Improving efficiency of non-Gaussian photonic circuit simulations

Zoltán Kolarovszki

June 21, 2022

Outline

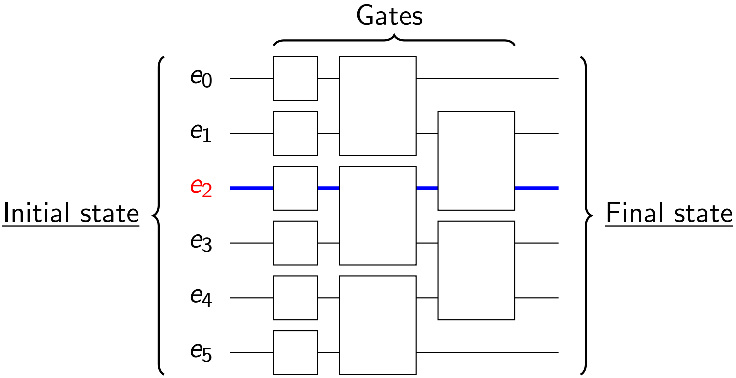
Modeling optical circuits

Tensorial computation

Proposed computation

Comparison

Modeling an optical circuit



$e_0, \ldots, e_{d-1}$ are orthonormal basis vectors corresponding to one particle in the **qumode** with the same label.

Example: e_2 corresponds to a system with a single particle in the 2nd qmode.

Fock basis states

Fock basis states can be built from $e_0, \dots, e_{d-1}$ using the symmetric tensor product. For $d = 3$, one can construct

$$e_0 \vee e_1 \vee e_1 \vee e_1 = e_0^{\vee 1} \vee e_1^{\vee 3} \vee e_2^{\vee 0} =: |\underbrace{1}_{\text{mode 0}} \underbrace{3}_{\text{mode 1}} \underbrace{0}_{\text{mode 2}}\rangle = |130\rangle. \tag{1}$$

Number of particles = sum of integers in the Fock basis state, e.g.

$$|4, 0, 1, 3\rangle \implies n = 4 + 0 + 1 + 3 = 8. \tag{2}$$

Considering only a fixed number of particles n the number of basis states is

$$\binom{d+n-1}{n} = \frac{(d+n-1)!}{(d-1)!n!} \tag{3}$$

Quantum states in the Fock representation

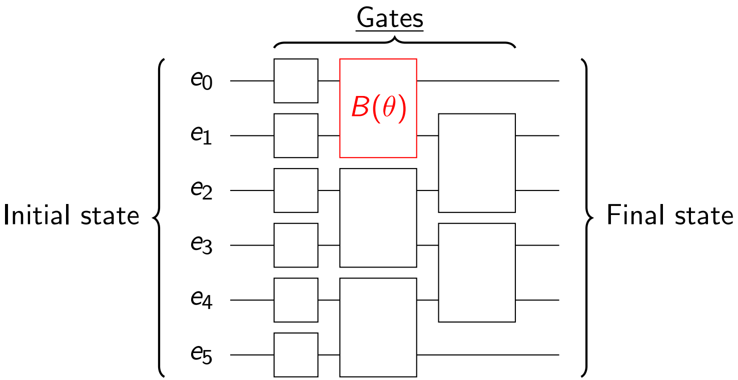
Quantum Information National Laboratory HUNGARY

Quantum states are combinations of Fock basis states, where the square of the **coefficients** sum to 1, e.g.

$$\frac{1}{2}|1,2,3\rangle + \frac{\sqrt{3}}{2}|0,2,0\rangle. \tag{4}$$

A generic pure state can be written as

$$|\psi\rangle = \sum_{\vec{n}\in\mathbb{N}^d} c_{\vec{n}}|\vec{n}\rangle, \qquad c_{\vec{n}} \in \mathbb{C}. \tag{5}$$


Gates
Initial state
e_0
e_1
e_2
e_3
e_4
e_5
$B(\theta)$
Final state

Quantum gates

Quantum Information National Laboratory HUNGARY

Quantum gates are maps between Fock basis states, e.g. the **beamsplitter gate** acts on the $|1, 0\rangle$ state as

$$B(\theta)|1, 0\rangle = \cos\theta \, |1, 0\rangle + \sin\theta \, |0, 1\rangle, \qquad \theta \in [0, 2\pi). \tag{6}$$

Generally, a general quantum gate G can be written as

$$G = \sum_{\vec{n}, \vec{m} \in \mathbb{N}^d} c_{\vec{n}, \vec{m}} |\vec{n}\rangle\langle\vec{m}|, \qquad c_{\vec{n}, \vec{m}} \in \mathbb{C}. \tag{7}$$

Mode-by-mode cutoff

No restriction on the number of particles $\implies$ infinitely many coefficients.

Idea: Cutoff.

Restrict the maximum number of particles **per mode** $\implies$ **"mode-by-mode"** cutoff.

With this consideration state can be approximated by

$$|\psi\rangle \approx \sum_{\vec{n} \in [0, c-1]^d} c_{\vec{n}} |\vec{n}\rangle, \qquad c_{\vec{n}} \in \mathbb{C}. \tag{8}$$

With this prescription, we will have c^d many coefficients.

Example: $(c = 5)$

$|2, 3, 4\rangle$ **can be** simulated, while

$|5, 6, 7\rangle$ **cannot be** simulated (since e.g. $7 \geq 5$).

Drawback of tensorial computation

c^d could be huge even for small systems!

Let $d = 7$ and $c = 5$. To store a quantum gate as a tensor of complex numbers of size 16 bytes, we need

$$(c^d)^2 \times 16 \text{ bytes} = 5^{14} \times 16 \text{ bytes} = 97656250000 \text{ bytes} \approx 90.9 \text{ GiB}$$

```
  File "/home/zk/Projects/piquasso/.venv/lib/python3.8/site-packages/strawberryfields/ba
ckends/fockbackend/circuit.py", line 446, in prepare_multimode
    self._state = ops.mix(self._state, self._num_modes)
  File "/home/zk/Projects/piquasso/.venv/lib/python3.8/site-packages/strawberryfields/ba
ckends/fockbackend/ops.py", line 120, in mix
    return np.einsum(einstr, state, state.conj())
  File "<__array_function__ internals>", line 5, in einsum
  File "/home/zk/Projects/piquasso/.venv/lib/python3.8/site-packages/numpy/core/einsumfu
nc.py", line 1359, in einsum
    return c_einsum(*operands, **kwargs)
numpy.core._exceptions.MemoryError: Unable to allocate 90.9 GiB for an array with shape
(5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5) and data type complex128
```

Active and passive gates

Quantum Information National Laboratory HUNGARY

Beamsplitter gate **conserves** the particle number:

$$B(\theta)|1,0\rangle = \cos(\theta)|1,0\rangle + \sin(\theta)|0,1\rangle. \tag{9}$$

However, the resulting state may have different number of particles than the initial state, e.g. the **displacement gate** produces particles as

$$D(\alpha)|0\rangle = e^{-\frac{|\alpha|^2}{2}} \sum_{k=0}^{\infty} \frac{\alpha^k}{\sqrt{k!}}|k\rangle \tag{10}$$

Passive gates: preserve the number of particles.

Active gates: create or annihilate particles.

Matrix representation for single particle

Fock basis states can be mapped to vectors of length d as

$$|1,0\rangle \mapsto \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{11}$$

$$|0,1\rangle \mapsto \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{12}$$

We can represent the **passive** gates as $d \times d$ matrices, i.e.

$$B(\theta) \mapsto \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} =: B(\theta)\big|_1 \tag{13}$$

Matrix representation with multiple particles

For a **multiparticle state** with n particles, passive gate representation of particle number n can be calculated from the 1-particle representation as

$$B(\theta)\big|_n = \left(B(\theta)\big|_1\right)^{\vee n}. \tag{14}$$

$\binom{d+n-1}{n}$ basis vectors $\implies$ Quantum gates are $\binom{d+n-1}{n} \times \binom{d+n-1}{n}$ matrices.

General matrix representation

A passive gate can be written in the Fock basis as

$$G = \bigoplus_{k=0}^{\infty} G|_1^{\vee k} = \begin{pmatrix} 1 & & & \\ & G|_1 & & \\ & & G|_1^{\vee 2} & \\ & & & \ddots \end{pmatrix} \sim \begin{pmatrix} * & & & & & & \\ & * & * & & & & \\ & * & * & & & & \\ & & & * & * & * & \\ & & & * & * & * & \\ & & & * & * & * & \\ & & & & & & \ddots \end{pmatrix}$$

Cutoff for total particle number

We can impose a **“system-wide” cutoff** for the total particle number. This yields square matrices of the form

$$G \approx \bigoplus_{k=0}^{c-1} G|_1^{\vee k} \sim \left(\begin{array}{cccccc|c} * & & & & & & \\ & * & * & & & & \\ & * & * & & & & \\ & & & * & * & * & \\ & & & * & * & * & \\ & & & * & * & * & \\ \hline & & & & & & \ddots \end{array}\right) = \begin{pmatrix} * & & & & & \\ & * & * & & & \\ & * & * & & & \\ & & & * & * & * \\ & & & * & * & * \\ & & & * & * & * \end{pmatrix}$$

$$\dim(G) = \binom{d+c-1}{c-1} \times \binom{d+c-1}{c-1} \qquad (15)$$

Memory usage using system-wide cutoff

Quantum Information National Laboratory HUNGARY

Let (again) $c = 5$ and $d = 7$. Then the memory usage of a quantum gate is

$$\binom{d+c-1}{c-1}^2 \times 16 \text{ bytes} = \binom{7+5-1}{5-1}^2 \times 16 \text{ bytes} \approx 1.66 \text{ MiB}.$$

Negligible in comparison with the previously obtained 90.9 GiB!

However, system-wide cutoff cuts out more coefficients $\implies$ **worse approximation**.

Example: $(c = 5)$

$$|4,3,2\rangle \in \textbf{mode-by-mode cutoff}, \text{ since } 4 < 5,\ 3 < 5 \text{ and } 2 < 5, \text{ but}$$
$$|4,3,2\rangle \notin \textbf{system-wide cutoff}, \text{ since } 4+3+2 \geq 5.$$

Argument for system-wide cutoff

Quantum Information National Laboratory HUNGARY

For active gates, contributions from higher particle numbers are (ususally) small.

Consider a simple displaced state

$$D(\alpha)|0\rangle = e^{-\frac{|\alpha|^2}{2}} \sum_{k=0}^{\infty} \frac{\alpha^k}{\sqrt{k!}} |k\rangle. \tag{16}$$

The particle number detection probabilities are

$$p(n) = |\langle n|\alpha\rangle|^2 = e^{-|\alpha|^2} \frac{|\alpha|^{2n}}{n!}, \tag{17}$$

which **quickly tapers off to** 0. Similar conclusion can be obtained for other systems.

2nd argument for system-wide cutoff

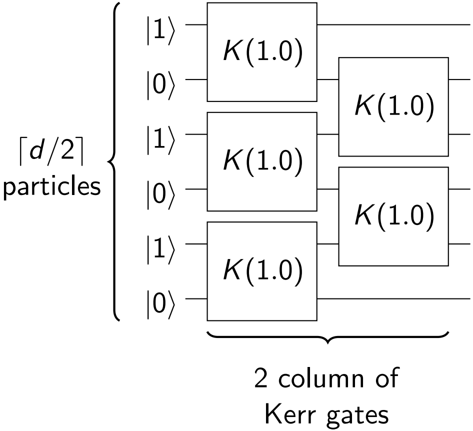
Faster simulation for Fock basis states and passive gates.

Passive gates $\implies$ number of particles are conserved.

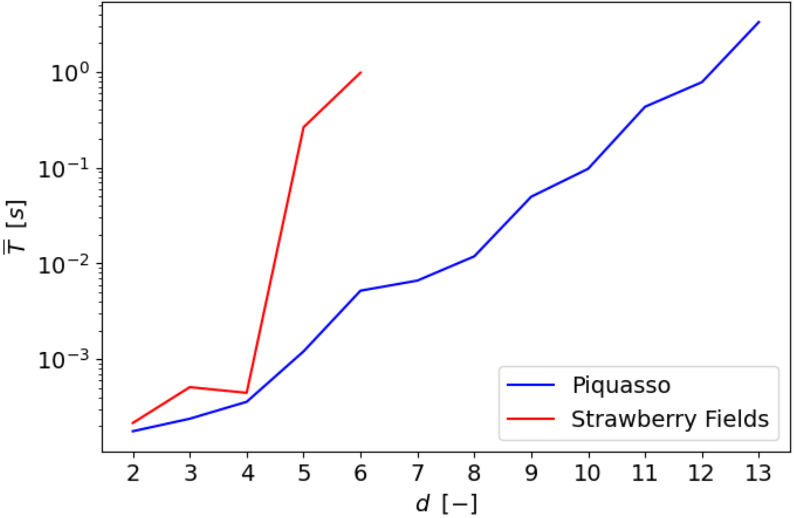
Fock basis states $\implies$ same results for identical cutoffs.

We have shown that the **memory usage** is significantly reduced.

We will show the same for the **computation time**.



Comparison of average computation times



Thank you for your attention!