Quantum State Engineering using Bayesian Optimization with Gaussian Process Priors

Amanuel Anteneh[1]

[1]Department of Computer Science, Georgia Institute of Technology

Abstract

We examine the method of using Gaussian processes as priors for Bayesian optimization to optimize a quantum optical circuit to engineer a particular quantum state. The objective function for this optimization problem becomes expensive to evaluate depending on the energy of the target quantum state and the number of oscillators (qumodes) in the optical circuit. This results in less data being able to be collected making Bayesian methods particularly appealing.

Contents

1 Introduction

The dynamics of many quantum mechanical systems can be described using the tools of elementary linear algebra. In this work we will focus on the quantum harmonic oscillator which is used in the study and realization of continuous-variable quantum computing using the tools of quantum optics [1, 2].

1.1 Quantum mechanics

The states of the oscillator can be described by N-dimensional *vectors* from $\mathbb{C}^N$ [3, 4]. For example if the oscillator has 0 photons (i.e. is in the vacuum state) and we choose $N = 4$ we can denote that state as

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

similarly for an oscillator which contains 1 photon we have

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

and so on up to $N - 1$ photons.[1] In principle these vectors are infinite dimensional but for practical computations one must always select a cutoff dimension N for the Hilbert space that appropriately describes the energy of the system. The operations (which we will call gates from now on) one can perform on these oscillator states are perfectly described by $N \times N$ *matrices* such as the photon annihilation and creation operators a and $a^\dagger$ which for $N = 4$ have the following matrix representations

$$a = \begin{bmatrix} 0 & \sqrt{1} & 0 & 0 \\ 0 & 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 & \sqrt{3} \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad a^\dagger = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \sqrt{1} & 0 & 0 & 0 \\ 0 & \sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{3} & 0 \end{bmatrix} \tag{3}$$

[1] Note that $\langle 0|$ denotes the conjugate transpose of $|0\rangle$ meaning $\langle 0| = |0\rangle^\dagger = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$

and act on the state vectors of the oscillator as

$$a\left|n\right\rangle = \sqrt{n}\left|n-1\right\rangle, \quad a^{\dagger}\left|n\right\rangle = \sqrt{n+1}\left|n+1\right\rangle. \tag{4}$$

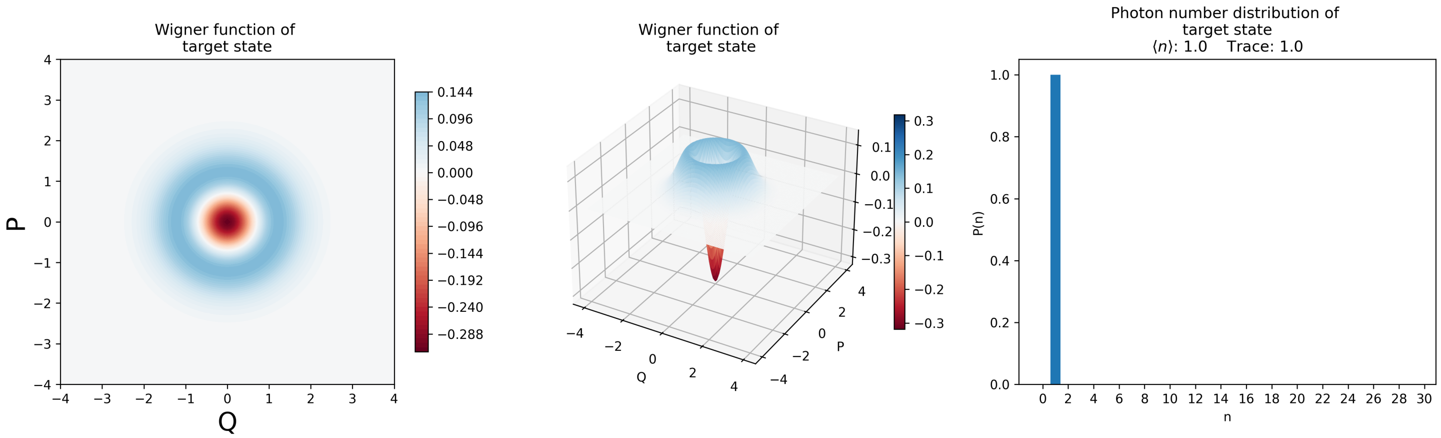


Figure 1: Wigner function and photon number probability distribution of the target oscillator state

1.2 Quantum computing

A necessary, but not necessarily sufficient, condition for a continuous-variable quantum computer to be able to achieve an exponential speed up over a classical computer is the use of non-Gaussian resources as the dynamics of a system involving only Gaussian states, gates and measurements can be efficiently simulated classically. This non-Gaussianity can be in the form of certain non-Gaussian quantum states like the Gottesman-Kitaev-Preskill grid states [5]. However, these states are not readily available so we can instead look towards non-Gaussian gates or measurements [6]. We actually need 4 distinct Gaussian gates combined with one non-Gaussian gate to achieve this exponential speedup [7]. In this work we will use one Gaussian gate and one non-Gaussian gate. The Gaussian gate we will use is the displacement gate which is defined by a single parameter α and is given by

$$D(\alpha) = e^{\alpha a^{\dagger} - \bar{\alpha} a} = \sum_{k=0}^{\infty} \frac{1}{k!}(\alpha a^{\dagger} - \bar{\alpha} a)^{k}. \tag{5}$$

In general α is a complex number but we will restrict it to be real for this work so $\alpha = \bar{\alpha}$. The non-Gaussian gate we will use is the Kerr gate which is defined by a single parameter κ

and is given by

$$K(\kappa) = e^{i\kappa(a^\dagger a)^2} = \sum_{k=0}^{\infty} \frac{1}{k!}(i\kappa(a^\dagger a)^2)^k \tag{6}$$

and is based on the self-Kerr interaction [8]. Our goal is to prepare a state $|\phi\rangle$ we desire by starting with an oscillator in the vacuum state and apply these gates successively b times in blocks as so

$$|\psi\rangle = \left[\prod_{j=1}^{b} K(\kappa_j)D(\alpha_j)\right] |0\rangle \, . \tag{7}$$

In this work we take $b = 4$ so we have parameters $\{\alpha_1, \kappa_1, \alpha_2, \kappa_2, \alpha_3, \kappa_3, \alpha_4, \kappa_4\}$. The problem one runs into is that finding what values of these parameters will give $|\psi\rangle = |\phi\rangle$ is highly non-intuitive.

2 Optimization problem & methods

2.1 Optimization problem

This problem can be framed as one of optimization. We would like to find values for the parameters $\mathbf{x} = [\alpha_1, \kappa_1, \alpha_2, \kappa_2, \alpha_3, \kappa_3, \alpha_4, \kappa_4]$ such that for a target quantum state $|\phi\rangle$ we maximize $|\langle\phi|\psi\rangle|^2$ which is the squared modulus of the dot product of the two quantum state vectors. We call this quantity the fidelity and it ranges from 0 (meaning the states are orthogonal) to 1 (meaning the states are identical). For the purpose of this work we will chose the target state to be the single photon state $|\phi\rangle = |1\rangle$. This is a non-Gaussian state which is currently only able to be created probabilistically in the lab.[2] So the optimization problem can be written as

$$\arg\max_{\mathbf{x}} f(\mathbf{x}) = \underset{\substack{[\alpha_1,\kappa_1,\alpha_2,\kappa_2,\\ \alpha_3,\kappa_3,\alpha_4,\kappa_4]}}{\arg\max} f(\alpha_1, \kappa_1, \ldots, \alpha_4, \kappa_4) = \underset{\substack{[\alpha_1,\kappa_1,\alpha_2,\kappa_2,\\ \alpha_3,\kappa_3,\alpha_4,\kappa_4]}}{\arg\max} |\langle\phi|\psi\rangle|^2 \tag{8}$$

where $|\phi\rangle$ is the target state vector $|1\rangle$ and

$$|\psi\rangle = [K(\kappa_4)D(\alpha_4)K(\kappa_3)D(\alpha_3)K(\kappa_2)D(\alpha_2)K(\kappa_1)D(\alpha_1)] |0\rangle \, . \tag{9}$$

[2] The curious reader may wonder why we simply don't use the creation operator to prepare the $|1\rangle$ state if we start with the vacuum state $|0\rangle$. This is simply because no feasible method currently exists for applying the creation or annihilation operators to arbitrary oscillator states deterministically in the lab.

In Fig. 1 we show the Wigner function, which is a quasi-probability distribution that describes quantum states, and the photon number probability distribution of the target state $|1\rangle$.

2.2 Why use Bayesian optimization?

The biggest obstacle for these quantum optimization problems stems from the fact that for a N-dimensional vector space and a quantum system of M oscillators the size of the matrices scales as $O(N^M)$ which is exponential in the size of the system. While we only consider the case of one oscillator in this work it is not uncommon that one would need to work with many more. This means that computation of the cost function can become extremely expensive since matrix multiplication has a complexity of $O(m^3)$ for $m \times m$ matrices. The common approach taken in the literature for these quantum optimization problems is to use gradient based optimization protocols [9, 10]. While these methods are quite successful they require the additional overhead of computing the derivative of the cost function in addition to the cost function itself thereby increasing the number of matrix multiplications performed.

By contrast, derivative-free optimization methods like Bayesian optimization avoid this overhead by assuming the function to be a black box and thus not providing any gradient information. While for many quantum mechanical phenomena we can derive and computationally evaluate an analytic expression for their dynamics there are also many cases in which an analytic solution (and therefore the respective derivatives) can not be found. Bayesian optimization is also useful here since it is better equipped to handle objective functions that are expensive to evaluate compared to most other derivative-free methods. Lastly gradient descent based methods are prone to converging to local minima whereas Bayesian optimization methods are designed to locate global minima [11].

2.3 Bayesian optimization

Algorithm 1 shows the pseudocode for the Bayesian optimization method. N is the maximum number function evaluations/queries we are allowed to perform. n_0 is the number of initial randomly selected points from the domain that we know the function value at. Given that Bayesian optimization assumes the objective function is a black box it relies on modeling the objective function using a surrogate model, which is commonly a Gaussian process, to locate the optimum.

2.3.1 Non-parametric Bayesian regression: Gaussian processes

A stochastic process can be viewed as a generalization of a probability distribution to the space of functions [12]. That is to say, where as sampling from a probability distribution

Algorithm 1 Bayesian optimization algorithm

```
Require: $n_0 > 0$ and $N > n_0$
1: Begin with Gaussian process prior on objective function $f$
2: Query $f$ at $n_0$ points to get $n_0$ initial data points
3: $n \leftarrow n_0$
4: while $n \leq N$ do
5:     Use newly acquired data and old data to update the posterior probability distribution on $f$
6:     Using the current posterior distribution compute the acquisition function $A_n(\mathbf{x})$ and find $\mathbf{x}^*$ which is the $\mathbf{x}$ value that maximizes $A_n(\mathbf{x})$
7:     Observe $y_n = f(\mathbf{x}^*) + \epsilon_n$ where $\epsilon_n \sim \mathcal{N}(\mu, \sigma^2)$ for noisy objective functions and $\epsilon_n = 0$ for noiseless objective functions
8:     $n \leftarrow n + 1$
9: end while
```

results in a scalar, or a vector of scalars in the case of multivariate distributions, sampling a stochastic process results in a *function*. They can therefore be thought of as probability distributions over functions. A Gaussian process, which we will denote $\mathcal{GP}(m(\mathbf{x}), \kappa(\mathbf{x}_i, \mathbf{x}_j))$, is a specific type of stochastic process and an example of a Bayesian non-parametric model [13]. It is a collection of random variables whose joint distribution is Gaussian and like it's multivariate probability distribution counterpart it is fully characterized by a mean function $m(\mathbf{x})$ and covariance function (also called a kernel) $\kappa(\mathbf{x}_i, \mathbf{x}_j)$. The mean function defines the mean of the Gaussian process at any given point $\mathbf{x}$ in the domain. The covariance function defines the covariance between the function values at particular points $\mathbf{x}_1$ and $\mathbf{x}_2$ i.e. $\text{cov}(f(\mathbf{x}_i), f(\mathbf{x}_j)) = \kappa(\mathbf{x}_i, \mathbf{x}_j)$. The covariance function is used to control the smoothness of the functions drawn from the Gaussian process by tuning its parameters such as the length scale parameter l of the radial basis covariance function we use later. As a concrete example in Fig. 2 we show the 95% confidence interval and 4 sample functions drawn from a 1-dimensional Gaussian process where $m(\mathbf{x}) = 0$, i.e. the mean at each $\mathbf{x}$ value in the domain is 0, and the covariance function $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ of the Gaussian process is the radial basis function (RBF)

$$\kappa(\mathbf{x}_i, \mathbf{x}_j; l) = \exp\left(-\frac{d(\mathbf{x}_i, \mathbf{x}_j)^2}{2l^2}\right) \tag{10}$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ is the Euclidean distance and we set the length scale l to 1. In Fig. 3a we can see how the Gaussian process prior is updated once we have observed new data. Notice how each of the sampled functions now interpolates each of the observed data points and the confidence region near those points is considerably narrower. This covers line 5 of Algorithm

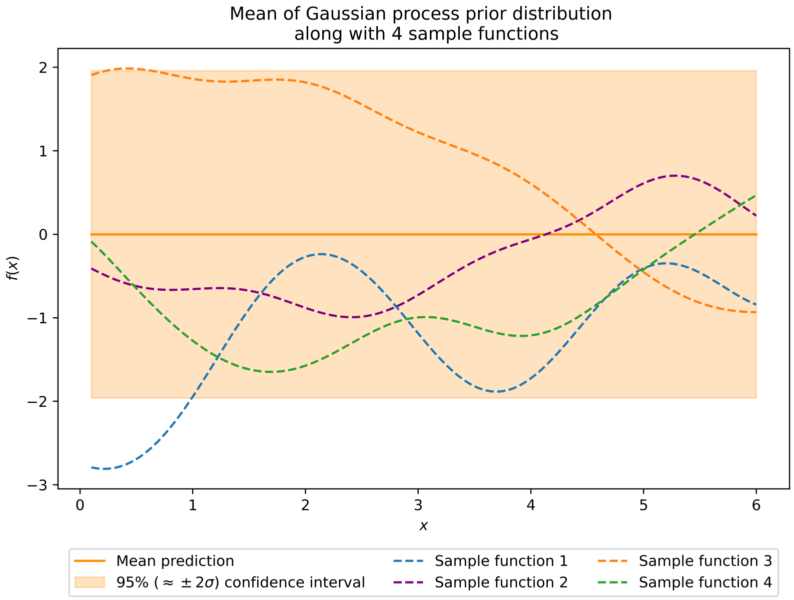


Figure 2: Gaussian process prior distribution over functions

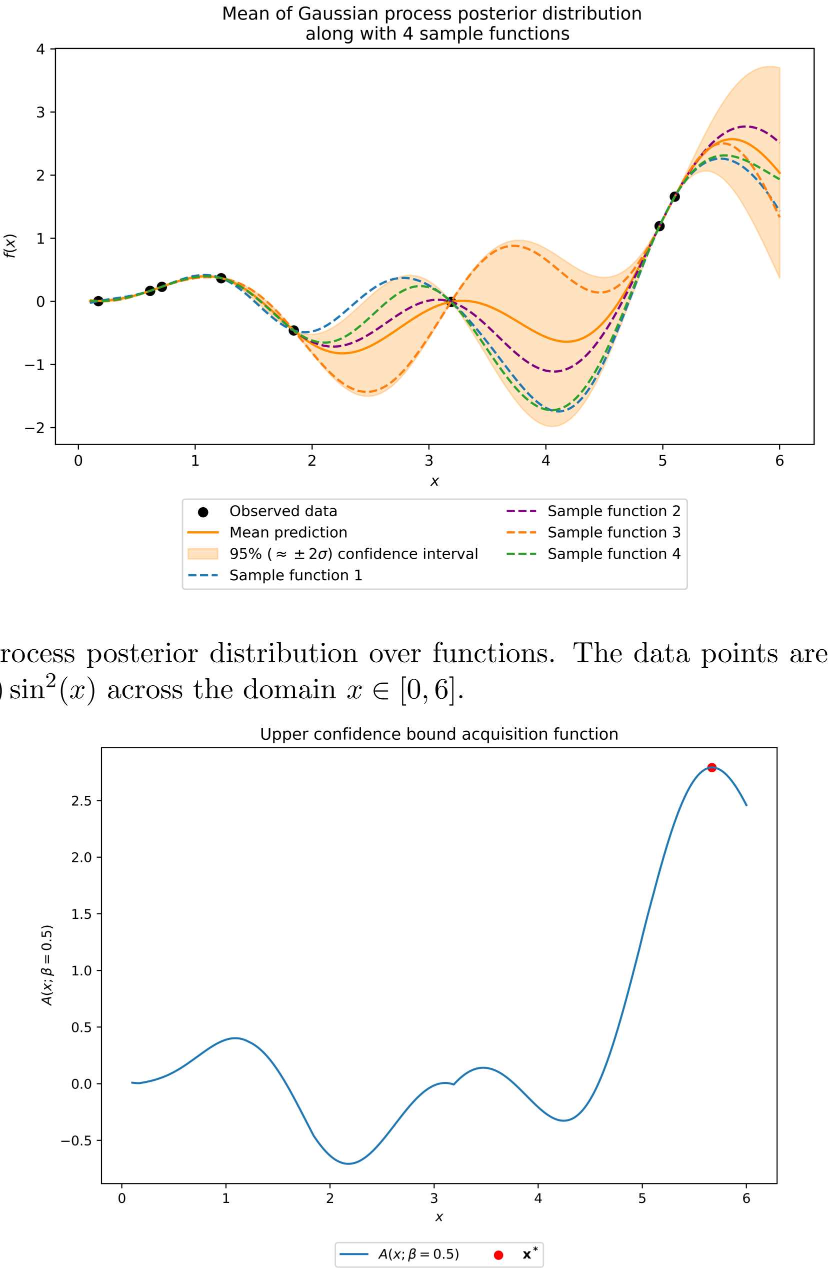
1 and is the essence of Bayesian optimization with Gaussian processes.

2.3.2 Acquisition functions

The acquisition function is used to help the algorithm determine what point in the domain to query next in the optimization process by assigning a score to each point $\mathbf{x}$. In contrast to the objective function the acquisition function is meant to be inexpensive to evaluate with well defined gradients allowing us to more easily find its extrema [11, 13]. There are many different acquisition functions used in Bayesian optimization such as expected improvement and probability of improvement [11]. However, for this work we use one of the simplest acquisition functions which is the upper confidence bound (UCB) defined as

$$A(\mathbf{x};\beta) = \mu(\mathbf{x}) + \beta\sigma(\mathbf{x}) \tag{11}$$

where $\mu(\mathbf{x})$ is the mean of the Gaussian process posterior at $\mathbf{x}$ and $\sigma(\mathbf{x})$ is the standard deviation or uncertainty of the posterior at $\mathbf{x}$. β is used to control the trade-off between exploration and exploitation [13]. An example of the UCB acquisition function for the Gaussian process posterior is shown in Fig. 3b. This covers line 6 of Algorithm 1.



(a) Gaussian process posterior distribution over functions. The data points are from the function $f(x) = x\cos(x)\sin^2(x)$ across the domain $x \in [0, 6]$.

(b) Acquisition function (UCB) for the Gaussian process posterior shown in Fig. 3a with $\beta = 0.5$. The red point denotes the location of $\mathbf{x}^*$ which will be the point queried in the next step of the optimization algorithm.

Figure 3: Gaussian process posterior and acquisition function

3 Results

To compare our results we use the derivative-free Nelder-Mead optimization method to optimize the objective in addition to using Bayesian optimization. We use `SciPy`'s implementation of the Nelder-Mead algorithm, the Python library `Bayesian Optimization` to implement the Bayesian optimization algorithm with Gaussian processes [14] and `QuTiP` to simulate the quantum dynamics [15]. For Bayesian optimization we begin with a Gaussian process prior with $m(\mathbf{x}) = 0$ and

$$\kappa(\mathbf{x}_i, \mathbf{x}_j; \nu, l) = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left[\frac{\sqrt{2\nu}}{l} d(\mathbf{x}_i, \mathbf{x}_j) \right]^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu}}{l} d(\mathbf{x}_i, \mathbf{x}_j) \right) \tag{12}$$

which is the Matérn covariance function where $\Gamma(\cdot)$ is the gamma function and $K_\nu(\cdot)$ is a modified Bessel function [12]. We set $l = 1$ and $\nu = 2.5$. For the UCB acquisition function we set β to 0.5 and 1.65 for the noiseless and noisy cases respectively. For both noise cases we set $n_0 = 50$ and $N = 200$.

3.1 Noiseless case

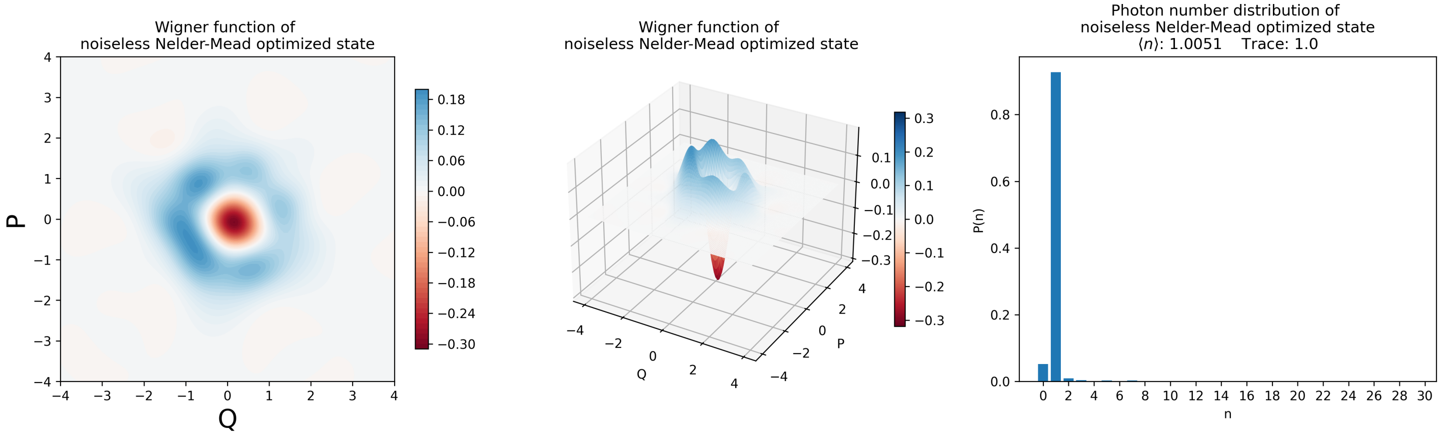


Figure 4: Wigner function and photon number probability distribution of the state found via the Nelder-Mead optimization method when using a noiseless objective function.

We will first consider the case where there is no noise, that is to say, the quantum optical circuit is perfectly isolated from interactions with the environment. Fig. 4 shows the Wigner function of the state prepared using the optimal parameters found by the Nelder-Mead optimization method. From Table 1 we can see that this state has fidelity ≈ 0.93 with the target state. Fig. 5 shows the Wigner function of the state prepared using the optimal parameters found by the Bayesian optimization method. From Table 1 we can see that the state has

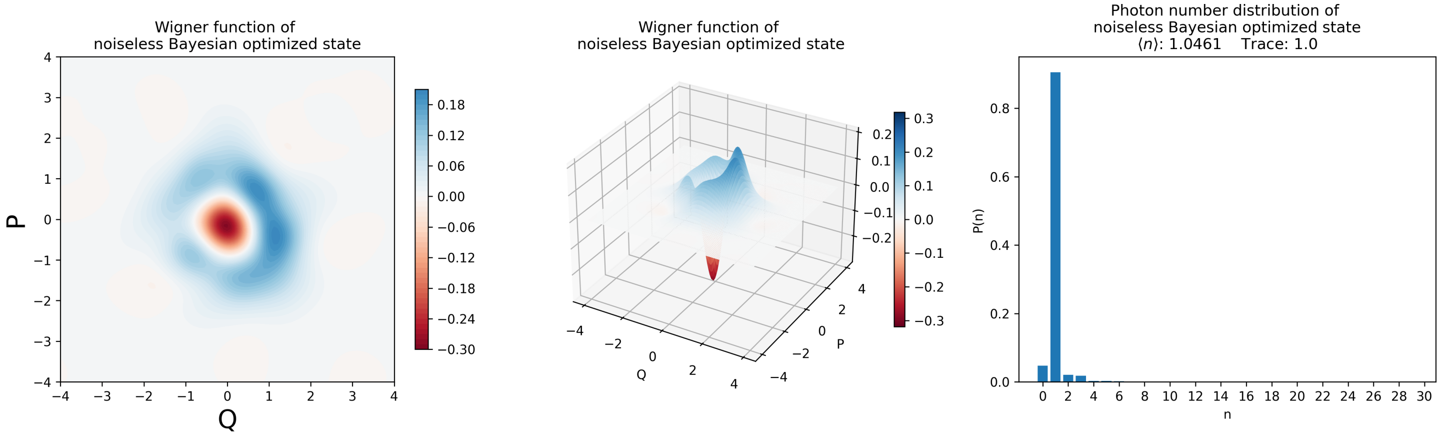


Figure 5: Wigner function and photon number probability distribution of the state found via the Bayesian optimization method when using a noiseless objective function.

	Final fidelity	# of function evaluations
Nelder-Mead	0.9271	446
Bayesian	0.9054	200

Table 1: Results from using the two algorithms to optimize the gate parameters with a noiseless objective function

slightly lower fidelity with the target state compared to the one found by the Nelder-Mead method. However, we can see that the Bayesian method required less than $\frac{1}{2}$ the number of function evaluations to find an optimum. Table 2 shows the values of the optimal parameters found by both optimization methods. Interestingly the two methods find quite different optimal values with the Bayesian method favoring values of the Kerr strengths κ_i with larger magnitude. A similar trend can be observed with the displacement strengths α_i which explain the higher energy, average photon number of 1.0461, of the state produced by Bayesian optimization.

3.2 Noisy case

When one is creating these states in the lab there will inevitably be imperfections in the experiment resulting in some noise in the objective function calculation. In quantum optical

	α_1	κ_1	α_2	κ_2	α_3	κ_3	α_4	κ_4
Nelder–Mead	0.9276	0.5311	0.5175	0.6032	0.3475	0.0648	0.0484	0.9562
Bayesian	0.0341	-1.3293	-0.7926	-0.4785	0.5493	-0.6107	-0.4167	2.3585

Table 2: Optimal gate parameters found when using a noiseless objective function

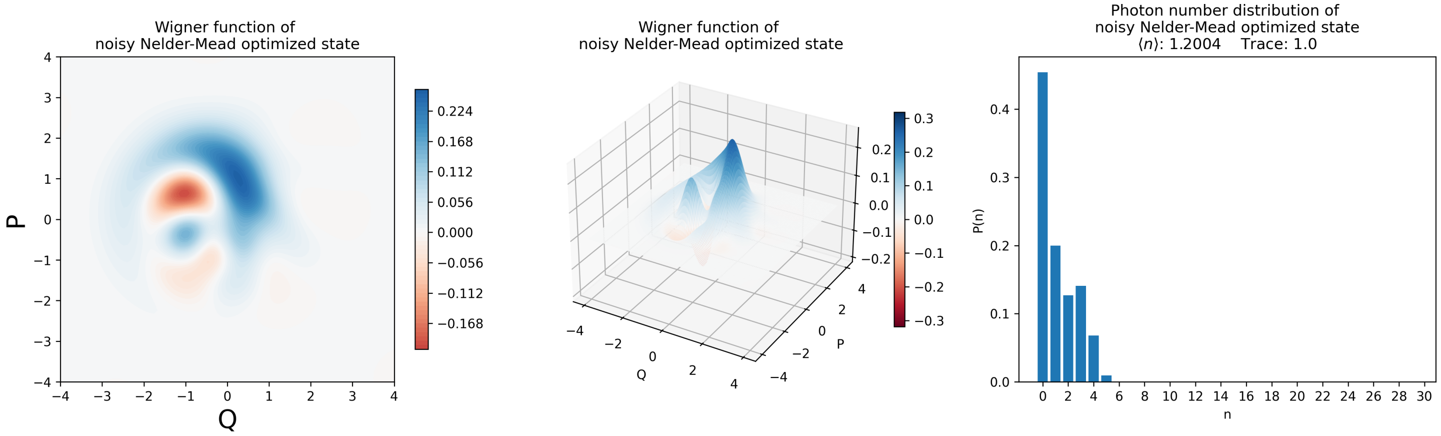


Figure 6: Wigner function and photon number probability distribution of the state found via the Nelder-Mead optimization method when using a noisy objective function.

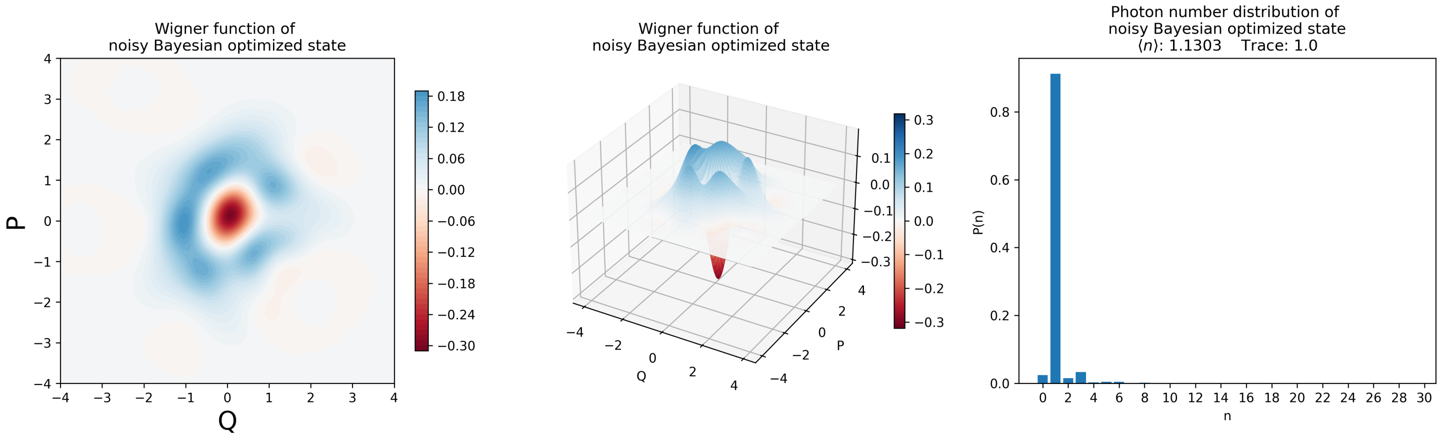


Figure 7: Wigner function and photon number probability distribution of the state found via the Bayesian optimization method when using a noisy objective function.

experiments the dominate source of imperfections is photon loss which can be modeled using the Kraus operators

$$L_l = \sqrt{\frac{\eta^l}{l!(1-\eta)^l}}\, a^l e^{\frac{1}{2} a^\dagger a \ln(1-\eta)} \tag{13}$$

where $\eta \in [0, 1]$ denotes the percentage of photons that are lost [16]. These Kraus operators transform a quantum state $|\chi\rangle$ as

$$\rho = \sum_{l=0}^{N} L_l \left|\chi\right\rangle \left\langle\chi\right| L_l^\dagger \tag{14}$$

	Final fidelity	# of function evaluations
Nelder-Mead	0.1999	2000
Bayesian	0.9118	200

Table 3: Results from using the two algorithms to optimize the gate parameters with a noisy objective function

	α_1	κ_1	α_2	κ_2	α_3	κ_3	α_4	κ_4
Nelder–Mead	0.5953	0.7352	0.0204	0.9857	0.8476	0.2145	0.1820	0.1826
Bayesian	0.2967	-0.0978	-0.4765	0.9171	0.6695	-0.9233	-0.3414	2.9007

Table 4: Optimal gate parameters found when using a noisy objective function

for a N-dimensional Hilbert space.[3] Now the objective function becomes

$$\underset{\substack{\alpha_1,\kappa_1,\alpha_2,\kappa_2,\\ \alpha_3,\kappa_3,\alpha_4,\kappa_4}}{\arg\max}\ f(\alpha_1,\kappa_1,\ldots,\alpha_4,\kappa_4) = \underset{\substack{\alpha_1,\kappa_1,\alpha_2,\kappa_2,\\ \alpha_3,\kappa_3,\alpha_4,\kappa_4}}{\arg\max}\ \mathrm{Tr}[\rho\tau] \tag{15}$$

where $\tau = |1\rangle\langle 1|$ is our previous target state in density matrix form,

$$\rho = \sum_{l=0}^{N} L_l\,|\psi\rangle\langle\psi|\,L_l^\dagger \tag{16}$$

is our output state that is degraded by loss η and $\mathrm{Tr}[\rho\tau]$ is the density matrix version of the fidelity of two quantum states.[4] We draw η from a normal distribution with mean 0 and standard deviation 0.1. Note we take the absolute value of the sampled η and truncate it to be ≤ 0.35 so that we only have a maximum loss of 35% which is realistic for state of the art experiments.

Fig. 6 shows the state prepared using the optimal parameters found by the Nelder-Mead optimization method. We can immediately see that the Wigner function bears little visual similarity to that of the target state and from Table 3 we can see that the resulting fidelity with the target state is ≈ 0.2 which is very sub-optimal. Perhaps worst of all we can see that 2000 function evaluation were performed to arrive at this sub-optimal result. The story is very different however with respect to the Bayesian optimized state shown in Fig. 7 which from Table 3 we can see has fidelity ≈ 0.92 with the target state. Clearly the Nelder-Mead method is much less robust to random noise in the objective function than the Bayesian

[3] Note that we use the density matrix formalism $\rho = |\chi\rangle\langle\chi|$ from here on out as photon loss results in quantum states that are statistical mixtures which can, in general, only be described by density matrices as opposed to state vectors.

[4] Note that at least one of the quantum states must be a pure state for this expression of the fidelity to be correct. If both states are mixed then the fidelity is given by the much more complicated expression $\mathrm{Tr}^2\left[\sqrt{\sqrt{\rho}\tau\sqrt{\rho}}\right]$.

method. This is likely due to the relative simplicity of the Nelder-Mead method which does not allow it to account for uncertainty in the value of the objective function at a specific point $\mathbf{x}$ unlike Bayesian methods. Furthermore, we can see that the Bayesian method only required $\frac{1}{10}$ the number of function evaluations to arrive at a much better optimum! Lastly, from Table 4 we can see that the Bayesian method again favors parameters with larger values of κ_i and α_i.

4 Conclusion

We have shown how a Bayesian optimization method can be used to optimize a quantum optical circuit and that this method greatly outperforms other derivative-free optimization methods when the objective function is noisy. However, it is important to note that Bayesian optimization currently performs poorly for high dimensional optimization problems i.e. where the number of parameters to optimize exceeds $\sim$20. On the experimental front the Kerr gate is very challenging to implement and control in an optical setting due to the weak strength of third (and higher) order non-linearities in quantum optics of which the self-Kerr effect is an example. Therefore in quantum optics non-Gaussian *measurements*, instead of non-Gaussian *gates*, like photon counting have been the point of focus for preparing non-Gaussian quantum states but we leave the question of how Bayesian methods can be applied in that setting for a future work. Lastly, we did not investigate how adding a regularization term that is a function of the gate parameters to the objective function effected the sparsity of the optimal gate parameters. This would be useful to explore given the aforementioned weakness of higher order non-linearities in quantum optics combined with the high error rates of current quantum gates.

5 Code availability

The Python code used to generate the data and plots in this work can be found at the following repository.

References

[1] O. Pfister, “Qubits without qubits,” *Science*, vol. 383, no. 6680, pp. 264–264, 2024.

[2] O. Pfister, “Continuous-variable quantum computing in the quantum optical frequency comb,” *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 53, no. 1, p. 012001, 2019.

[3] D. J. Griffiths and D. F. Schroeter, *Introduction to quantum mechanics.* Cambridge university press, 2018.

[4] J. S. Townsend, *A modern approach to quantum mechanics.* University Science Books, 2000.

[5] D. Gottesman, A. Kitaev, and J. Preskill, “Encoding a qubit in an oscillator,” *Physical Review A*, vol. 64, no. 1, p. 012310, 2001.

[6] M. Walschaers, “Non-gaussian quantum states and where to find them,” *PRX quantum*, vol. 2, no. 3, p. 030204, 2021.

[7] S. Lloyd and S. L. Braunstein, “Quantum computation over continuous variables,” *Physical Review Letters*, vol. 82, no. 8, p. 1784, 1999.

[8] C. C. Gerry and P. L. Knight, *Introductory quantum optics.* Cambridge university press, 2023.

[9] M. Kudra, M. Kervinen, I. Strandberg, S. Ahmed, M. Scigliuzzo, A. Osman, D. P. Lozano, M. O. Tholén, R. Borgani, D. B. Haviland, *et al.*, “Robust preparation of wigner-negative states with optimized snap-displacement sequences,” *PRX Quantum*, vol. 3, no. 3, p. 030301, 2022.

[10] J. M. Arrazola, T. R. Bromley, J. Izaac, C. R. Myers, K. Brádler, and N. Killoran, “Machine learning method for state preparation and gate synthesis on photonic quantum computers,” *Quantum Science and Technology*, vol. 4, no. 2, p. 024004, 2019.

[11] P. I. Frazier, “Bayesian optimization,” in *Recent advances in optimization and modeling of contemporary problems*, pp. 255–278, Informs, 2018.

[12] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*, vol. 2. MIT press Cambridge, MA, 2006.

[13] R. Garnett, *Bayesian optimization.* Cambridge University Press, 2023.

[14] F. Nogueira *et al.*, “Bayesian optimization: Open source constrained global optimization tool for python,” 2014.

[15] J. R. Johansson, P. D. Nation, and F. Nori, “Qutip: An open-source python framework for the dynamics of open quantum systems,” *Computer Physics Communications*, vol. 183, no. 8, pp. 1760–1772, 2012.

[16] M. Eaton, C. González-Arciniegas, R. N. Alexander, N. C. Menicucci, and O. Pfister, “Measurement-based generation and preservation of cat and grid states within a continuous-variable cluster state,” *Quantum*, vol. 6, p. 769, 2022.