Simulating a Photonic Quantum Computer

FockWits

IBM Qiskit camp, Feb 2019

Motivation

Why is this interesing?

- Exploration of photonic quantum computer (CV) concepts on a discretized system

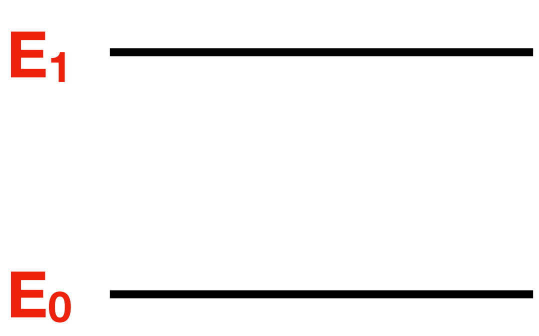
Possible applications:

- Hamiltonian simulation
- Gaussian boson sampling

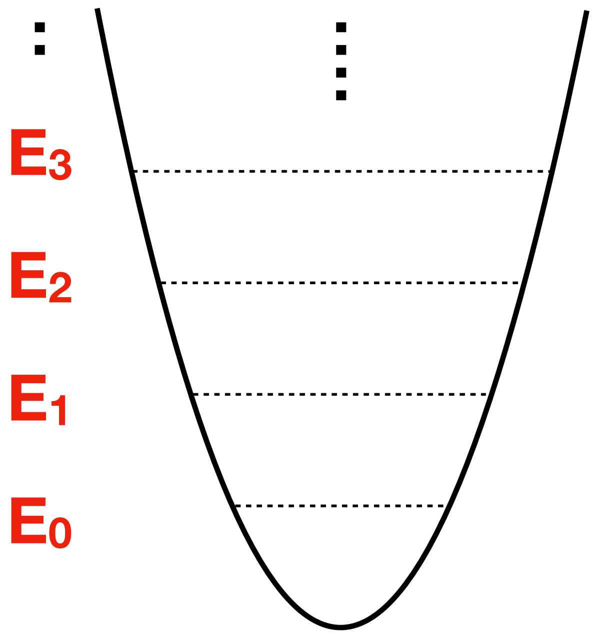
WTFock is CV?

- CV is a paradigm of quantum computing based on qumodes
- A qubit is a two level system, whereas a qumode has the properties of a quantum harmonic oscillator (ie. it has an infinite set of equally spaced energy levels)

Qubit: Two level system



Qumode: Infinite level system



Embedding

- It is not technically feasible to construct a 'true' qumode on a qubit-based system as it would require an infinite number of qubits.
- We can, however, approximate a qumode by using a finite set of qubits to model the lowest n levels, and truncating the rest.
- The number of levels we include will affect the error in our approximation
- Some example choices of embedding are shown below:

Qubits	$\|00\rangle$	$\|01\rangle$	$\|10\rangle$	$\|11\rangle$
Qumode	$\|0\rangle$	$\|1\rangle$	$\|2\rangle$	$\|3\rangle$

2 qubits per qumode

Qubits	$\|000\rangle$	$\|001\rangle$	$\|010\rangle$	$\|011\rangle$	$\|100\rangle$	$\|101\rangle$	$\|110\rangle$	$\|111\rangle$
Qumode	$\|0\rangle$	$\|1\rangle$	$\|2\rangle$	$\|3\rangle$	$\|4\rangle$	$\|5\rangle$	$\|6\rangle$	$\|7\rangle$

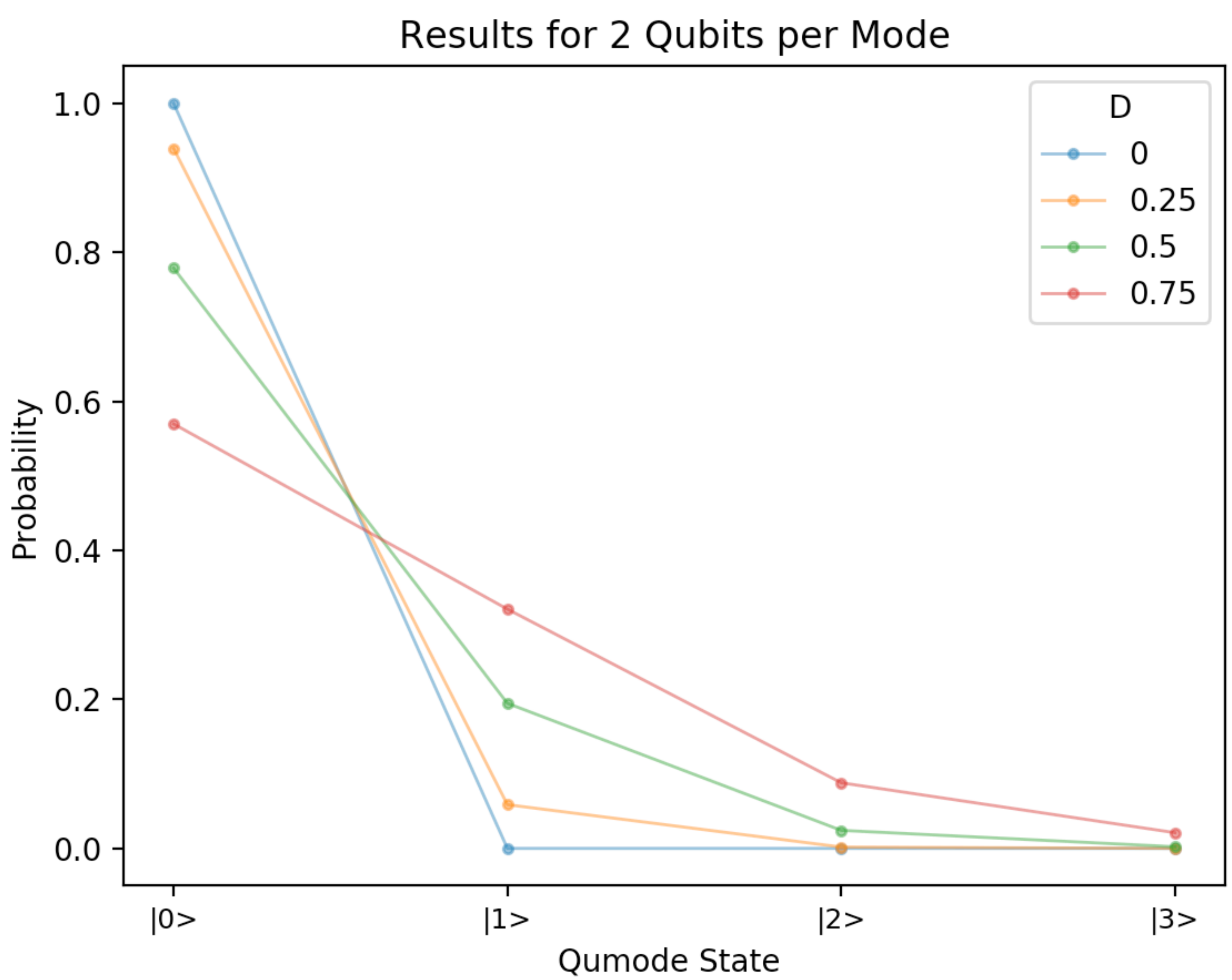
3 qubits per qumode

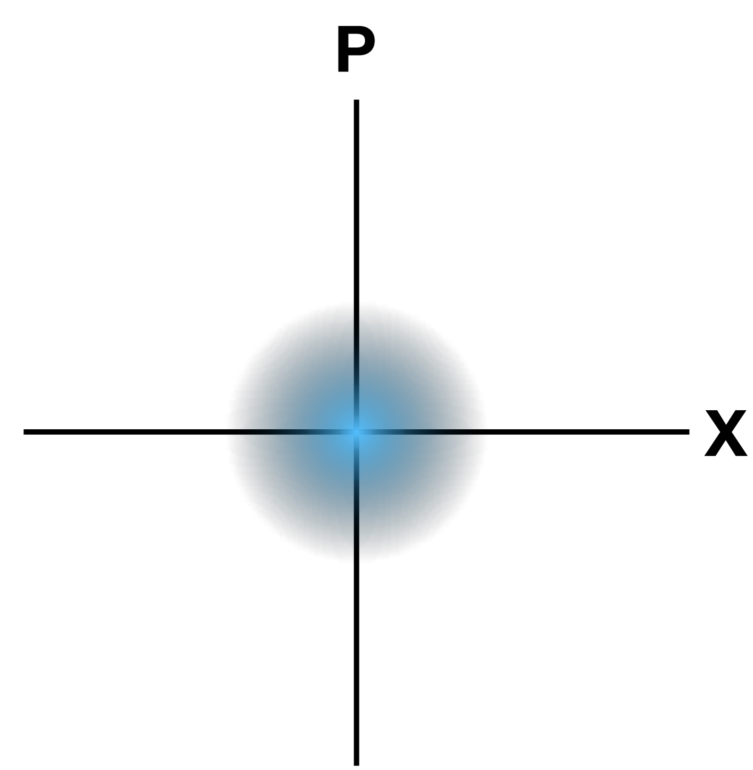
Gaussian Transformations

- We construct our states in the Fock basis by using creation and annihilation operators ($\hat{a}$, $\hat{a}^\dagger$) in a qubit representation, which act to add or subtract a photon.
- For example $D(\alpha) = \exp(\alpha \hat{a}^\dagger - \alpha^* \hat{a})$
- We construct our unitary matrices based on these qumode operations, as shown below

Vacuum State

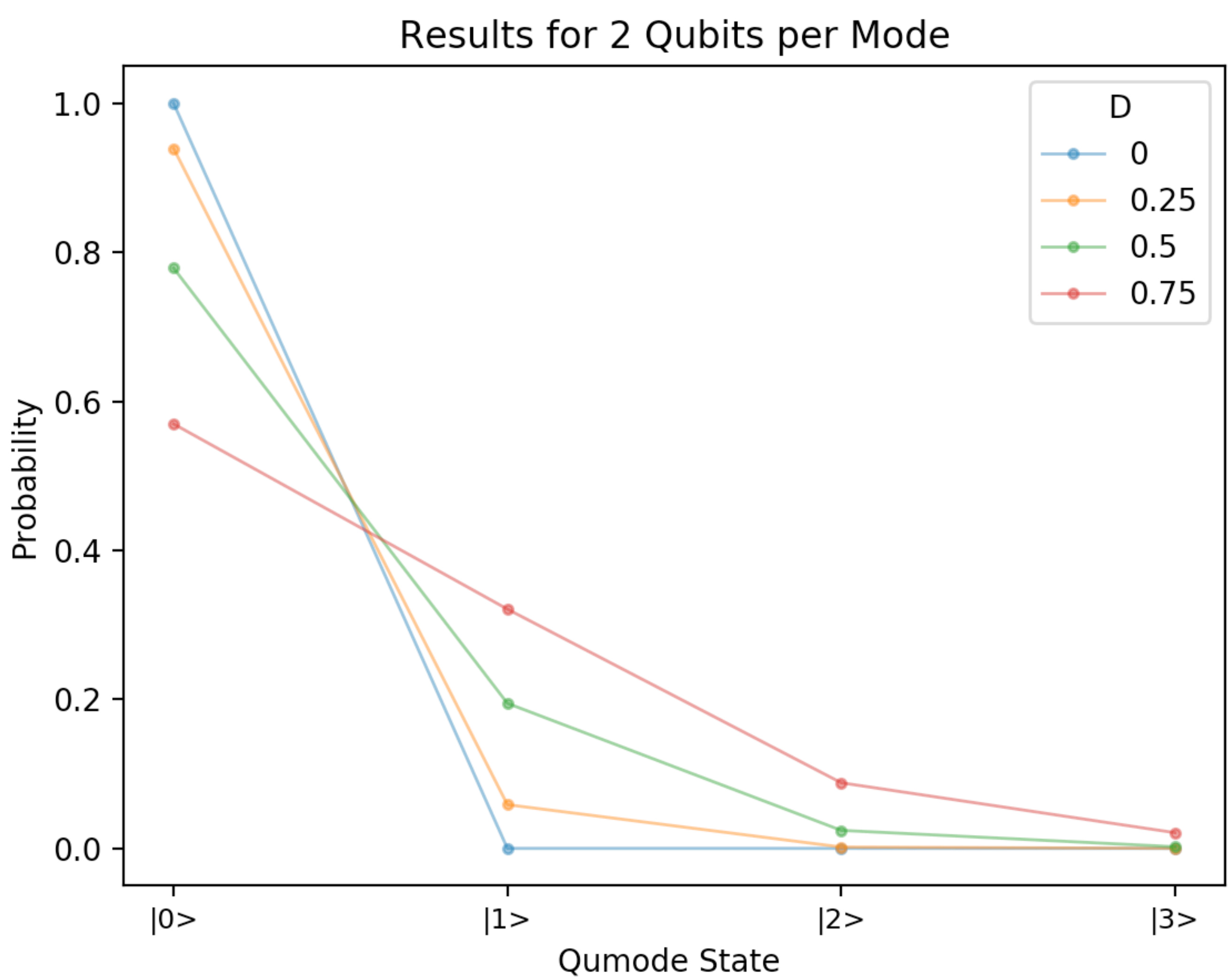
- The vacuum state of our qumode is encoded as the ground state of all qubits in the mode.
- In phase space this is represented as shown below (right)

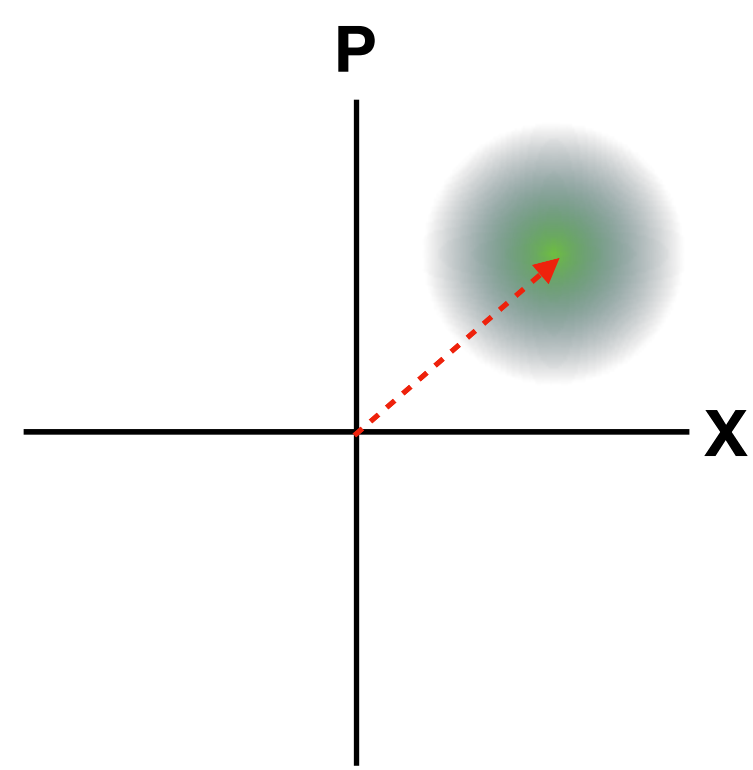




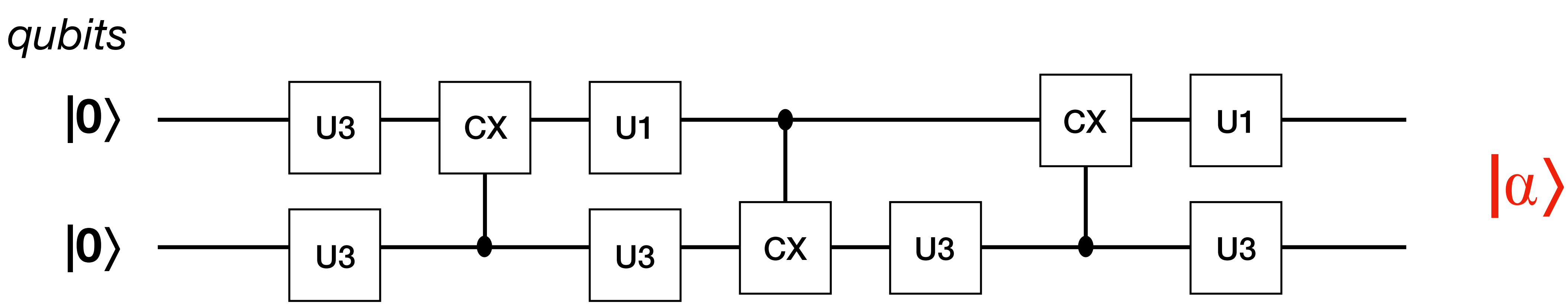
Displacement Operators

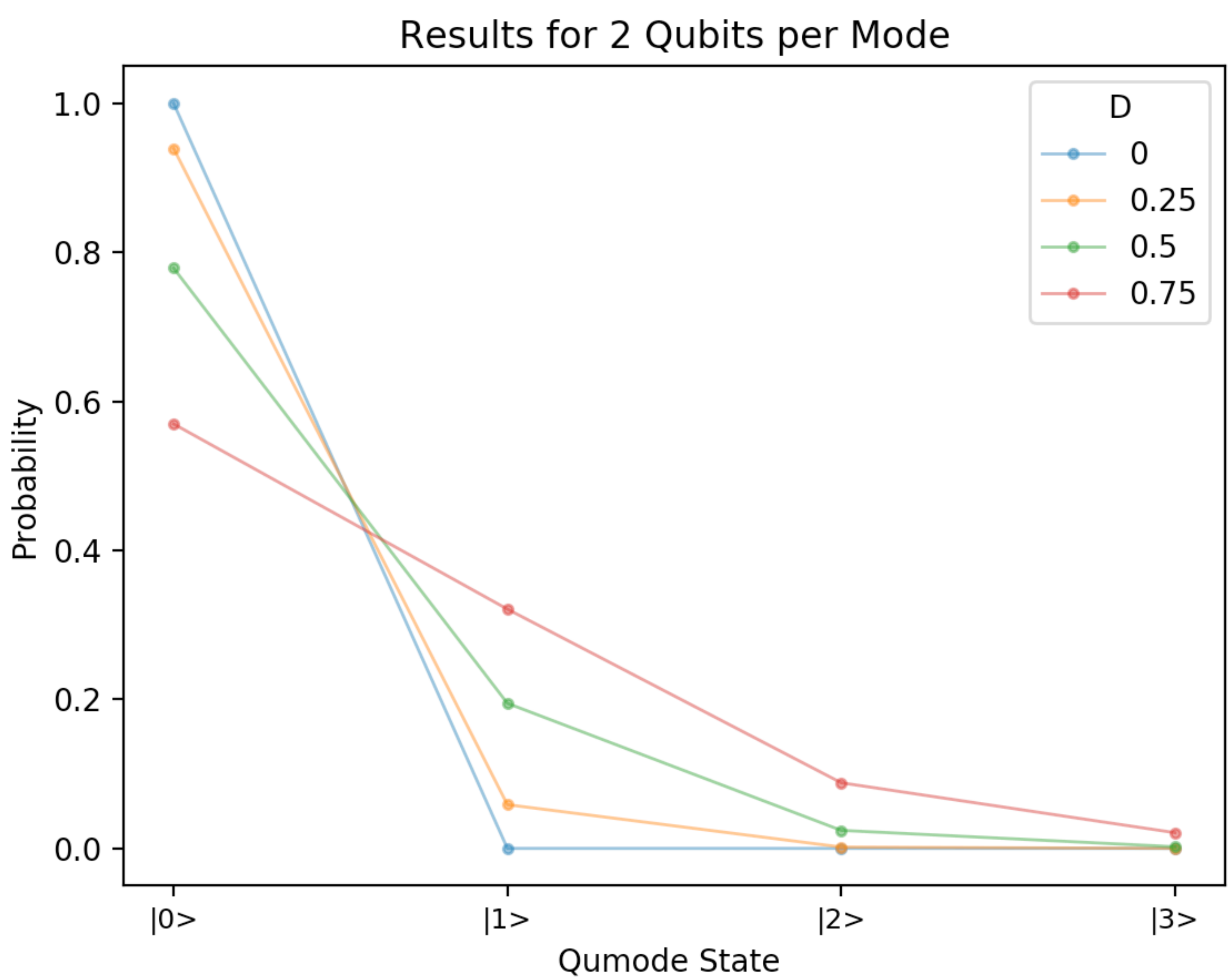
- The displacement operator is $D(\alpha) = \exp(\alpha \hat{a}^\dagger - \alpha^* \hat{a})$
- This, acting on the vacuum state gives a coherent state $|\alpha\rangle$ which has non-zero probabilities for excited qumode states, as shown below (left)

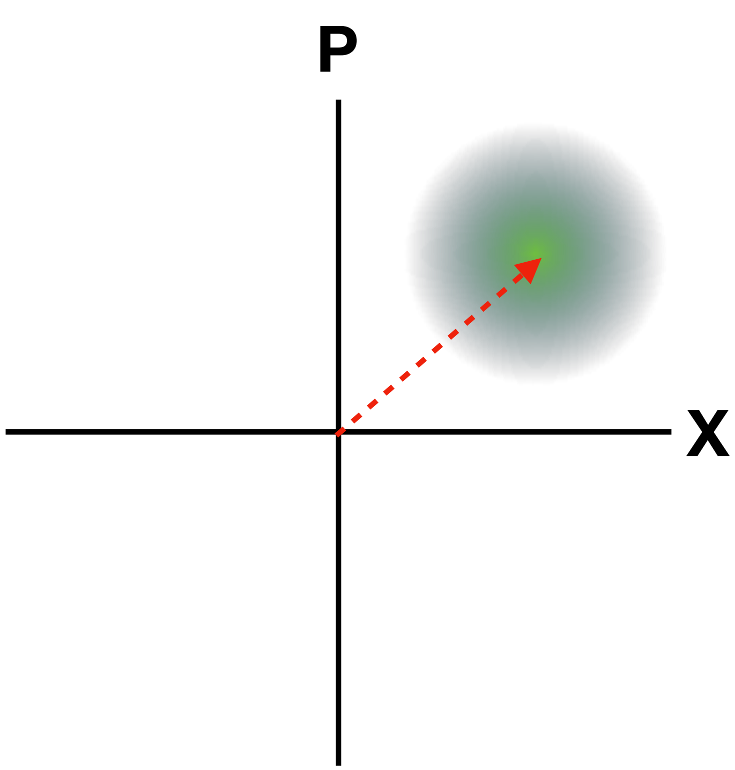




Displacement Operators

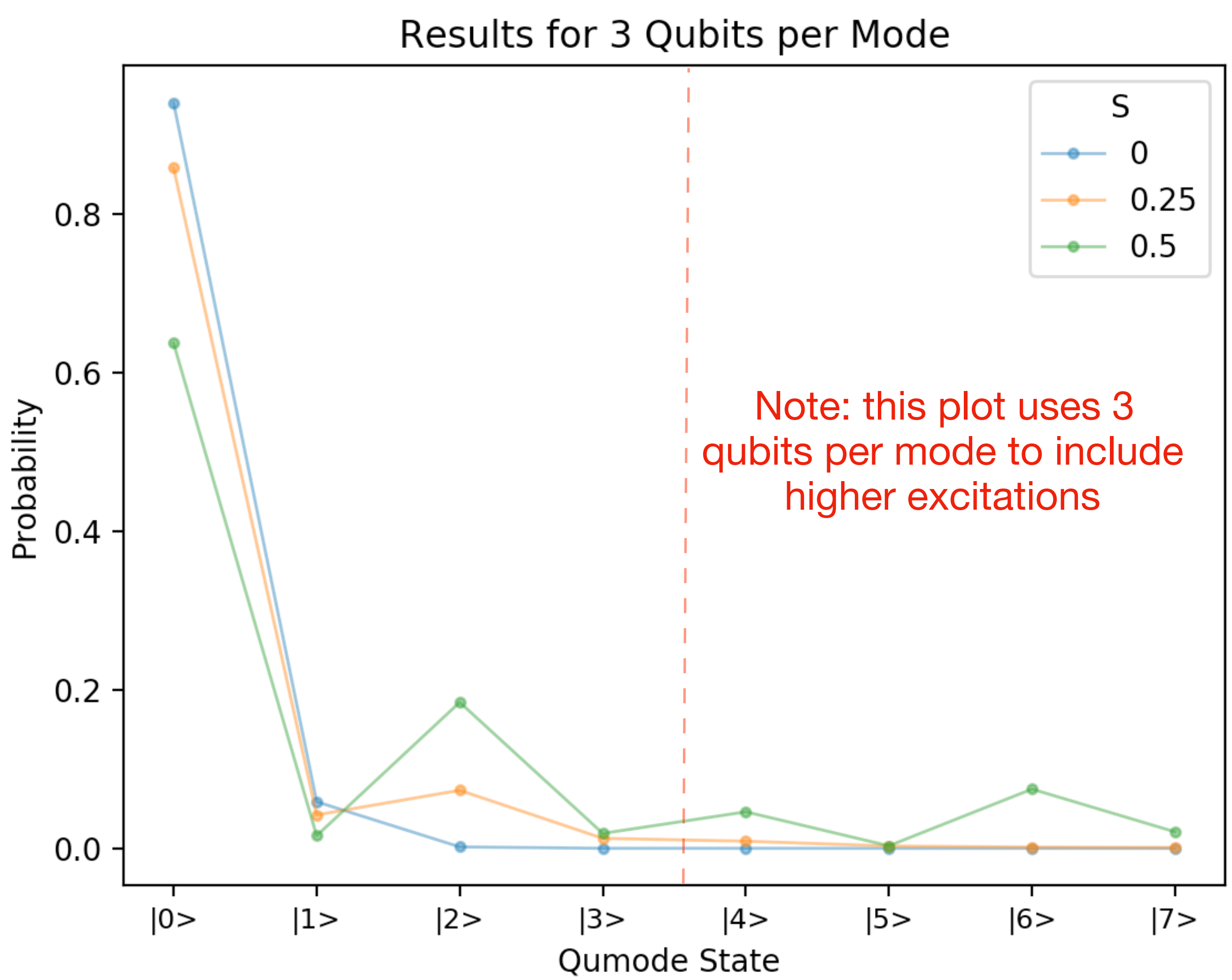


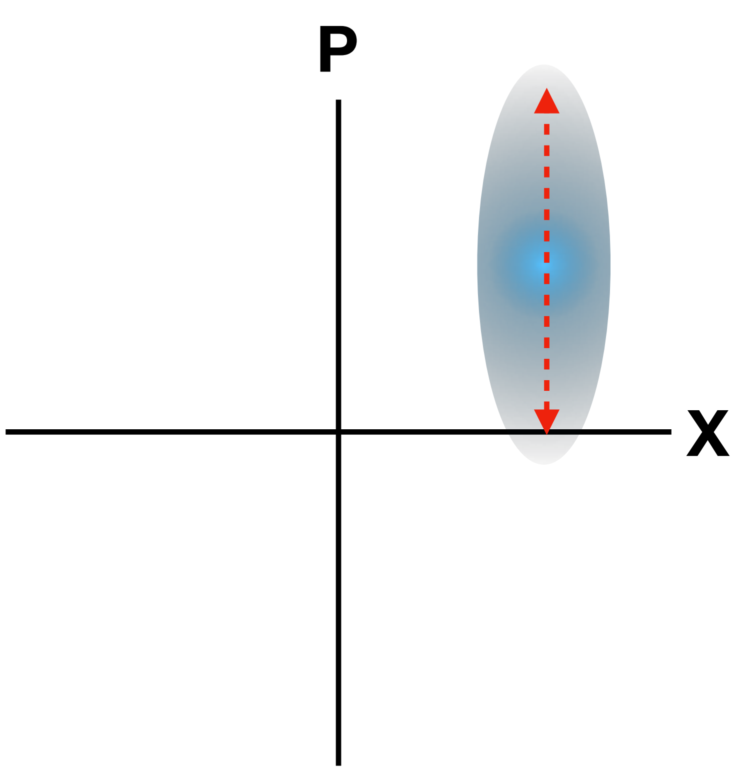




Squeezing Operations

- The squeezing operator is $S(z) = \exp\left(\frac{1}{2}\left(z^*\hat{a}^2 - z\hat{a}^{\dagger 2}\right)\right)$
- We act this on the coherent state and the results for different squeezing strengths are shown below (left).





Overview of gates

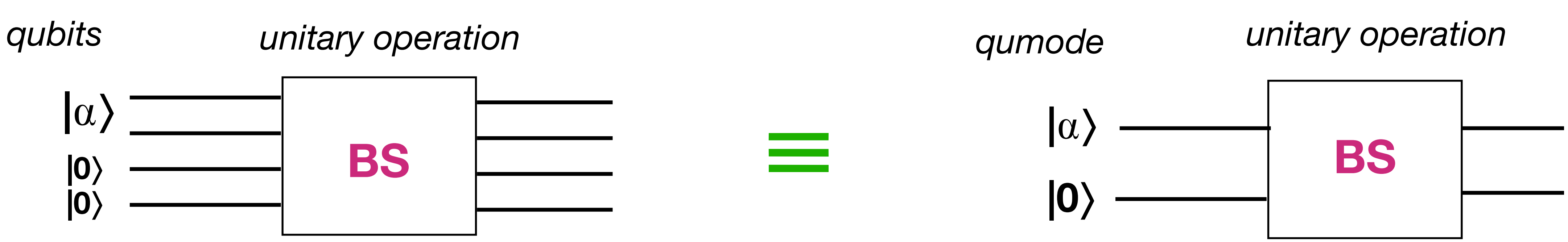
We developed a library of unitary CV operations to represent qumode gates:

- ☑ Single qumode gates: displacement (D), rotation (R), squeezing (S), Kerr (K, non-gaussian)

Beam Splitter (BS)

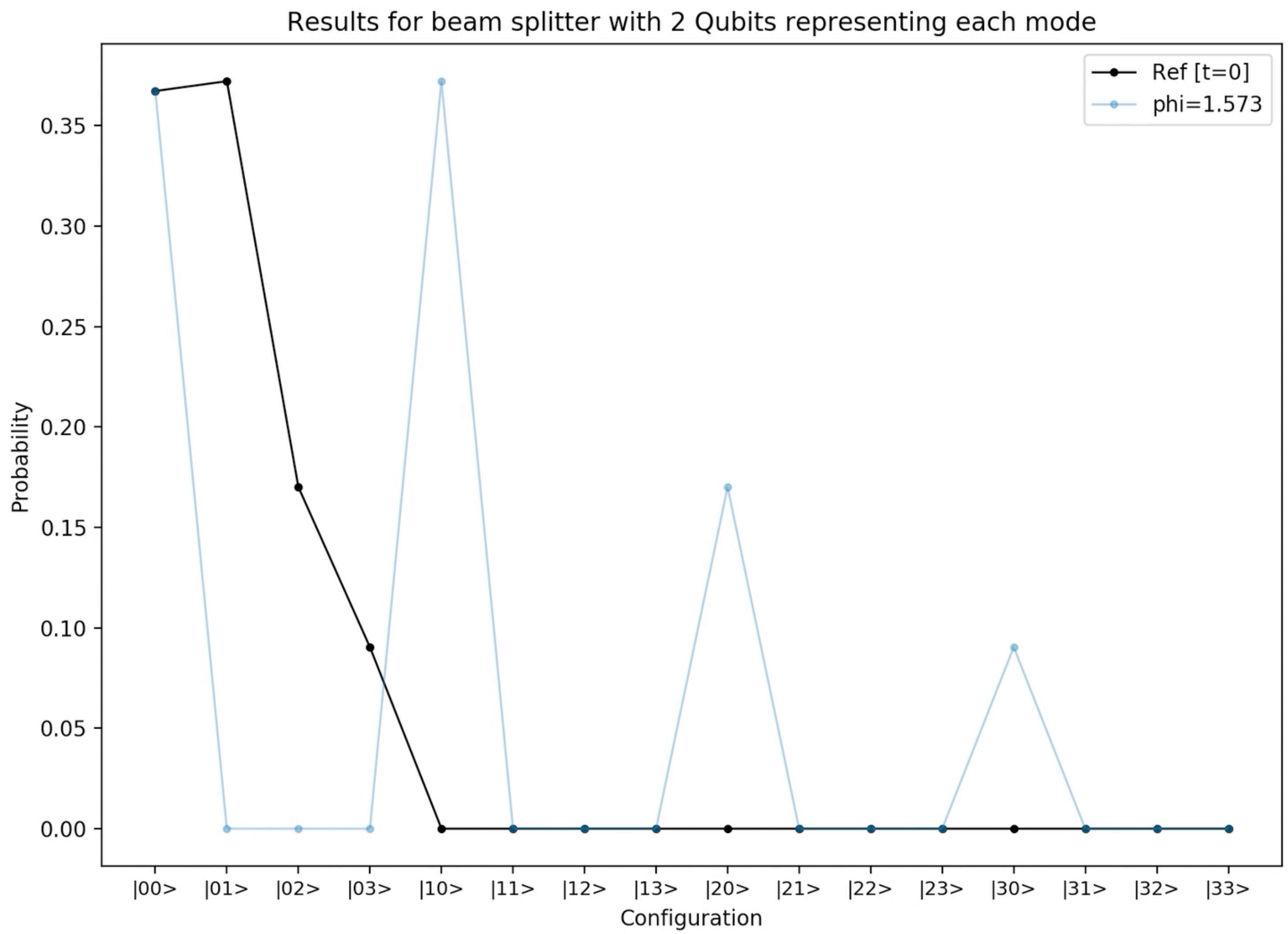
- A beam splitter is a multi-qumode operation which is defined as $B(\theta, \phi) = \exp\big(\theta(e^{i\phi}\hat{a}_1\hat{a}_2^\dagger - e^{-i\phi}\hat{a}_1^\dagger\hat{a}_2)\big)$

- It is a continuous counterpart to the CNOT gate. It is the only two qumode gate required for a universal CV gate set.



Total BS

- Movie shows how the beam splitter shift the coherent state from one mode to another in oscillating fashion.



Overview of gates

We developed a library of unitary CV operations to represent qumode gates:

- ☑ Single qumode gates: displacement (D), rotation (R), squeezing (S), Kerr (K, non-gaussian)

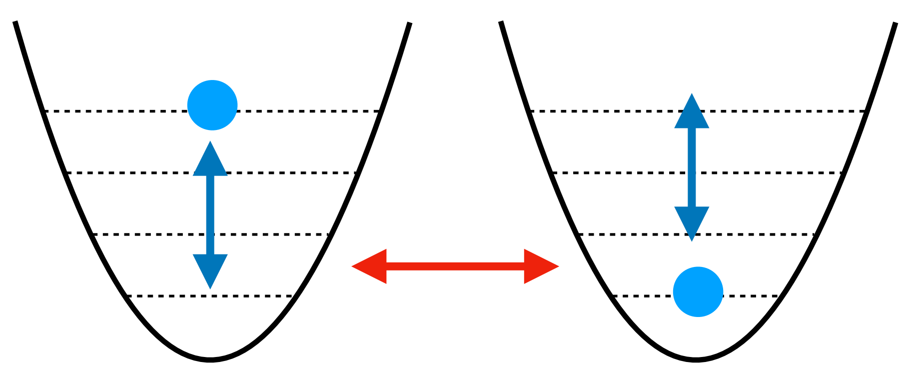
- ☑ Two qumode gates: beamsplitter (BS), two mode squeezing (S2)

Implementation on qiskit

```
# ==== Initialize circuit =====

qr = QuantumRegister(n_qubits_per_mode*n_qumodes)
cr = ClassicalRegister(n_qubits_per_mode*n_qumodes)
circuit = QuantumCircuit(qr, cr)
cv_circuit = CVCircuit(circuit, qr, n_qubits_per_mode)

# ==== Build circuit ====

cv_circuit.initialize([0,0])
cv_circuit.DGate(alpha, 0)
cv_circuit.BSGate(phi, (0,1))
circuit.measure(qr, cr)
```

Bose Hubbard Model

"The problem of simulating the dynamics of quantum systems was the original motivation for quantum computers and remains one of their major potential applications." - Berry et al.



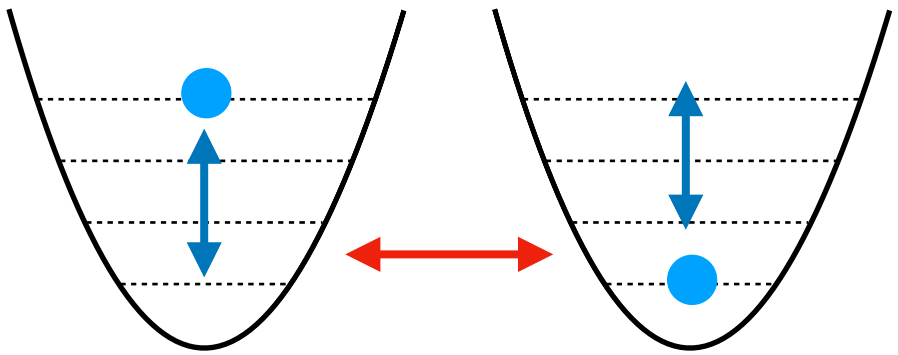
Bose Hubbard Model

Mobility of bosons on lattice

On-site interaction

$$H = J \sum_i \sum_j A_{ij} \hat{a}_i^\dagger \hat{a}_j + \frac{1}{2} U \sum_i \hat{n}_i (\hat{n}_i - 1)$$

$$e^{-iHt} = \left[\exp\left(-i\frac{Jt}{k}(\hat{a}_1^\dagger \hat{a}_2 + \hat{a}_2^\dagger \hat{a}_1)\right) \exp\left(-i\frac{Ut}{2k}\hat{n}_1^2\right) \exp\left(-i\frac{Ut}{2k}\hat{n}_2^2\right) \exp\left(i\frac{Ut}{2k}\hat{n}_1\right) \exp\left(i\frac{Ut}{2k}\hat{n}_2\right)\right]^k + \mathcal{O}\left(t^2/k\right)$$



Bose Hubbard Model

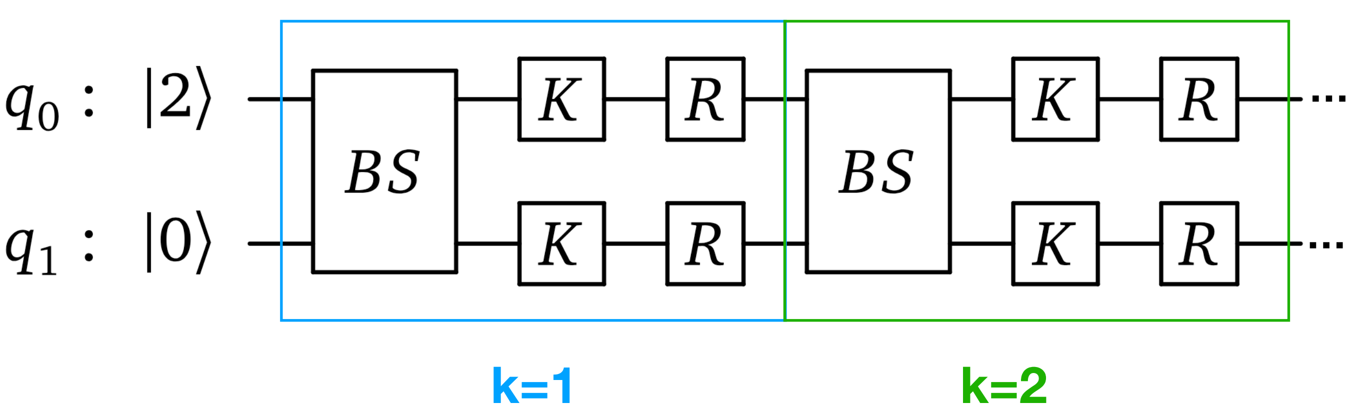
Mobility of bosons on lattice

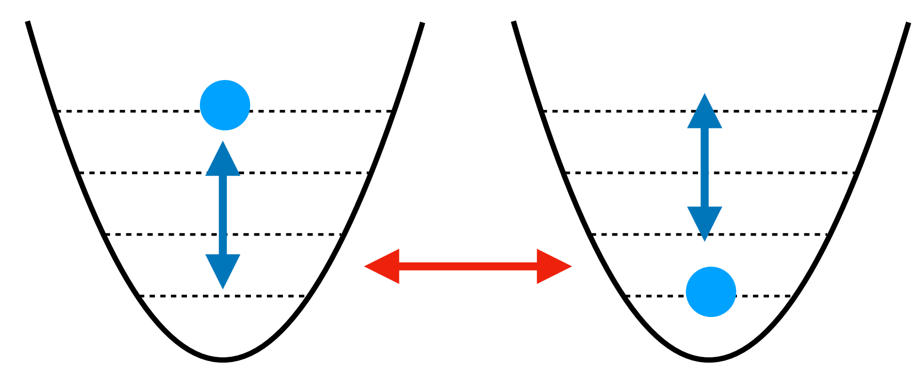
On-site interaction

$$H = J \sum_i \sum_j A_{ij} \hat{a}_i^\dagger \hat{a}_j + \frac{1}{2} U \sum_i \hat{n}_i (\hat{n}_i - 1)$$

$$e^{iHt} = \left[BS\,(\theta, \phi)\,(K(r)R(-r) \otimes K(r)R(-r))\right]^k + \mathcal{O}\left(t^2/k\right)$$

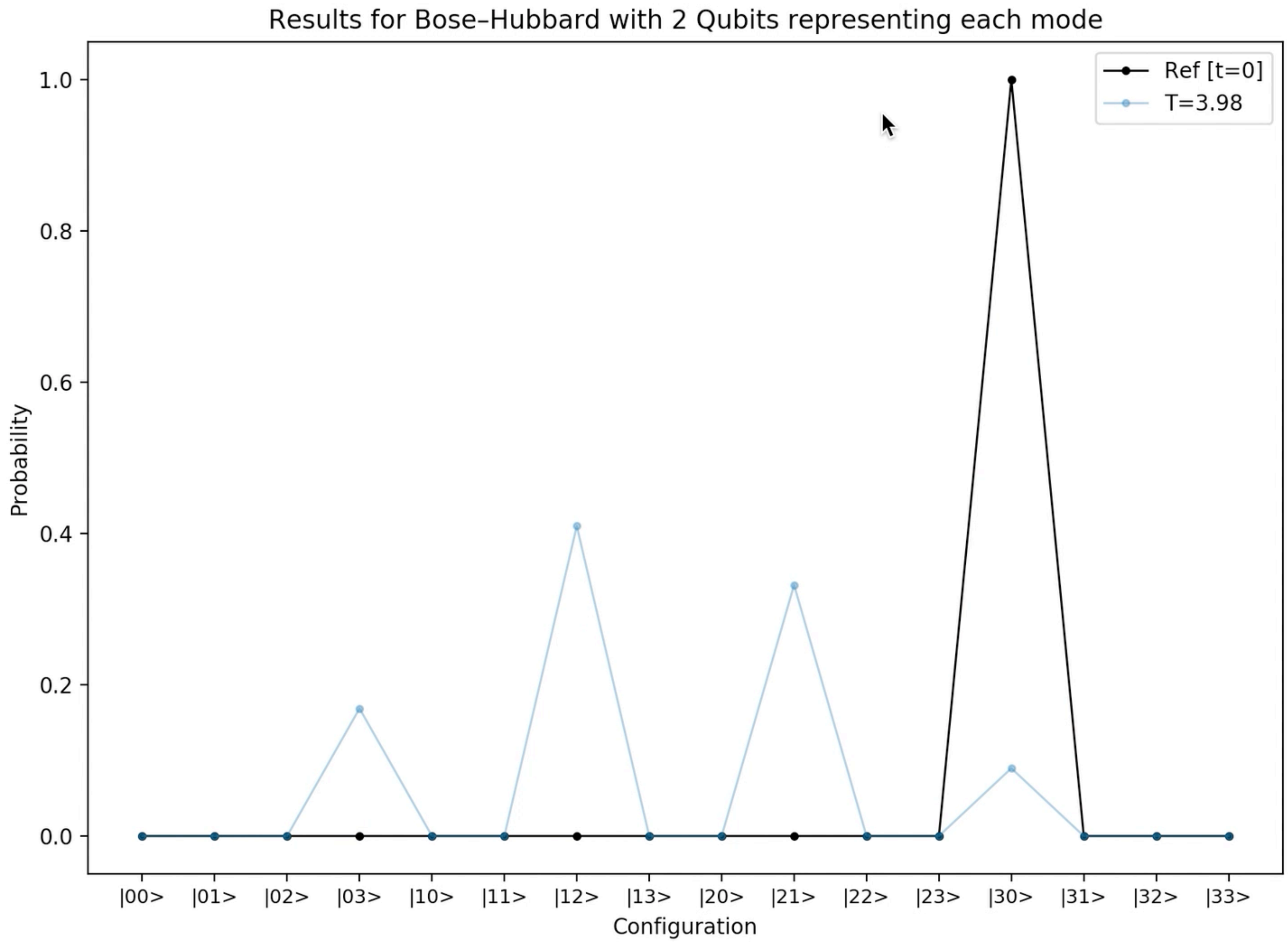
where $\theta = -Jt/k$, $\phi = \pi/2$, and $r = -Ut/2k$.





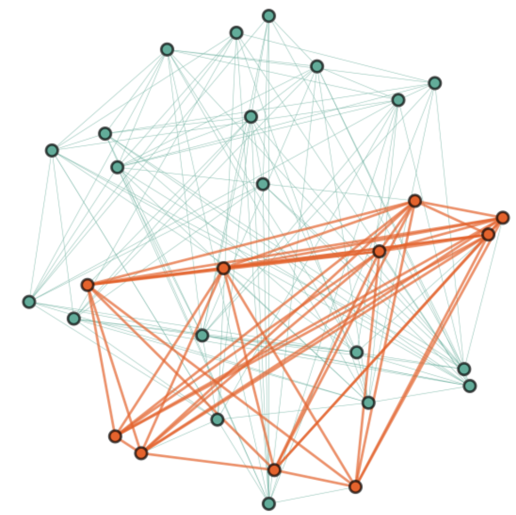
Bose Hubbard Model

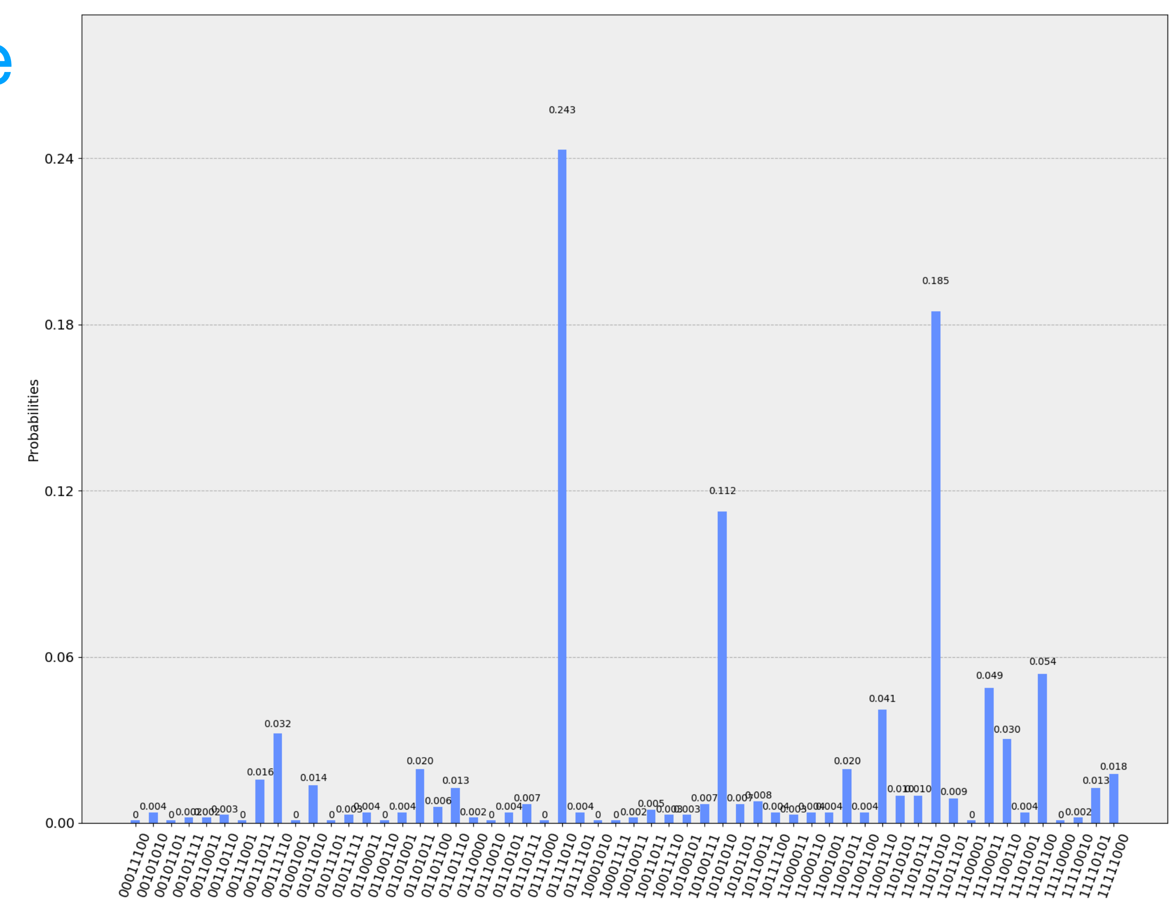
Using this model we can observe the propagation of phonons (vibrational modes) between two qumodes

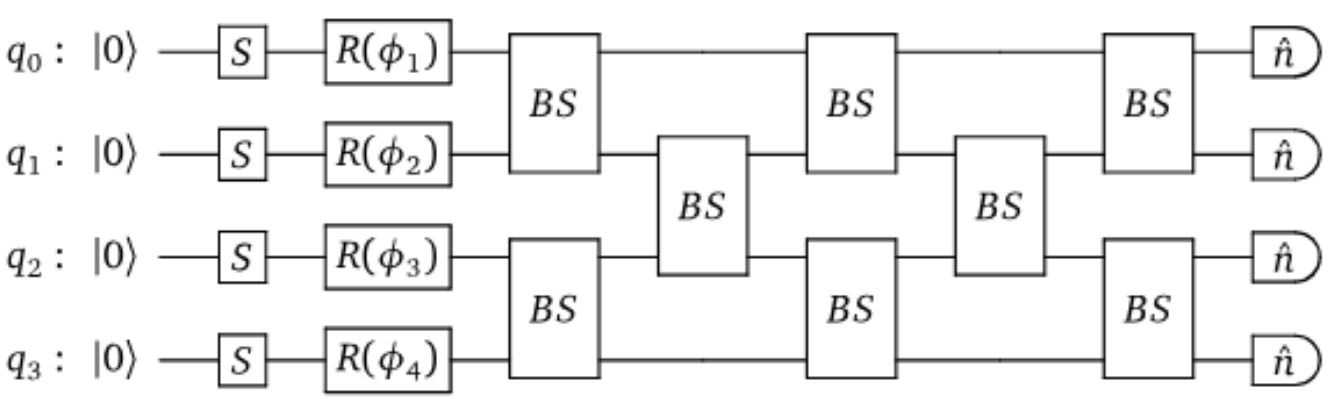


Gaussian Boson Sampling

- Finding the subgraph with the most connections (densest) within a graph
- It was implemented on qiskit using the circuit (below).
- The results show that the most probable outcomes are the densest subgraphs







Outlook

- We would like to implement full decomposition of our unitary matrices to better understand circuit depths required and inefficiencies of modelling a CV system using qiskit.

- In addition, we would like to investigate the inaccuracies that arise from our various truncation regimes.

Meet the FockWits

- Eric Brown - Creative Destruction Labs
- Arthur Pesah - University of Toronto
- Steffen Cruz - Solid State AI
- Abhi Rampal - Solid State AI
- Tommy Moffat - TKS (Honorary member)

Thanks to the Qiskit team!