Simulating SYK Wormhole Teleportation and Learning Mutual Information on an IBM Eagle Processor

Oscar Scholin
Advisor: Professor Ami Radunskaya

April 12, 2024

Motivation

What is the most significant technological development since the Industrial Revolution?

Motivation

What is the most significant technological development since the Industrial Revolution?

1. Atomic weapons?

Motivation

What is the most significant technological development since the Industrial Revolution?

1. Atomic weapons?
2. Going to the moon?

Motivation

What is the most significant technological development since the Industrial Revolution?

1. Atomic weapons?
2. Going to the moon?
3. Tik-Tok?

Motivation

What is the most significant technological development since the Industrial Revolution?

1. Atomic weapons?
2. Going to the moon?
3. Tik-Tok?

the computer.

Motivation

How do we compute?

Motivation

How do we compute?

Current computers are built on *classical* logic — true (1) or false (0).

How do we compute?

Current computers are built on *classical* logic — true (1) or false (0).

The universe is *not* classical, but *quantum* — how could we hope to simulate *non*-classical using classical?

Motivation

How do we compute?

Current computers are built on *classical* logic — true (1) or false (0).

The universe is *not* classical, but *quantum* — how could we hope to simulate *non*-classical using classical?

Think of how impactful the computer is in your daily life. What if we change the very basis of how we compute?

Question of the thesis

How to program wormhole teleportation on a quantum computer?

How to program wormhole teleportation on a quantum computer?

1. build simulation environment based on the description given in the paper by Jafferis et al 2022 [2],

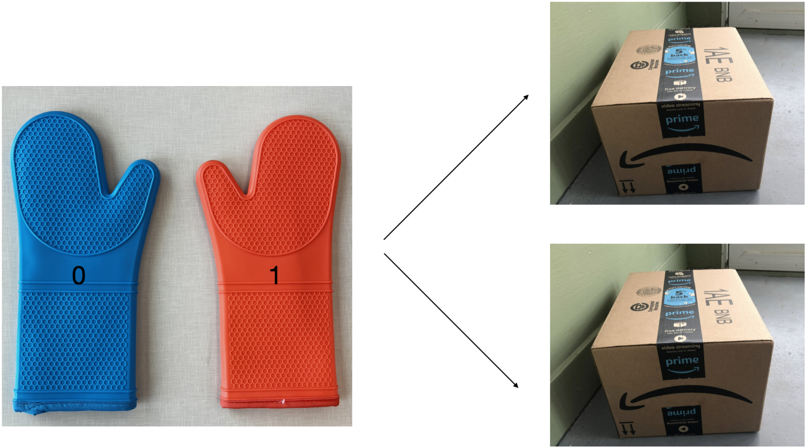
Question of the thesis

How to program wormhole teleportation on a quantum computer?

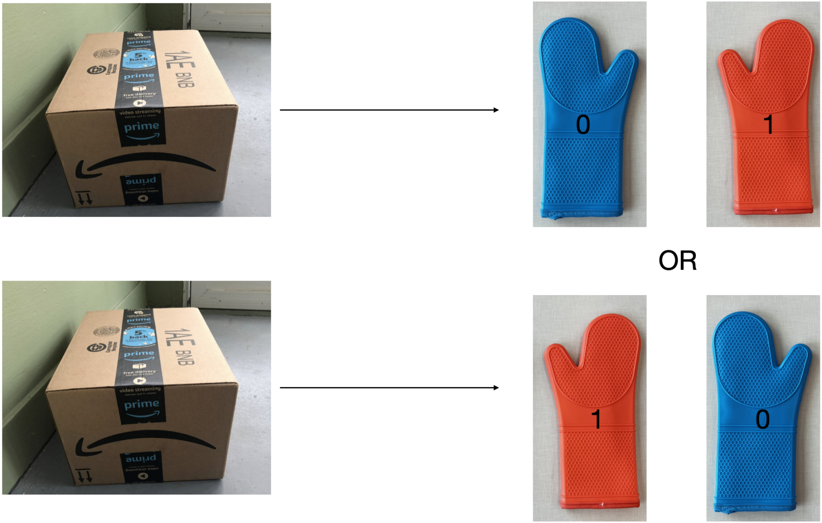
1. build simulation environment based on the description given in the paper by Jafferis et al 2022 [2],
2. develop better unbiased machine learning procedure to simplify the simulation model.

Quantum Oven Mits

Quantum Oven Mits



Quantum Oven Mits



Quantum Oven Mits

This is an example of correlation: opening one box will immediately tell you whether the other mitt is a 0 or a 1.

Quantum Oven Mits

This is an example of correlation: opening one box will immediately tell you whether the other mitt is a 0 or a 1.

Quantum entanglement is when information is stored not in the particles themselves but *only* in their correlations.

Quantum Oven Mits

This is an example of correlation: opening one box will immediately tell you whether the other mitt is a 0 or a 1.

Quantum entanglement is when information is stored not in the particles themselves but *only* in their correlations. These correlations are persistent even as we change the basis of measurement.

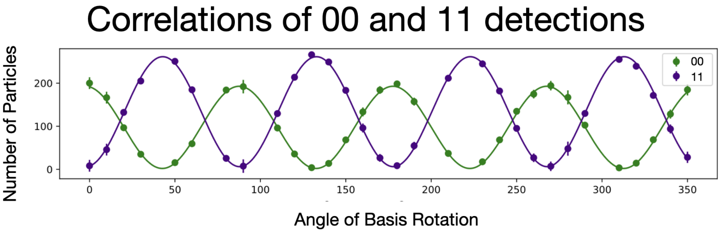


Figure: Measuring entangled particles.

Qubit

*Quant*um means *quant*ized:

Qubit

*Quant*um means *quant*ized: states are discrete. Like the oven mits, $|0\rangle$ and $|1\rangle$.

Qubit

*Quant*um means *quant*ized: states are discrete. Like the oven mits, $|0\rangle$ and $|1\rangle$. We can capture the inherent indeterminateness by writing a state $|\phi\rangle$ as a *superposition* of basis states:

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

Qubit

Quantum means *quantized*: states are discrete. Like the oven mits, $|0\rangle$ and $|1\rangle$. We can capture the inherent indeterminateness by writing a state $|\phi\rangle$ as a *superposition* of basis states:

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

where $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$.

Qubit

Quantum means *quantized*: states are discrete. Like the oven mits, $|0\rangle$ and $|1\rangle$. We can capture the inherent indeterminateness by writing a state $|\phi\rangle$ as a

superposition of basis states:

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

where $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$.

If the state is 2-level, i.e. consisting of only $|0\rangle$ and $|1\rangle$ for some α, β, then we call $|\phi\rangle$ a *qubit*.

$|\cdot\rangle$ is a column vector, e.g. in the standard basis,

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2}$$

(3)

$|\cdot\rangle$ is a column vector, e.g. in the standard basis,

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2}$$

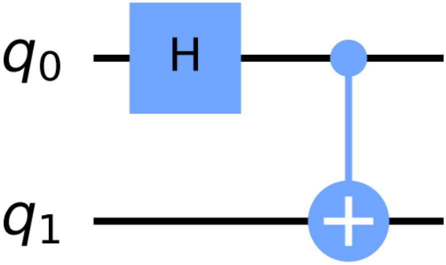


Figure: Creating entanglement!

$$\tag{3}$$

$|\cdot\rangle$ is a column vector, e.g. in the standard basis,

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \text{ and } |1\rangle = \begin{bmatrix}0\\1\end{bmatrix} \tag{2}$$

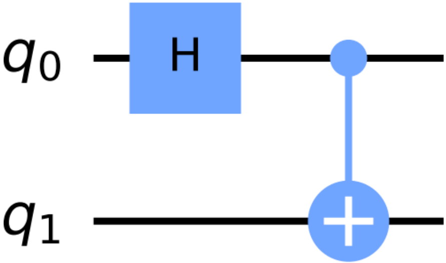


Figure: Creating entanglement!

$$|0\rangle \otimes |0\rangle \tag{3}$$

$|\cdot\rangle$ is a column vector, e.g. in the standard basis,

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2}$$

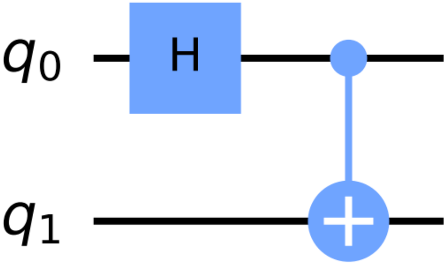


Figure: Creating entanglement!

$$(H \otimes I)|0\rangle \otimes |0\rangle \tag{3}$$

$|\cdot\rangle$ is a column vector, e.g. in the standard basis,

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \text{ and } |1\rangle = \begin{bmatrix}0\\1\end{bmatrix} \tag{2}$$

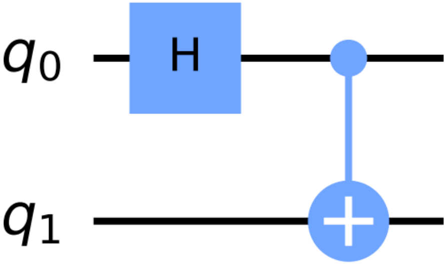


Figure: Creating entanglement!

$$\text{CNOT}(H \otimes I)|0\rangle \otimes |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle) \tag{3}$$

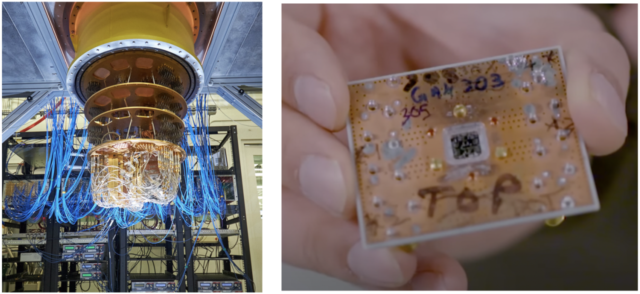


Figure: (*left*) Google's Sycamore processor. (*right*) IBM 4-qubit chip.

What is a wormhole?

And now... for something completely different...

What is a wormhole?

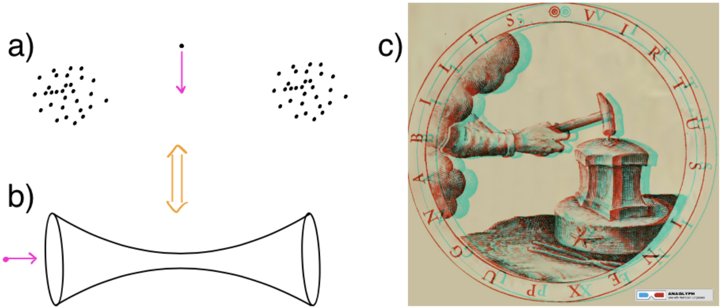


Figure: Visual representation of the AdS/CFT correspondence.

SYK Hamiltonians describe the dynamics

Take one of those clouds of particles. We can describe how the system evolves in time in terms of a *Hamiltonian* $\hat{H}$, which is the operator from Schrödinger's equation [2].

$$ \tag{4} $$

SYK Hamiltonians describe the dynamics

Take one of those clouds of particles. We can describe how the system evolves in time in terms of a *Hamiltonian* $\hat{H}$, which is the operator from Schrödinger's equation [2]. Consider an interaction of 4 particles:

$$\hat{H} = \psi_i \psi_j \psi_k \psi_l \tag{4}$$

SYK Hamiltonians describe the dynamics

Take one of those clouds of particles. We can describe how the system evolves in time in terms of a *Hamiltonian* $\hat{H}$, which is the operator from Schrödinger's equation [2]. Consider an interaction of 4 particles:

$$\hat{H} = \mathcal{J}_{ijkl}\psi_i\psi_j\psi_k\psi_l \tag{4}$$

SYK Hamiltonians describe the dynamics

Take one of those clouds of particles. We can describe how the system evolves in time in terms of a *Hamiltonian* $\hat{H}$, which is the operator from Schrödinger's equation [2]. Consider an interaction of 4 particles:

$$\hat{H} = \sum_{i=0}^{N_m-1} \sum_{j=i+1}^{N_m-1} \sum_{k=j+1}^{N_m-1} \sum_{l=k+1}^{N_m-1} \mathcal{J}_{ijk} \psi_i \psi_j \psi_l \psi_k, \tag{4}$$

where $\mathcal{J}_{ijkl}$ drawn from normal distribution with mean 0 and variance $\sigma^2 = \frac{3! J^2}{N_m^3}$ for $J = 2$.

Entropy is information

Von Neumann Entropy is

$$S_X = -X \ln X \tag{5}$$

Pomona College

Von Neumann Entropy is

$$S_X = -X \ln X \tag{5}$$

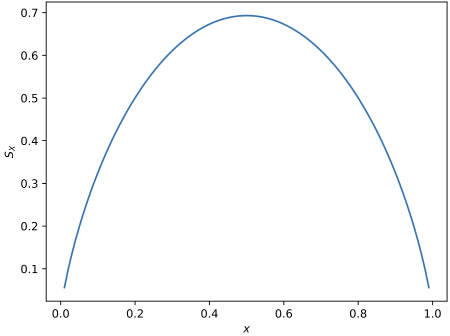


Figure: Entropy of $x, 1-x$.

Von Neumann Entropy is

$$S_X = -X \ln X \qquad (5)$$

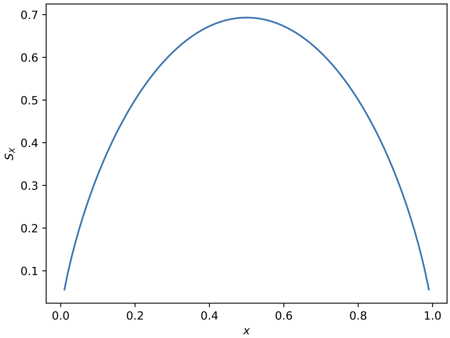


Figure: Entropy of $x, 1 - x$.

S is maximized when both states are equally likely, i.e. most unpredictable.

Von Neumann Entropy is

$$S_X = -X \ln X \tag{5}$$

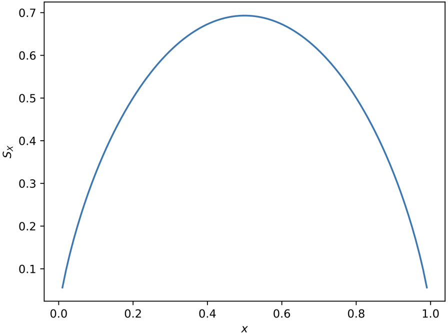


Figure: Entropy of $x, 1-x$.

S is maximized when both states are equally likely, i.e. most unpredictable. The more you don't know, the more each observation tells you.

Entanglement is entropy

We can generalize this idea of entropy to quantum states, where ρ is a *density matrix* $\rho = |\phi\rangle|\phi\rangle^\dagger = |\phi\rangle\langle\phi|$:

$$S_\rho = -\operatorname{Tr} \rho \ln \rho \tag{6}$$

Entanglement is entropy

We can generalize this idea of entropy to quantum states, where ρ is a *density matrix* $\rho = |\phi\rangle|\phi\rangle^\dagger = |\phi\rangle\langle\phi|$:

$$S_\rho = -\operatorname{Tr}\rho\ln\rho \tag{6}$$

We can quantify entanglement based on the amount of *mutual information* shared between qubits within an overall system:

$$I(P:T) = S_{\rho_P} + S_{\rho_T} - S_{\rho_{PT}}, \tag{7}$$

where S_{ρ_P}, S_{ρ_T}, $S_{\rho_{PT}}$ are *reduced density matrices* of ρ. This means we are isolating particular subsystems at a time.

Wormhole protocol

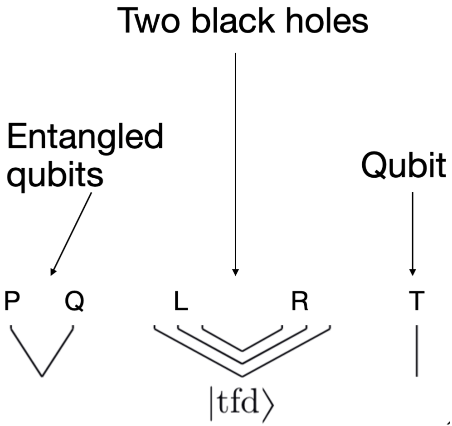


Figure: Wormhole teleportation, [2]

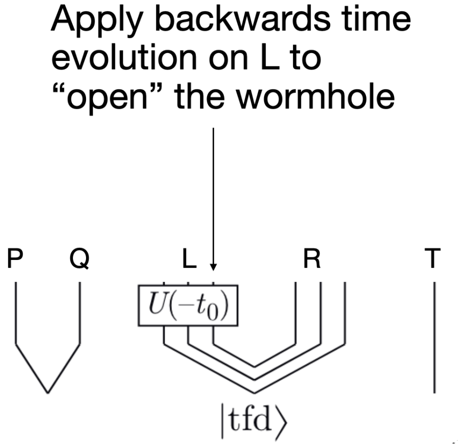


Figure: Wormhole teleportation, [2]

Wormhole protocol

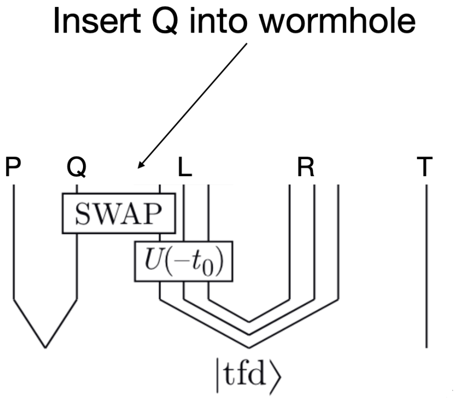


Figure: Wormhole teleportation, [2]

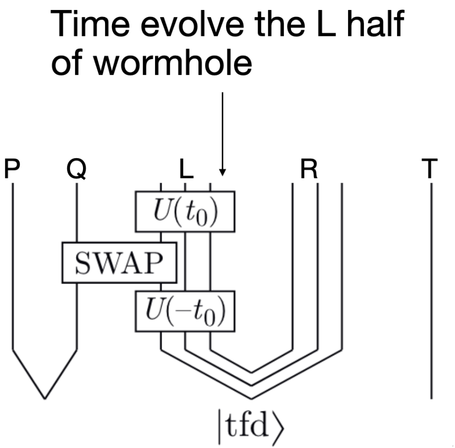


Figure: Wormhole teleportation, [2]

Wormhole protocol

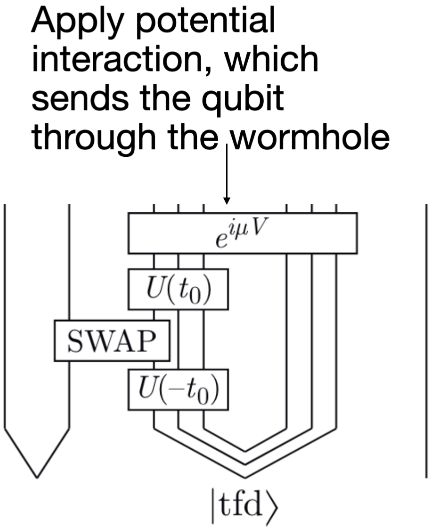


Figure: Wormhole teleportation, [2]

Wormhole protocol

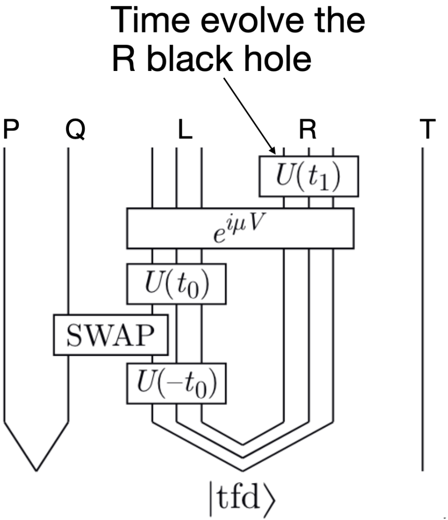


Figure: Wormhole teleportation, [2]

Wormhole protocol

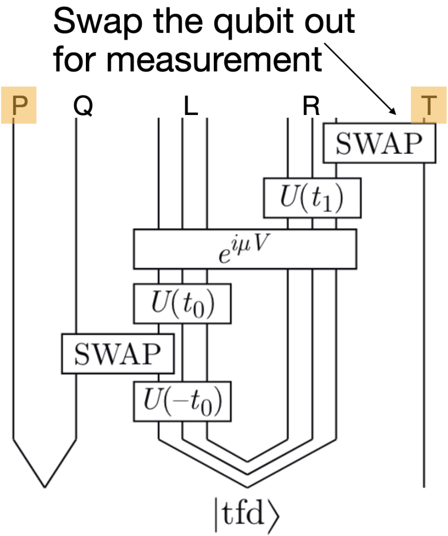


Figure: Wormhole teleportation, [2]

Coding in Qiskit

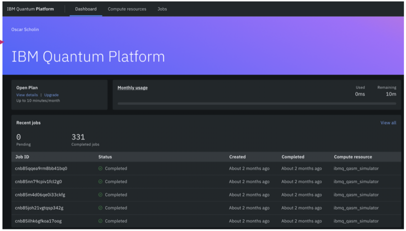
```python
def protocol_round(H_R, tfd, expV, tev_nt0, tev_pt0, t, display_circs=False, save_param=None):
    '''Implements a single round of the the wormhole protocol for the SYK model with n_majorana fermions at time t, interaction param mu,
    and inverse temperature beta.'''
    ## STEP 1: generate TFD and apply negative time evolution on L
    # make tfd
    total_circuit = QuantumCircuit(2*H_R.num_qubits + 2)

    ## STEP 2: swap in register Q of bell pair into tfd
    total_circuit.h(0)
    total_circuit.cx(0, 1)

    # swap in the bell pair
    total_circuit.swap(1, 2)

    # set the tfd within the larger full circuit with registers P, Q before it and T at the end
    for gate in tfd.data:
        qubits = [q.index + 2 for q in gate[1]]
        total_circuit.append(gate[0], qubits) # start at 2 to account for the extra registers

    # apply backwards time evolution to L part of tfd
    for gate in tev_nt0.data:
        qubits = [q.index + 2 for q in gate[1]]
        total_circuit.append(gate[0], qubits)

    ## STEP 3: apply forward time evolution to L part of tfd
    for gate in tev_pt0.data:
        qubits = [q.index +2 for q in gate[1]]
        total_circuit.append(gate[0], qubits)

    ## STEP 4: apply expV to all tfd
    for gate in expV.data:
        qubits = [q.index + 2 for q in gate[1]]
        total_circuit.append(gate[0], qubits)

    ## STEP 5: apply forward time evolution by t1 to R part of tfd
    tev_pt1 = time_evolve(H_R, tf=t)
    for gate in tev_pt1.data:
        qubits = [q.index + H_R.num_qubits +2 for q in gate[1]]
        total_circuit.append(gate[0], qubits)

    ## STEP 6: SWAP out qubit (skip, since we'll just measure on the last qubit)

    I = compute_mi(total_circuit, display_circs=display_circs, save_param=save_param)
    if display_circs:
        print('Mutual info:', I)
    # print(f'Mutual info at t = {t}: {I}')
    return I
```

Coding in Qiskit

backend

```python
def compute_mi_actual(circuit, backend, shots=10000):
    ''' Computes mutual info between first and last qubit using Qiskit Experiments for state tomography.'''

    # Setup state tomography on the first and last qubits
    tomo_experiment = StateTomography(circuit, measurement_indices=[0, circuit.num_qubits - 1])

    # Run the state tomography experiment
    experiment_data = tomo_experiment.run(backend, shots=shots).block_for_results()

    # Access analysis results directly
    analysis_results = experiment_data.analysis_results()

    # Process analysis results as needed
    tomo_result = analysis_results[0].value

    # Compute the mutual information
    # Get the reduced density matrices
    rho_P = partial_trace(tomo_result, [0])
    rho_T = partial_trace(tomo_result, [1])
    rho_PT = tomo_result  # The joint state of the first and last qubits

    # Compute the mutual info
    mutual_info = entropy(rho_P) + entropy(rho_T) - entropy(rho_PT)
    return mutual_info
```

Free cloud API access!



Results of the simulation

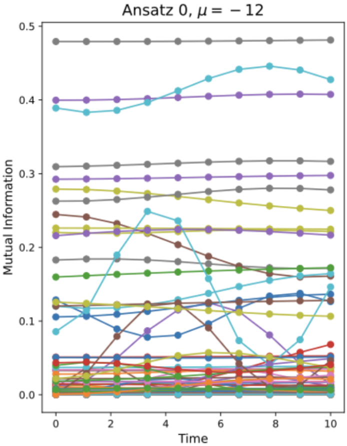
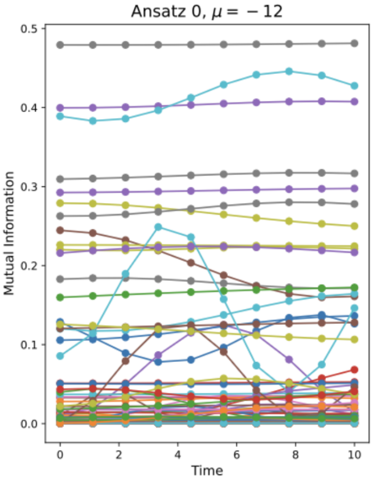


Figure: Classical simulations, with $N_m = 10$. This translates to 333 CNOT and 171 U3 gates and 13 qubits.



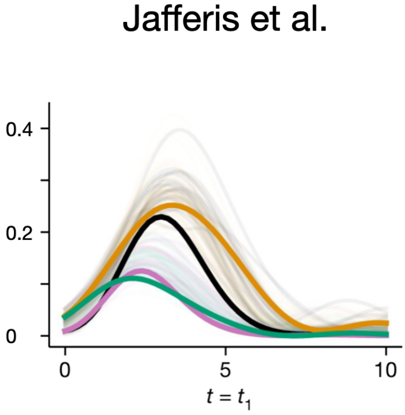


Figure: Classical simulations, with $N_m = 10$. This translates to 333 CNOT and 171 U3 gates and 12 qubits—not achievable with today's quantum computers. Comparing to [2].

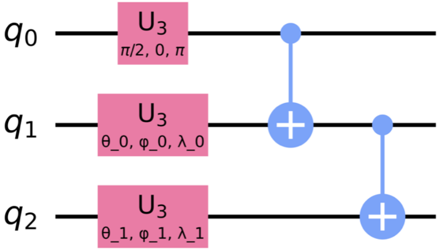
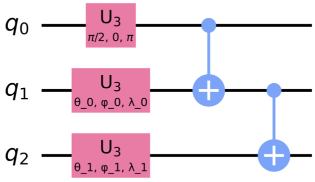


Figure: Guess this circuit, learn the optimal parameters to match the desired mutual information values using stochastic gradient descent.

Real quantum data



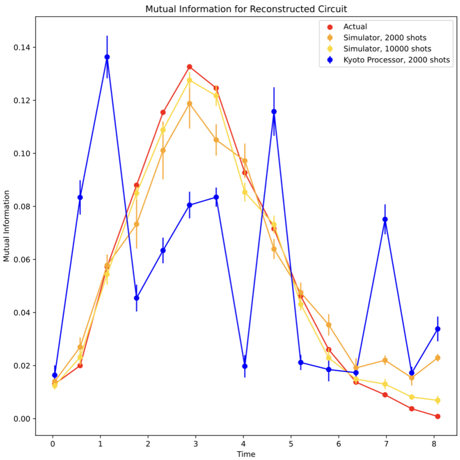


Figure: Using the learned parameters, calculate the mutual information with simulated noise, and then with real quantum data.

Check out my GitHub!

References

[1] Jason Alicea. "New directions in the pursuit of Majorana fermions in solid state systems". en. In: *Reports on Progress in Physics* 75.7 (June 2012). Publisher: IOP Publishing, p. 076501. ISSN: 0034-4885. DOI: 10.1088/0034-4885/75/7/076501. URL: https://dx.doi.org/10.1088/0034-4885/75/7/076501 (visited on 09/16/2023).

[2] Daniel Jafferis et al. "Traversable wormhole dynamics on a quantum processor". en. In: *Nature* 612.7938 (Dec. 2022). Number: 7938 Publisher: Nature Publishing Group, pp. 51–55. ISSN: 1476-4687. DOI: 10.1038/s41586-022-05424-3. URL: https://www.nature.com/articles/s41586-022-05424-3 (visited on 06/14/2023).

[3] Michael E. Peskin. *An Introduction To Quantum Field Theory*. Boca Raton: CRC Press, Jan. 2018. ISBN: 978-0-429-50355-9. DOI: 10.1201/9780429503559.

Thanks for listening!

Thank you to Dr. Rad for all her patience and advice!

New simplification algorithms?

1. Using time evolution to convert $\hat{H}$ into a circuit, then use ridge regression to set many of the coupling constants to 0.
2. Use Markov-chain Monte Carlo instead of optimization via gradient descent.

What might be wrong?

Tried the following ansatzes to VQE:

0. `EfficientSU2` from Qiskit, which is 2 series of CNOTs and single qubit rotations,
1. an initial layer of single qubit rotation gates followed by a chain of CNOTs with additional rotation gates,
2. like (1) but with a CNOT chain on the L,
3. like (2), but CNOT chain applied to L and R

We measured the error relative to the actual ground state as:

$$e_r = \frac{|\hat{E}_0 - E_0|}{|E_0|}. \tag{8}$$

Ansatz	Mean	SEM
0	0.2	0.1
1	0.3	0.1
2	0.2	0.1
3	0.4	0.2

Table: Relative error of ground state energies for 100 repetitions, computing both the mean and standard error of the mean.

What is a traversable wormhole?

According to general relativity,

$$E^2 = m^2 + p^2 \tag{9}$$

where m is an object's rest mass and p is the magnitude of its relativistic momentum. Thus $E > 0$ always.

However, in QFT, negative energy is possible because of virtual—for example, in Hawking radiation. Negative energy is essential to counteract the force of gravity that would act to close the wormhole.

What are Majorana fermions?

Majorana fermions. Here, N_m are the number of particles, which we require to be even. Majorana fermions are half-integer spin particles that must satisfy

$$\psi = \psi^*, \tag{10}$$

where $*$ denotes the anti-particle [3]. These states are relevant to consider because of how they appear in 2D materials like graphite [1]. Equation 10 implies

$$\{\psi_i, \psi_j\} = \delta_{i,j}, \tag{11}$$

where the curly brackets denote the anti-commutator, $\{A, B\} = AB - BA$, and

$$\delta_{i,j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else,} \end{cases} \tag{12}$$

is the Kronecker delta. Equation 11 is the definition of Clifford Algebra, so we can find mathematical objects that satisfy this relation and are useful to us.

Encoding the particles as qubits

The Majorana fermions can be encoded as follows (1 fermion = 2 qubits), where there is an implicit tensor product ($\otimes$) between the matrices:

$$\begin{aligned} \psi_L^j &= Z^{\lfloor \frac{j+1}{2} \rfloor} X I^{N_q - 2 - \lfloor \frac{j+1}{2} \rfloor} \\ \psi_R^j &= Z^{\lfloor \frac{j+1}{2} \rfloor} Y I^{N_q - 2 - \lfloor \frac{j+1}{2} \rfloor}, \end{aligned} \tag{13}$$

where X, Y, Z are Pauli matrices:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \tag{14}$$

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \tag{15}$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{16}$$

The potential coupling is a 2-fermion operation:

$$V = \frac{1}{4N_m} \sum_{j=0}^{N_m-1} \psi_L^j \psi_R^j. \tag{17}$$

The thermofield double state is:

$$|\,\mathrm{TFD}\rangle = e^{H\beta} |n\rangle_L |n\rangle_R, \tag{18}$$

Variation Quantum Eigensolver assumes a set circuit and solves for the parameters in order to create the desired circuit. For this simulation, by [2] we can implement the TFD state by implementing the VQE to create the target state:

$$H_{\mathsf{TFD}} = H_L + H_R + i\beta V. \tag{19}$$

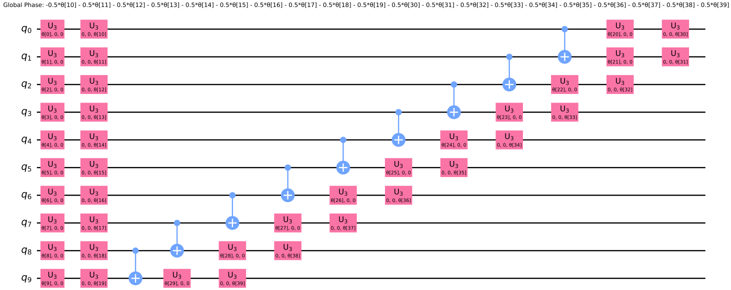


Figure: EfficientSU2 circuit from Qiskit.

What is Trotterization?

Break time evolution into discrete steps.

$$\lim_{n\to\infty}(e^{iAt/n}e^{iBt/n})^n = e^{i(A+B)t} \tag{20}$$

Density matrices and reduced density matrices

Say we have a 2-level system with a density matrix

$$\rho = \begin{bmatrix} \rho_{11} & \rho_{12} & \rho_{13} & \rho_{14} \\ \rho_{21} & \rho_{22} & \rho_{23} & \rho_{24} \\ \rho_{31} & \rho_{32} & \rho_{33} & \rho_{34} \\ \rho_{41} & \rho_{42} & \rho_{43} & \rho_{44} \end{bmatrix}. \tag{21}$$

Then the reduced density matrix of the first subsystem is

$$\rho_1 = \mathrm{Tr}_2(\rho) = \begin{bmatrix} \rho_{11} & \rho_{13} \\ \rho_{31} & \rho_{33} \end{bmatrix} + \begin{bmatrix} \rho_{22} & \rho_{24} \\ \rho_{42} & \rho_{44} \end{bmatrix} = \begin{bmatrix} \rho_{11} + \rho_{22} & \rho_{13} + \rho_{42} \\ \rho_{31} + \rho_{42} & \rho_{33} + \rho_{44} \end{bmatrix}, \tag{22}$$

and the reduced density matrix of the second subsystem is

$$\rho_2 = \mathrm{Tr}_1(\rho) = \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} + \begin{bmatrix} \rho_{33} & \rho_{34} \\ \rho_{43} & \rho_{44} \end{bmatrix} = \begin{bmatrix} \rho_{11} + \rho_{33} & \rho_{12} + \rho_{34} \\ \rho_{21} + \rho_{43} & \rho_{22} + \rho_{44} \end{bmatrix}. \tag{23}$$

Full simulation results

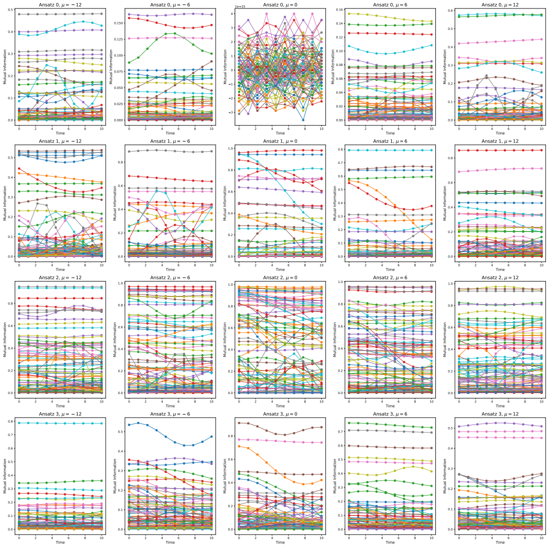


Figure: Full simulation results