

Simulating Quantum Drude Oscillators on a photonic quantum computer

Matthieu Sarkis*

*Department of Physics and Materials Science
University of Luxembourg, L-1511,
Luxembourg City, Luxembourg.*

Adel Sohbi†

ORCA

(Dated: December 12, 2022)

I. INTRODUCTION

II. DEFINITION OF THE MODEL

$$\begin{aligned}
 H &= \frac{1}{2} \sum_{i=1}^N (x_i^2 + p_i^2) + \sum_{i<j} \gamma_{ij} x_i x_j \\
 &= \sum_{i=1}^N \left(a_i^\dagger a_i + \frac{1}{2} \right) + \sum_{i<j} \gamma_{ij} x_i x_j \\
 &= \sum_{i=1}^N \left(a_i^\dagger a_i + \frac{1}{2} \right) + \frac{1}{2} \sum_{i<j} \gamma_{ij} (a_i + a_i^\dagger) (a_j + a_j^\dagger)
 \end{aligned} \tag{1}$$

with

$$a_i = \frac{x_i + ip_i}{\sqrt{2}}, \quad a_i^\dagger = \frac{x_i - ip_i}{\sqrt{2}}. \tag{2}$$

III. PHOTONIC CIRCUIT

The circuit implements a unitary $U(\theta)$ acting on an input state that we simply take to be the vacuum state $|0\rangle$. The state prepared by the circuit is therefore given by

$$|\psi(\theta)\rangle = U(\theta)|0\rangle. \tag{3}$$

IV. VARIATIONAL ALGORITHM

We define the following loss function:

$$\mathcal{C}(\theta) := \langle \psi(\theta) | H | \psi(\theta) \rangle \tag{4}$$

with the Hamiltonian defined in eq. (1). In order to compute this loss, one therefore has to measure both the photon number operator on each channel, as well as the position quadrature operator on each channel.

Algorithm 1: Computation of the energy using photon numbers and quadratures

Parameters: reference statevector $|0\rangle$, circuit U

Result: Value of the energy E

Prepare statevector $|\psi\rangle = U|0\rangle$;
Measure the position quadratures x_i ;
Prepare statevector $|\psi\rangle = U|0\rangle$;
Measure the photon numbers n_i ;
Compute the energy E with eq. (1);
return E .

Algorithm 2: Computation of the energy using coherent state basis

Parameters: reference statevector $|0\rangle$, circuit U

Result: Value of the energy E

Prepare statevector $|\psi\rangle = U|0\rangle$;
Perform heterodyne measurement on each channel to get α_i ;
Compute the energy E with eq. (1);
return E .

Algorithm 3: Computation of the energy using quadratures

Parameters: reference statevector $|0\rangle$, circuit U

Result: Value of the energy E

Prepare statevector $|\psi\rangle = U|0\rangle$;
Perform homodyne measurement on each channel to get the position quadratures x_i ;
Prepare statevector $|\psi\rangle = U|0\rangle$;
Perform homodyne measurement on each channel to get the momentum quadratures p_i ;
Compute the energy E with eq. (1);
return E .

Questions that should be addressed:

- Are alg. (1) or (3) the best since one needs to prepare the state twice to measure both the position quadratures and the photon numbers. Maybe a single measure in the coherent states basis would be possible? This can be achieved by a heterodyne measurement. With strawberry fields it can only be

* matthieu.sarkis@uni.lu

† sohbi@kias.re.kr

Algorithm 4: Computation of the loss**Parameters:** $M \in \mathbb{N}$ **Result:** Value of the loss \mathcal{C} Initialize $\mathcal{C} \leftarrow 0$;**for** $j = 1$ *to* M **do** Compute the energy E with alg. (1), (2) or (3); Update the loss $\mathcal{C} \leftarrow \mathcal{C} + E$; **end for** $\mathcal{C} \leftarrow \mathcal{C}/M$;**return** \mathcal{C} .**Algorithm 5:** Training of the parameterized photonic circuit**Parameters:** $N_{\text{steps}} \in \mathbb{N}$, initial parameters $\theta_0 \in \mathbb{R}^K$, learning rate $\eta \in \mathbb{R}_+$ **Result:** Optimized hyperparameters $\theta \in \mathbb{R}^K$ Initialize hyperparameters $\theta \leftarrow \theta_0$;**for** $i = 1$ *to* N_{steps} **do** Compute the loss \mathcal{C} with alg. (4); Compute the gradient $\nabla_{\theta}\mathcal{C}$ with shift rule and alg. (4); Update the parameters $\theta \leftarrow \theta - \eta \nabla_{\theta}\mathcal{C}$; **end for****return** θ .

performed in the Gaussian or Bosonic backends, but not the tensorflow backend...

- What is the role of the parameter M in alg. (4)? Could one safely set $M = 1$ and still hope for

convergence, a bit like in reinforcement learning? Setting $M = 1$ is of course a very rough estimate of the expected value, but since it is embedded in the training loop, maybe this estimate is actually enough?

- In strawberryfields the training procedure is managed by the tensorflow backend. Is it actually implementing the shift rule? Double check that.

After measuring the amplitude α_i on each channel with heterodyne detections, the measured energy reads

$$E = \sum_{i=1}^N \left(|\alpha_i|^2 + \frac{1}{2} \right) + 2 \sum_{i < j} \gamma_{ij} \text{Re}(\alpha_i) \text{Re}(\alpha_j) \quad (5)$$

The expected value of the energy in state $|\psi\rangle$ is obtained by averaging the result of $M \in \mathbb{N}$ such measurements:

$$\langle \psi | H | \psi \rangle = \frac{1}{M} \sum_{j=1}^M E_j + \mathcal{O} \left(\frac{1}{\sqrt{M}} \right) \quad (6)$$

A very rough estimate would consist in setting $M = 1$.

V. CONCLUSION**ACKNOWLEDGMENTS****DATA AVAILABILITY****CODE AVAILABILITY**

The reader will find an open source python code accompanying this paper following this github repository.

[1] O. Kiss, F. Tacchino, S. Vallecorsa, and I. Tavernelli, "Quantum neural networks force fields generation," *Machine Learning: Science and Technology*, vol. 3,

no. 3, p. 035004, jul 2022. [Online]. Available: <https://doi.org/10.1088/2632-2153/ac7d3c>