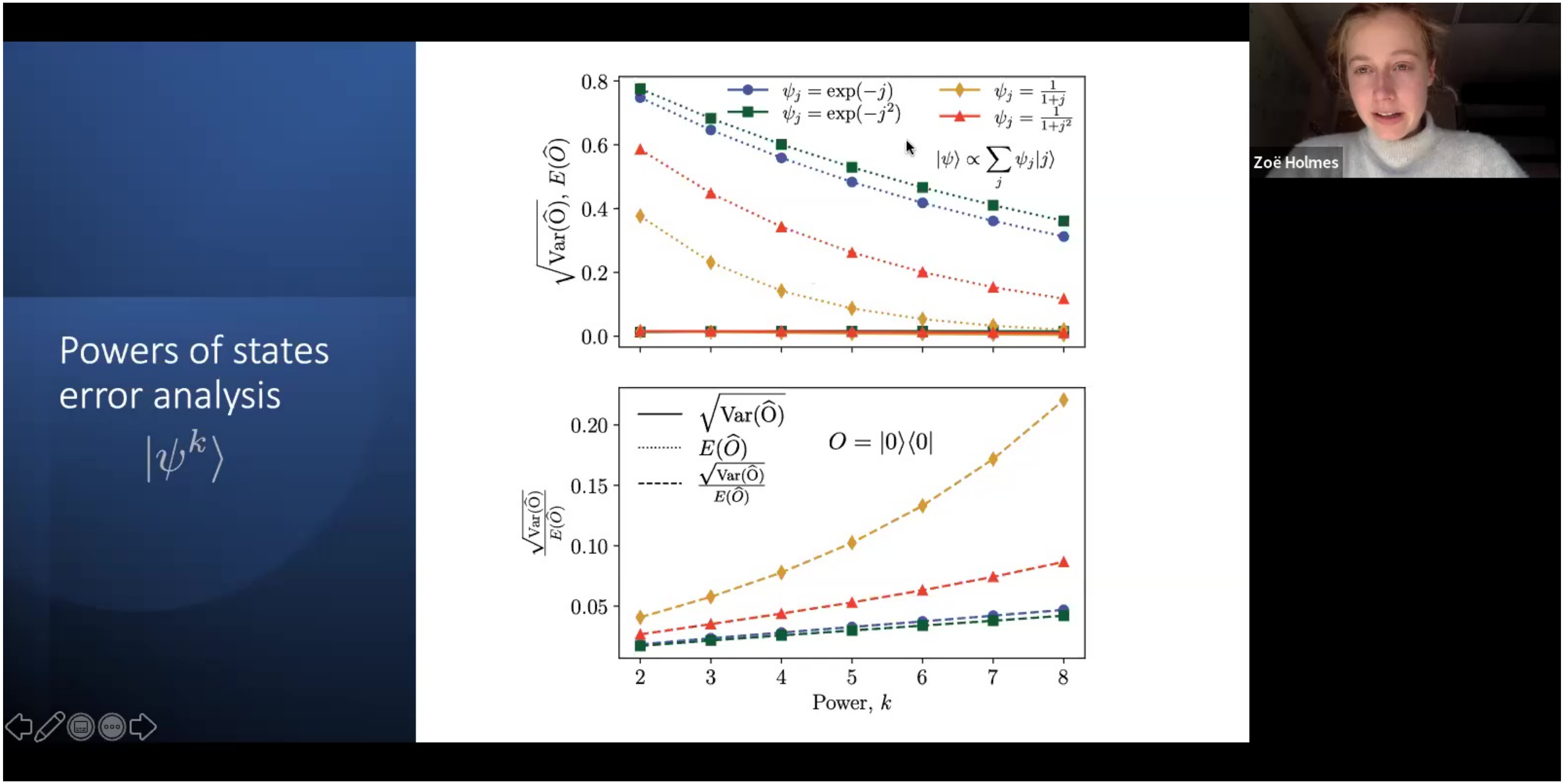
The variance (and sampling complexity) depends on the measurement O performed on the weighted state

To compare the sampling complexity of different algorithms it is helpful to derive an operator independent bound on the variance:

$$\mathrm{Var}(\widehat{XY}) \leq 2\mathrm{Var}(\widehat{X})||Y||_\infty^2 + 2\langle X\rangle^2 \mathrm{Var}(\widehat{Y}) \leq 2\langle X^2\rangle$$

Assuming $||Y||_\infty \leq 1$

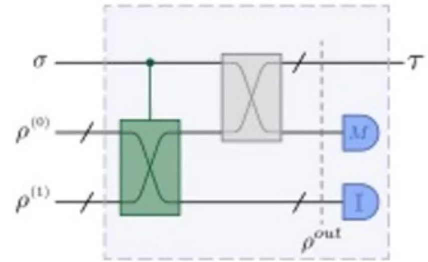
➡ Variance of weighted state circuit bounded as: $\mathrm{Var} \leq \dfrac{2\mathrm{Tr}[\rho^{\mathrm{out}}(\mathbb{I} \otimes M \otimes \mathbb{I})^2]}{s}$



Powers of states error analysis
$|\psi^k\rangle$
$\psi_j = \exp(-j)$
$\psi_j = \exp(-j^2)$
$\psi_j = \frac{1}{1+j}$
$\psi_j = \frac{1}{1+j^2}$
$|\psi\rangle \propto \sum_j \psi_j |j\rangle$
$\sqrt{\mathrm{Var}(\widehat{O})}, E(\widehat{O})$
$\sqrt{\mathrm{Var}(\widehat{O})}$
$E(\widehat{O})$
$\frac{\sqrt{\mathrm{Var}(\widehat{O})}}{E(\widehat{O})}$
$O = |0\rangle\langle 0|$
Power, k
Zoë Holmes

Optimizing initial state for linear combination of states analysis

$$|\psi\rangle = \alpha_0|\psi^{(0)}\rangle + \alpha_1|\psi^{(1)}\rangle$$



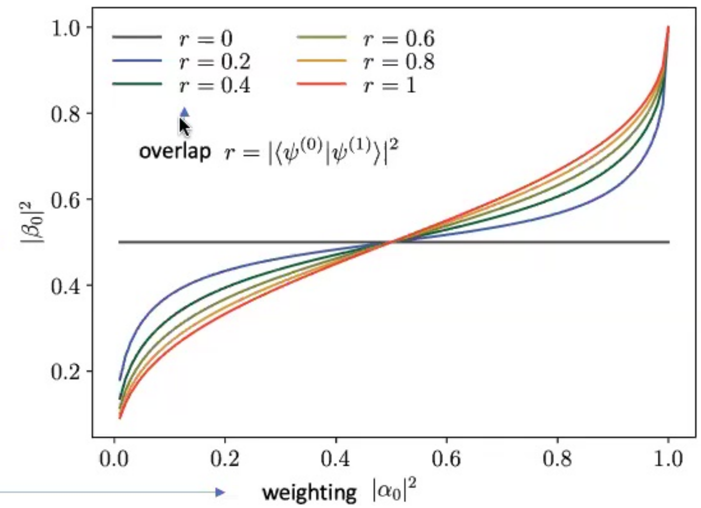
Recall there is freedom in how to pick the initial state σ

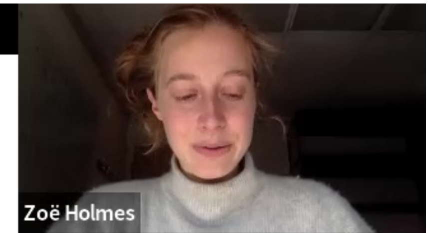
Suppose, as before we write:

$$\sigma = |\beta\rangle\langle\beta| \text{ with } |\beta\rangle = \beta_0|0\rangle + \beta_1|1\rangle$$

Then we find that....

Optimum choice in β_0

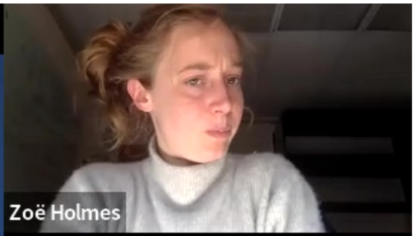




Summary

On nonlinear transformations in quantum computation

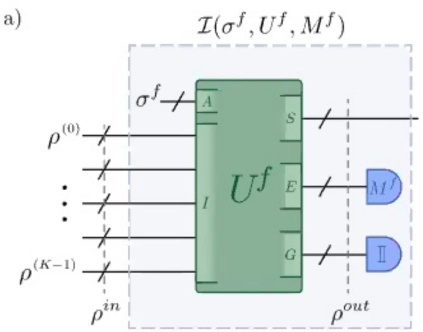
arXiv:2112.12307



"A solution searching for a problem?"

1) Introduced the weighted state framework

- Weighted states play a similar role to standard density matrices...
- But are liberated from the constraints of positivity, Hermiticity and normalization
- and hence can be generic functions of input states



2) Introduced three primitives for implementing non-linear operations

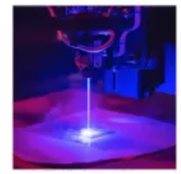
- Quantum Hadamard Product $\rho^{(0)} \odot \rho^{(1)}$
- Generalized Transpose $\rho_{\sigma}^{(T)} = \sigma \odot \rho^T$
- Quantum State Polynomial $\alpha_{00}\rho^{(0)} + \alpha_{11}\rho^{(1)} + \alpha_{01}\rho^{(0)}\rho^{(1)} + \alpha_{10}\rho^{(1)}\rho^{(0)}$

Combined: $|\psi\rangle \rightarrow |f(\psi)\rangle = \sum_i f(\psi_i)|i\rangle \text{ where } f(\psi_i) = \sum_j \alpha_j \psi_i^j$

Summary

On nonlinear transformations in quantum computation
arXiv:2112.12307

"A solution searching for a problem?"



1) Introduced the weighted state framework

- Weighted states play a similar role to standard density matrices…
- But are liberated from the constraints of positivity, Hermiticity and normalization
- and hence can be generic functions of input states

2) Introduced three primitives for implementing non-linear operations

- Quantum Hadamard Product $\rho^{(0)} \odot \rho^{(1)}$
- Generalized Transpose $\rho_{\sigma}^{(T)} = \sigma \odot \rho^{T}$
- Quantum State Polynomial $\alpha_{00}\rho^{(0)} + \alpha_{11}\rho^{(1)} + \alpha_{01}\rho^{(0)}\rho^{(1)} + \alpha_{10}\rho^{(1)}\rho^{(0)}$

Combined: $|\psi\rangle \rightarrow |f(\psi)\rangle = \sum_i f(\psi_i)|i\rangle$ where $f(\psi_i) = \sum_j \alpha_j \psi_i^j$

a)